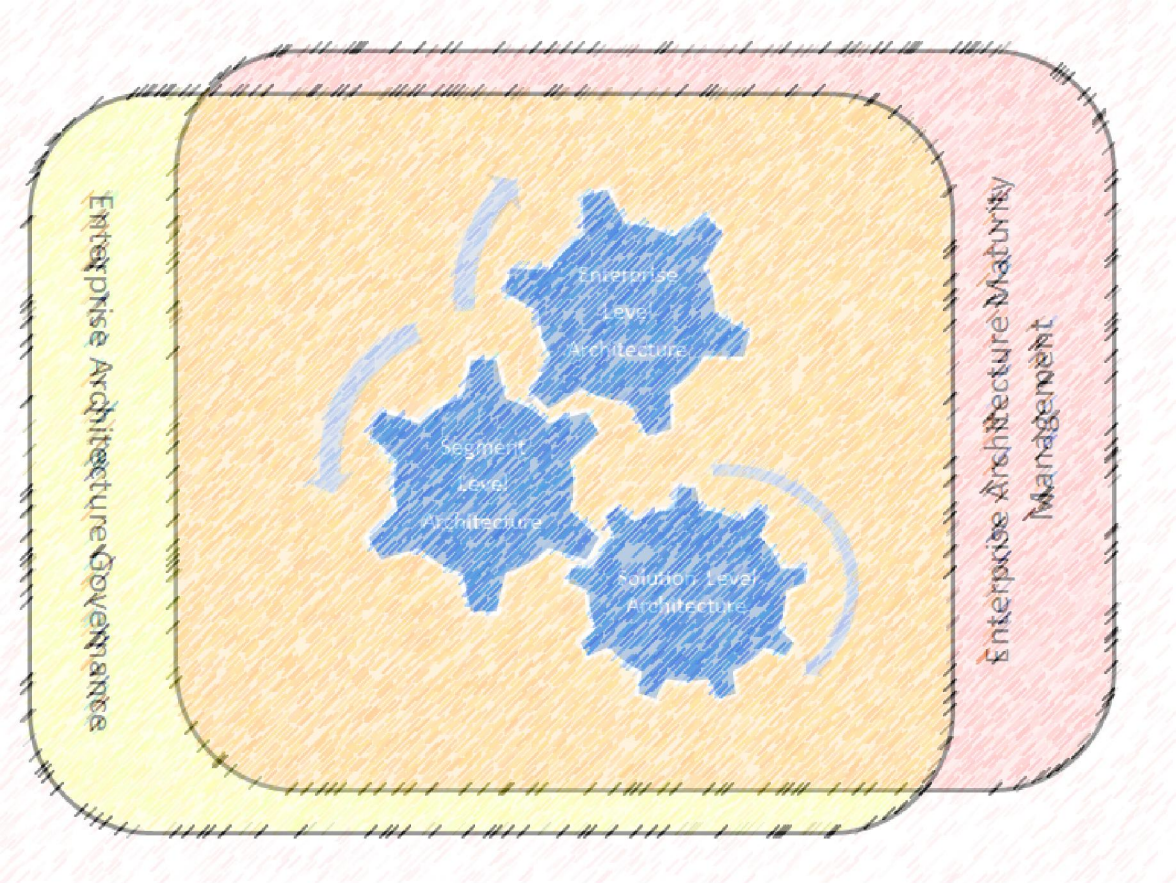


An Enterprise Architecture Practitioner's Notes

VOLUME 4: GOVERNANCE & MATURITY MANAGEMENT

Matthew Ford Kern

Copyright 2014 to 2017

$200.00
ISBN 978-1-4951-7312-7
20000>
9 781495 173127

Nature of Release	Changes Made	Author Initials	Date
Initial Release	Initial Release	MFK	**2015**
Editorial Changes	Comment incorporated	MFK	2016

Title: An Enterprise Architecture Practitioner's Notes

Subtitle: Volume 2: Solution Level Architecture

Self Published, 2015

Matthew Ford Kern, MsEA BsEE CEA(DODAF, FEAF and Zachman), CISSP-ISSAP PMP

ISBN 978-1-4951-7313-4

Price for spiral bound copy, or commercial purposes: $200

Use as an eBook free for education and government.

CONTENTS

VOLUME 4: GOVERNANCE AND MATURITY MANAGEMENT

This series of books will describe what I view as the totality of enterprise architecture, in the broad meaning of that term. It will consist of five volumes as follows:

1. Enterprise Level Architecture

2. Segment Level Architecture

3. Solution Level Architecture

4. Governance and Maturity Management

5. The Business Environment

While the series will address enterprise architecture (EA) in the broad sense, this volume will address enterprise level architecture in the narrow sense. It is this narrow activity that gives the whole its name, purpose and organizing principles.

These volumes originated as blog posts on my personal website, and later on LinkedIn™. I wrote these to popularize and clarify the deep understanding of enterprise architecture of my government colleagues, who invented it. It seems to me that knowledge of the subject has spread from its global hub city, Washington DC, out to the rest of the globe. We know it best here.

As for myself, my main contribution at the enterprise level is simply popularizing, restating the work of the various founders of EA, of whom I know many. Some are friends; all are colleagues. They created a comprehensive vision of EA, and I follow it. This is mostly true of EA maturity management and EA governance as well. In regard to solution architecture and segment architecture, I have contributed some innovations.

I did contribute the organizational structure for the book, the “Five Activities Model.” It is a small and obvious addition to the work of Dick Burk while at the Office of Management and Budget (OMB).

This work is not another framework and is not intended to replace them. It is designed to, instead, provide what they do not: perspective. This particular volume draws mainly on Federal Enterprise Architecture Framework (FEAF) and FEA concepts, with Zachman’s concepts, as those are more directly applicable to this level of architecture and scope of effort.

Each day I hope to do something useful. It is not an ambitious philosophy, but it helps in consulting. With this book I also hope that I have done something useful. Please let me know if I have.

SECTION 1: INTRODUCTION TO VOLUME 4

The section "My Ideas: Mostly Not" describes the limits of my contribution to the art relative to the founders. The other four sections describe the underlying model on which the five volumes are based.

In "The Five Activities…" the basic notion of dividing EA into these five bins is described and then in "Enterprise, Segment, Solution" the preceding model of OMB and Burk is recapped and simplified. In "Correspondence to Management Activities" the link of each of the OMB levels to the PMI levels of management is described. Lastly in "Strategic, Operational and Tactical Thinking" the correspondence to levels of planning is implied.

New material was added for this volume. In "Why Do Architecture" I address the attitude that architecture is not required. In "Introducing Architecture…" I give some notes on initial implementation of an architecture program. In "…Carts and Horses" I address the phenomenon where some organizations promote parts above the whole. In "Achieving Success" I suggest that architecture is used to achieve some goal or goals.

QUESTIONS FOR SECTION 1

1. Do these five levels cover all of enterprise architecture? If not, what is left out?
2. Does the three-level model correspond to contracts and "statements of work" you have seen?
3. How often is it true that the segment architecture works for the program manager? Should it happen more often in an ideal situation? What of the other levels?
4. Have you seen other authors use a different order of operational and tactical planning? Have you seen them refer to all planning as strategic? Is that useful?
5. Why do architecture, in your own words?
6. Is it possible to have architecture without a goal or goals?

1.1 MY IDEAS: MOSTLY NOT, MAY 25, 2015

Several people suggested I should write a book, notably Dale Chalfant, in Detroit (a fine architect), who convinced me. I had toyed with and resisted the idea for several years, as most of what I have learned has come from those founders of enterprise architecture and systems engineering and whomever else I have read. I have added little to the body of thought, a bit here or there, but nothing like their sweeping insights.

You could say that mostly I simply popularize (simplify, clarify and restate) what my government and beltway friends created and regarding enterprise level architecture that would be fair. I think in solution architecture I may have innovated more.

> I really don't have that many ideas of my own regarding EA, and of those I have, only a few are profound.

I would like to list some of the folks whose ideas I have borrowed, restated and maybe extended a small bit. They are the real source of what I say. They are not in order of precedence, just random order as I thought of them.

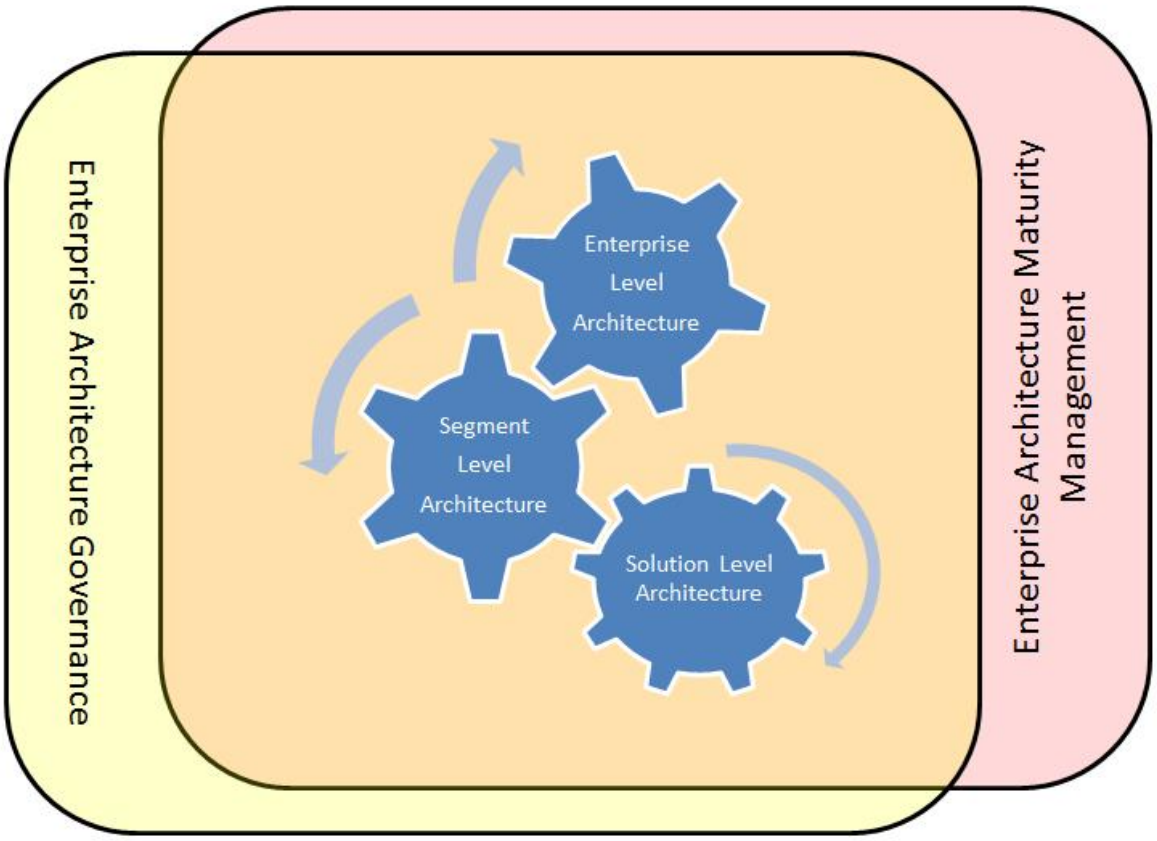
- John A. Zachman, whose ontology I use to think about architecture
- Kathie Sowell, the "mother" of DODAF, who ran the project at MITRE
- Mike Tieman, who rewrote FEAF 1.1 and whose thinking I admire greatly

- Felix Rausch and Beryl Bellman who made it possible for me to get an education in enterprise architecture
- Lee Smith, the first Chief Architect at DHS and lion tamer, from whom I learned many lessons concerning governance
- Rob Thomas, who wrote so much of the early material for FEA and FEAF. I met him recently, a great man.
- Manny DeVera, contributor to FEAF 1.1 and a great guy. I worked with Manny a bit at the Federal Emergency Management Agency (FEMA).
- Ira Grossman, who popularized and supported EA for years. I met him on EDMICS and worked with him a bit at FEMA
- Dr. Scott Bernard, a brilliant guy. He was the brains behind FEAF II. Currently Chief Architect at OMB.
- Bobby Jones, who can sell enterprise architecture like no one else. I worked for Bobby at FEMA for a bit.
- Stephen Spewak, who died early and who I never met. I regret that.
- Bradford Rigdon, who chaired the panel for NIST and whose team first used the term “enterprise architecture" defining it by context. I never met him either.
- Richard Burk, former Chief Architect at OMB and a great guy. I quote him often.
- Kshemendra Paul, another former Chief Architect at OMB. I sat in class with him at FEAC, a brilliant guy.
- Randy Hite, who worked tirelessly at the US Government Accountability Office (GAO) for so many years and wrote the Enterprise Architecture Management Maturity Model (EAMMF), a monument to good government.
- Skip Boeteger, my sounding board and more senior colleague. We share neurons, I think.
- I have to add John Tieso. I just saw him at the Business Process Management (BPM) conference, which reminded me. I sometimes forget he is a great architect, as he pretty much agrees with me on nearly everything. I suppose, ironically, that if I had a bigger ego I would remember John is a great architect more often.

It strikes me, having written the list, how many of these folks are friends. Also, of those remaining, how many I wish I knew better. Surely I left some out, and I will have to edit them in. (If I did leave you out, it was probably simply my brain misfiring.)

Regardless, they thought the profound thoughts and I followed. I hope I was a good student. If you like what I have said, seek these folks out. I learned from them.

1.2 THE FIVE ACTIVITIES OF ENTERPRISE ARCHITECTURE, APR 25, 2015



Enterprise Architecture is pretty darned simple if you have a good model to explain it. Without a good model everyone starts arguing. In 1989 the Federal Enterprise Architecture Framework (FEAF) described a three-layer model that was pretty simple. In 2006 Burk at OMB described it even more clearly. Three layers—easy.

People talked about two other important activities, governance and process improvement of EA. I wrote a paper and added those a few years ago. Five activities—simple.

So for just a moment ignore those with partial views, axes to grind or strange garage-grown frameworks and let me explain the five simple activities of a complete enterprise architecture effort.

ENTERPRISE LEVEL ARCHITECTURE

The term enterprise refers to either a whole organization or some hard effort. Here we refer to the whole organization. This has nothing to do with coding information systems or other details, and everything to do with keeping an inventory of all the important features in the enterprise to be transformed and updating this based on plans. It also involves keeping a schedule (roadmap, transition plan) for efforts to change the enterprise. This level supports the portfolio management

efforts encompassing all transformational investments (programs, projects) in the enterprise—all of them.

SEGMENT LEVEL ARCHITECTURE

A "Line of Business" can be a product line or a line of services or some mix. A "Segment" is either one of those or some other large internal effort used across a wide range of lines. The segment architecture describes things like the customers (or stakeholders), the value chain, the logistics chain, the distribution chain, the production line, etc. These usually correspond to a "Program" and so segment architecture usually supports "Program Management." Most importantly, this activity must propose the business cases for the improvements to be funded in the portfolio and implemented in real projects.

SOLUTION LEVEL ARCHITECTURE

The solution is some system created to effect transformation. In changing the organization or the line of business, something must often be automated, centralized, decentralized, constructed, moved or otherwise revamped. The solution architecture describes how that is built, moved, changed etc. Each such thing is a system and a project to be completed.

ENTERPRISE ARCHITECTURE GOVERNANCE

To make all this work you must have a governance structure to tie architecture with real implementation. Otherwise all the stray cats go their own way. You need at least three levels: to approve the portfolio decisions, to approve the business cases to change the segments, and to approve the changes in the solutions (systems).

ENTERPRISE ARCHITECTURE MATURITY MANAGEMENT

All these things are processes. The organization needs some means to manage, standardize and improve these processes.

CONCLUSION

The Five Activities Model is a simple way to understand enterprise architecture.

1.3 ENTERPRISE, SEGMENT, SOLUTION, JAN 14, 2015

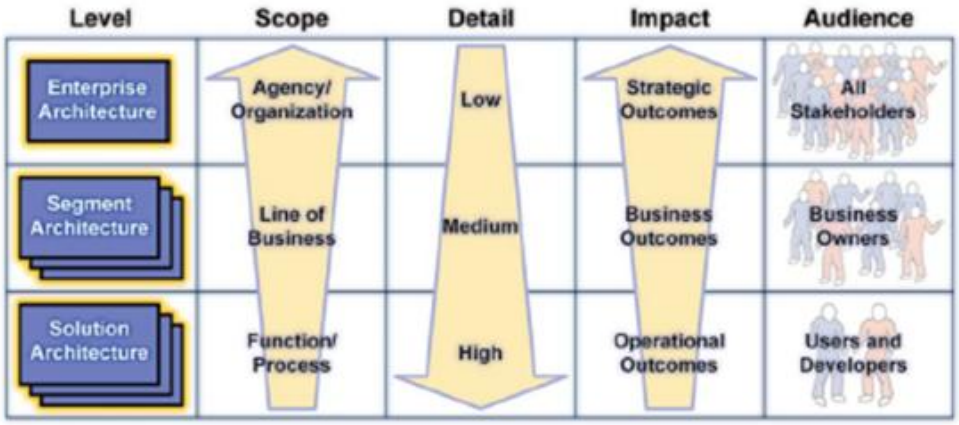


FIGURE 1 THREE LEVELS OF ARCHITECTURE FROM 2007 FEA PRACTICE GUIDANCE OF US GOVERNMENT OMB

This section will describe the basic differences in the three levels of architecture presented first in early material on the FEAF and FEA. The three levels were again described by Burk in the 2007 and 2008 FEA Practice Guidance. This model is extremely important in differentiating the types of work in architecture and minimizing redundancy of effort.

The descriptions here are based in part on my own understanding of architecture and experience. For other views, you might check FEA practice guidance, the early FEA documents on establishing enterprise architecture and FEAF v1.1.

I find that this old material is poorly understood outside DC. Even in DC, some practitioners have an inadequate understanding due to lack of education or training. Consequences of mashing the levels together with fuzzy thought processes include less effective architecture, reduced cost effectiveness, poor clarity, redundancy and excess work. Therefore, I find this material important to all practitioners.

ENTERPRISE LEVEL

As shown in the accompanying image, the enterprise level of architecture is intended to be shallow but broad. The intended scope is the entire enterprise—the complete agency, department, corporation or whatever you are charged with. It is focused on strategic outcomes based on strategic planning. The process of ensuring investments and architecture support strategy is called alignment.

The enterprise level of architecture supports the choice of transformation investments. In the Project Management Body of Knowledge (PMBOK) this activity is described as portfolio management, and in the US Government it is called Capital Planning and Investment Control (CPIC). Transformation expenditures in EA are treated as investments and are expected to produce a return on investment (ROI). Comparative management of investments ensures high ROI and controlled risk.

To minimize unneeded depth (detail) and maximize utility, simple inventories of the major elements of the enterprise are kept at the enterprise level (composition). Relationships between elements of the inventories are kept (structure) to understand the effects of change. By comparing the current enterprise to the target enterprise (the composition and structure after investments are applied), you can determine remaining gaps.

Other than inventory lists and their relationships, the main artifact at the enterprise level is the transition plan or roadmap, a schedule of initiation and completion of each investment leading to the target state. (One good practice is to include stage gates, color coded for systems development life cycle (SDLC) stages and initial operating capability/full operating capability (IOC/FOC) on the investment lines of this schedule.) Artifacts should not include the types listed for lower levels of architecture, as these would be redundant, unless a clear need exists.

Vision, standards, principles and other guidance are commonly produced at this level for consumption by the levels below.

For more on a minimalist approach to the enterprise level see:
https://www.linkedin.com/pulse/20140727145732-86002769-very-lean-enterprise-architecture?trk=mp-reader-card

SEGMENT LEVEL

The segment level of architecture is less broad and more detailed than the enterprise level. It is also wider and less deep than solution architecture. The segment level is focused on the operational mission and on operational plans. One primary purpose of this level is to produce the business plans that propose new transformation investments (to be reviewed and selected at higher levels). The segment level also introduces customer focus and ensures individual systems add value to the operations.

The segment level describes lines of business. This would include the products that compose a product line or the services that compose a line of services. It would also include the mix of products and services in a line of business.

The governance body that most often appears at this level is that which makes stage-gate review decisions, which oversees lower systems engineering and subsumed solution architecture. Some SDLC context is often applied.

Coverage might include the supply chain, the manufacturing line, the value chain, the distribution chain, markets and customers. Segment architecture is best when focused on the value delivered to the customer, or in government, the value delivered to the citizen.

Various operational diagrams are the main artifacts of segment architecture. The value chain diagram is of particular note. The key artifacts do not include redundant listed inventories of what exists in the enterprise nor roadmaps.

Three kinds of segments are often described. The first includes all the mission segments, a.k.a. the core business of the organization. The second category includes all support operations, such as human resources. The third includes any internal initiatives to provide a common resource to the organization, such as an enterprise service bus (ESB) in IT, or a fleet of cars and trucks for non-IT.

Each may have many component solutions implemented as projects, so the segment level can be said to correspond to the program level in the PMBOK.

For more on a customer focused approach to Segment Architecture see: https://www.linkedin.com/pulse/20140727163249-86002769-customer-centric-enterprise-architecture?trk=mp-reader-card

SOLUTION LEVEL

The solution level of architecture describes the particular detailed implementation plans of one project or investment. Often this may be produce by the contracted company implementing the plan, unlike the levels above. Only at this level is discussion of servers, virtual servers, programming languages and features (Struts, Hibernate, etc…) is appropriate.

The governance body most often associated with this level of architecture is the CCB (configuration control board). Decisions supported by this level of architecture are considered tactical in the enterprise context.

A wide range of artifacts are possible at the solution level describing operations, business process, databases, software structure or service-oriented architecture (SOA) services, component applications, ESBs, and other such details of implementation. These artifacts should not be redundantly reproduced at higher levels. Higher level artifacts are commonly referenced.

Any solution exists within the context of improvement of a segment.

INTEGRATION

In the enterprise repository the segment artifacts are commonly attached to the enterprise. The solution artifacts are attached to the segment in the EA repository as well. Segment artifacts are filed and kept together by some mechanism in the repository, as are solutions. Repository tools such as Troux Architect™ are designed to do precisely these things.

In a medium or large enterprise, different teams may produce different instances at different levels of architecture. The enterprise level is most often reserved for the employees of the enterprise. This solution level is often contracted out with the solution, producing innovation and other advantages. Guidance on these different levels helps to streamline these distributed efforts.

TERMINOLOGY

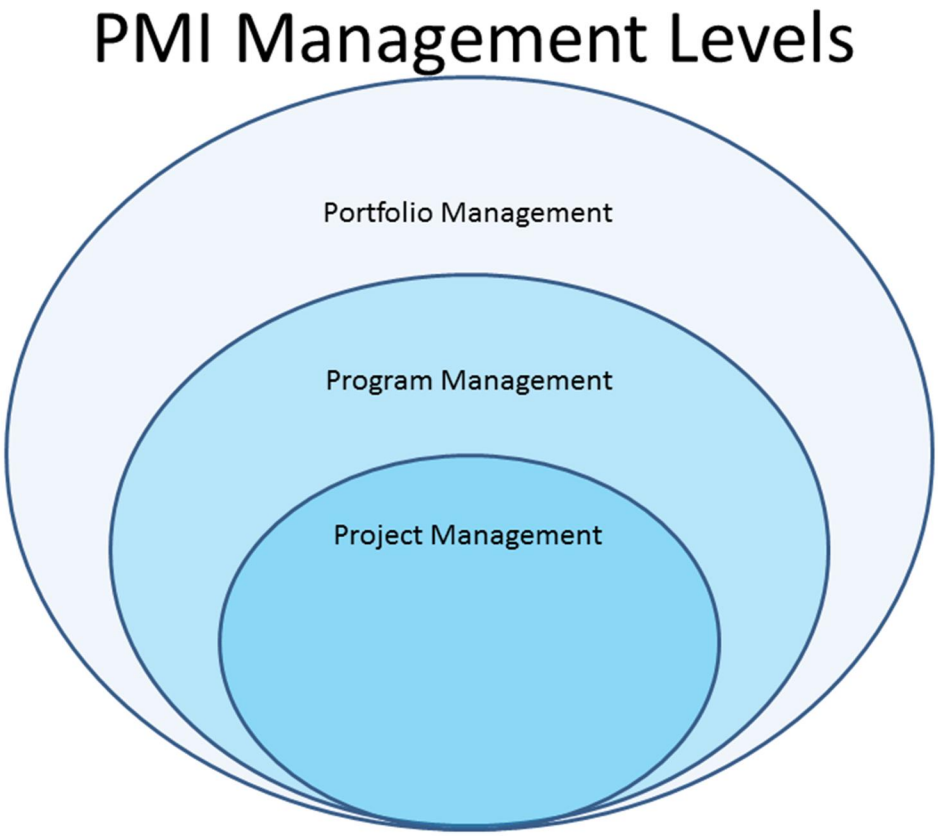
In a loose sense all three levels are referred to as enterprise architecture. In a strict sense, only the enterprise level is included in that term.

I have written a paper connecting this 15-plus-year-old model with the two other most common activities, EA governance and EA internal practice maturity. You can find that here:

http://www.unauthorizedprogress.com/images/EA_as_5_activities_2014.pdf

I hope this helps.

1.4 CORRESPONDENCE TO MANAGEMENT ACTIVITIES, NEVER POSTED



In the previous section, we saw the three levels of enterprise architecture described by Burk at OMB in 2006 and originally introduced in the early FEAF circa 1999. These have a direct correspondence to the three levels of management described by the Project Management Institute™ (PMI) that are the fundamental subject matter of the PMP™ certification.

PORTFOLIO MANAGEMENT

The portfolio management activity is described by PMI as responsible for business leadership, alignment, value orientation, program selection and portfolio adjustment. This corresponds closely to the enterprise architecture activities described in US OMB Circular A-130 and described by OMB/Burk in 2006.

PROGRAM MANAGEMENT

PMI describes the program management activity as providing business sponsorship, ownership of benefits, benefit streams, comprehensive ownership of the business system, and multiple projects. This corresponds well to descriptions of segment architecture and the improvement activities of a line of business.

PROJECT MANAGEMENT

Project management is described by PMI as providing delivery of capabilities, budget and schedule. Solution architecture as commonly describes support those items. This includes descriptions by Burk at OMB.

MERGING

Architecture Level	Scope	Detail	Impact	Audience	Customer Management Level
Enterprise Architecture	Corporation or Agency	Low	Strategic Outcomes	All Stakeholders	Portfolio Management
Segment Architecture	Line of Business	Medium	Business Outcomes	Business Owners	Program Management
Solution Architecture	Function or Process	High	Operational Outcomes	Users and Developers	Project Management

If we merge the information in the OMB table (previous section) with the PMI information, we get a table like the one above. It implies that the enterprise architecture activity supports the portfolio management activity and the portfolio manager. Further the segment architecture activity supports the program manager. Lastly the solution architecture activity supports the project manager.

While these are not absolute rules embedded in any law or policy, they seem to be important guidelines.

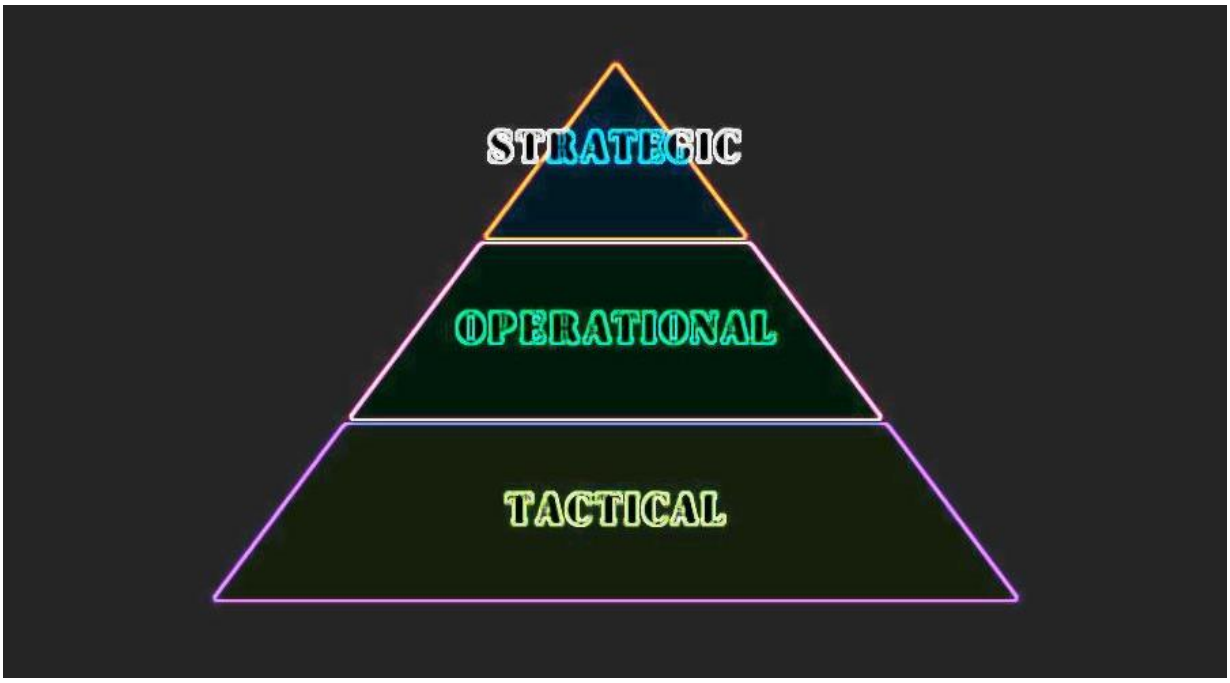
MEASURING SUCCESS

Customer Management Level	Architecture Level	Scope	Detail	Outcome Measures
Portfolio Management	Enterprise Architecture	Corporation or Agency	Low	Strategic Goals & Objectives
Program Management	Segment Architecture	Line of Business	Medium	Organizational Performance Measures & KPIs
Project Management	Solution Architecture	Function or Process	High	MOPEs, MOPs and KPPs (INCOSE)

If we add common material on how success is measured to the appropriate levels, we get the table shown above. The correspondences here are rough, as there is overlap between the system engineering measures of the International Council on Systems Engineering (INCOSE) and the common business nomenclature of performance indicators and the Key Performance Indicator (KPI) versus the INCOSE Measure Of Effectiveness (MOE).

Ignoring the semantic overlap, there appears to be a hierarchy of measures that can be used to measure the success of management and architecture.

1.5 STRATEGIC, OPERATIONAL AND TACTICAL THINKING, JUL 19, 2014



Are you really a strategic thinker? Is your plan really strategy? Probably not, based on the predominance of mislabeled plans and concepts: Many use the word "strategic" as a synonym for "important." While strategy is widely acknowledged to be important, the words are not synonymous. Those who misunderstand the term or who misuse it are unlikely to produce strategy.

STRATEGY

Both time-frame and scope are associated with strategy. If it affects the entire organization and covers a period of years, it may be strategy. Examples of strategy include what markets you will compete in and which you will exit; what are your competitive advantages shared across the company; where you will invest in capacity and where you will divest; and the fundamental purpose (mission, goals) of your company or organization.

TACTICS

Tactics involve point approaches to local problems or situations. Tactics may be reusable for a common problem or situation. Tactics are usually rapid compared to strategy and do not describe activities covering years before fruition. Examples of tactics are Standard Operating Procedures (SOPs) and choice of and purchases or acquisitions of services or products.

OPERATIONAL THINKING

This lies between tactics and strategy, affecting perhaps an entire product line but not the organization or the business processes used repeatedly and changes to them. Examples may include new features or improved performance of a product or a single line of products among many.

MISUSE

Now let's examine some common misuse of the term "strategy." A vendor wants you to have a "mobile strategy." This may well be strategic to the vendor, who sells mobile services or devices, but it is not about your market positioning, your markets, or your major investment areas. It is at best operational to you and perhaps tactical. Everyone is using this stuff; there is no competitive advantage.

Six Sigma or Lean or Agile are said to be strategic and may provide competitive advantage. Adoption and implementation of these may rise to a strategic goal to provide competitive advantage, but once adopted these are operational issues.

STRATEGY FORMAT

In the US Government it has become common to create recurring yearly strategies in the form of a list of broad goals, subdivided into concrete objectives, perhaps associated with some performance measures. Supporting policies are often not included (perhaps due to the complexity of their approval). Sequences of actions are left to operational plans. In commercial use a strategic plan may commonly include all three and are more often confined to a single issue.

CONCLUSION

Perhaps the quickest way to indicate your strategic irrelevance is to improperly indicate that your tactical advantage is strategy. Those trained in strategy can spot the difference. Try to use the terms correctly and you may be better respected by your audience. If you are a CxO or vendor, or anyone between, misuse of the term "strategy" will likely hurt you more than helping you.

1.6 WHY DO ARCHITECTURE?, JUL 5, 2015

I have left out one topic. I am remiss. This particular topic mystifies me, not the answer itself but all the controversy.

I started out my paid career as a technician. I would repair and maintain complex electronic systems designed by others, much of it past the end of its predicted product life. To do this I needed drawings, lists, matrices and documents which are the artifacts of engineering. Without these the complex and expensive systems I was charged to maintain would remain broken as I could not figure out what might be wrong, I could not troubleshoot.

> No manager was both so bold and so dim as to suggest that engineering documentation was not needed for my function.

Later, I was an electronics design engineer and manager of engineering. I was charged with producing prototypes and their associated drawings, lists, matrices and documents. Without these, the production engineering function would have nothing to optimize, and technicians no means to repair. The repair function slowly changed, as all electronics became commodity and broken items were wastefully thrown away.

Electronics design moved to the Pacific Rim and I left it behind. No matter how good I was, the exchange rate (Dollar to Yen etc.) became so unfavorable that they could hire multiple engineers there to replace one of me, and for the same money.

> Now managers sometimes spoke of outsourcing all design and the provider keeping the documentation.

I began to work on system integration. I would design systems, do site surveys, and produce drawings, lists, matrices and documents. I was paid to do this. Without such documents the large complex systems we produced could not be implemented, fielded. We had to build them and make them work, then repair and maintain them. I had moved from single boards and units to entire groups of computers and peripherals as building blocks, but the artifacts of design remained the same.

> No manager responsible for design and fielding of such complex systems argued that the artifacts of design were not needed. But customers who had outsourced the function were sometimes uninterested in those artifacts.

In the latest leg of my career I have managed the enterprise, often working for the customer. Here there are many large complex systems, which may be interconnected. Those involved in building and interconnecting such systems produce artifacts such as drawings, lists, matrices and documents, to manage the design, construction, fielding and maintenance of the systems and their interconnection. The customer is less interested in the artifacts, so long as the provider is doing the job.

> Managers today routinely question the need for design artifacts regarding the enterprise.

So I can tell you these experiential facts and opinions from my career:

Opinion: Architecture and design are mostly synonymous, although some academic minds insist on slicing them apart via tiny nuance.

Fact (by observation): If you are directly responsible for designing, building, fielding and maintaining complex equipment or systems and competent in doing so, you will use drawings, lists, matrices and documents. You will not often question their use if you produce more than a prototype.

Fact (by observation): If you are a customer who has outsourced these functions you will not care as much. So long as the provider has done their job you may not care at all.

Fact (by observation): At some point in the lifecycle of complex equipment or systems you will need to upgrade, modify, maintain or otherwise change them. If they are sufficiently complex, and the original designers have long gone, you will suddenly realize that you need those artifacts (drawings, lists, matrices and documents). You will be unable to meet customer needs without them, and you will be out of business with your present technology. On that day you will become aware that you needed architecture or design documentation.

Fact (by observation): This can all be mitigated if you intentionally plan to throw away all you have and start from scratch every few years. (Zachman refers to this as "scrap and rework".)

Zachman identifies two key factors: Complexity and Change. As complexity increases, scribbling notes on whiteboards and post-it notes will fail as engineering methods. When change occurs you will be unable to use your documentation to manage it. This is true regardless of scale.

Complexity arises from producing systems and equipment to aid in achieving complex goals and objectives, BTW. You produce sophisticated tools to support sophisticated use. Excellence, quality, or cost-effectiveness require some sophistication.

So, as a manager of a complex system or equipment you may have a short-term view and expect to move on before the disaster strikes. You may then argue against engineering documentation, as it is an expense for an item you will not require before transferring the obligation elsewhere. It will not be needed if all your technology and its configuration is commodity to be thrown away repeatedly.

You do architecture or design of an item not for its own sake. You do it so that the item, be it process or box, supports effort to produce product or service. That way you make money, for example. You initially produce it so that it exists at all, then later so it is better. You will stay ahead of competition that way. You want to produce that process or service with lower cost, higher quality, or higher throughput. These are outcomes, to produce the product or service with higher throughput, quality or lower cost. To simply produce, it is a capability.

If you are the customer, buying the product or service, you may discard the documentation as if it were a microwave oven or dishwasher in your home. You will assume the manufacturer has a copy. However if it was custom built for you they may not keep your documentation. If you fire the first vendor, the second vendor will need the documentation. If your enterprise falls apart after the last person in charge has left, the new person may also desire documentation. If the mission or market changes, you will want the documentation. When you cannot serve the mission or the customer any more with the undocumented mess that has been left behind, you will realize you need the documentation.

If you do not believe these drawings, lists, matrices and documents are required for your large and very complex project, system, equipment or enterprise then I request you do one of the following:

Go work for my company's competitors or hostile nations and apply your views there;

Get out of the way of those doing required work, even if you do not understand it, as your presence is not constructive;

Stick around long enough to take responsibility for your lack of foresight;

Take up used-car sales or tele-evangelism, go far away and quit playing with managing technology before you damage something.

1.7 INTRODUCING ARCHITECTURE TO YOUR ORGANIZATION, JUNE 7, 2015

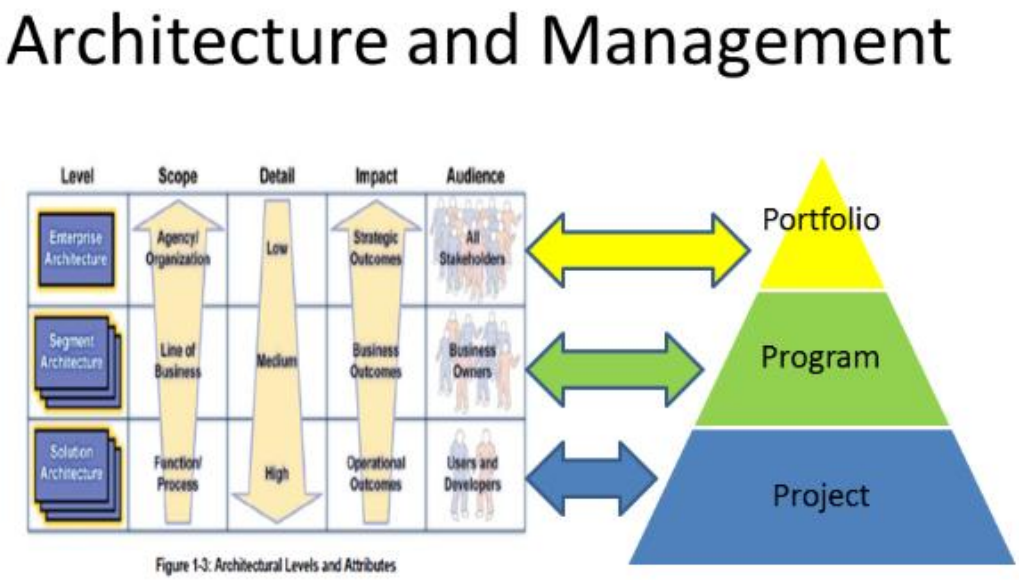


Figure 1-3: Architectural Levels and Attributes

Have you heard the story of the Emperor of China asking the great sage Lao Tzu to explain all the secrets of the universe, while the emperor stood on one foot? An impatient and impetuous young emperor thinking it was all that simple is the point of the story. I was recently asked how to implement architecture in a huge contract gone rogue, with many systems thrown together at the last minute to meet deadlines. They seemed to want a simple formula.

Let's assume you have no architecture practice, which is probably untrue and people are probably building architecture in little dark corners because they have to, but let our assumption stand. In my mind, the first step to introducing architecture to your organization is to understand that there is not one architecture. I have listed elsewhere the five activities within enterprise architecture, and so near as I can tell they have different purposes and processes, and implementing each is different.

ENTERPRISE LEVEL ARCHITECTURE

The enterprise level architecture practice supports portfolio management of transformation investments, it has a yearly cycle, following the budget. Its main risk is "boiling the ocean" or trying to do too much architecture centrally. Burk described this level as most strategic, broad but shallow.

Here the term "enterprise" is specifically applied to the whole organization. I am not using the broad meaning, all of this stuff, at all levels, as that would not aid description.

Oddly the startup of this level may be easiest to describe. Rob Thomas II and company described this well for the Federal Government in a good document: A Practical Guide to Federal Enterprise Architecture. Read that.

SEGMENT (LINE OF BUSINESS) LEVEL ARCHITECTURE

Someone must first divide the enterprise into lines of business, and cross-cutting efforts. This must be done at the enterprise level, and everyone must agree to the divisions. Then you can kick off each one of these efforts, associated with an overall transformation program for each.

The segment or LOB architecture serves the program. A clear, 1 to 1 relationship is best. To do this well I suggest you ignore the FSAM and DSAM of the US Federal Government and instead look to the "Mission Architecture" or "operational architecture" efforts of DoD in DODAF. They do this well. An example can be found here: http://www.rand.org/content/dam/rand/pubs/research_reports/RR200/RR261/RAND_RR261.pdf

Segment or LOB architecture effort should produce business cases proposing needed operational improvements. This level of architecture has moderate breadth, and moderate depth. As for how to proceed, the FSAM, the DSAM and DoDAF all have processes for that. There will be one to many segments in an organization.

SYSTEM OR SOLUTION LEVEL

Systems are oddly problematic, and you have to choose boundaries. Once you do, you can define them through architecture. This kind of architecture serves a project intended to construct the system and make it operational. The architecture then supports maintenance and operations. There may be many systems in a segment or LOB. Not all may need the same level of architectural documentation.

How do you proceed? First get an SDLC or equivalent like PMBOK or the TOGAF ADM (first used in TAFIM years ago to describe system construction). Then follow that. Stop by INCOSE in your search.

ARCHITECTURE GOVERNANCE

To institute architecture governance is a bit tricky. It must fit into your corporate governance, or customer governance. The main problem in this is reigning in runaway governance, as everybody wants the power to control something. Go for simplicity. The governance glues the levels above together. Have a look at my post here for a minimalist viewpoint. Here is a guy with a 5 point approach. You can find advice all over the web, but try the early FEA documents to understand how to get started related to architecture.

Starting the architecture above and the governance that uses it is a "chicken and egg" problem. Start somewhere and keep moving. Do not wait for all conditions to be perfect.

MATURITY MANAGEMENT

How do you institute process improvement of architecture itself? You need an independent process. Look here for some guidance. There will be audits, self assessments, analysis, and process improvement. (You can do it yearly in the lull between budget cycle support at the enterprise level.)

GENERAL RULES

Here are some general rules:

Architecture is lists, matrices, drawings, documents. Do lists first, documents last to the extent you can.

FEAF v1 points out that in architecture, business drives data drives systems drives technology and infrastructure. The original NIST model puts middleware and data exchange between business and systems, and databases between systems and platform. Follow advice like this as to what precedes what.

Zachman often says don't do all the architecture for everything, do it as needed. Do that.

Do architecture at the lowest level that makes sense. The upper levels drive the lower, but are shallower. The upper levels only become shallower at the top by doing architecture at the lowest level reasonably.

STAFFING

At the enterprise level the basic FEAF v1 team has about 6 persons. You have 1 for the inventory of business functions (top level processes), another for the list of data assets, another for the list of systems, another for the standards and approved products, a performance architect and a chief. Tiny organizations may need less. You can hit ten if you do some other sophisticated things, like governance and maturity management and performance management. If you need far more than that you may be "boiling the ocean".

At the segment or LOB level you will have a large team at first, and then less later. This can be 20+ people to describe the architecture in terms of production line, supply chain, distribution chain, value chain, product mix and more. Choose what you need but do not skimp on the initial staffing. The ROI is high here, and this affects your core businesses.

A system can often have one architect. Several systems may share one. Large database centered enterprise software may have 2 or 3 (system, software, database). Unless you have a system of systems, or SOA environment, or some such, more is not advisable; and if you do have SOA or ESB or SOS it is probably a segment and you screwed up the system boundaries. (All the ESB, EAI, ETL, data-warehouse and DataMart efforts are best mixed into a single program for several reasons.)

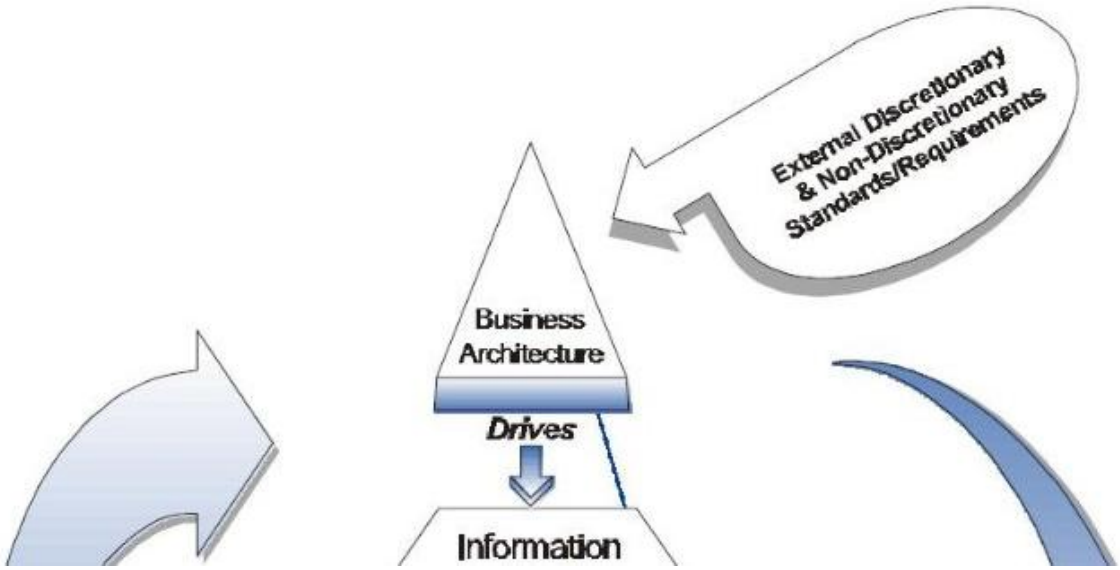
Somewhere, above all this effort, you need a chief architect or chief systems engineer to sort it out-maybe both. Don't skimp on this guy.

If you have all these folks and more, and nothing is getting done, the organization and scopes and authorities are not clear. It’s a management problem, not an architecture problem. Fix it.

CONCLUSION

Initiating an architecture practice in your organization is not a single, simple formula.

1.8 EA, BA, DA, CARTS AND HORSES, JUNE 28, 2015



Recently I saw a presentation from a BPM guru that completely mischaracterized EA as IT only architecture. The ignorant presenter was also not aware that Business Architecture is a component of Enterprise Architecture. This rendered the presentation worthless.

I have seen Business Architecture create a body of knowledge emphasizing such misinformation, and organizations elevating business architecture outside of Enterprise Architecture. Such a huge step backwards will not create a seamless application of technology to the mission, or in support of strategy.

> Separating BA from EA is not constructive to the management of either technology or modern business.

Data Architecture has also been oft promoted to outside of Enterprise Architecture. This also rewrites history, and works against a unified architecture.

The value of architecture will not be achieved by attempts to split it into fiefdoms. Holism is key. Examples of splitting out this and that component and giving it dominant political power or supremacy over EA holism are also examples of bad management.

If you want real results, keep the components of Enterprise Architecture in their context and relative position within it. The relationship was described in the first EA document (NIST SP 500-167), and in the early FEAF, and is implicit in DODAF, etc. Do not mistake political power grabs with effective management or effective architecture.

1.9 ACHIEVING SUCCESS, APRIL 12, 2015

Success is one of those words that loses meaning without a context. People argue endlessly regarding what success means, and it is meaningless without the key. This post is for younger folks and anyone else who do not have that key.

I don't want to string you along: There is no success without a goal. When you achieve your goal you have success. There, the mystery is gone. You can look at me, and plenty of old guys, and say "he is no billionaire, he is not successful". Hey, that was not my goal. If that is your goal, the same approach will work. Go for it.

To make your life successful, follow the same advice that generations in the past have followed. Here it is: use the planning cycle. It's not obsolete. It's not irrelevant, because our time is full of chaos and turmoil. It's not optional. It's not a peripheral part of the life of the driven and successful. It is a core activity if you want to succeed.

Let me describe the cycle you use for success:

- Set (or adjust) a goal or goals. Pick carefully. You are going to achieve these if you pick obtainable goals, and you need to appreciate them once you have attained them. These goals must reflect who you are, your values, what has meaning to you - as they will be all that for you in the end.
- Make plans. Include contingencies, risk mitigations. The harder the goal the better the plan must be. Make plans that are real, concrete, and attainable. Do not be timid. Be objective.
- Execute the plan. Be brave. Stick to it. Never give up. Work hard. Ignore distractions.
- Evaluate and consolidate. Check where you are versus your goals. Evaluate the effectiveness of your plan. Be brutally honest. Be factual. Then go back to the top.

Pick something like a yearly schedule for the cycle. There are other descriptions of the cycle, most are fine. Now go do it.

Here is the hard part of this advice: Others who have done this tell me that there are very few things in life more important to your success. They say learning to read well, study, write well and speak well may be such a thing, but it can be achieved by this cycle. They tell me you can be tall or short, red or green, LGBT or straight, woman or man, Catholic or agnostic, born poor or born rich, big or small, ugly or good looking, and this approach will work anyway. So do not whine and complain about whatever disadvantage life has given you, we all have disadvantages and things to overcome. Shut up and get moving. Stop waiting for someone else to fix your life, it's yours to fix by achieving goals. If someone does help you, be grateful but do not expect it or rely on it.

Some may have achieved more this way, some a bit less perhaps, but still they achieved success. This advice works for almost anyone, as well as for companies and government organizations. It applies to whole countries and civilizations. If you have no goals you are unlikely to achieve them. If you do, you have a good chance.

SECTION 2: TECHNICAL LEADERSHIP

2.1 MANAGEMENT VS. LEADERSHIP, MARCH 15, 2015

There is a natural tension between management and leadership. It is hard to do both at once. It raises conflicts of interest. It confuses the issues.

FULL DEFINITION OF MANAGEMENT

1: THE ACT OR ART OF MANAGING : THE CONDUCTING OR SUPERVISING OF SOMETHING (AS A BUSINESS) 2: JUDICIOUS USE OF MEANS TO ACCOMPLISH AN END 3: THE COLLECTIVE BODY OF THOSE WHO MANAGE OR DIRECT AN ENTERPRISE

From http://www.merriam-webster.com/dictionary/management

FULL DEFINITION OF LEADERSHIP

1: THE OFFICE OR POSITION OF A LEADER 2: CAPACITY TO LEAD 3: THE ACT OR AN INSTANCE OFLEADING 4: LEADERS <THE PARTY leadership>

From: http://www.merriam-webster.com/dictionary/leadership

WHO DO YOU REPRESENT?

When you act as manager, controlling and directing resources, you act for the business or organization. You seek to achieve their goals, follow their policies. But when you act as a leader, you represent the team or group you lead. You seek to organically motivate the team, the individuals to do what the team has decided in the way they have decided. Instead of telling people to comply with a plan or be fired, you now listen and seek progress based on the individual strengths and motivations of members.

> Mixing the two is a sort of compromise, a conflict, a corruption- and you will sometimes know it in your gut.

PROJECT MANAGEMENT

The modern corporation, company or organization has many policies designed to restrict and limit individuals, to disadvantage them. Most employees or members are hard pressed to find motivation or agreement to achieve goals in the presence of the naked corporate structure and positions. It is up to a project manager, as an example, to cause these folks to achieve goals on schedule. The PM may have to bend or break rules to humanize the organization enough to achieve anything. The organization will have no responsibility to back these compromises. He or she must be both leader and manager and exceed authorized limits and ethical practices to succeed.

SUSTAINABLE ORGANIZATIONS:

Attempts to rectify this contemporary bureaucratic nonsense include trying to introduce culture to the corporation, which often creates something like a cult with suits. There are also team based methods incorporated into production, which is like fruit growing from a thorn bush. There is talk of corporations replacing nations, sort of a fascism without a land or tradition. What is unsustainable will not be sustained. Some speak of eliminating the employee relationship, and destroying social institutions that what keep us from anarchy. Something will eventually change, but in the meantime the line manager is caught in a conundrum, and at the pivot point of historic change.

…AND?

Good luck with that. For what its worth, here is my advice: Only your individual integrity can protect you from these forces of the times. Only traditional societal values can guide you between the landmines. Know yourself before you try to manage or lead others.

22 POWER, LEADERSHIP, MANAGEMENT & PROFESSION, JULY 19, 2015

Many spend their lives seeking power. Many others are aware of it and take it as a fact of life. Some have examined it more closely. Some have spent their lives examining it. Early in my life I came to grips with the topic after some study and contemplation, due to various contemporary circumstances. Some of you may not have had that experience.

Power relates to leadership and management, even technical leadership. I will write about it briefly here, in summary only. My views here may reasonably be described as a bit traditional or "old school". These are only my own opinions, although perhaps shared by others and written about over generations in various similar forms. I hope they may be of use to you.

Some say power may be divided into two types. The first is power over others, and the second is power over yourself. Others do not divide it at all, some speaking only of power over others and ignoring the second. Some have divided it three ways. I will use the common division into two parts, and then I will relate it to management and leadership.

POWER OVER OTHERS

Power is often defined, without regard to another type of it, simply as the ability to control or influence others. To attain such power you control or use something the other party needs: water, food, money, drugs, approval, almost anything. By withholding it (in fact or by implication) until the other person gives you what you want you can exercise power. This type of power is widely discussed and studied.

Power over others has an effect on its wielder. It is seductive, addictive, and it disempowers you by habituating your dependency on others. Psychologists point out that feeling powerful in this way decreases the mirror system of neural activity responsible for sympathy or empathy, you become less connected. The old adage has it that power corrupts, speaking of power over others.

In a real and physiological sense, a psychological sense, an ethical and moral sense, this power over others can be seen as an illusion diminishing yourself(Power over others contains a kind of weakness. Further such pursuits, if excessive, divert your efforts from other goals and efforts. Power over others is a fact of life, it cannot be avoided, but it has its costs.

POWER OVER YOURSELF

Power over yourself comes from effort. You must understand yourself, develop self control, develop skill, develop your abilities, reduce your unhealthy dependencies and needs. This kind of power is the cornerstone of various lifestyles, archetypical life patterns, philosophies, and religions. It is a lifelong pursuit, requiring dedication and hard work. The idea is that if you improve your skills and abilities you can do more to help yourself and others. Conversely, if you have no skills or abilities you can do little for anyone.

This type of power is also widely discussed and studied.

MANAGEMENT AND POWER

Management of people is to wield power over others to a purposeful end, a defined goal, often one of benefit to some organization or higher manager. Management must remain circumspect with regard to wielding the power given or may loose judgement and violate ethics. Management ethics is an important and recurring topic. The use of power to manage is also widely discussed and studied. Be careful.

LEADERSHIP AND POWER

Leadership is not management. To lead is to reflect the group welfare of followers, not the advantage and benefit of the more powerful few. Management and leadership can only coexist if the arrangement is beneficial for both the powerful and the follower. Those who are enthralled with or addicted by power over others are not generally suitable as leaders.

Examining this allows one to understand the old saying that good leaders do not desire to lead, do not desire followers. Look for that.

THE PROFESSIONAL

The doctor, lawyer, engineer, priest or reverend, following a profession or professional ethics, seek power over themselves. They seek to increase their own skill and ability to aid others and themselves. It is antithetical to for such professionals to seek power over others. From such a viewpoint, the term "professional manager" can be a bit of an oxymoron, or at least problematic in definition and execution. There is a conflict. The professional may hold power over others in some disdain, a thing to be used only when required.

IN SOME SENSE TO HAVE A PROFESSION IS TO BE CALLED TO A PATH OF IMPROVING YOUR OWN SELF, TO DEVELOP SOME SKILL OR ABILITY IN YOURSELF.

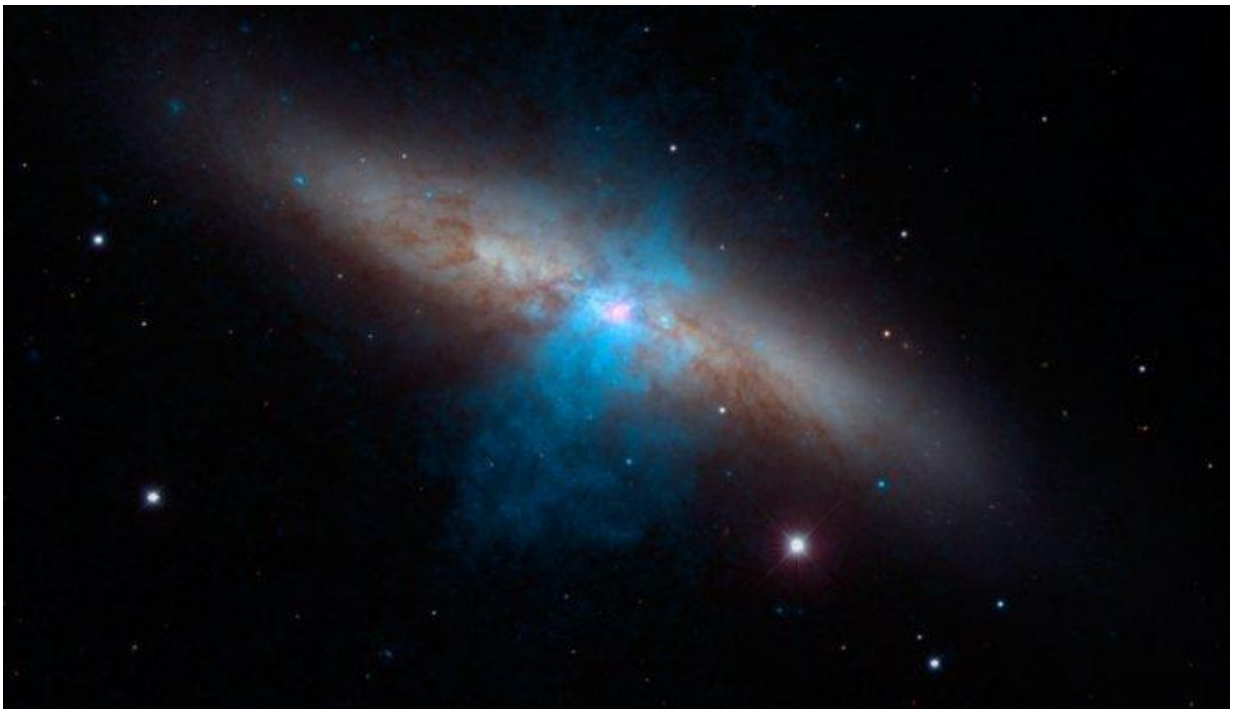
CONCLUSION

To be a professional and a leader is a natural combination. To be a manger and a leader can only occur in circumstances advantageous to all parties. To be a professional and a manager is also limited to such circumstances. These limitations occur via the use of power. Ethics governs if you can be two at once.

There you have it, some "old school" views about power and its use in relation to some ordinary traditional values. I do not claim to be a paragon or great example of leadership, management, power over myself or power over others. I do seek to practice these occasionally as required to get by in life. I hope you may find my rather basic understanding and summary of the topic to be valuable.

Go do good things.

2.3 GEEK LEADERSHIP, AUGUST 20, 2015



In technology we seek to hack the laws of the universe, to produce some useful gadget or program or material or what-have-you that is of greater use than the standard stuff supplied by nature. These things we create are tools to better the condition of mankind. We are toe builders of tools for Homo-Faber. Our tools are the basis for the use of others creating greater tools to achieve some ultimate objective.

To succeed we are focused on the product, the service, the output produced by our efforts. Some say it is obsession, some single-mindedness. This preoccupation with the quality of what is produced is seen by some as a fault, a problem in the so called "geek" who lives in pursuit of endless improvement of that product. Others see it as the same motivation as 6-Sigma, TQM and other incremental improvement efforts, applied to the individual.

Such single-mindedness is not without cost. The devoted technologist may not care much for social interaction. He or she may ignore or devalue others perceptions of these efforts, or of these methods. Indeed a wide range of less technical persons, less devoted, use these and any other perceived weaknesses to socially marginalize and devalue the technical devotee. They seek to obtain power over the technologist, directly or indirectly.

(But Sheldon does not exist. Sheldon is an amusing fiction. The true geek is not a narcissistic egotist, he is too busy ignoring such issues in favor of the product.)

So who should lead the technologists? Often these devotees of technological improvement must work in teams, someone must lead? In earlier years technologists would elect the best, the most capable among them selves to lead. They would seek the person most likely to lead to that

breakthrough, that next improvement, the common value they sought to achieve. This lead to great technological progress.

Recently technologists are lead by those less dedicated, those not devoted. This has followed the rise of the MBA and the PMP. Other considerations have gained equal ground to technological progress, such as political acceptance. As a result the degree of technological improvement, the degree of innovation, has decreased.

While some technological progress is meaningless, lacking purpose, and should not be funded, this does not mean teams of technologists should be less effective. While customer satisfaction is important, this does not mean that diminished focus on excellence should occur. The complexities of a tricky but ingenious approach should not be dismissed by leaders incapable of grasping it.

Our wellbeing as a culture, our economy, is dependent on the innovation and prodigious output of geeks. To improve our lot, we should return to the environments that demonstrably created great success, and to technological leadership by technologists.

On the other hand, the understanding of the roles of technologists has degenerated to childish simplicity. Anyone using a computer has become a programmer. Greater sophistication in the different roles and responsibilities of different disciplines must be reestablished.

Without these changes, the present trajectory will hold. Technological excellence and rate of innovation will remain low.

2.4 LEARN LEADERSHIP: HOW TO FOLLOW, JUNE 30, 2015

Much has been written about leadership. Too much. It has been raised to a mystical art, and equated with all sorts of strange unrelated things including the various agendas of psychologists, sociologists, and managers. There is money in leadership as a topic, and it is very popular. You could read about leadership all day and barely have time for lunch. You can pay big money to learn it, to hear about it in conferences and seminars.

However one key prerequisite skill of leadership is less discussed: following. Followership may be taken as synonymous with service including: service to others, service to the nation, service to a cause. Before you can be a good leader, a credible leader, you must serve. Those who claim leadership but did not serve at any time, in any capacity, are not credible leaders.

In Japan there is a philosophy that a leader serves his constituents. This is a martial philosophy, born of life and death leadership and decisions under duress. You can also find it in Matthew 23:11. There are hundreds of other references, I think, in many cultures. Most military, ex-military, Peace-Corps, Boy-Scouts, Priests, Nuns and similar recognize the truth in it. Those who do not may perhaps be more about power than about leadership.

There is a sort of brotherhood (brother-and-sisterhood?) of those who have served in one way or another. They are tolerant and respectful of a broad range of types of service. They carry no cards, wear no insignia, but can recognize each other on the street, in the meeting, or on the conference call. They respect each other, are courteous to each other, listen to each other, rely on each other. No amount of money can buy you into that group. No amount of criticism from outside diminishes a member. It is an honor, a privilege. It comes with a lifetime membership and a money-back guarantee (its free, sort of).

Hardship and adversity do not diminish the lifelong status of the follower, of service. Attempts to serve when you are impaired, disadvantaged, ill, disabled, crushed by circumstance, downtrodden by injustice or other factors do not reduce the respect you have won by service and following. To the contrary, such adversity increases the respect you will find from others who have served. Admitting powerlessness can even, sometimes, rouse them to additional effort on your behalf. It is like a lifetime get-out-of jail card in some ways, albeit strictly limited by the amount of service you have rendered. This is as it should be. Ignore those who would limit such justice, they probably have not served more than themselves.

There are now articles about followership in magazines, and books to buy. Yet fewer claim this skill. It seems to me that if you have done enough of it, you may not need the books. The only true path to learning leadership is to follow, to serve.

We currently teach our children to lead. I am not so sure we teach them to follow, to serve. Expecting to lead is often arrogance. Expecting to lead when you are younger than the leader is often doubly so. Contracts and other factors can influence that, though. For example: the prime contractor supplies the leadership in most cases in my industry. Ignoring that is arrogance as well.

So what does it take to follow, to serve, to selflessly contribute? This is a nearly spiritual subject. (It takes some spiritual self-development to follow, or to lead.)

- Humility: It is not about you, it is about (the mission or other similar concept)
- Loyalty: A good follower is loyal to the leader and to the team.
- Trust: A follower must trust both leader and team.
- Honesty: You cannot follow without speaking the truth when asked. Leaders cannot lead without hearing the truth. The team cannot run correctly on falsehoods.
- Charity: You must be willing to serve and devote your efforts to the good of others, subordinating your own rewards to the structure and rules of the team.
- Courtesy: Those who follow must respect the team and the leader and themselves in speech, writing and actions.
- Following Orders: To follow you must do your best at what is needed, when directed by others and when not yet directed (initiative), so long as it is lawful and just.

Without these traits you are not prepared to follow, nor to lead. To follow is to develop these traits. If you have not followed, devoted a bit of your life to some cause outside yourself, do not presume you are prepared to lead others. If you see a leader who cannot follow, reject that leader. It is the duty of the follower to reject the bad leader, and we all learned that from Hitler or Stalin or another like them. This is the simple truth of following and leading.

2.5 THE TOP LEADERSHIP SCHOOL, NOVEMBER 11, 2014

Many universities are now offer leadership education. Many MBA programs include leadership training. Leadership is in short supply, according to management experts. Where is the very best leadership education and training available?

> The US military receive the finest and most comprehensive leadership training in the world, and every Veteran has it.

There are a set of diplomas, a set of experiences, that most recruiters ignore through ignorance, in the possession of every veteran. These document leadership training unequaled in civilian life. It is a national disgrace that so many recruiters lack basic competence in recognizing and rewarding these credentials.

ENLISTED BASIC TRAINING

Starting at the first moment of arrival, you are evaluated for leadership potential. Can you follow instructions? Are you reliable? Can you stay calm? Can you surmount challenges? It is common to identify several individuals in every class as basic training leaders. They are given extra responsibility and grilled, examined, evaluated even more than their peers. This most fundamental school offers weeks of full immersion, night and day leadership training and evaluation for minimum suitability. If you are not suitable, you do not graduate and you are sent home.

All unsuitable candidates do not get their honorable discharge. If you have no business in leadership or teamwork, you are "sent packing", "weeded out", "kicked to the curb".

ENLISTED TECHNICAL SCHOOLS

Once you have been found to be minimally suitable for further responsibility, you are shipped to a school to learn a trade. Again, some are selected for leadership and responsibility and examined beyond their peers. Others may demonstrate academic or technical leadership. All evidence of exceptional promise is documented. Some such schools may last a year or more, full time, every weekday. Most are college level courses, with transferable credits.

After technical school each enlistee has been comprehensively examined for potential, and that is fully documented.

NCO SCHOOLS

Those persons demonstrating superior ability to lead and perform are promoted to Non-Commissioned Officer. Each NCO has proven and demonstrated the capability to do the job and lead others in doing the job. Those not proven capable are sent home, but usually still have an honorable discharge. All NCO training explicitly includes leadership training, real leadership experience, and evaluation as a leader. Most are college level courses, with transferable credits.

Recruiter: does your candidate's resume say "NCO"? If so they are a proven leader.

Senior NCO Schools: Are proven, year after year, to be among the most capable you are promoted to higher NCO grades. If you are not selected, you are sent home. Others may opt to leave as well. However if you remain you will be sent to senior leadership training. This training is college level advanced leadership, with credits. At this point, the majority (over 90 percent) have moved on to higher paying or less demanding work, or have been eliminated as not the very best leadership material.

Recruiter: Was your candidate a senior NCO? How many did he manage and lead? Dozens? Hundreds? What was his financial responsibility? Millions? Tens of millions? More? All typical.

OFFICER TRAINING

Officer training in leadership is as comprehensive, but starting at a higher academic level than enlisted leadership education and training. The military academies are the pinnacle of leadership training and evaluation in the military. The very same process of retaining only the most promising candidates occurs for officers. The famous draconian policies of GE have no greater effectiveness, and are less selective.

AUXILIARIES

Various auxiliaries offer additional leadership training and experiences. This includes ROTC and Civil Air Patrol. Lest you discount this, I will offer my own experience in search and rescue as a teenager responsible for my part in assuring the lives and welfare of our team as well as victims. You can accumulate thousands of hours of classroom training and real leadership experience in these programs- potentially more than in any college degree.

OPPORTUNITY

In civilian life you may be responsible for opening the store or repairing the tire in your early 20's. In Military life you might see that, or you might be responsible for the lives of your squad under fire, or for the operation and stewardship of millions of dollars of equipment. You will often have much more responsibility, more leadership scope, much faster.

COMPONENTS OF LEADERSHIP

In the US Military you learn to respect those you work with, be polite, and follow directions (courtesy and discipline). You learn responsibility and real consequences. Veterans learn to set and achieve goals. They learn to communicate effectively and unambiguously. They learn to perform, and meet standards set. They learn to work under pressure, and achieve results regardless. Veterans will probably have learned some theory of motivation and leadership psychology, and will have had opportunity to apply that.

APPLYING THIS LESSON

If you are a recruiter, and you have two candidates before you, one veteran and one not, with similar experience otherwise then you must remember to apply the hundreds or thousands of hours of additional leadership training and experience of the veteran. If you have difficulty understanding or recognizing the additional qualifications, ask the candidate for clarifications or look them up. It is all documented somewhere. Remember that to the Veteran this level of skill and training may seem routine, but to the civilian it may be incomparable. If you do not bother to understand and accept the additional qualifications, shame on you. It is your duty as an American, and by law in most cases.

If you are a student trying to learn leadership, do not discount the finest leadership training and education you can undertake: the US Military.

If you are a citizen decrying our lack of leadership in the USA, examine why we are under-utilizing our best and most highly trained leaders: veterans.

Have a happy Veterans Day.

2.6 FIVE SIMPLE RULES, JUNE 1, 2014

When I was a boy I joined Civil Air Patrol, and each week we had a class called “Leadership Lab”. I had chances to use leadership in search and rescue, drill (marching) and even cleaning the CAP building. Later I took yet more leadership classes as an NCO in the Air Force. I must have spent 1000 hours or more learning leadership. Since that time I have had 30 years of applying leadership to my work in technology, where I often manage something.

I am not saying that I am a great leader. I know some great leaders, and I will sometimes defer to them when they are present and leading. However I am competent. I have lead groups, small and large, for many years.

I have used five simple rules to explain and think about the topic of leadership. I do not think just anyone is suited to leadership, but if you are then these five rules will take you far.

1) You must identify the group you lead. They may have a name and identity or you may have to give them one.

2) You must share your group's values. You must be a spokesman or at least an example for their ethics and morals. It helps if you are honest about it.

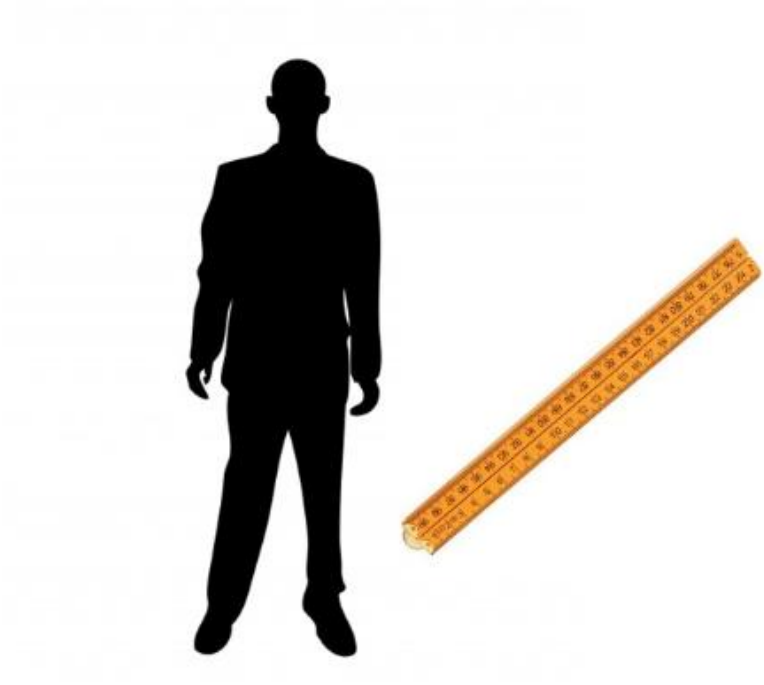
3) You must share their mythology and symbolism. If they see the flag as a symbol of their values, you must also see it that way.

4) Your actions must be directed by these values and symbols. You must have integrity. If you do not they will eventually find out that you are a fraud and toss you aside.

5) You must put the constituents interests ahead of, or at least on par with your own interests. If you do not they will eventually find out that you are insincere and toss you aside.

The greater extent to which you represent them (your "followers" or "constituents"), the more they will trust you to lead them. These rules are not complex, but they are valid. If you have the proper skills you can follow these rules and become an effective leader. If you do not, you can still use these rules to evaluate the leaders you have and learn to pick better leaders for yourself.

2.7 MAN-UP, MARCH 14, 2015



Life is short. Look the truth in the eye and face it.

MANHOOD

You may not have heard this, so I decided to take a moment and post it. Here is some advice on being a **man** that is not commonly discussed any longer. It is both American and traditional, so not particularly politically correct or current perhaps. I have summarized it and put it in my words. I am definitely not saying that I am a good example, just that this is something to strive for, a bar we **men** should try to reach. Do with it as you wish.

FOUR CATEGORIES

As a man you must manage 4 categories of effort:

- **Responsibilities:** Important things asked of you that you have agreed to do, verbally or on paper.
- **Commitments:** Important things not asked of you, but that you have agreed to do anyway.
- **Obligations:** Important things others expect you to do, without your explicit agreement.
- **Indulgences:** Things you do for yourself alone.
- As a boy you focus on indulgences. As a man you must balance all these things out to maintain your spirit and your reputation. Obviously responsibilities come first, then commitments: these are your word. Obligations, unless they are ridiculous, come before indulgences. Yet you must have enough indulgences to keep your spirits up, but not too many. You only have so many hours in a day, and so many days on earth, so use them

wisely. Your accomplishment is also limited by the skills you have acquired, and those you have not. Do not take on more than you can accomplish.

SERVICE

Boys serve mostly their own interests, and as they grow they learn service to family and friends, maybe community. As men we do not serve ourselves, we serve others. If you serve only yourself, no matter your age, you are a boy and not a man. Unless you dedicate your life to something in particular, there is a sort of default hierarchy:

- Wife and children
- Wider family and friends
- Your community
- Your country
- The world

The way the hierarchy works is like this: If there is a conflict between something higher and lower on the list pick the one on top. Things you add to life to accomplish your service to these, like your company, exist ultimately to serve this list and not for their own sake.

As a man, the rules of the four categories above may sometimes come before the hierarchy below them. When that happens, it is a hard day.

WOMEN:

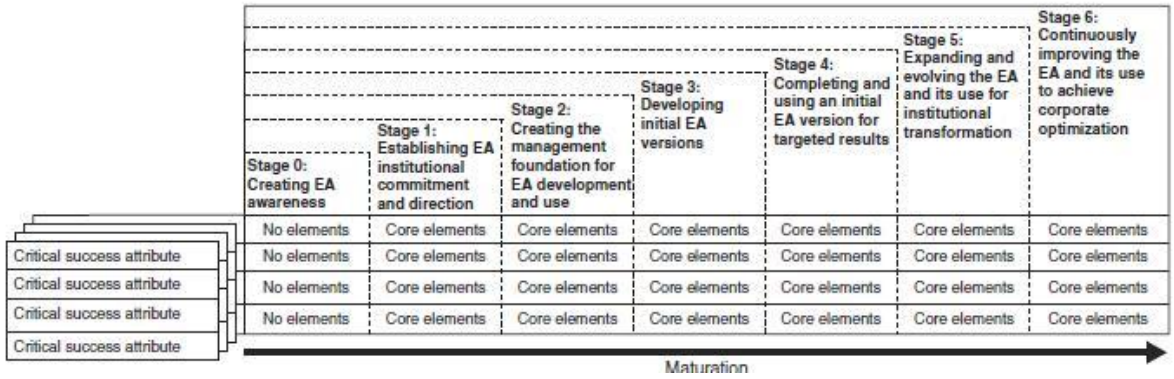
Women have their own way, their own rules, and not being one I will not try to describe it. Women can adopt our rules, our standard, but something important might be lost in doing so. Some say their purpose in life can be much more than ours.

CONCLUSION

It is best to never judge another unless there is some real need to do so. Others have different circumstances than you, and you have no idea what they struggle with. It is best to measure others rarely and yourself often in life, and I do it yearly. Go measure yourself and leave me out of it.

SECTION 3: ENTERPRISE ARCHITECTURE CONTINUOUS IMPROVEMENT

3.1 CONTINUOUS IMPROVEMENT OF ENTERPRISE ARCHITECTURE, AUGUST 2, 2014



"Maturity management" refers to process improvement. Processes are said to be "mature" when they are effective, stable and under continuous revision for detailed improvement. It is not unreasonable to expect EA to exhibit this same kind of process improvement and effectiveness it purports to provide to the enterprise as a whole.

ANALOGY

Probably the best known process improvement framework in technology is CMMI. CMMI applies to software development and not EA. However, like CMMI, we must initialy write down our processes, concepts, methods and make them repeatable to begin organized and controlled improvement. In the end both CMMI and EA maturity management seek to establish continuous process improvement..

FRAMEWORKS

Writing down your process, methods, concepts produces what is called a "framework" in EA. "Framework"is a quirky "term of art" that we use. If you meet an enterprise architect, EA company, consultant, Chief Architect who does not know, use or recommend frameworks then you can immediately infer that that what they have or offer is "immature", as writing down what you do is a first step in producing better EA processes. Fire them.

TAILORING

You can grab one of many preexisting EA frameworks and start with that, or you can make one up. In either case, you must then tailor and improve your framework over time to meet local needs, improve effectiveness, reduce errors, streamline effort, etc.

You must keep in mind that a PM framework is NOT and EA framework. An auditing framework is NOT and EA framework. A framework for maintain your car is NOT an EA framework. If you produce your EA framework on your own, without reusing a reputable and proven one from elsewhere, from scratch, you may be throwing away many thousands of hours of free work, and you may produce something that is NOT an EA framework.

Not all EA frameworks on the web are equal. The three that can trace roots back to the beginning of EA are DODAF (and your local national variant), FEAF (and your local national variant), and TOGAF

(also started with the help of the US DOD). The rest deserve a bit more examination for suitability before adoption. All will be tailored to your local organizational needs.

YOU MAY START WITH FEAF OR DODAF OR TOGAF OR WHATEVER, BUT OVER TIME TAILORING AND CONTINUOUS IMPROVEMENT MEAN THAT ALL MATURE ENTERPRISES MOVE TO A (MORE OR LESS) HYBRID FRAMEWORK.

EAMMF

We need an example framework for improving our EA framework. EA has frameworks for every little thing.

The Enterprise Architecture Management Maturity Framework, produced by US GAO is probably the very best example of a an organized method to improve EA processes. There are alternatives. Version 2.0 is the grand opus of Randy Hite, before he retired at GAO, and is a thourough and excellent body of work. It is based on the oversite and management of hundreds of organizations using EA in the US Federal Government. That is a big sample.

Yes, you could apply it to industry. Honest.

http://www.gao.gov/new.items/d10846g.pdf

SEVEN STAGES

The EAMMF lists seven stages of maturity, describing a clear path to success.

- Stage 0: Creating EA Awareness
- Stage 1: Establishing EA Institutional Commitment and Direction
- Stage 2: Creating the Management Foundation for EA Development and Use
- Stage 3: Developing Initial EA Versions
- Stage 4: Completing and Using an Initial EA Version for Targeted Results
- Stage 5: Expanding and Evolving the EA and Its Use for Institutional Transformation
- Stage 6: Continuously Improving the EA and Its Use to Achieve Corporate Optimization

In stage one you select a framework. In stage 6 you have achieved continuous improvement of your customized framework.

FOUR MANAGEMENT ACTIONS

EAMMF lists four management actions identified as critical to success in whatever management initiative you may have, including EA.

- Demonstrates commitment:
- Provides capability to meet commitment
- Demonstrates satisfaction of commitment
- Verifies satisfaction of commitment:

CORE ELEMENTS

The EAMMF has 50 core elements, detailed items in your checklist for the success of EA.

> "THE 59 CORE ELEMENTS ARE COLLECTIVELY THE EA PRACTICES, STRUCTURES, ACTIVITIES,
> AND CONDITIONS THAT, WHEN PROPERLY EMPLOYED BASED ON THE UNIQUE FACTS AND
> CIRCUMSTANCES OF EACH ORGANIZATION AND THE STATED PURPOSE OF ITS EA PROGRAM,
> CAN PERMIT THAT ORGANIZATION TO PROGRESS TO INCREASINGLY HIGHER STATES OF EA
> MANAGEMENT MATURITY AND THEREBY MAXIMIZE ITS CHANCES OF REALIZING AN EA'S
> INSTITUTIONAL VALUE." US GAO

INDIVIDUAL POWER

in immature processes and organizations individuals act as "heroes" to assure work is completed. In mature processes and organizations this is less true, and the process assures correctness or completion. Also, individual decisions may become less prevalent or important because the process exposes fact and basis for decision. In EA the chief architect is less decision maker and more facilitator as the process matures.

Sometimes a Chief Architect or big-name consultant resists maturity to retain self-importance, ego satisfaction, or control. You may want to avoid such persons as your Chief Architect or lead consultant.

CONCLUSION

It is reasonable to expect EA to provide improvement and tailoring of the EA framework used in any organization. There are maturity frameworks to provide a path to improvement of EA in your organization, and one example is described here.

3.2 EA FRAMEWORK EVOLUTION, OCTOBER 4, 2014

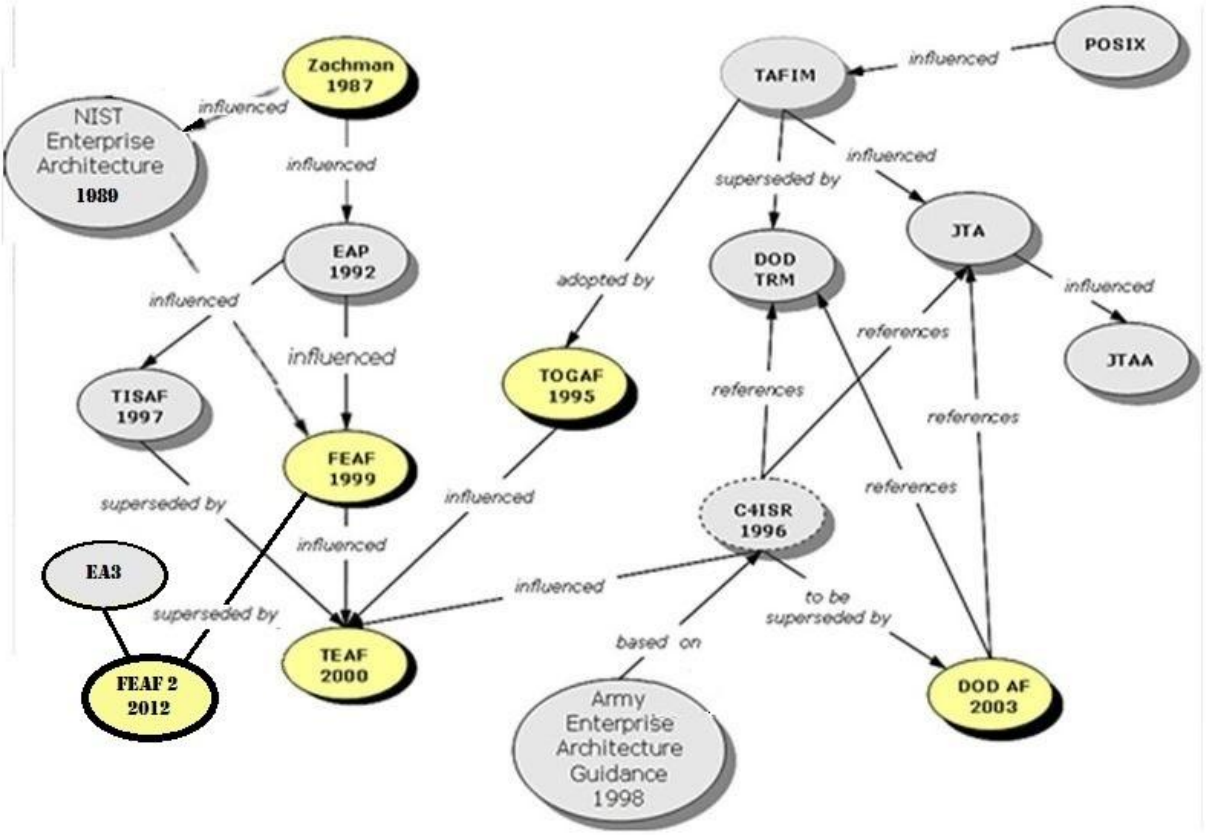


FIGURE 2 THIS IMAGE COURTESY OF THE ARMY VIA ROY ROEBUCK AND TWEAKED BY HIM, THEN ME. THANKS ROY.

I am of the belief that it is disingenuous to claim an example of a human creation before that thing was ever created. The term "enterprise architecture" was first created and used in 1989. However the history of enterprise architecture frameworks goes a bit further back, but is not enterprise architecture per-se until 1989.

We will stay focused on base frameworks, not the special purpose frameworks associated with EA. There are four highly influential enterprise architecture frameworks, and two worthy of honorable mention. Just six in total. The rest is noise.

ZACHMAN'S ZIF

John A. Zachman wrote a papers in 1987 concerning a framework to improve the construction of individual information systems by vendors, especially IBM. In 1989 he participated in an United States NIST conference where the new and different concept of enterprise architecture was proposed. In this new concept the customer (not the builder) would manage all the information systems they owned, and architect a roadmap to improved organizational performance. John's framework was not selected as the exemplar, but was applicable to both the original smaller

problem and the larger scope. (John Zachman did not fully acknowledge the connection of his framework and the EA concept officially until the 1990s, after others had done so.)

The ZIF went through at least 3 major improvement cycles, influenced many other frameworks, and became one of the most important EA frameworks in the world today.

(Please note that my personal opinion is that all John's ideas belong to John and IBM, and not the fraudsters who are attempting to masquerade as John. My regards and support go to the Zachman family in that dispute.)

TOGAF

TOGAF (The Open Group Architecture Framework (tm)) started life as the US DoD TAFIM. TAFIM was a standard environment for building intelligence and C2 applications, a sort of "workbench" for analysts. When the future of TAFIM was bleak due to funding, DoD sent the material to The Open Group(tm) where it became the basis of one of the most successful EA frameworks. Like Zachman's ZIF, the TAFIM was not, at first, about the customer managing all of their IT assets and architecting a future of the whole as a set.

Most recently, as I understand it, the whole body of DODAF was also donated to The Open Group(tm) and TOGAF due to the fine work work of Walt Okon.

DODAF

The United States Department of Defense Architecture Framework was created with the aid of MITRE (a well known non-profit company that exists for the public interest) and project manager Kathie Sowell. It brought together elements of a prior framework called C4ISR (for building command, control, communications, intelligence, surveillance and reconnaissance systems) with elements of USAF IDEF (for use in computer aided manufacturing- not shown) and dozens of DoD interests. In the beginning it provided a means to build a "system of systems" using asynchronous messaging, but it now provides a means to depict and analyze almost any operational need.

As mentioned above DODAF has apparently been donated to TOG and presumably will be absorbed into TOGAF.

DoD has spawned dozens of related frameworks (such as MODAF in the UK) worldwide. They all exchange ideas. DODAF has ideas from many countries and global participants. As a family, these related frameworks are very widely used.

FEAF

The Federal Enterprise Architecture Framework was the work of many including Mike Tieman, Ira Grossman, Manny Centano, John Zachman and many others. It drew directly from NIST work (and the NIST framework- I added 1989 to the bubble there) and from the start was focused on managing the enterprise as a whole. It used a subset of the ZIF tailored after the thoughts of Spewak (1992).

FEAF version 1 has been widely adopted by other civil governments. It was enormously successful and stable for over a decade. It is still in wide use today.

I added FEAF 2 to the diagram as a distinct entity, as it draws heavily on Scott Bernard PhD's work EA3. Thus the FEAF 2 is essentially a different framework It also merges parts of the FEA reference models, not originally a framework buy a metadata taxonomy for reporting, into the FEAF 2 itself. (Note that I added FEAF 2 and EA3 to the diagram.)

The FEA, not a framework, also contains practice guidance and implementation guids (I note esp. Burk and Thomas among my favorites).

TEAF

The Treasury Enterprise Architecture Framework is worthy of mention as it was an early attempt to merge ideas from DODAF and FEAF. It failed and the TEAF is no longer used, but it did try. This attempt to unify the EA frameworks of the US Government is now connected to FEAF 2 and "The Common Approach" to enterprise architecture of OMB.

TEAF is an example of the trend of major framework consolidation underway today. Do not try to implement the TEAF, just remember consolidation is my recommendation.

Enterprise Engineering: James Martin adopted the broad work of Australian Clive Finkelstein called Information Engineering, a predecessor to enterprise architecture in the era of integration via database. Missing the EA revolution and late to the game James Martin created Enterprise Engineering. It is not quite an enterprise architecture framework, lacking detail and having a wider scope to include culture.

Enterprise Engineering is notable, but the work of Finkelstein is as much so or even more notable in contributing to that lineage.

INTENTIONALLY OMITTED

GERAM is sometimes listed as an EA framework, but I do not see it as one. It is merely the work of several vendor consortia (AMICE, CIMOSA, GRAI) to organize product architecture providing standard products for integration. (In this case the target was CIM.) CSC has one of these as an integrator, as do many other vendors. These are not means for the customer to organize and control the entire diverse set of systems procured from diverse vendors, and I do not include them as important EA frameworks. They are not means to effect or enable business transformation.

There is also a thing called "Agile Enterprise Architecture", which seems focused on software and not the broad problems of IT. If you are interested in this area I would steer you more towards SAFe and less toward Ambler & McGovern which strikes me as useless. At least SAFe and Dean Leftingwell has adopted a three tier system similar to FEA, and hardware/business analysis/etc. could be grafted on. Remember software is a part of one of the 4 pillars of the FEAF, for example, and not all of the scope of EA.

There are a wide range of other frameworks, small or little used. Among those I recommend Roy Roebuck's work, as he very much knows what he is doing. He now calls it an Enterprise Management Framework. Some of the others are produced by persons without credentials in, education concerning or experience with real EA and you might easily buy a package full of nonsense.

REMARKS

Enterprise architecture frameworks are consolidating. They are in a period of consolidation and not wild innovation. If you are off playing with flaky little frameworks you may not get very far in the industry right now. The major frameworks probably have everything you need. Other bits may be added to them via tailoring, including a minimalist lean approach.

A further word on bloating: Most of the frameworks are becoming bigger and less useful. Small, narrow purpose frameworks that work together are better than huge frameworks whose application is unclear. When you tailor your chosen major framework you are likely to reduce more than you add.

Lastly, SOA is well managed by FEAF, DODAF and TOGAF- even Zachman if applied with a bit of art. SOA is not in any way distinct or separate or outside of EA.

If you want to read further, you might try this paper: http://ggatz.com/images/SOA_COMPARE.pdf

3.3 CHOOSE FIVE FRAMEWORKS, OCTOBER 4, 2014

Choose Five Frameworks

Your Choices (Some of Them...)	FEAF	DODAF	TOGAF	Zachman	EAMMF	Martin	ISO 38500	CMMI	Custom	FEA
Enterprise Level										
Segment Level										
Solution Level										
Governance										
Maturity Management										

To develop a mature enterprise architecture practice you must choose a framework. Many authors have said this in many documents. Some practitioners are in denial and will never develop a mature practice: too bad for them. For most of us we have accepted that to improve and manage your concepts, processes and methods you must first write them down. Having written them down, you now have a framework. A good many of us have also faced that it is better to start with someone else's mature work than doing it over from scratch.

Let's assume the readers of this post are rational creatures of the latter sort.

FIVE AREAS

I have written elsewhere that EA actually covers 5 areas. You are actually choosing for all five areas. You can choose to ignore that, and select a single framework to cover everything, but that is also a choice.

NOT ALL THE SAME

It turns out not all frameworks are the same. Some are stronger here, others there.The EAMMF (Enterprise Architecture Maturity Management Framework) is only intended to manage the process of documenting, tailoring, improving your chosen framework(s). If you choose only the

EAMMF, you may spend a very long time tailoring and creating the missing elements to fit your needs.

Some will not choose the EAMMF even for its intended purpose. Some will insist on using CMMI for EA maturity, even though it was developed for software development maturity management and is not perfectly suited to EA. There is no accounting for taste.

DODAF STRENGTHS

Some might protest and say their framework is already all inclusive. However you cannot show me the view of all the databases in DoD used to mange redundancy of that data storage- it's not in there. DoD has some other processes and registries for that. You can choose that associated stuff to go with your DODAF- or not. Frankly those other mechanisms are not that mature, and some other frameworks might help. In fact DoD has chosen an international standards based framework for IT governance.

DODAF is very good at solution and segment architecture.

The best part of all this is that they have written it all down, quite explicitly. That's why it works.

ENTERPRISE LEVEL

Zachman and FEAF are best, in my opinion, at the enterprise level. If you want to just inventory the big items and stop, look at these.

SAFE OR AGILE

Some will insist on using Agile or SAFe based frameworks for EA, despite the fact that these address primarily software and leave many questions unanswered. How will you handle hardware standards and communications standards and routers and... yes those are all int the scope of EA. Choose Agile and you may have gaps or spend your next decade writing to fill them. What about that COTS and legacy software?

TOGAF

TOGAF covers everything! No it doesn't. The ADM is a solution development process. Enterprise and segment processes are primarily driven by yearly budget shedule. I do not find much recognition of that in TOGAF.

ISO IS COMPREHENSIVE

Is it? It does not yet handle EA governance processes well IMO, and focuses too heavily on custom software development. You can disagree... it’s your choice.

PARTIAL COVERAGE

You are only going to have architecture for solutions? Fine by me. Forget about enterprise transformation and management to achieve strategy. You will only manage what you choose to manage, and develop processes for.

COMPATIBILITY

Whatever you choose it all must work together. It must work with your other corporate processes. That will not happen out of the box. Have fun making it all fit together seamlessly.

OTHER

Your audit, SDLC and acquisition management processes must also fit with EA. If they do not fit together well watch your chances of transformation decrease. Did I mention security yet?

CONCLUSION

When you choose your framework(s) you may need an expert. Tailoring takes some expertise too. If you choose to not have a single framework, whoever chose was probably not that expert.

That geeky dude who says you are just going to use Agile for everything- you know him right? Make him show you his management degree. Have him explain how that choice will satisfy your corporate controls for fiduciary responsibility and ROI, and how it will control redundant or mission irrelevant development. While you are at it, ask him how it will focus on mission related improvements, and measure those organizational gains. When you frame a house, perhaps it is not best to use painters and gardeners to do the framing eh?

Remember that guy who choose poorly in the Indiana Jones movie? Don't be him. It will cost you. Choose well. If you need some help call your friendly neighborhood real enterprise architect, they do this kind of stuff. They will be happy to help.

I'm not suggesting a mountain of paper either. You should specify just enough to cover what you will really do. 'All of it. In a smaller company you may not need a distinct segment architecture, for instance. Like Einstein said, make it as simple as possible but not one bit simpler. Or not, its your choice.

3.4 PERIODIC & CONTINUOUS IMPROVEMENT OF ENTERPRISE ARCHITECTURE, FEBRUARY 15, 2016

So you have identified a few frameworks that work together as how you will operate, or maybe one. Now you are getting started. After a while you will discover that the framework, out of the box, does not fit with how your organization does governance / investment management / LOB (Line of Business) management, or something else. Also, the framework you choose has 27 different ways to do x, and you need one. Furthermore, how it does something else is less effective than how some other framework does that.

Well, you need to manage your framework. You need to tailor it to how you work, and how the organization works, and then you need to revise it when better methods are found. You need to eliminate parts you do not need. If you do not need to do that, you are very lucky and I don't really believe it, but fine. Move on. If you do find yourself here, read on.

PERIODIC REVIEW

One way to handle this is to get together yearly, or quarterly, or as often as you need, on a regular schedule to revise and update the framework. You can keep a big framework document or just an SOP (Standard Operational Procedures) document for enterprise architecture that references parts in the big framework document. Either way, you edit it in the periodic meeting.

CONTINUOUS IMPROVEMENT

Alternatively you can revise it on the spot any time a problem is found. You halt work and change it right there. More likely you revise at the end of each action in a "lessons learned" meeting, and call that continuous.

SELF ASSESSMENT

If you have a big organization you can send out a survey, get an opinion on how you do each part of enterprise architecture, then update your documents based on that.

AUDIT

Instead of a self assessment and survey, you can hire or assign some team to review activity in a big organization. Then you take the results and update your documentation.

MATURITY

Ultimately you want to get better. You want to use performance measures and real data to tell you what to change. You want to monitor effectiveness using some formal framework like the EAMMF (Enterprise Architecture Management Maturity Framework) to guide you to being better. You might also modify that framework with experience.

In the end state your enterprise architecture practice should be documented, repeatable, produce real results, fit within the organization and use the best known methods and processes.

3.5 AGILITY VS MATURITY 1, NOVEMBER 20, 2014

Organization Size vs Average or Overall Process Maturity for Optimal Performance

Very Small	Small	Intermediate	Large	Very Large
Very High Maturity	Very High Maturity	Very High Maturity	Very High Maturity	Very High Maturity
High Maturity	High Maturity	High Maturity	High Maturity	High Maturity
Moderate Maturity	Moderate Maturity	Moderate Maturity	Moderate Maturity	Moderate Maturity
Low Maturity	Low Maturity	Low Maturity	Low Maturity	Low Maturity
Very Low Maturity	Very Low Maturity	Very Low Maturity	Very Low Maturity	Very Low Maturity

In "agility" we attempt to keep processes lightweight. This allows an organization to stay nimble, to react to differing conditions. In process maturity we attempt to make processes repeatable to ensure repeatable results, repeatable levels of quality. How do we reconcile these two concepts in an enterprise?

MATURITY AND AGILITY

Do you remember CMMI? A few short years ago the way to build better software was to have mature processes.

http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

http://cmmiinstitute.com/get-started

However we not have efforts to produce greater agility:

http://en.wikipedia.org/wiki/Agile_software_development

http://agilemanifesto.org/

AT ODDS

While CMMI is about processes and tools, the Agile manifesto declares that it focuses on individuals and interactions. Individual heros are a hallmark of immature organizational development.

While mature organizations measure progress by organizational performance improvement, Agile principles say working software is the primary measure of progress, regardless of if it improves the organization.

While mature organizations plan for change, Agile embraces changing requirements without control, and focuses on customer collaboration over contractual terms.

ORGANIZATIONAL DEVELOPMENT

For many years smaller organizations have been thought to be more flexible, more nimble, more responsible to the market. Larger organizations have been said to offer economies of scale and the opportunity for funding quality efforts and greater product or service quality.

http://smallbusiness.chron.com/advantages-large-business-21007.html

http://www.feverbee.com/2010/01/bigorganizationsonlinecommunity.html

http://en.wikipedia.org/wiki/Organization_development

http://www.scienpress.com/Upload/AMAE/Vol%203_5_8.pdf

OVERALL MATURITY

Any organization may be characterized by its overall maturity. Identify all the processes used in the organization (mostly the same processes as the next organization), and then characterize the overall maturity of those processes. Use CMMI or whatever yardstick is handy. This is the average process maturity across all organizational processes.

Maturity vs Size: Many years ago, building workflow systems, I learned a simple truth. The overall maturity an organization can afford depends on size. Process maturity takes work, effort, labor hours. If you automate it the result is technology purchase, customization and maintenance expenses traded for some effort, not all effort. Bigger organizations can simply afford more maturity across more processes.

> The optimal level of maturity follows a curve based on organizational size. You can have too much or too little. There is a spectrum of agility vs maturity.

One interesting observation is that even in the very largest organization you can produce more process, more policy, more rules than the participants can bear. You can always increase to the point of oppression.

ADVANTAGE, BIG AND SMALL

For many years management has observed that small companies are more nimble due to reduced overall process rigidity, and large companies are more stable and have

higher quality by increased overall process rigidity. This observation is not, and never was, incorrect.

POLICIES AND RULES

Policies and rules are part of process maturity. Bigger organizations can afford more.Too many policies and rules in smaller organizations produce excess burden, excess effort, overhead, inefficiency when they create too many policies and rules to improve process maturity. Too much maturity will make employees and management unhappy.

Too much maturity is oppressive, and too little is unprofessional or undisciplined. It is all relative to size.

INCREASING AND REDUCING MATURITY

I have written about which processes should be the focus of improvement. If you need to choose what processes to improve, follow that advice. It is transformation, this improvement of processes, so target increased organizational performance. If you need to decide which processes to leave flexible, choose those not related to increased organizational performance.

COMPLIANCE

Most organizations have the reverse pattern. They have more process maturity, more rules, more policies related to supporting processes that do not improve organizational performance. Some of this is due to legal compliance and necessary. Some of this is due to personal psychological need for control and is not constructive to the enterprise.

LARGE ORGANIZATION AGILITY

Agility in larger organizations is not usually achieved by ignoring all process, or leaving all processes at low maturity. Instead maturity is focused on fewer key processes. In lean manufacturing, Kanban and similar mechanisms there is an extreme focus on quality management processes. Statistical methods are used, root cause analysis, and a range of other techniques you can find in any book on the subject. These count as processes, and they are rigid or mature in a correctly functioning lean operation. Quality comes from process maturity.

SMALL ORGANIZATION MATURITY

Small organizations prosper by picking a small number of key business processes to document and improve the quality of. They do not prosper by huge manuals full of policies and rules, processes and penalties that are largely unused and annoy employees in the course of daily business.

Fashion: It is fashionable to focus on agility, reduction of overall maturity, in business. You will move to the bottom of the optimal range on the curve. This will inevitably decrease quality of some processes. To make this work you must pick and choose which processes will have controlled quality, repeatable results. These processes should be associated with your core business and the quality of products or services produced.

MAKING BIG INTO SMALL

There are management methods to make a large organization function as many smaller organizations in a loose confederation. These can change the optimal point on the curve for agility vs maturity. However the overarching top organization must work to reduce the number of imposed high maturity processes, policies and rules which it applies to the small subordinate organizations or effectiveness will be lost.

A common trick to extend organizational size without oppressive impact is to have level 1 produce policies, rules and processes that only ever apply to management and reporting in level 2 organizations. Level 2 produces policies, rules and processes for M&R in level 3 organizations only. All official communications is barred between levels 1 and 3, it must pass through 2. This protects the lower organization. While lightening the burden of maturity this causes other reductions of agility, but may be better than alternatives.

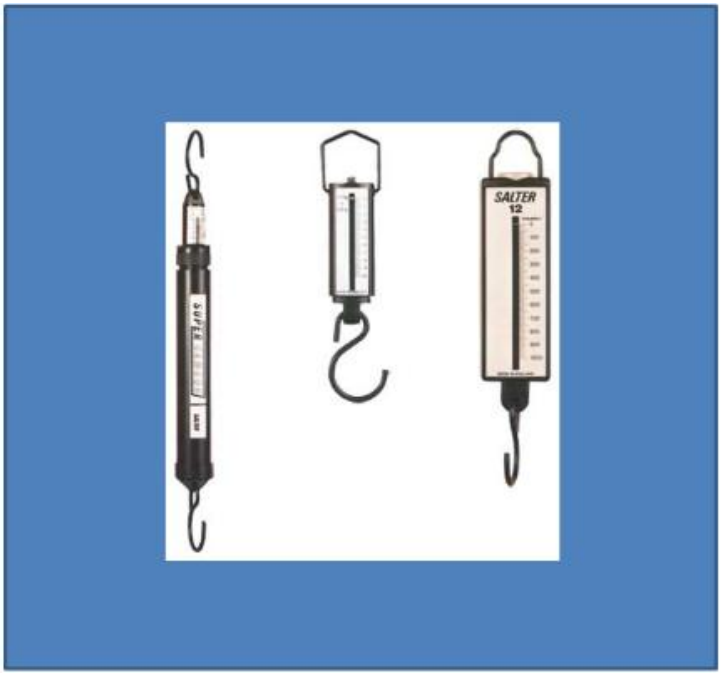
CONCLUSION

There is an agility/maturity spectrum. Your organization has an optimal level based on size. You can make a big organization function like many small ones, but you must impose minimum rigidity from above.

"Agile" efforts undermine organizational and process maturity. Maturity advantages are not conferred by "Agile" methods. Your organization can be one, or the other, and not both.

(You will not find that in your textbook. Put that in your pipe and smoke it.)

3.6 AGILITY VS MATURITY 2, MARCH 22, 2015



Which is more desirable for software development, agility or maturity? It is a valid question, agility is roughly the opposite of maturity in the CMMI sense. If you can only have one, which will it be?

AGILITY & MATURITY?

Well, although the Agile manifesto places processes as secondary behind individuals, interactions, and fluffy bunnies playing with butterflies... people ignore that and combine something like Agile with CMMI. (I suppose they do not take the Agile Manifesto very seriously. I am sure they do take the market momentum and hype seriously, to sell anything it had to be Agile for a long period.) Dr. Dobbs reported the phenomenon in 2012.

- 127 respondents indicated that their organizations were doing agile projects.
- 13 (10%) indicated that they were exclusively doing CMMI-compliant agile projects
- 57 (45%) said they were exclusively doing non-CMMI agile projects
- 57 (45%) said that their organization had both CMMI compliant and non-CMMI projects.
- The survey, which also asked about traditional projects, found the following success rates:
- 57.6% for CMMI-compliant traditional projects (107 responses)
- 56.8% for non-CMMI Agile projects (114 responses)
- 54.8% for non-CMMI traditional projects (172 responses)
- 53.4% for CMMI-compliant Agile projects (70 responses)

Dr. Dobbs went on to say CMMI decreases your chances on Agile projects, the truth is plainly that CMMI has higher success rates. Dr. Dobbs should have reported that

Agile decreases your success on CMMI projects. The success rates were too low and too close anyway. Something else is apparently going on. Our Agile focus is a red herring, distracting us from the real issues.

http://www.drdobbs.com/architecture-and-design/dr-dobbs-agile-newsletter-0208/206800401

SOMETHING ELSE:

What is that other thing that is going on? The answer is not hard to find. Poor requirements cause project failure, Note that I did not say "user stories", I said requirements. 'Shocking.

From a different source:

> THERE IS ENOUGH EMPIRICAL EVIDENCE TO SAY THAT POOR REQUIREMENTS CONTRIBUTE TO THE MAJORITY OF PROJECT FAILURES. LOOK AT THESE STUDY CONCLUSIONS PUBLISHED OVER A 13 YEAR PERIOD BEGINNING IN 1995:

...and...

> REQUIREMENTS PROBLEMS HAVE BEEN PROVEN TO CONTRIBUTE TO 20-25% OF ALL PROJECT FAILURES. THE AVERAGE PROJECT OVERRAN ITS BUDGET 189% AND ITS SCHEDULE BY 222%—CHAOS REPORT / THE STANDISH GROUP 1995

...and...

> REQUIREMENTS ERRORS ACCOUNT FOR 70% TO 85% OF REWORK—LIFFINGWELL, 1997

...And...

> POOR REQUIREMENTS ACCOUNT FOR 71% OF PROJECT FAILURES—GRADY, 1999

...And...

> BETWEEN 40 AND 60 PER CENT OF ALL SOFTWARE DEFECTS CAN BE ATTRIBUTED TO BAD REQUIREMENTS—ABBOTT, 2001

…And…

> ONLY 34% OF PROJECTS EXPECTED TO FINISH ON TIME; 52% HAD PROPOSED FUNCTIONALITY; 82% HAD TIME OVERRUNS; 43% HAD BUDGET OVERRUNS—THE CHAOS CHRONICLES / THE STANDISH GROUP 2004

…And…

> FLAWED REQUIREMENTS TRIGGER 70% OF PROJECT FAILURES—INFOTECH RESEARCH, 2005

…And…

> GAPS IN THE TECHNICAL REQUIREMENTS ACCOUNTED FOR MORE THAN 70% OF PROGRAM PROBLEMS—UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE, 2008

http://ultimatesdlc.com/blame-troubled-projects-business/

THE REAL TRUTH

There is a series of real studies years ago that said if errors are caught earlier in the lifecycle they cost less to fix. No study has ever contradicted this. There was no magic exception from God that made Agile or Scrum work by different rules.

- http://csse.usc.edu/csse/TECHRPTS/1984/usccse84-500/usccse84-500.pdf
- http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670.pdf

Better Engineering, better requirements, make better software with less failure. Its not popular, but it is still true.

BY CIRCUITOUS MEANS

We find that errors detected earlier reduce failure. As Agile delays specification until programmers are on-site coding, this does not bode well for ever achieving higher rates of success. On the other hand the success of improved maturity methods based on an SDLC is well documented, and is

compatible with removing errors earlier in the process. Equivalent documented success is not present for Agile methods.

http://www.sei.cmu.edu/library/assets/evidence.pdf

Further I have written elsewhere about the need for corporate controls that Agile seems to bypass in real applications. Not all software projects have ROI as measured by the yardstick of improved organizational performance. Not all business cases for software development efforts should be approved. In fact many are wastes of money, not aligned with strategic goals, the mission, or producing redundant systems.

https://www.linkedin.com/pulse/article/enterprise-level-anti-patterns-kern-msea-cea-pmp-itil-cissp-issap/edit?trk=pulse-edit-nav_art

MY PREFERENCE IS SUCCESS AND QUALITY

I believe in engineering, not manifestos and fluffy bunnies making people feel better. I personally feel better when I deliver success and high quality software. I like the quality definition in System Engineering, where the product meets the requirements. We might add the notion that the requirements need to be correct.

Quality definitions as aggregated by INCOSE:

> QUALITY: THE COMPOSITE OF MATERIAL ATTRIBUTES INCLUDING PERFORMANCE FEATURES AND CHARACTERISTICS OF A PRODUCTION OR SERVICE TO SATISFY A CUSTOMER'S GIVEN NEED. (DSMC)

...and...

> QUALITY: THE COMPOSITE OF ALL THE ATTRIBUTES OR CHARACTERISTICS, INCLUDING PERFORMANCE, OF AN ITEM OR PRODUCT. (DODD 4155.11), (MIL-STD- 109B), (FDA 90-423)

...and...

> QUALITY: THE TOTALITY OF FEATURES AND CHARACTERISTICS OF A PRODUCT OF SERVICE THAT BEAR ON ITS ABILITY TO SATISFY STATED OR IMPLIED NEEDS. (ISO 8402)

...and...

> QUALITY: CONFORMANCE TO CUSTOMER REQUIREMENTS AND KNOWN EXPECTATIONS. (SNL EPS) QUALITY THE DEGREE TO WHICH A SYSTEM, COMPONENT, OR PROCESS MEETS SPECIFIED REQUIREMENTS. (IEEE 610.12-1990)

...and...

> QUALITY: THE TOTALITY OF FEATURES AND CHARACTERISTICS OF A PRODUCT OR SERVICE THAT BEAR ON ITS ABILITY TO SATISFY STATED OR IMPLIED NEEDS. (WG6)

...and...

> QUALITY: THE TOTALITY OF CHARACTERISTICS OF AN ENTITY THAT BEAR ON ITS ABILITY TO SATISFY STATED AND IMPLIED NEEDS. (ISO 8402)

...and...

> QUALITY: THE TOTALITY OF CHARACTERISTICS OF AN ENTITY THAT BEAR ON ITS ABILITY TO SATISFY STATED AND IMPLIED NEEDS. (ISO 8402), (IEC 1508)

https://www.incose.org/ProductsPubs/pdf/techdata/ERTC/GlossaryDefnsOfTerms_1998-10_TWG.pdf

CONCLUSION:

Process maturity beats Agility. Process maturity and an SDLC have the potential to reach higher rates of success and higher quality of product than Agility. Agile is, and always was, in my humble opinion, an irrelevant backwater in the stream of progress,

a dead end. Scrum and DevOps show promise, but only once divorced from Agile baggage and mysticism.

3.7 BUILDING COMMUNITY, NOVEMBER 4, 2014

We do not seem to spend much time on communities anymore. That's a pity, because we need it more than ever. The very idea of community is so important to our lives, and our quality of life.

NEIGHBORHOODS

Do you remember when you were a child, that feeling of belonging you had in your neighborhood? The friends and acquaintances you had? Being able to seek help from the neighbor's house? Trusting that they would treat you well because they lived so near? There are those stories of neighbors helping to erect your barn, or rebuild your house after a fire. People from next door would bring you food when someone died. We were all connected by proximity.

FAMILY

Families used to be closer when they lived near each-other. The people you knew best remained the people you know best. Tied by blood, you could trust family.

Industrial Revolution

The industrial revolution scrambled people, scattering them across the land. You had to move for the work. That has escalated, and now your home is so impermanent, the next job may move you somewhere.

INTERNET

With the Internet we may rarely even go outdoors. Why spend time with neighbors, friends, family unless they are online? Something is lost without physical presence though.

Commitment

We had more commitment and loyalty to individual people when we moved less and spent more time together, I think.

Common Values

We used to discuss things more, and share our values more. Your family would have values, and you would share in those. Your town would have values, and you would share in those too. We spent more time discussing and agreeing on things with people we knew would not disappear with the next move, or would not "unfriend" you.

Understanding

By discussing things in person, watching non=verbal clues, exchanging handshakes and hugs, and seeing the person the next day and every day after, we spent more time understanding them. Understanding leads to tolerance. We accepted our neighbors, families and friends more in less mobile times.

Connection

Because we understood our people, shared common values, were committed to them I believe we formed deeper and more lasting connections. In earlier decades sociologists would write about this, and lament its passing.

Consequences

Today our country is torn by division. Some suggest nations, based on community and shared values or traditions, are obsolete and should be replaced by global unelected

government not responsible to the people.. Some think to replace community with corporations- whose actual values demonstrated in policy are not reciprocal. Documents based on our shared values and commitment, like the Constitution, are ignored. Oaths by our leaders are ignored. Integrity is scarce because people are temporary.

CONCLUSION

Our world will not improve until we form longer, deeper relationships, commitments, to people. This is true if you are an individual, a family, a group of friends, a school, a government or a company. That may not be popular, but it seems to be true. If you really want to improve things then I urge each one of you to work on that. It is the root cause of many ills, and one of our greatest means for improvement.

3.8 EXCESSIVE COLLABORATION, JANUARY 17, 2015

Much of our current corporate culture discussion revolves around seeking consensus in large groups. This is used to gain buy-in and cooperation from stakeholders, mostly. That approach deserves some examination.

1) First, groups often make bad decisions. Remember the Abilene Paradox? If you haven't heard of this, you probably never studied management.

- https://www.youtube.com/watch?v=Y5d0vf6SOh0
- https://www.youtube.com/watch?v=z_iGdiYO7gI
- https://www.youtube.com/watch?v=O73aRfL3xvw

2) Second, group averages are proven to provide better predictions. One common example is economics. However, almost everyone forgets that this works among of surveyed groups of **experts**, not random stakeholders who show up with varied backgrounds.

- http://en.wikipedia.org/wiki/Consensus_forecast
- http://en.wikipedia.org/wiki/Delphi_method

In consensus decision making, a group of novices or journeymen can make a decision approaching the insight of an expert. This is not exceeding the insight of an expert. This is not innovating or transcending current understanding as a genius. At best you will mimic the current state of the art.

- http://en.wikipedia.org/wiki/Consensus_decision-making
- http://en.wikipedia.org/wiki/Genius
-

3) Evidence, science and engineering trump consensus. A quick example, the consensus held that non-white ethnicity were intellectually and moraly inferior in the 1900s. At the time of Columbus it is said the consensus was that the world was flat.

4) The following innovations ran counter to consensus:

- Einstein and relativity
- Louis Pasteur and germs as the basis of disease
- Johannes Kepler and the earth rotating about the sun
- Tesla, Marconi and Radio
- Armstrong and FM Radio

Today we call this stuff disruptive technology and hold it in esteem. It runs contrary to consensus.

CONCLUSION

Ask yourself, do we want to be a culture driven by fact, evidence, science and engineering, one capable of innovation and transcending current technological and social norms, or a tribe using groupthink to dictate our limits. You pick.

The next time you are in a meeting jammed full of 20 or 40 other professionals, with no action items and no status and no agenda but to achieve consensus, remember this post. Remember it as you all sit there burning thousands per minute. Remember as the time that makes up your professional and private life drains down the gutter, forever lost.

3.9 THE LEAN IPT, MAY 22, 2015

I was on a routine conference call with my Fed colleague and friend Tom McCarty yesterday. The call concerned setting up a potential IPT (Integrated Project Team) for a big program we are connected with. There were several others on the call, but I am crediting Tom with connecting these ideas together- its my post.

An Integrated Project Team or IPT is a collaborative structure for managing some part of a large program. The approach was tried for many years with great success at DoD, and is now encouraged across DHS. All the relevant parties meet in a recurring meeting to coordinate and review complex issues addressing many roles on the program. An example would be an IPT for Test and Evaluation of a new aircraft. A test may involve several vendors, engineering, testing, several operational support elements, etc.

An IPT differs from an informal working group by having a charter (a distinct scope and purpose) and meeting minutes (notes) to track real accomplishment and responsibility.

We proposed an IPT to address some long standing issues. Tom connected the idea of an IPT with an Agile Core Team. Genius... ok, maybe not but pretty darned smart. Of course I don't like using "Agile" as a term, so I am saying "Lean IPT"- its my post.

The idea is that instead of getting all peripheral stakeholders in the room each week, instead get only a core team in there on a recurring basis. Then add the others required as specific issues show up. This improves productivity, prevents wasted participation and is superior for a range of potential IPT types (focuses)..

The idea is so good, I am writing about it. Others should try it. When you do, if it works, thank Tom. If you have trouble making it work, call me and offer money (just kidding).

Governance & Maturity Management

3.10 THE BEST ARCHITECTURE IS NOT EMERGENT, AUGUST 3, 2014

<Image GFDL Public License, found on Wikipedia>

Cisco Command Line

I recently took a Cisco related bootcamp to satisfy my yearly ISSAP training. I spent time with the Cisco command line. Not meaning to pick on Cisco, they have been enormously successful; However that success has meant that many companies and products were acquired.

> THE RESULT IS THAT THE CISCO COMMAND STRUCTURE IS A HODGE-PODGE OF DIFFERING SYNTAX, QUIRKS AND ODDITIES.

Now almost anything can have an architecture. An architecture is a set of elements and the relationships between them. The Cisco command set, taken as a whole, may be viewed as an architecture.

> THE ARCHITECTURE OF THE CISCO COMMAND LINE, EMERGING AS A COURSE OF CIRCUMSTANCES, IS NOT VERY GOOD.

If it were better you could grasp it in far less time, and apply it more easily.

IBM 360 SERIES

The processor architecture of the IBM 360 series was a hallmark of logical design when it was created. Eventually, as a 32 bit architecture it required a big kludge to access a 64 bit address range. Different 32 bit address ranges were windows into the 64 bit address space. A similar method was used for the 8080 and Z-80 of early personal computers. These emerged as hardware fixes to aging processors with aging instruction sets and register structure. Neither architecture is widely used today.

A SOFTWARE PRODUCT

I cannot say what product, but I worked to add significant features to a 20+ year old software architecture. It was written for DEC VAX VMS, ported to UNIX, then adapted to Windows Servers and Client Server. The architecture that emerged was so difficult to modify that the new features were "bolted on" as distinct external modules.

THE WINCHESTER HOUSE

A legendary non-software example of emergent architecture is full of doors that open to walls, halls that lead nowhere, inaccessible rooms. All designed without a plan beforehand.

PRINCIPLES BEHIND THE AGILE MANIFESTO

The Web Page so titled says the following. "The best architectures, requirements, and designs emerge from self-organizing teams". This has been modified a bit from its original form stressing emergence as better than planning.

Few truly believe that software teams should execute without significant planning, analysis, design, architecture and requirements discovered beforehand. Only trivial front-end applications can be built in the spirit of the principle of Agile.

THE BEST ARCHITECTURE IS PRODUCED BY SIGNIFICANT ANALYSIS AND FORETHOUGHT

This is true for all but the trivial project, and even more true the more complex a software system you might construct. Some will argue, but most engineers I know agree. In fact, that might be a workable definition of engineering and engineering has produced results for thousands of years. Others are true believers of the mythology of a manifesto, a tool of the radical nut job, used by the communists and religious fringe groups.

This goes double for requirements and designs.

I will stand with the engineers. I prefer the application of science to mythology and manifesto myself.

3.11 EVALUATING ENTERPRISE ARCHITECTURE METHODS

Not all enterprise architecture methods are equal. Not surprisingly, some are more effective than others. This often depends on context. Let's examine a few examples. First, we must have a purpose for enterprise architecture to evaluate against. (We will compare methods, in total, not just frameworks.)

PURPOSE

We will evaluate methods for application in the US Federal Government as our example. Enterprise architecture, as Zachman said, is the thing between strategy and execution. It is used to transform organizations from less effective to more effective, a process often involving technology. We will very briefly evaluate enterprise architecture methodologies as a means to effect transformation to achieve strategy by means including technology.

EVALUATION FRAMEWORK

We will target use in the US Federal Government. We will evaluate for:

1. Process Completeness (using the 5 activities model);
2. Scope (Using the scope model here);
3. Compliance (using Federal Government mandates as an example);
4. Strategic Impact;
5. Maturity (How long have they been used).

ZACHMAN'S METHODS

Zachman's methods have high maturity, broad scope, proven strategic impact, moderate compliance (some areas are simply not addressed), and a very limited process model (compared to the 5 activities). That gives us roughly 3.5 of 5.

Let me be clear that no practice is complete without using Zachman's ontology to think about architecture and to understand its data and metadata. It is unequalled. But it is, by itself, not complete for this purpose.

HEFFNER'S METHODS

Stephen Heffner has defined enterprise architecture to include data architecture and process management. He excludes IT, strategy and transformation from EA. His method has process and maturity (long practice) nailed, but lacks strategic impact, compliance (no transformation, the purpose of EA in the Federal Government), and has partial scope at best. That would be 2 of 5.

GEWERTZ'S METHODS

My friend and colleague Marc Gewertz is on a quest to define his method. It is not fully published but I have seen some of it. His work will be my example of a brand-new method, still shiney and still in the box. His methods have process and scope. His compliance is as yet uncertain, but he has recently redefined roles and it looks a bit gloomy in that direction. Strategic impact should be high, but maturity is low (its new). I will estimate 3 of 5, but its preliminary.

(There are a great many little "grow your own" frameworks and methods that would not fare as well as Marc's, as he has education and background in this area. Some might score very low for this application, as low as 1. Pick one and try it.)

DOD'S METHODS

Hey, process completeness is pretty good if you include JCIDS as part of EA. Compliance has some issues in the area of CPIC and other corners, but is otherwise not too bad. Scope is great. Maturity is high. Strategic impact is questionable, as DODAF and JCIDS do not seek to fulfill the strategic plan but instead focus on mission. My call is 4 to 4.5 of 5, so flame me via email.

FEA

The US Government's Federal Enterprise Architecture, taken as an umbrella, has a huge range of documentation and method described. The process model is nearly complete with guidance in the areas of enterprise, segment and solution architecture, maturity management (OMB, GAO and others), and governance (CPIC, early guidance on other areas). Maturity was high, but recent discontinuous changes have been setbacks (FEAF II does not show enough commonality with FEAF 1.x, new OMB Circular A-130, various other bits). Strategic impact is high, if you bother to do it right. It is the standard for compliance. Again, I would call it 4 to 4.5 out of 5.

CONCLUSION

We have briefly evaluated the methods of different enterprise architecture "experts" and groups for use in Federal Government practice. None were perfect. To get better you ***must*** adopt a method

and then improve it, tailor it for your organization. Much of DoD's success has been through this process. Combining bits of this or that is best, and as I have pointed out parts of up to 5 frameworks may be required for completeness.

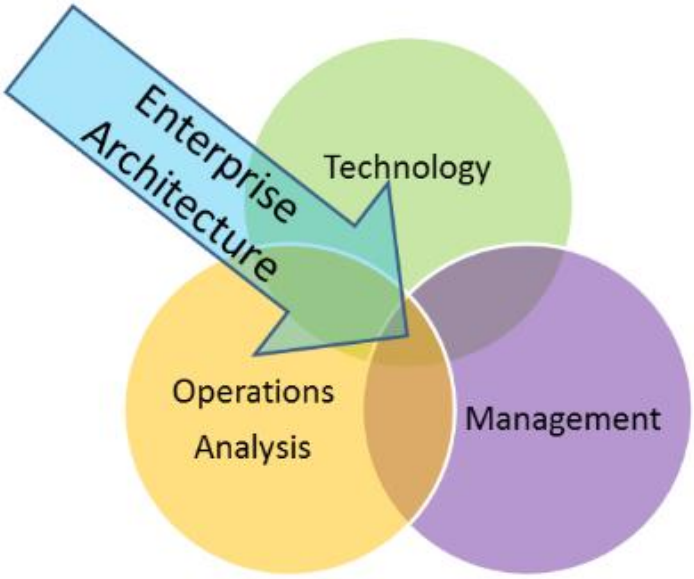
You may notice that the methods designed to work in the US Federal Government seem to score best for use in the US Federal Government. That should be obvious. You must tailor enterprise architecture methods for your particular use, over time. Maturity counts, don't switch things around too much or you fall backwards.

Over 20+ years there has been a massive methodology and framework shakeout, and most methodologies have been obsoleted. Roger Sessions did a comparisonof frameworks some few years ago, only covering the four still relevant in his eyes. (He left out DODAF and included Gartner. He also reviewed FEA instead of FEAF, 3 apples and an orange, but we all have our viewpoints.) The days of new EA methodologies are long over. In general, ignore the smaller frameworks and cultish methodologies, unless you have some strong reason to use one, as they are probably doomed to remain on the fringe.

The only methods we really need to concern ourselves with these days are the ones that are demonstrably, provably, better in some way than the existing ones we use. All the rest are just a waste of our time. I suggest that you evaluate methods for your particular purposes before you adopt them.

SECTION 4: ENTERPRISE ARCHITECTURE RELATED GOVERNANCE

4.1 EA IS TRANSFORMATION, NOVEMBER 29, 2014



I did a Venn Diagram thing like this before, emphasizing the overlap between the "four pillars of FEAF". This one is different. I'm going to try a new way of explaining this. https://www.linkedin.com/pulse/article/20140816204807-86002769-enterprise-architecture-purpose?trk=mp-reader-card

ENTERPRISE ARCHITECTURE

Enterprise Architecture: The study of enterprise architecture (specificaly at the enterprise level) mixes management, technology and some operational analysis. It is not the study of software structure. It is not. Get over it. If you thought that you were wrong.

EXAMPLES:

National University: You can get a Masters in Engineering Management with specialization in Enterprise Architecture from NU. I did that. All three areas are well covered.

Griffith University: In Australia they have a University with a nice program in EA combining some management study and some technology study (too much software flavor for my tastes). Operational analysis classes are electives. Fair.

Penn State: At Penn State you can get a Masters in EA. The program covers management and technology. It covers logistics and supply chain operations, but not enough business process analysis for my tastes. Fair.

So what do you need to know in these three areas I mention? Well EA is about transformation, keep that in mind. We are transforming the enterprise.

MANAGEMENT

Management: You have to know about several things in this area.

- Strategy, as EA implements strategy (but does not produce the strategic goals of the organization)
- Portfolio management (usually implemented adjacent to EA but closely tied in)
- Budgeting (a wee bit)
- Planning and scheduling of huge efforts over years of time (transformation is often not rapid)
- Cost management, value management and or ROI
- Business plans, reading and evaluation thereof
- Organizational change
- Governance

OPERATIONAL ANALYSIS

Operational Analysis: I broke this out as a separate category to emphasize it. In EA you have to know about the following:

- Business Process Reengineering and Process Improvement
- Supply chains, value chains, markets and customers, distribution
- Metrics
- Evaluation
- Alignment: efficiency and inefficiency, effectiveness and ineffectiveness
- Security
- Vision: how the place will operate when you are done

Operational analysis is key. The point is to improve enterprise operations. Any technology employed is to improve enterprise operations. Management is targeted at the enterprise operations that must be improved.

TECHNOLOGY

Technology: This is always overemphasized. Further the specific areas EA is involved in are rarely made sufficiently explicit. Lets see if I can do better.

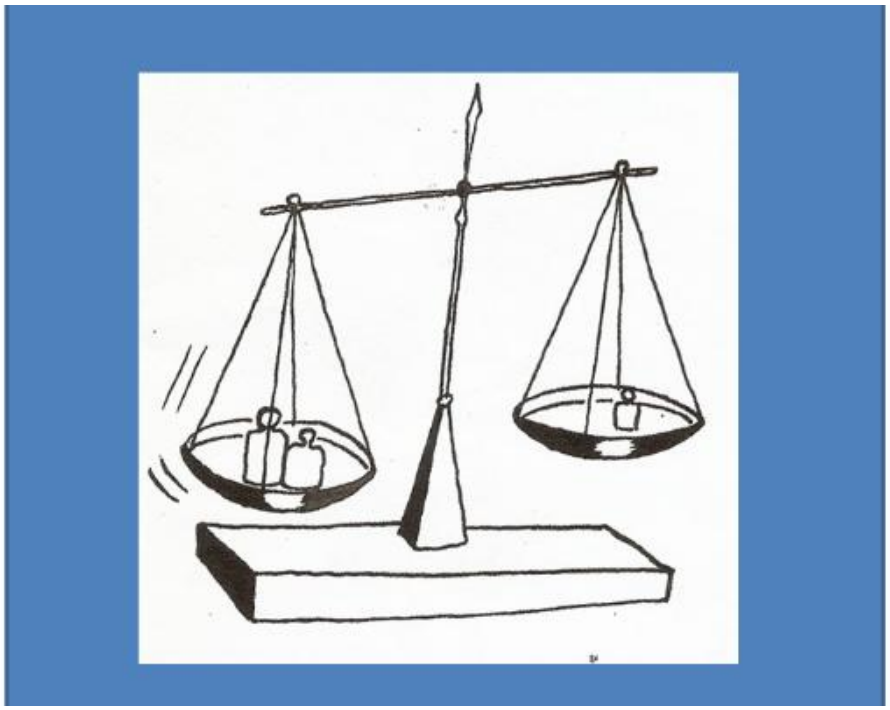
- Integration and integration technologies: SOA, ESB, EAI, MOM, interconnection and interaction, workflow systems
- Functional coverage of applications and solutions- what they do not how they do it. Identification of redundancy or gaps
- Data architectures, conceptual data models and overlaps in coverage
- Infrastructure, servers and networks and capacity
- Security
- Standards

- Management of a list of accepted or approved products
- Principles: what is good in a solution

If you are looking for more detail in the work you do as an architect, go to the segment (LOB) or solution level. It is not in here.

By making operational analysis explicit, it is possible to focus the dialog on operational improvement. This may allow a greater clarity in description of EA. It is an experiment in description, of a sort.

4.2 ENTERPRISE TRANSFORMATION VS IT, JANUARY 31, 2015



What is the relationship between transforming the enterprise for greater performance, and information technology?

THE TRANSFORMED ENTERPRISE

Once someone identifies the strategic goals, and supporting new or changed operational missions or lines of business (lines of service, product lines) are identified, and the vision of the new enterprise is described, you then have to construct that new, transformed enterprise and have its component parts meet the performance goals established.

This may require completely new business processes, new assembly lines, new lines of service. Alternately the old assembly lines, lines of service and business processes may be changed. They may have steps simplified or eliminated by new tools, capabilities or by automation. They may require **Material Solution.** If only changes to process and procedure are required, and no new stuff is needed then you have a **non-material solution.** Either way that solution may be designed by **solution architecture.**

MATERIAL SOLUTION

Any material solution, or set of new things needed to support new processes, new service or product production can be categorized asIT material solution or non-IT material solution. A non-IT material solution might be a new fleet of trucks, or a new building. An IT material solution contains software and hardware and peripherals of various kinds. You must choose to ***build or buy*** any material solution, including an IT material solution.

IT MATERIAL SOLUTION

An IT material solution will support the new processes, production of products, and delivery of services of the new transformed enterprise. It will be evaluated by if it succeeds in supporting the new performance goals for producing products, delivering services or improving processes.

THREE KINDS

There are three main kinds of material IT-solution involved in supporting enterprise transformation. These are listed here (you just may not ever see it anywhere else, so pay attention):

1. Enterprise Software
2. Integration
3. Data Integration

These three cover 80% or more of transformation activity support through IT. The rest is mostly covered by peripherals and devices, like large format plotters for engineering documents or cell phones or PBX systems combined with the items above.

ENTERPRISE SOFTWARE

Enterprise software is a large number of forms connected to a big database. Apps may be included in the package. This software automates processes, designs or manages product production, automates and delivers all or parts of services. These can be commercial off the shelf (COTS) or custom. It may be SAAS (Software As A Service) as well.

INTEGRATION

Integration connects enterprise software packages together, or to devices or peripherals. Integration uses integration technology such as an enterprise service bus (ESB), enterprise application integration, EAI, message oriented middleware, service oriented architecture, workflow packages and BPM, CORBA, MS Pipes, etc. Integration can use web services or RPCs or something else entirely. Integration automates processes across enterprise software applications.

DATA INTEGRATION

Data integration is the name used here for dashboards to monitor performance measures, data warehouses, DataMarts, big data collections, sweeping and collecting data from the internet or social media, or whatever. This can support transformation as well by improving performance management or collecting new kinds of data to create new efficiency.

CONCLUSION

It is overly simplistic to say that IT is enterprise transformation. That position is nonsense from the uninitiated. There are both material and non-material solutions. Among material solutions there are IT and non-IT solutions. Among IT solutions three kinds dominate.

It is very short sighted to concentrate on IT only, although most transformation will include and involve IT in some portion of the effort. The future may place other technology at the forefront, say genetically modified organisms or cyborgs or self-contained self-directed robots treated as non-IT capital expenditure or biotech mechanisms to grow your products (say beer or medicines). In the

future these kinds of investments may dominate transformations of enterprises, and commoditized IT (agile or not) may become far less important.

4.3 EXAMPLES OF ORGANIZATIONAL TRANSFORMATION, FEB 15, 2016

We speak of transforming the organization to increase performance. You may wonder what we mean by that. Here are a few examples:

- FedEx had no nationwide overnight shipping. They decided that they would produce that service. To do it they created a national hub where all the airplanes flew to each night, then out again. They also created a barcode and computer and conveyor line infrastructure to route packages to the right airplane. With that and a bit more they captured the overnight shipping business.
- Ford Motor Company had a quality problem in the 1980s. They created a program and some policies to change the problem. They started measuring quality in new ways, changed suppliers, changed processes, and implemented continuous change to maintain that. Today they have no quality problem.

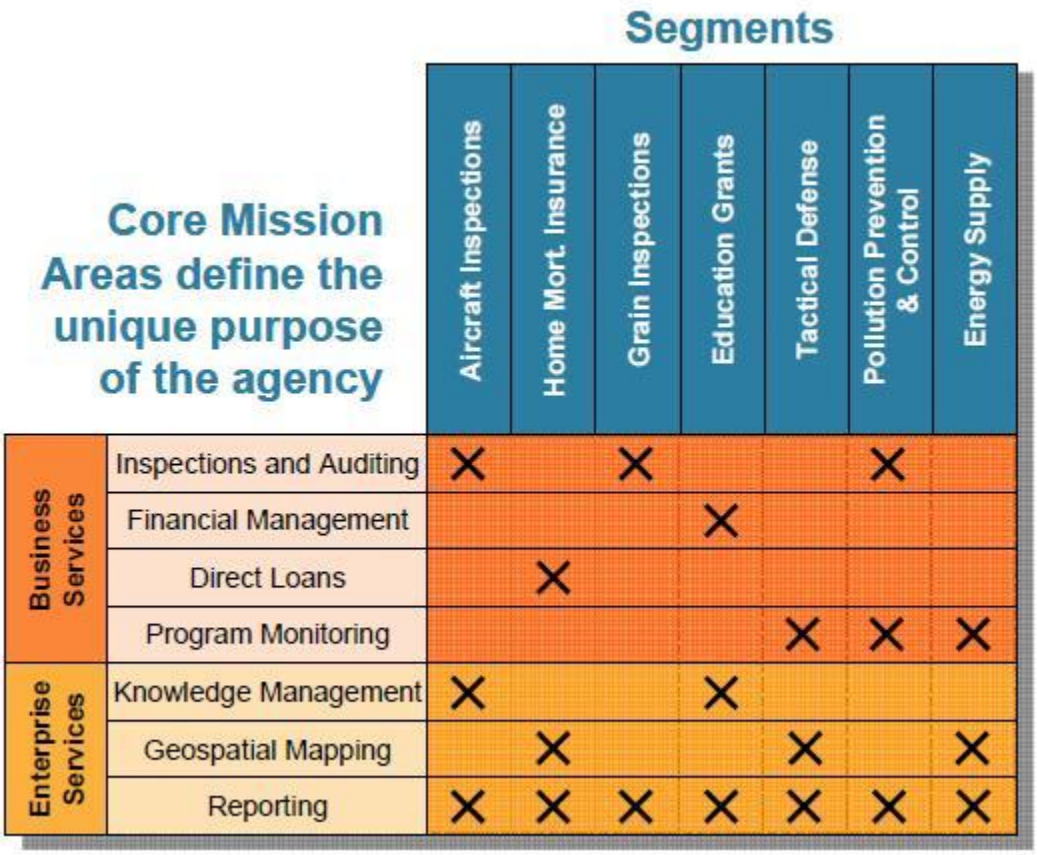
- DoD changed how it did logistics through a process called Business Transformation Architecture.
- Verizon quality initiatives have allowed it to dominate the cellular business.
- Automation at R.R. Donnelly saved millions and improved turnaround time.
- Progressive Insurance improved its operations creating significant growth not shared by other companies in their industry.

There are many examples of such transformation, from sweeping down to narrow scoped change. Change can be focused on different objectives. You can rank these objectives in importance in categories as follows:

1. The most impactful kinds of transformation implement a new line of business or business model. Amazon and FedEx did this, as did those who invented the automobile and the personal computer. This is "revolutionary" or "discontinuous" improvement.
2. Creating a means to improve over time falls in this middle ground. Efforts like TQM, Lean, Kan Ban or Agile allow you to keep changing the enterprise over time. This is "continuous" improvement.
3. The least impactful transformations are single occurrence improvements. You just perhaps save costs or eliminate a redundancy. You do the same things, just less expensively, or create an incremental new product.

Notice the lack of discussion about IT. IT is a supporting element in organizational transformation. One does not seek to buy IT, one seeks to improve the organizational performance which icidentaly requires some specific IT.

4.4 WHERE TO START, SEPTEMBER 20, 2014



<The image shown is from the 2006 FEA Practice Guidance of the US OMB.>

You desire to use enterprise architecture to transform your organization and achieve higher performance. Where do you start?

In 2005 and 2006 a gentleman named Richard Burk wrote practice guidance for US Federal Government enterprise architecture. He had two key insights. I have written elsewhere about the three level model of EA, but we are not looking at that here. We will stick to the middle level, the "segment" architecture, organized around a "line of business".

NOT ALL EQUAL

Burk identified that not all such "Segments" or "Lines of Business" are equal for the purposes of transformation. Some encompass the "core mission areas" of a government agency, analogous to the "core business" of a commercial organization. By improving those areas you can directly improve the performance of your organization.

Burk's second great observation is that to improve organizational performance, focus on the "core mission" segments (core business). These core business areas are usually delegated to some operational unit and not the central office of the corporation or government department. These are where the greatest potential gains are found.

SUPPORTING SEGMENTS

Supporting segments, those not related to the core business, come in two flavors according to Burk. These are 1) Business Services and 2) Enterprise Services.

BUSINESS SERVICES

Human resources, financial management, and legal services are examples of potential business services. These often represent opportunities to save money and consolidate at the central office or headquarters. The same goes for the department (not bureau) level in the Federal Government. Such centralization is a secondary target for performance improvement, and really only saves cost. Once centralized, no matter how good your HR function may get (assuming it is adequate), it is unlikely to have a strong impact on your core business.

ENTERPRISE SERVICES

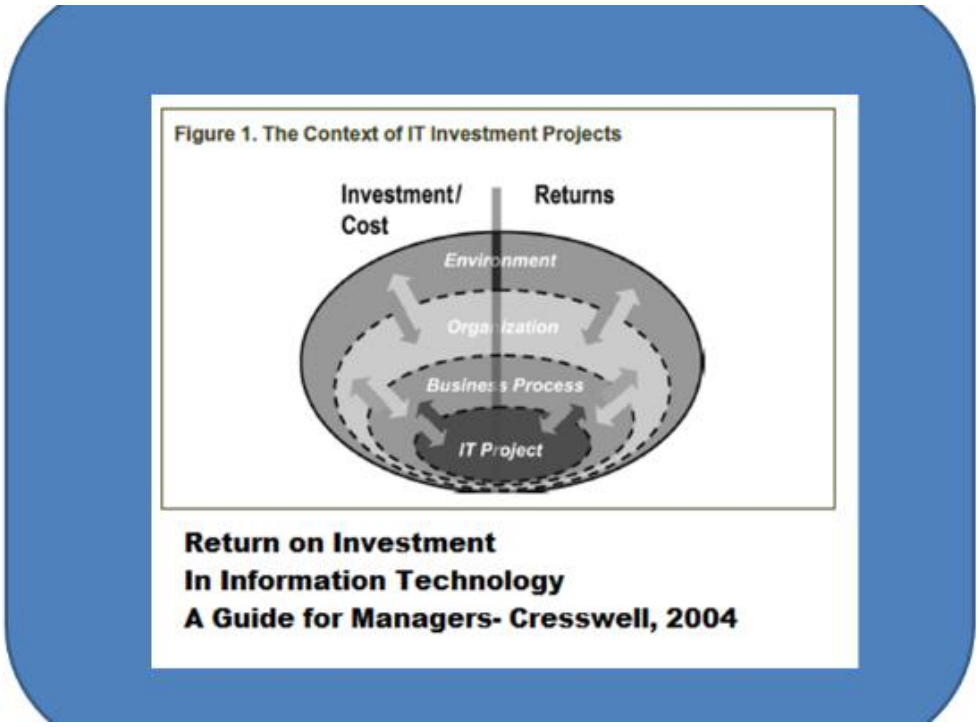
This seems to be an attempt by Burk to simplify a bit of poor terminology used earlier in the federal EA. Enterprise Services seem to be the same as Cross Cutting Measures. In both cases these are bits of technology infrastructure that touch nearly every operation in the enterprise. Examples are IPv6, your ESB or integration technology and your Data-Mart/ Data Warehouse/ Dashboard systems.

To Burk this is a kind of segment. To those familiar with the older nomenclature, these efforts are orthogonal to segment investments. In either case, they are geeky technical efforts that can improve operations everywhere. This kind of investment is the tirtiary target of improvement, last in line.

ANSWERING THE QUESTION

Where do you start to improve with enterprise architecture? Start with your primary target, core business segments. When you get that under control, move to centralizing business services and moving them out of lower level organizations. Do not focus on performance improvement above adequacy of those business services. Lastly look at cross-cutting measures aka enterprise services, things like an ESB or a dashboard. This is the order in which you will see the greatest improvements first.

4.5 THE BIG PICTURE OF EA



Some various people have posted on the "big picture" of enterprise architecture. I thought I might try to add a bit of clarity and reality to this subject. The "big picture" in EA has to do with achieving the strategic goals of the organization, as expressed in the strategic plan. These goals support achieving the organizational mission.

If you tell me the strategic goals of the organization, then I can tell you the primary goals of its enterprise architecture.

People seem to get confused, thinking EA is different things in different places. It is all the same thing once you grasp this picture. You are providing support for the mission of the organization by achieving the strategic goals through detailed execution. As John Zachman said: "Architecture is the thing between strategy and execution". You may have heard it here first, but now you know- so stop making up weird new stories.

"Architecture is the thing between strategy and execution." John A. Zachman

You can paraphrase this approach by describing this support as improving the performance of the organization. That means we perform the mission more effectively. That means execute the

strategic goals, which usually provide new performance targets. That means executing the objectives beneath the goals, and the subdivisions of that, all the way down and with their respective organizational performance measures or KPIs (Key Performance Indicators). Systems supporting the organizational improvement have MOEs (Measures of Effectiveness), MOPs (Measures of Performance) and TPMs (Technical Performance Measures) right out of the INCOSE guidance aligned as described here. System performance itself means little until it supports organizational performance improvement, they are related by operation.

So let's look at that process from the bottom up. You may have a project, and because you may think EA is always about writing code you want to know how your software relates to the big picture. A guy named Cresswell wrote a paper about it, about ROI in software investments His illustration above is similar to mine below, as used in other posts.

Taking Cresswell's point of view, the software supports a business process. The process supports the organization in producing something. That something is delivered to the environment (thus realizing the organizational mission in a way that supports the strategic plan). (There are methods bypassing the business process analysis and reengineering, but the business process method is by far the most common.)

If you do this thinking like a computer science geek, you say you have a portfolio of software development efforts. If you do this thinking like an enterprise architect, you say you have a portfolio of organizational transformation investments, and some of those might have a software component as a part of the effort.

Making the software (or other widget or tweak) have a lasting effect on organizational performance is an **OUTCOME**. Fitting all the pieces together to achieve that outcome is ***alignment***. The set of concepts at different levels from strategic goals down to software features is called the **LINE OF SIGHT**. An architecture mixing organizational performance (business) of the

whole enterprise with technology (including but perhaps not exclusively software) is an***enterprise architecture***. Each change initiative made to improve the performance of the enterprise is an ***investment*** in that enterprise. Because that changes the enterprise it is ***transformation***. The measurable outcomes produced from the whole effort, including software and business process and policy and other changes, is **RETURN ON INVESTMENT (ROI).** Because the enterprise is large and has many products and pieces, the enterprise architecture often has many **COMPONENT ARCHITECTURES**. The whole is named enterprise architecture, as it the topmost architecture in the hierarchy of component architectures (by convention).

This whole notion was constructed in government with the aid of top experts from industry. That means some of the language is slanted towards government. For example, measuring ROI requires both tangible and intangible costs and benefits, but in government tangible benefits are few (besides cost savings), and so an emphasis is places on measuring organizational performance as return. (Note that system performance is not return until it supports organizational performance.) This all works perfectly fine in commercial space. If a strategic goal is top line or bottom line growth, or cost reduction, you can achieve and measure that.

If you can identify to me what success the organization has had in achieving its strategic goals using technology, and how it has improved from those efforts, then we will be talking about the ROI (Return on Investment) of its enterprise architecture.

Sometimes organizations ask the question "Should we do enterprise architecture?" Most often their organizations are currently attempting to apply ubiquitous technology to achieve their organizational goals and support their mission. Therefore they are perhaps doing some trivial kind of enterprise architecture, using amateurs, maybe scribbling on paper napkins, maybe keeping notes in their heads, and usually doing it very badly. They may achieve very poor results (low ROI). The discipline of enterprise architecture as practiced by qualified enterprise architects offers improved methods, repeatability, and rigor and should improve ROI.

To fully eliminate enterprise architecture of any sort, try eliminating any expectation of operational relevance for your technology initiatives and avoiding all types of business transformation. Sometimes organizations do ask the question "We have been doing enterprise architecture and it does not seem to work, should we stop doing it?" Clearly if it is not working you are using poor methods, have left out key components or are using unqualified people. Simply achieving business (mission) goals, and achieving ROI for your transformational investments, is not magic nor impossible if you are serious about it. If you refuse to expend any effort on enterprise architecture you will simply do it badly, not achieve your goals, not improve mission performance.

Sometimes people call enterprise architecture by some other name, perhaps with some partial scope like "digital transformation". You can call it Suzy but it is still enterprise architecture.

That is the "big picture" of enterprise architecture. There is little more to it, at this summary level. It is not different everywhere (although implementation, mission and goals may be different), and it is about 25 years old (not new). It may be a bit unusual to see it all in one place, this "big picture". If you had not heard of it, that does not mean it did not exist.

4.6 THE ENTERPRISE TRANSFORMATION OFFICE, FEBRUARY 8, 2015

A PMO or Project Management Office is an element of the enterprise that maintains project management standards. Enterprise Architecture plans and monitors enterprise transformation. The two may be (and have sometimes been) combined in an EAPMO.

Other approaches to enterprise transformation or improvement fall far short of the breadth of vision and potential gains of agility, efficiency and effectiveness offered by the EAPMO organized to combine oversight of all transformative projects. I suggest the term **Enterprise Transformation Office or ETO** for this kind of organization.

PMO: The project management office (1,2) commonly maintains standards for project management, program management and portfolio management. It is a single unique operation within the enterprise. As the enterprise should have only one portfolio to avoid suboptimization, there is discussion as to if the PMO should perform the portfolio management for the organization.

EA

Enterprise Architecture, as described in FEA and FEAF, is decision support for organizational portfolio management. It is closely connected to portfolio management in its main or top-level process. (3,4,5,6,7,8,9, 10) EA and portfolio management should be organizationally close, tied together, to achieve gains through transformation.

TRANSFORMATION

Transformation changes the enterprise, improving operations or creating new capabilities. The primary means of causing this transformation is the project, possibly organized into programs, and managed by the portfolio. This is differentiated from functional or operational management which

oversees steady-state operations in the enterprise. The primary purpose of the PMO is to achieve organizational transformation. (11,12,13,14, 15, 16) Again, organizational proximity would help.

PORTFOLIO MANAGEMENT

There are strong reasons to integrate, or tightly relate the portfolio management function with its enterprise architecture decision support. These are intended to operate in a tight cycle. If you also agree that the PMO should perform the organizational portfolio management, then there is significant potential synergy in combining the PMO, EA and portfolio management in a single organizational unit. (17,18)

Such an organization would contain experts in the distinct disciplines of EA, project management and portfolio management, presumably in distinct sub-units, located together for organizational efficiency. It would possibly be headed by the person responsible for organizational transformation.

When these functions are not combined significant synergy, efficiency and effectiveness can be lost.

(In the US Federal Government portfolio management is called CPIC. The relationship between EA and portfolio management is described in OMB Circular A-130, and this relationship is the policy of the US Federal Government. Program and project review is also describe there as the "control" and "evaluate" phases of CPIC. Compliance with law and policy requires that these be treated together for all IT. Best practice would have these functions combined for non-IT as well, all transformational efforts of any kind.)

ROADMAP

A roadmap is a schedule for transformation efforts. The enterprise transformational roadmap is a central artifact for both a PMO and EA. (19) Combining would again improve effectiveness.

WASTE

Placing EA in the PMO can reduce waste. (20,21)

GOVERNANCE

Unifying EA and PMO governance can save effort. (22,23)

STANDARDS

The PMO promotes and possibly audits for standards applied to projects and programs. EA is intended to be the standards clearinghouse for the organization in various frameworks. This function can and should be unified. (24)

SDLC

The system lifecycle is a standard that should be managed and enforced by EA and PMO efforts, together. (25)

VISION\

A key product of EA is the vision of the target (to-be) enterprise. This is what the programs and projects are attempting to create. There is bennefit in having that function performed in the PMO. (26)

EA PRINCIPLES

Principles describe how projects and programs will perform transformation. There is benefit in having the PMO select and promote EA principles. (27)

STRATEGY

EA seeks to create a portfolio that implements the strategic plan. The PMO also seeks to create and management the portfolio of projects and programs to implement the strategic plan. There is benefit in unifying or centralizing these functions. (28, 29, 30)

BUSINESS AND TECHNICAL DRIVERS

EA tracks the forces that drive organizational change, attempting to accommodate, leverage and even anticipate these forces. It is advantageous to merge this function into the PMO. (31)

INNOVATION

An ETO style EAPMO can reduce barriers to innovation and improve agility. (32)

Acquisition or Procurement: Each project or program must select implementation methods for its bit of organizational transformation. Some efforts may be outsourced. Some may be purchased. Some may be developed within the organization. Many require purchase of supporting software or other elements. With an ETO/ EAPMO acquisition or procurement is tactical support for the strategic efforts of the ETO, applied to the project or program. In effect, the ETO implements "strategic acquisition" or replaces any notion of or independent implementation of "strategic acquisition". However the ETOhas greater potential to do so in an effective and compliant way vice modern current organizational standards. (33)

AUDITS AND IV&V

The ETOis the proper place to sponsor or conduct audits and IV&V (Independent Verification and Validation) Efforts, in addition to program or project reviews and progress reporting. (34)

CONCLUSION

If you perform portfolio management, enterprise architecture and management oversight of programs and projects then you should probably combine these functions into an Enterprise Transformation Office style EAPMO. There is significant overlap of function, and significant synergy to be achieved.

REFERENCES:

1. http://www.slideshare.net/Gartner/gartner-enterprise-programme-management-office-functions-and-capabilities

2. http://en.wikipedia.org/wiki/Project_management_office
3. https://www.linkedin.com/pulse/article/enterprise-segment-solution-kern-msem-bsee-cea-cissp-issap-itil-pmp/edit
4. https://www.linkedin.com/pulse/article/parsimony-vs-comprehensive-ea-matthew/edit
5. https://www.linkedin.com/pulse/article/20141129174141-86002769-ea-is-interdisciplinary/edit
6. https://www.linkedin.com/pulse/article/20141004183219-86002769-choose-five-frameworks/edit
7. https://www.linkedin.com/pulse/article/20140816204807-86002769-enterprise-architecture-purpose/edit
8. https://www.linkedin.com/pulse/article/20140816144513-86002769-enterprise-architecture-scope/edit
9. https://www.linkedin.com/pulse/article/20140622135614-86002769-enterprise-architecture-concepts/edit
10. https://www.linkedin.com/pulse/article/20140622135425-86002769-enterprise-architecture-origins/edit
11. https://www.linkedin.com/pulse/article/enterprise-transformation-matthew-kern/edit
12. https://www.linkedin.com/pulse/article/integration-system-engineering-enterprise-matthew/edit
13. https://www.linkedin.com/pulse/article/20140920150603-86002769-where-to-start/edit
14. https://www.linkedin.com/pulse/article/20140920122233-86002769-transformation-investment-portfolio/edit
15. https://www.linkedin.com/pulse/article/20140803161040-86002769-the-value-of-enterprise-software/edit
16. https://www.linkedin.com/pulse/article/20140723111655-86002769-enterprise-transformation-methods/edit
17. https://www.linkedin.com/pulse/article/enterprise-architecture-portfolio-management-matthew/edit
18. https://www.linkedin.com/pulse/article/20140826222008-86002769-what-is-suboptimization/edit
19. https://www.linkedin.com/pulse/article/20140920032120-86002769-enterprise-architecture-roadmap/edit
20. https://www.linkedin.com/pulse/article/20140802151937-86002769-change-without-purpose/edit
21. https://www.linkedin.com/pulse/article/20140921200744-86002769-10-ways-to-botch-ea/edit

22. https://www.linkedin.com/pulse/article/20140801112523-86002769-enterprise-architecture-governance/edit
23. https://www.linkedin.com/pulse/article/20140802133423-86002769-enterprise-architecture-maturity-management/edit
24. https://www.linkedin.com/pulse/article/20140921131156-86002769-approved-technologies-standards/edit
25. https://www.linkedin.com/pulse/article/20141127015951-86002769-system-engineering-inefficiency/edit
26. https://www.linkedin.com/pulse/article/20140912112037-86002769-enterprise-architecture-vision/edit
27. https://www.linkedin.com/pulse/article/20140921154042-86002769-enterprise-architecture-principles/edit
28. https://www.linkedin.com/pulse/article/20140919090447-86002769-outcomes-vs-capabilities/edit
29. https://www.linkedin.com/pulse/article/20140822173711-86002769-culture-vs-strategy/edit
30. https://www.linkedin.com/pulse/article/20140719145138-86002769-strategic-operational-and-tactical-thinking/edit
31. https://www.linkedin.com/pulse/article/20140912125408-86002769-enterprise-architecture-drivers/edit
32. https://www.linkedin.com/pulse/article/20140830141415-86002769-innovation/edit
33. https://www.linkedin.com/pulse/article/20140704182755-86002769-cots-gots-vs-mots-vs-custom-developemnt/edit
34. https://www.linkedin.com/pulse/article/20140726190101-86002769-performance-measures-in-the-enterprise/edit

4.7 MANAGEMENT LEVELS & ENTERPRISE ARCHITECTURE

The enterprise contains three kinds of management. Beneath executive leadership functional management handles ongoing operations. All activities with a beginning and end are transformational.

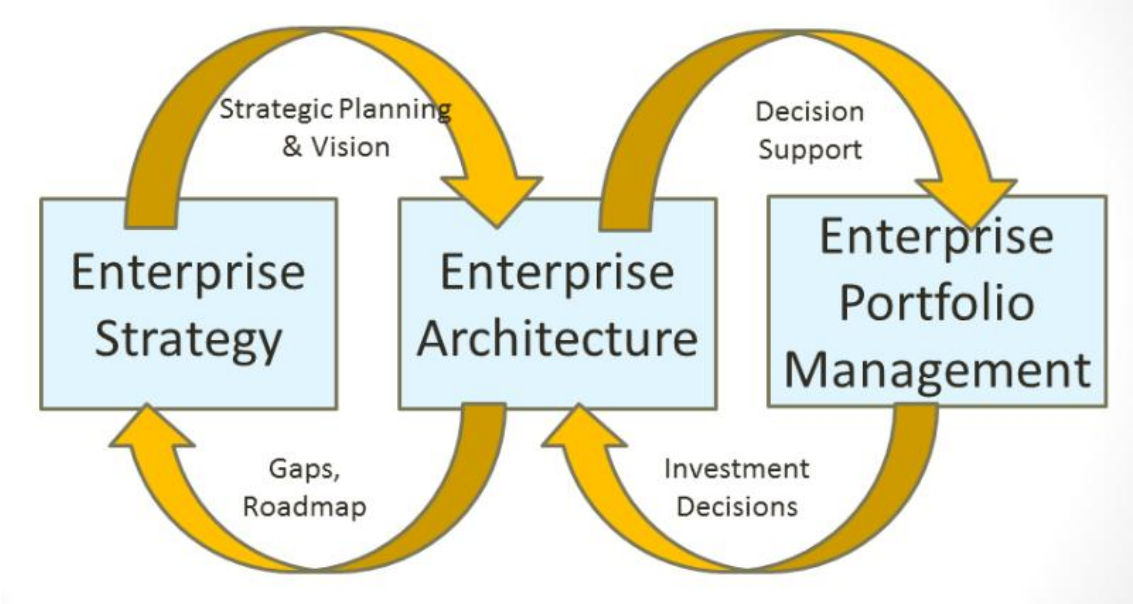
Management Level	Architecture Level	Planning Level	Results	Impact
Portfolio	Enterprise	Strategic	Strategic Outcomes	Highest
Program	Segment	Operational	Operational Improvements	Moderate
Project	Solution	Tactical	Capabilities	Lowest

Let's look at the relationship between management and enterprise architecture. The odds are that your organization has some executive leadership. Most do, though in some you might wonder. Beneath executive leadership are two kinds of management. Functional management runs ongoing operations, anything that is not changing in the period of interest is ongoing operations. Projectized management runs transformational activities, anything with a beginning and end. (This is a bit of a simplification, but will do for our purposes.)

In projectized or transformational management there are three levels. These are portfolio management, program management and project management. You can find these in the PMBOK. (As for PRINCE II, you are on your own. I'm not certified in that, and do not write about it.) The three levels of architecture originally come from the FEA and FEAF 1.x framework. Here I will describe the natural relationship between these PMBOK and FEA levels, which I have seen from performing each and all of them over my career.

PORTFOLIO MANAGEMENT

Portfolio management is the top level of transformational (projectized) management. It has the greatest impact on the organization. In the US Federal Government the portfolio management function is called CPIC. The portfolio should be driven by organizational strategy and the strategic planning function.



Between the strategic planning function and the portfolio management functionlies enterprise level architecture. This is one activity of five in enterprise architecture. As John Zachman identified, architecture lies between strategy and execution, and here you can see that implemented in practice. The enterprise level architecture applies the meaning of "enterprise" as the whole organization. Enterprise level architecture is broad but shallow, and can be made very lean.

In the US Federal government this relationship between strategy and CPIC (portfolio management) has been policy for 15 years and more, and is now law. This is perhaps because portfolio management is so very effective, and government acquisition based on needs has been so very ineffective in terms of costs and results.

(Note: This level corresponds roughly to one alternative view of enterprise architecture purpose in LePalme's 2012 paper: **"Effectively execute(s) and operate(s) the overall enterprise strategy for maintaining a competitive advantage by aligning the business and IT strategies such that the proper IT capabilities are developed to support current and future business needs."**)

Just as the portfolio consists of multiple programs and projects, the enterprise architecture sits above multiple segments and solutions. If implemented well there can be a direct correspondence.

PROGRAM MANAGEMENT

A program consists of multiple projects that are related to the same goals. The program has projects which transform a product line, service line, or line of business. Program management is supported by segment architecture. There arethree kinds of segments, and by implication three kinds of associated programs. These are "core mission", "business services" and "enterprise services".

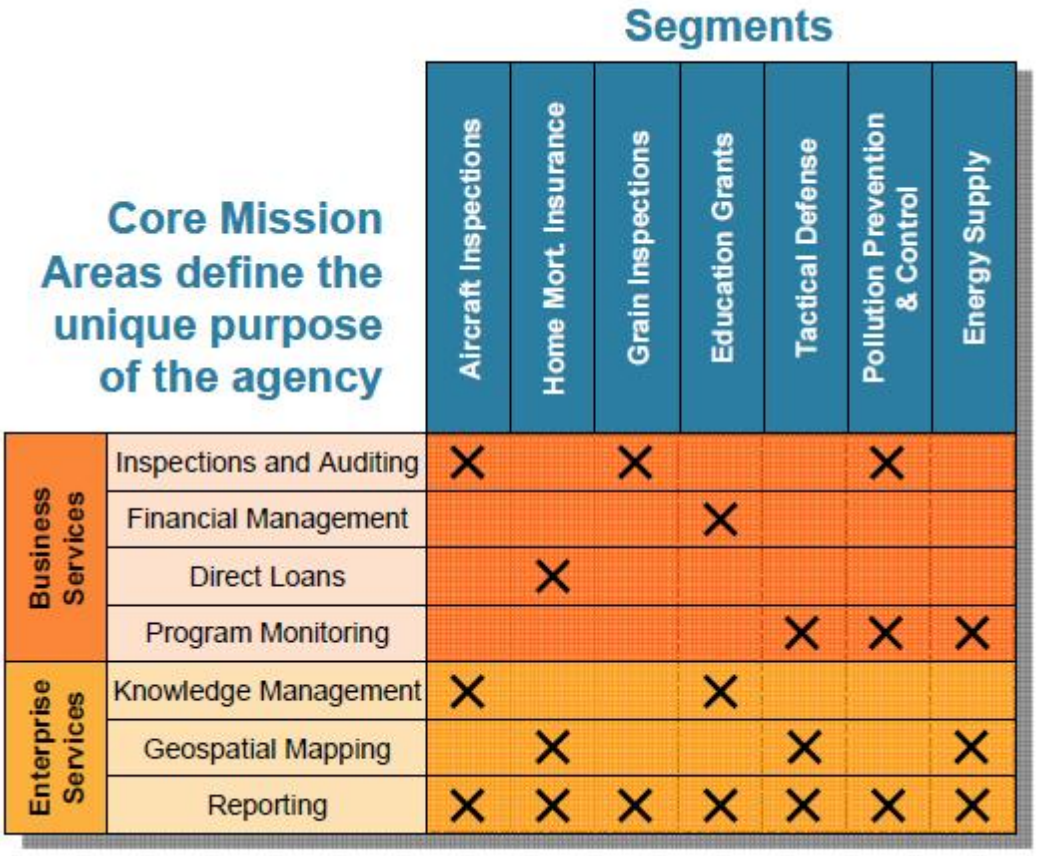


FIGURE 3 IMAGE FROM OMB 2006 FEA PRACTICE GUIDANCE, WHICH SEE.

(Note: In total all the segments should cover all the organizational functions. This level corresponds to LePalme's alternative purpose for enterprise architecture stated as: "EFFECTIVELY IMPLEMENT(S) THE OVERALL ENTERPRISE STRATEGY BY DESIGNING THE VARIOUS ENTERPRISE FACETS (GOVERNANCE STRUCTURES, IT CAPABILITIES, REMUNERATION POLICIES, WORK DESIGN, AND SO ON) TO MAXIMIZE COHERENCY BETWEEN THEM AND MINIMIZE CONTRADICTIONS")

Segment and solution level architecture use that other meaning of "enterprise", referring to an undertaking. They are part of the broad meaning of enterprise architecture, all 5 activities.

PROJECT MANAGEMENT

Project management may produce a system, solution or capability. This management level has the least impact on the organization. Solution Architecture aka system architecture supports project management.

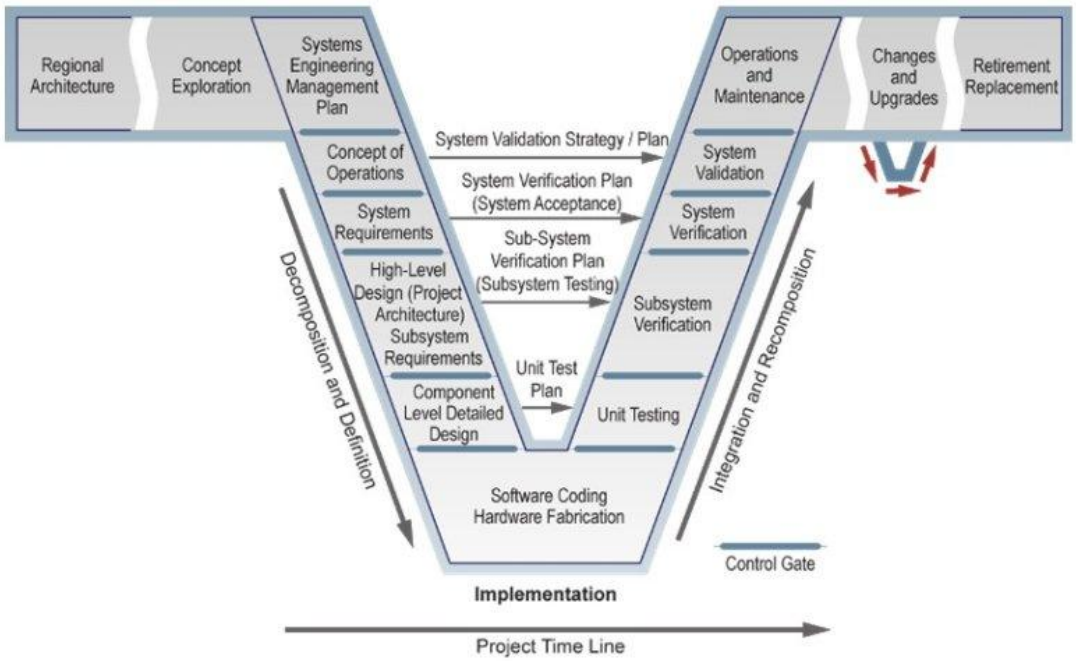


FIGURE 4 (IMAGE FROM DEPARTMENT OF TRANSPORTATION, AS AN EXAMPLE OF ARCHITECTURE WITHIN SYSTEMS ENGINEERING V PROCESS.)

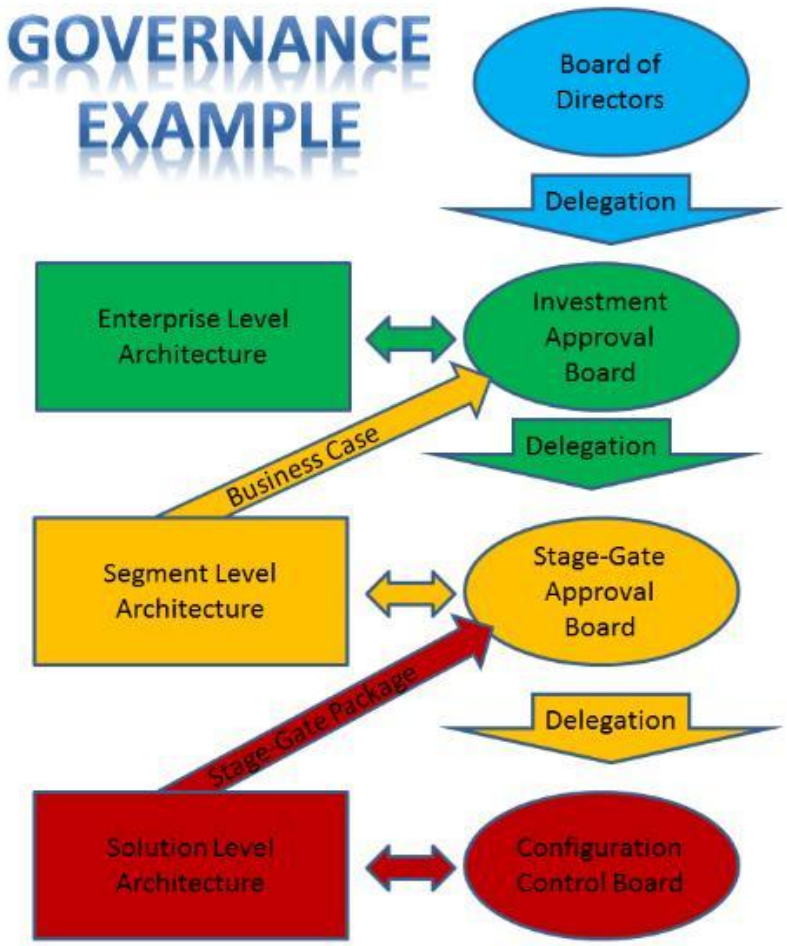
I[illegible] e processes. Management approach should be determined by project type. Management needs a blueprint of what is being produced. Solution architecture involves identifying system and subsystem boundaries and allocating requirements. Solution architectures may vary in quality. Solution architecture takes effort and forethought.

(Note: A properly aligned project and solution architecture will "Help(s) the organization innovate and adapt by designing the various enterprise facets to maximize organizational learning throughout the enterprise" -Lapalme, 2012.)

INTEGRATING LEVELS

Processes for architecture should be managed together for improvement. Levels are coordinated through governance. There may be a central repository, and management of which architectures are approved for inclusion.

4.8 ENTERPRISE ARCHITECTURE GOVERNANCE, AUGUST 1, 2014



Enterprise architecture related governance is an obscure topic for some architects. I will present an example with some simple and direct logic here, hoping to clarify the topic. I will use the FEA as the context for the example, but I will use common terminology from business. In the early literature of FEA 3 governance bodies, committees, were mentioned in relation to enterprise architecture. I will update the names and functions based on subsequent developments, common practice, and mention four committees.

THE BOARD OF DIRECTORS

The ***Board of Directors*** is responsible for top-level oversight of the organization. It has a role in oversight of enterprise architecture and transformation efforts. In early government EA documents an executive advisory board was mentioned, having a vaguely analogous function but existing within a government body.

INVESTMENT APPROVAL BOARD

The ***Investment Approval Board*** has delegated authority from the Board of Directors to approve transformation investments. This body is usually labeled an investment review board, but I want to emphasize its authority to approve investments without further review or approval. It is the body that makes decisions on portfolio management (aka CPIC). Investment decisions ARE related to investment phases, in CPIC these are Select, Control and Evaluate. Select is initial selection a set of transformational investments (portfolio) based on cost, risk and return. It

approves business cases for a transformational investment. It reports portfolio activity to the higher board.

STAGE GATE APPROVAL BOARD

The ***Stage-Gate Approval Board*** reviews project and program documentation to authorize that project or program to proceed (and spend money associated with the next phase). Stage-gates are usually associated with an SDLC, an expansion of the investment life cycle. This board has delegated authority (from the Investment Approval Board) to oversee the "control" and "evaluate" decisions of the Investment Approval Board, the natural function of SDLC oversight. This usually happens by review of documents in a stage-gate package, containing all the information required for the decision. It reports approval activity to the higher board.

CONFIGURATION CONTROL BOARD

The **Configuration Control Board** oversees and approves changes of configuration related to a transformational investment. This may be changes to technological infrastructure or the processes that use it in some cases. Its authority is delegated from the Stage-Gate approval board, to oversee changes between stage-gates.

ENTERPRISE LEVEL ARCHITECTURE

Enterprise Level Architecture Role: These lower boards operate to ensure compliance with the standards, approved technologies, vision and principles published in the EA repository. The enterprise architecture itself is a decision aid to the Investment Approval Board, and is updated based on approval decisions for transformational investments (business cases).

SEGMENT LEVEL ARCHITECTURE

Segment Level Architecture Role: The segment architecture acts as a decision aid to the Stage-Gate approval board, and assists in production of transformational business cases for higher review. Stage-Gate Approval Board activity may result in changes to the segment architecture.

SOLUTION LEVEL ARCHITECTURE

Solution Level Architecture Role: The solution architecture is a decision aid for the Configuration Control Board, and aids in the production of stage-gate packages. configuration control board approvals or denials may update the solution architecture.

Conclusion: Once again this is mostly activity you should be doing anyway in a large organization, not extra work. The main point is how the activity fits together in the context of enterprise architecture.

4.9 ARCHITECTURE REVIEW BOARD

There are many who think they are in the business of enterprise architecture, but are not really and perform only solution level architecture. Those architects may find this material new or perhaps even offensive. For others it will be a description of well-worn daily tools. I will present my take on this topic, a practical and simple approach. I will try not to bore the practitioners, nor confuse those less exposed.

BLUF (Bottom Line Up Front): The Architecture Review Board should have sole and final authority to approve architectures and promote them to a position in the repository. Only approved architectures should be used in the other governance boards (i.e. Board of Directors, Investment Approval Board, Stage-Gate Approval Board, Configuration Control Board).

So then let me explain. The enterprise should have an enterprise architecture repository. This is where all official, reviewed, approved artifacts should reside. Viewing access to it should be ubiquitous to assure compliance. The structure of the repository should echo the hierarchical structure of your architectures.

I have elsewhere described four governance bodies for a minimalist and idealized implementation. These would exist otherwise, despite EA absence, and do not include an ARB. Minimalization is good, as the problem of "governance sprawl" can easily bring your enterprise to a crawl. These review bodies correspond to the three levels of management in transformation (project, program, portfolio). This correspondence leads to simplification and streamlining. As described here andhere, solution architecture supports a project, segment architecture supports a program, and enterprise level architecture supports the portfolio. This avoids one of the most common failures of enterprise architecture: "Boiling the Ocean".

So you have many architectures to contain in your repository. They exist in a hierarchy. Lower level ones must be reviewed for compliance with higher level architectures and other direction. Somebody must review.

You have a few options:

1. You can move to the mountains to meditate, give up EA as too hard, and eat only local vegetables. If you get there first, remember to save other enterprise architects who give up a nice space nearby so you can argue.

2. Review can be based on the honor system. That does not work well in big, complex organizations. Cowboys run off to produce duplicative, poorly aligned, poorly scoped and suboptimized solutions reducing the overall organization's effectiveness to feed their fiefdoms.
3. You can go with the sprawl and have many committees that do almost nothing and waste precious resources, dooming your enterprise to remain ineffective.
4. You can have each of the other boards I have described before review their level of architecture and submit it up. That can work. In that case you do not need an ARB. In my view this is nirvana, but may be difficult to attain.
5. You can have an ARB to review architectures. The post you are reading describes this option.

(Frankly, if you do not have any governance bolted securely to architecture there is little reason to do any higher level architecture. People will ignore it.)

In these options, an orthogonal choice exists as to whether to split the review into technical and management sessions. I argue against the governance sprawl. If you are going to have a board, give them the power to do something. Review should be limited to compliance with the architectures and guidance of higher levels, and should exclude pet rants. Managers at lower levels should retain sufficient authority and leeway to succeed. Excessive constraint can be a big problem.

TOG has a nice description of the responsibilities of an ARB which I cannot argue much with. (If the rest of their stuff was as good, I would be overjoyed.) Who staffs an ARB can be a problem. As John Tieso points out, some political animals must exist on this board and must show up when reviewing the top level architectures. All of the specialities required for complete review should be included somehow. I suggest the "Lean IPT" concept as an approach.

Lastly there is the problem of logistics. Does the board review artifact-by-artifact or in complete packages for example? Put that in the charter you wish you did not have to write. At this point the description seems complete but for details.

I have run an ARB for 2+ years, and participated in a few. It is hard to make an ARB work. Betting your EA program on the success of the ARB is risky. Other means may prove less difficult.

Note that this view differs from that in Rob Thomas & Company in "Practical Guide to Federal Enterprise Architecture", a truly great document which you should read, but it is dated. Much experience has happened since.

Also note that making it all work is about more than acquisition. You must acquire the correct things. See Enterprise Software; The value of enterprise software; Code relevance; & this on moving Beyond mere tactical devOPS...

Another note: There are significant questions about authority and responsibility in an ARB. Portfolio management has the authority to supply or deny funding, and EA supports that authority. Program management has the authority to monitor project progress toward transformation of real operations, including stopping your pet project at the next stage gate, and segment architecture supports that authority. The Project does not have infinite authority to do whatever it likes at its own pace. It is the management that has the authority, and the architecture

at those higher levels express the plans you shall adhere to. This is the basis of authority transferred to an ARB, in reviewing, checking and harmonizing those plans so everything moves in the same direction.

4.10 TRANSFORMATION INVESTMENT PORTFOLIO, SEPTEMBER 20, 2014

Transformation Investment Selection

Investment Name	Risk	Tangible Cost	Intangible Cost	Tangible Return	Intangible Return	ROI
Investment 1	High	1	Low	5	High	High
Investment 2	Low	2.5	Low	10	High	High
Investment 3	Medium	3	Medium	11	High	High
Investment 4	Low	2	Medium	9	High	High
Investment 5	High	4	Medium	12	High	Medium
Investment 6	Medium	3	High	8	Medium	Medium
Investment 7	High	2	Medium	4	Medium	Medium
Investment 8	Low	1	High	2.5	Medium	Medium

Enterprise architecture is dependent on its primary corporate control for success. That control is portfolio management, Lets briefly examine the portfolio management of transformational investments.

TRANSFORMATIONAL INVESTMENTS

Transformational Investments: In enterprise architecture each effort to transform the enterprise is treated as an investment. Each has a business case. The business case identifies the summary of the investment, the major risks, the costs, the returns and the total ROI (Return on Investment). Investments have a lifecycle in which they are selected, then controlled then evaluated after implementation (terminology taken from US Government CPIC or Capital Planning and Investment Control).

Investments should be very large projects, programs, or sets of these. Investments should be broad in scope, spanning perhaps multiple organizational elements (and capabilities if used) to address the improvement of a complete line of business or cross-cutting measure. (see my discussion on these elsewhere.)

RISK

Risk: The risk of an investment is a summary of all the individual risks, impacts and probabilities for that investment. You can "monetize" this by expressing the risk as the statistical ***expectation of costs for mitigation*** The subject of risk management is certainly outside the scope of this small text, but the most important point is that it is managed here in portfolio management, in program management, and in project management. It is also managed in the scope of system engineering, especially in regard to technical risks. It is not particularly addressed within the scope of enterprise architecture itself.

TANGIBLE COSTS

Tangible Costs: The tangible costs are the quantitative identifiable costs of the investment, in terms of dollars. These costs, for the transformation investment, should be total lifecycle costs and not simply implementation costs or purchase price. Tangible costs should include all aspects of the transformation, including labor and reserves based on risk.

INTANGIBLE COSTS

Intangible Costs: The intangible costs of the transformation investment include all qualitative understanding of losses or expenditures related to the investment. This might include goodwill or reputation or lost opportunity.

TANGIBLE RETURNS

Tangible Returns: The tangible benefits include all quantifiable benefits acrued from the investment. In the Federal Government most of the benefits you might expect in industry do not apply, as sales, margins and revenue are not available. However some few Federal agencies are funded by fees. The biggest remaining tangible benefit is cost savings from operations, which is often emphasized. It is good to save those taxpayer dollars.

INTANGIBLE RETURNS

Intangible Returns: Intangible returns are all those non-quantifiable benefits of the investment. In the US Federal Government EA is all about the intangible returns, as performance improvements to business processes in quality and timeliness (and throughput) are all usually intangible. For example reductions in complaints concerning immigration services would be intangible, as would speed of response in a natural disaster.

Intangible returns are measured by performance measures or performance indicators.

RETURN ON INVESTMENT

Return On Investment (ROI): This is the (returns minus the costs) divided by the costs. It is a calculated value. Where most returns are intangible, or where costs are largely intangible, it may be appropriate to substitute a qualified estimate of the ROI as high, medium or low. This often happens in the Federal Government.

INVESTMENT SELECTION

Investment Selection: Early on, the best investments must be selected. Some selection criteria should be officially established and maintained. In the US Federal Government for example, the policy indicates that the highest ROI investments should be selected, without regard to risk.

In the graphic above I show the usual means to select investments. You line them up with in a table with the best at the top. You select the top several until you reach the budget, then stop. Some management reserve should be removed from the budget before calculation of selection. The red line in the graphic indicates the budget level. There are tools to automate this.

The idea is to remove as much subjective tinkering as possible, thus reducing management favoritism, fraud, waste, and abuse of funds.

INVESTMENT CONTROL

Investment Control: Selection only occurs once in a single investments lifecycle. After it is selected an investment is implemented, and during implementation it is controlled. Control involves evaluating the program and project management that implements the investment, using tools such as Earned value Management.

INVESTMENT EVALUATION

Investment Evaluation: Once the investment is in operation it is no longer controlled, and the project or program manager often sent off to implement something else. During operations and maintenance the investment is evaluated. In the US Federal Government evaluation is required yearly by policy. Evaluation involves measuring the performance measures or performance indicators and comparing them both to predicted performance in the business case and the history of performance improvement.

In evaluation, when a transformation investment fails to perform as needed after a period of satisfactory performance an end of life evaluation should be made.

COMMON ERRORS

Common Errors: Common errors in this process include:

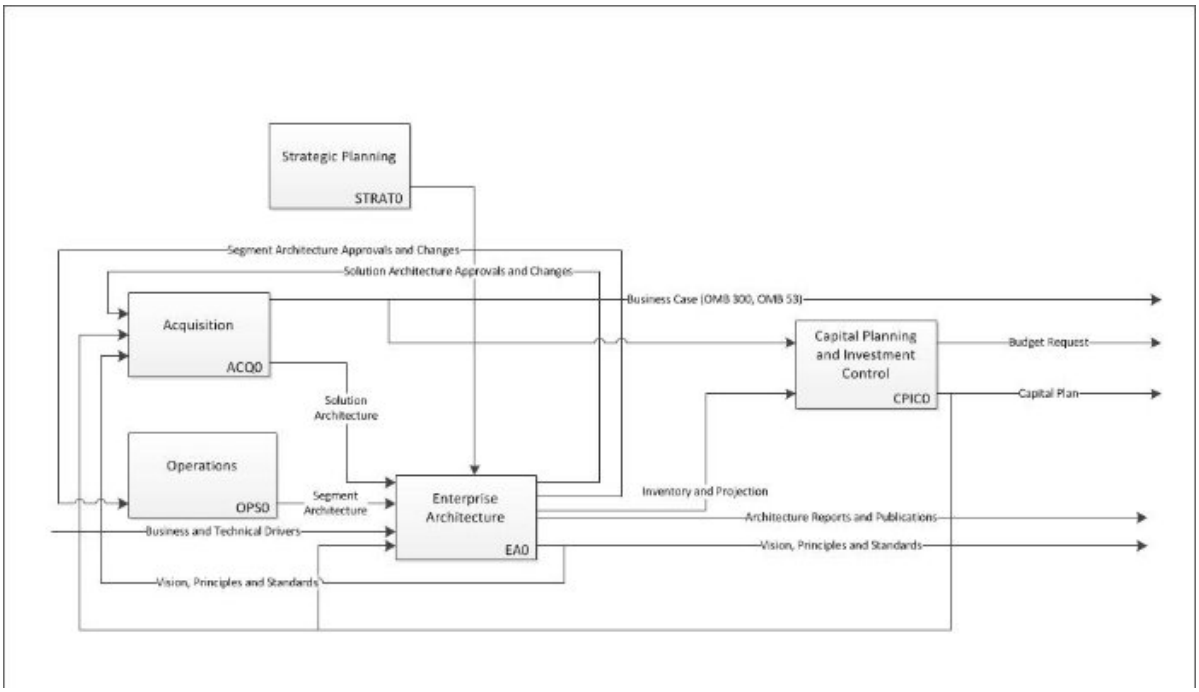
- Not organizing portfolio management as the top level decision authority concerning investment. For example elevating acquisition, a tactical concern, to a strategic level above portfolio management is a common error.
- Thinking small: Investments are large and sweeping. A portfolio of hundreds of investments is unmanageable, and only a few dozen should exist. Each of these should have clear strategic impacts larger than one capability or one performance measure or one project.
- Zombie Investments: This is a term for those transformational investments that have not performed, and for which funding is reduced to the point that they will never produce results of strategic impact. Zombie Investments are common in organizations with weak management that can not ever cancel a program or project. It is a practice that wastes money and resources.

CONCLUSIONS

Conclusions: Transformational investment management is often also called IT investment management, because nearly all transformation initiatives involve business process improvement via automation. Concepts and tradeoffs are very similar to those in the PMBOK for portfolio management, with some small tweaks easily rectified. This is the top process for managing transformation in the enterprise, and EA cannot succeed without it.

This is a brief introduction to the subject of portfolio management for transformational investments. It is not intended as comprehensive.

4.11 ENTERPRISE ARCHITECTURE & PORTFOLIO MANAGEMENT, JANUARY 18, 2015



How are enterprise architecture and portfolio management connected? This is a topic which many practitioners fail to grasp. Here is a diagram, with a simple explanation. We will use the example of the US Federal Government, as documented in OMB Circular A-130, the top policy on enterprise architecture.

PORTFOLIO MANAGEMENT

If you will look at the activity (block) titled "Capital Planning and Investment Control", or CPIC, this block is intended to be portfolio management. In this block you select, control and evaluate investments in enterprise transformation (performance improvement). Note the input labeled "inventory and projection", this is the top level current and target architecture, distilled down to what you have today and what you desire to have at the planning horizon. The gap is analyzed for transformation opportunities.

There should be only one portfolio:https://www.linkedin.com/pulse/20140826222008-86002769-what-is-suboptimization?trk=mp-reader-card

GAPS

The transformation opportunities (gaps) are compared to business cases (marked as OMB Form 53 and OMB Form 300 in the diagram). Those business cases with the highest ROI (Return on Investment are selected. Presumably redundant investments to fill gaps already closed will be excluded.

See here for selection by ROI:https://www.linkedin.com/pulse/20140920122233-86002769-transformation-investment-portfolio?trk=mp-reader-card

CAPITAL PLAN

The output of portfolio management (CPIC) is the capital plan. When the capital plan changes, the target enterprise architecture changes to reflect the new future state. (Note the line back to inputs to enterprise architecture.) The new gaps compared to strategy will be highlighted. (Note the input from strategic planning into enterprise architecture, as a control.) This new architecture is sent to CPIC/portfolio management to describe the current state, future state and gaps as "inventory and projection".

LOOP

This is the feedback loop that powers enterprise transformation. EA identifies what is needed, and corresponding business cases are selected in portfolio management (CPIC). Again, you can find all this in OMB Policy.

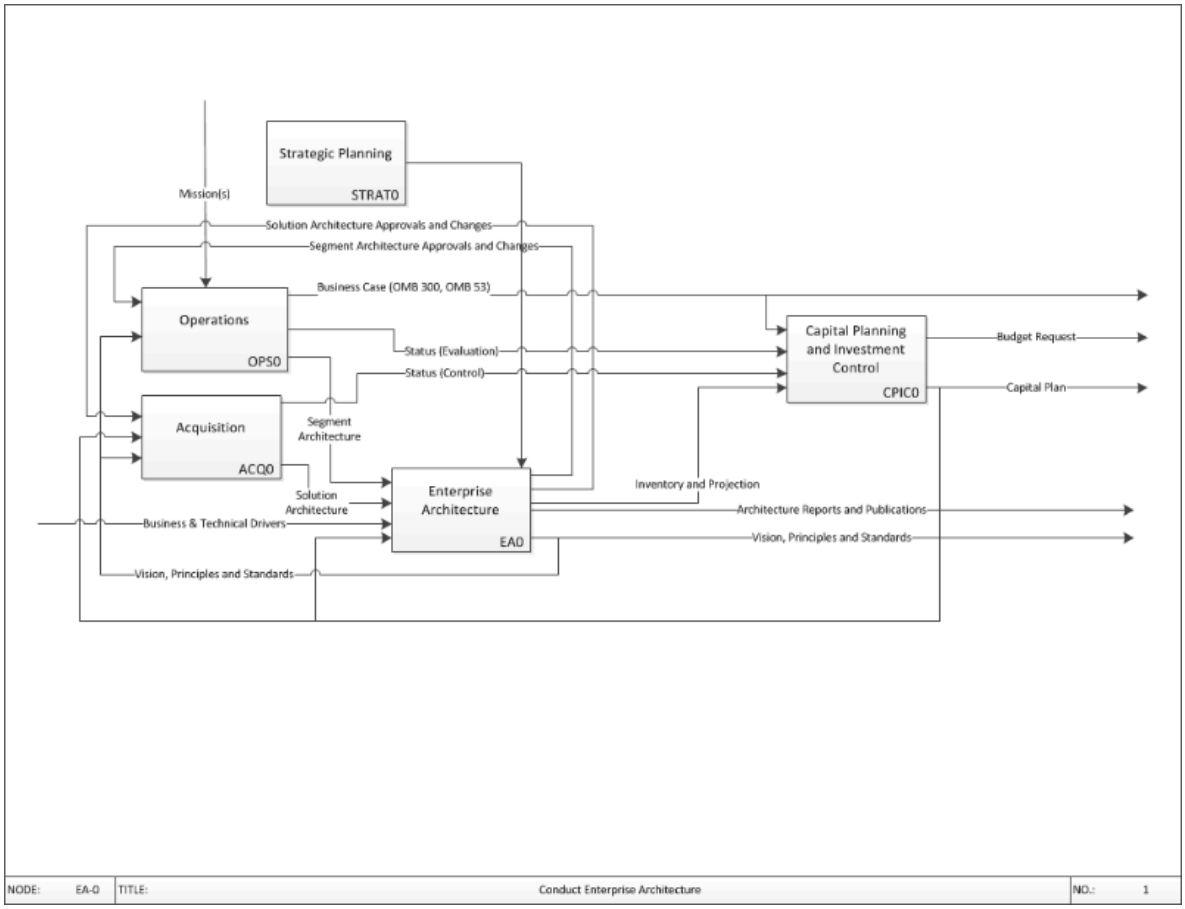
FAULT

What happens if this feedback loop is missing? Well, people develop architectures that do not reflect the capital plan. Also capital allocation does not focus on the gaps identified in EA. In other words, expenditures are unrelated to architecture.

NOTE

The original diagram shows business cases coming from acquisition. This corresponds to the notion of "Big A Acquisition". Such an approach is not recommended outside DoD, and business cases should come from operations with the segment architectures. (DoD only acquisition laws cause this rift.) The astute reader will note that it won't work quite right as depicted above. Operations know what they need, not acquisitions. Acquisition should be driven by EA, and acquisition should not drive EA.

Here is the diagram with the error corrected, and operational missions put into perspective:



CONCLUSION

There it is, the core of EA that everyone should know but few understand. It is simple, really. Now go do good things!

You can implement this in the 3 level FEA model for example: https://www.linkedin.com/pulse/enterprise-segment-solution-kern-msem-bsee-cea-cissp-issap-itil-pmp?trk=mp-reader-card

The top level of EA can then be streamlined as in: https://www.linkedin.com/pulse/20140727145732-86002769-very-lean-enterprise-architecture?trk=mp-reader-card

TERMINOLOGY

Some notes on terminology follow...

1. A TRANSFORMATION INVESTMENT is an approved and implemented business case, hopefully causing improved performance of the enterprise
2. A transformation portfolio is a set of business cases, and is the subject of portfolio management. The other "portfolios" listed below are NOT the subject of CPIC/portfolio management.

3. Transition plan or roadmap is a schedule for implementation of transformation investments, not shown, omitted for clarity
4. CPIC is the term OMB uses for portfolio management
5. A SEGMENT is a line of business, a product line or service line or a mixture. A segment may also be a cross cutting measure, a large reusable technology initiative used by many other internal investments
6. INVENTORY AND PROJECTIONS is a term I have used for the current and target architectures, consisting of...
    - The application portfolio is the inventory of all enterprise applications in the current and target architectures. At the enterprise level this is the application architecture. This name is used by some, but is very confusing.
    - The data asset portfolio is the inventory of all enterprise data stores in the current and target architectures. At the enterprise level this is the data architecture. This name is used by some, but is very confusing.
    - The business activity portfolio is the inventory of all enterprise operational functions. At the enterprise level this is the business architecture. This name is used by some, but is very confusing.
    - The technology portfolio is the inventory of all approved enterprise technologies It is embodied in an artifact called a TRM and another called a standards profile. At the enterprise level this is the technology architecture. This name is used by some, but is very confusing.
    - The information exchange portfolio is the set of standards used for information interchange used at interfaces. It is "data in motion" as opposed to "data at rest" in the data asset portfolio. This does not exist in FEA but did in NIST, and it is a good idea if you are using SOA.
    - Segments, the list of lines of business (mission and supporting) and cross-cutting technology initiatives reused by other investments.

4.12 MANAGING TRANSFORMATIONAL INVESTMENTS

In the US Government the new FITARA law emphasizes portfolio management. In the US Government such portfolio management has gone by the nomenclature CPIC (Capital Planning and and Investment Control) for many years. Since the CIO Act, the federal CIOs have been responsible for transformation of the organization, and CPIC has had the notion of IT as embedded in investments to improve (transform) the organization.

This stuff is more than easily adopted by industry. It can also be easily adopted by other governments. To be effective both CPIC and enterprise architecture need each other, so if you do implement thes bolt them together. CPIC without EA is blind, and EA without CPIC has no real effect on the organization.

To do this portfolio management aka CPIC has been described as having three phases. These have been named Select, Control and Evaluate. Let me describe some of the basics of how each is performed.

SELECT

In the select phase you choose what to spend money on, what to invest in. This approach is in contrast to older, inferior methods based on surveying needs and managing acquisition as if it were the mission itself. To select transformation initiatives you compare business cases.

Each business case describes its cost, its return on investment, its risks and its schedule. Each should state break-even dates, outcomes, and return on investment. Apples to apples, side by side, you compare these and select those with the best return, given reasonable risk. Simply needing something is not enough, as there are many needs and fewer funds. You must compete and produce the documented best way to serve the mission, not just the way with the political backing.

You know what risk is reasonable by comparing investment risk to the organizational risk tolerance. You know if the investment produces return in reasonable time by comparing to the organizational planning horizon.

Costs and returns will have a tangible and intangible portion. In government, where the top line and bottom line are absent, intangible returns will be predominant other than cost savings.

Enterprise architecture can tell you which investments cover which strategic goals, and which will be left unaddressed. An enterprise architecture roadmap will also tell you which goals will be addressed in which year.

What you do not do is survey needs and compare those needs. Return outweighs needs. You do not produce need statements, business plans subsume that information.

CONTROL

The "control" phase applies to the period of time during which you are implementing your investment, and achieving the outcomes desired by constructing any supporting capabilities. In this phase you monitor the programs and projects that implement these things.

Monitoring construction consists of two parts: monitoring management and monitoring technical progress. The former is accomplished by tools like EVMS and management reviews. (In EVMS it is important to monitor when the outcomes will occur, not just activity as in Agile. Otherwise this tool becomes worthless.) The later is commonly accomplished by an SDLC, SELC and stage-gates. An SDLC has phases of completion, and before you can move th the next phase thus spending more money the proper results from the prior phase must be reviewed and accepted. If any of this is missing, you are doing a poor job monitoring.

Monitoring supports control. The control occurs when you KILL or END project and program funding for non-performing projects. You do not reduce funding to produce "zombie projects" underfunded and incapable of ever producing a return. Zombie projects are a form of government waste and abuse of funds.

EVALUATE

Once the construction or implementation is completed you move the thing to operations. The evaluate phase applies to operations and maintenance. In this phase you monitor that the KPIs of the organization are being affected as predicted in the transformational business plan. When the organizational performance falters, something has gone wrong and corrective action must occur. A new investment may be required, marking end-of-life for the current investment.

To evaluate organizations may use operational evaluations, scored exercises, dashboards, analytics and similar tools. Enterprise architecture can analyze and identify if a transformational initiative has met end of life. The objective is to kill an ineffective investment, stop spending money on it.

CONCLUSION

Whereas in the past government waste in acquisitions was rampant, the methods of CPIC aka Investment Management offer a more modern and data centric approach. Billions of dollars in taxpayer funds might be saved. (If applied in industry, that would be investor funds.)

This stuff is not hard to describe, nor to perform. However parties desiring to retain control over public funds, retain power over pet initiatives, or repurpose funds and resources to serve their careers will resist these methods. Penalizing such actions would improve acceptance.

4.13 ENTERPRISE ARCHITECTURE VS THE BUDGET, JUL 2, 2015

Figure 2. Planned Maturation of Agency Enterprise Architecture

How does enterprise architecture drive the budget? Does it drive the budget at all? I will describe some of the basics of the intended approach of the US Government.

Let's talk about a generic government entity. You can translate for commercial industry, it will not be hard. The budget consists of three kinds of items. There are transformational effort expenses, operations expenses and miscellany (read as overhead).

- **Transformational Effort Expenses** are synonymous with all internal projects and programs, of all kinds. The purpose of any such internal effort is to improve the organization.
- **Operations Expenses** are related to conducting business, or performing the mission. They are not used to improve the organization, but are related to supplying the service or goods that are the purpose of your enterprise.
- **Miscellaneous Expenses** are all costs neither operational nor transformational. Senior management hours are such a cost.

We will break out costs this way to illustrate some important (and obscure) points. You're agencies' accountants probably have other breakouts, for other purposes. (In government transformational costs are large, BTW.)

How to link EA and budget, as intended:

- The purpose of enterprise architecture is to plan your transformational efforts. Your planning horizon for EA should be far enough into the future for you to cover the procurement and implementation of transformational efforts and their budgets.
- Each year you should have a new EA with the proper planning horizon to produce the budget.
- The link from EA to budget is portfolio management (aka CPIC in government) which will select all the transformational efforts to be implemented, and all those to be discarded.
- All selection decisions in portfolio management should cause update of the EA and the "roadmap" or "transition plan", which should cover from now to the planning horizon.
- Only transformational efforts in the architecture, approved by portfolio management, should be implemented (see exception below).
- Each budget should have a discretionary transformational component. There should be a CIO discretionary budget, for example. It should never be the largest component. Policy should set the maximum size, perhaps by percentage. When that limit is exceeded reviews and audits should occur, and lawyers should show up. Congressmen and Senators should be informed, if Federal Government.
- The OMB Form 300 is to be used as a budget request. Some are approved by the investment board, some rejected. The approved ones are to be submitted to OMB, and the budget should reflect those submissions.

HANKY PANKY

There is an enormous amount of hanky-panky (unusual deviation) in the budget of most government agencies. CPIC and EA were meant to help correct that. But political influence still drives the budget more than rigor and method.

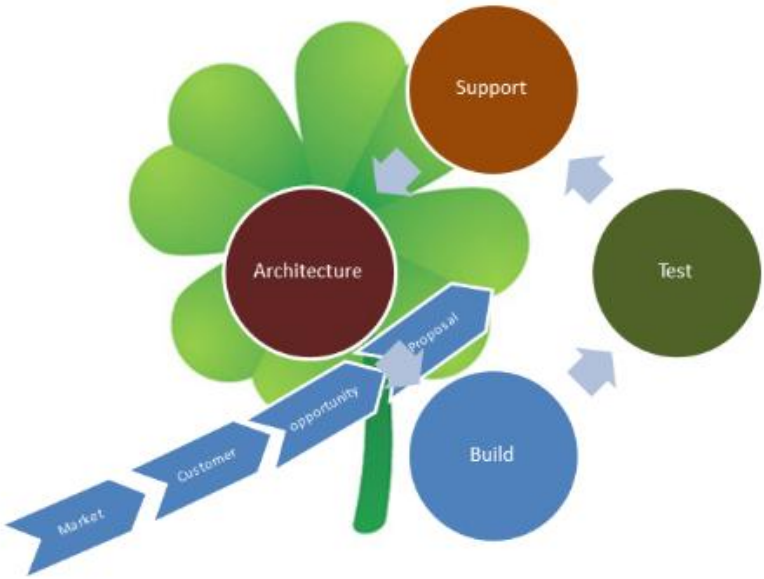
To help fix that, miscellaneous expenses should be kept to under some target. The ratio of transformational to operational expenses should be managed and analyzed. This ratio should generally align to how badly the agency performs now, and how well it is projected to perform. A simple, single clear hierarchy of accounts should lead from any outlay to the transformational expense item that covers it.

Proper application of enterprise architecture in your organization should reduce operational expenses (per unit delivered), reduce miscellaneous expenses (again per unit), improve quality of service (or product) to the taxpayer/customer, or increase the volume (capacity) of service (product) provided. Any single transformational effort (investment) should be expected to contribute to one or at most two of these categories. Those who are confused about the ROI for enterprise architecture as applied in the US Federal Government have missed this point. I attribute widespread confusion on this point to equally widespread lack of proper enterprise architecture education.

These benefits from correctly implemented EA, integrated into governance, can also be expected in commercial organizations if desired. There is no theoretical barrier.

Good Luck! For more official USA OMB information on this topic, please look here.

4.14 INTEGRATION AUDITS USING CLOVER, JANUARY 31, 2015



The Clover (tm) method is a simplified means to perform audit checks of an integration project or program evaluating documentation and process adequacy. It covers the complete integration lifecycle quickly, while using a completely different structure to assure that you look at the problem anew, without the suppositions and expectations of the existing management context (even if that is very sparse or nonexistent). It is effective cheap, and relatively quick. It is proven, having been in use since the 1980s.

> Clover asks the question: How are you managing this integration effort? If you perform integration as a service to others, it can equally ask: How do you manage the technical part of this business?

LIFECYCLE

In Clover the integration lifecycle is divided into six stages. If one of the stages does not apply, do not use it. The stages are:

- Capture Customer (Pipeline)
- Architecture (Cycle)
- Build (Cycle)
- Test (Cycle)
- Support (Cycle)
- Remove Customer (Pipeline) (not covered here)

NOTES:

1) The four stages Architecture, Build, Test, Support form a meta-cycle. However all four stages are assumed to continue concurrently.
2) The Remove Customer stage is used rarely for unprofitable or troublesome customers. It will not be further described here.
3) Al six stages have four sub-stages. Always four. This helps you memorize the thing for easy use.

LISTS OF SUB -REAS:

- Capture Customer: Identify Market, Identify Customers, Identify Opportunities, Propose
- Architecture: Current Inventory, Target Inventory, Plans, Design
- Build: Design, Construct, Test, Evaluate (covers both Agile and Waterfall development just fine)
- Test: Vendor Integration Test, Customer Integration Test, Cut-Over Test, Performance Evaluation (covers both incremental and full scope testing)
- Remove Customer: not covered here.

AREAS OF INTEREST

Identify the areas of interest you want to assure are covered in your analysis. Start with software development, data management, infrastructure, business process and technology. Optionally ad culture and human factors or aesthetic design. Consider adding military DOTMLPF topics: http://en.wikipedia.org/wiki/DOTMLPF

MATRIX

A Clover matrix is formed with the sub-stages as rows and the interest areas as columns. In each cell of the matrix you list the documents and processes that support that cell, that interest area at that lifecycle stage. Fill in the matrix.

Use some notation to show if processes are broken or documents incomplete (or junk documents). A good example is to list these in red. A single document can cover multiple cells, within reason.

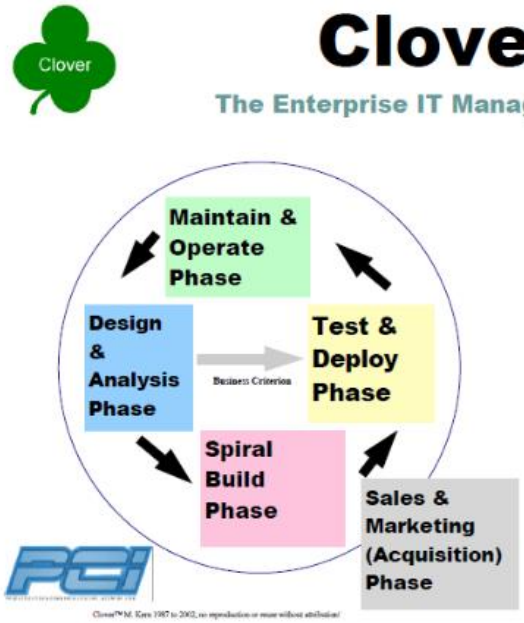
ANALYSIS

One complete you look at the clover matrix, and you note what cells have no processes or documentation. These are gaps. Produce a report showing the gaps, listed by lifecycle stage. Include in the report cells supported by broken processes incomplete documents.

CONCLUSION

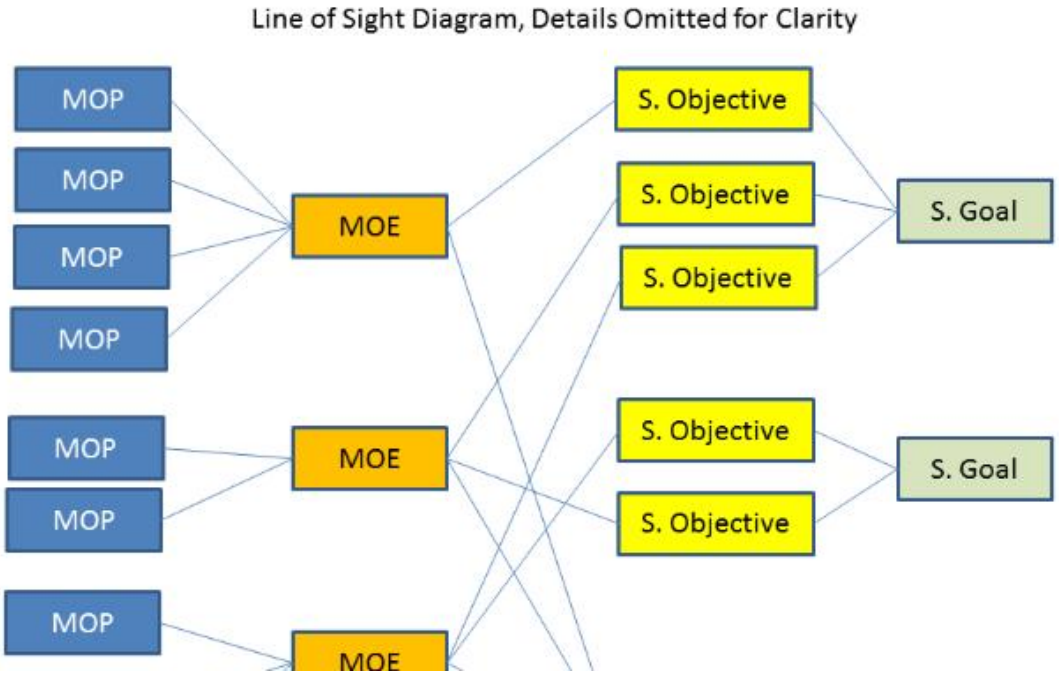
Using the Clover method you can analyze an integration operation and write the report in a very short period of time. The Clover method works very well for a fast and coarse grained 3rd party evaluation. (I once evaluated CMMI activities using Clover, for example. It took a few days, part time. The results were very enlightening.)

If you perform integration and you have no cross-check mechanism like this, you ought to.



Sub-Phases and Areas					
(Checklists)	**Network**	**Platform**	**Software**	**Data**	**Business**
AS-IS	CHK1-1N	CHK1-1P	CHK1-1S	CHK1-1D	CHK1-1B
Vision	CHK1-2N	CHK1-2P	CHK1-2S	CHK1-2D	CHK1-2B
TO-BE	CHK1-3N	CHK1-3P	CHK1-3S	CHK1-3D	CHK1-3B
Architecture	CHK1-4N	CHK1-4P	CHK1-4S	CHK1-4D	CHK1-4B
Design	CHK2-1N	CHK2-1P	CHK2-1S	CHK2-1D	CHK2-1B
Build	CHK2-2N	CHK2-2P	CHK2-2S	CHK2-2D	CHK2-2B
Mitigate	CHK2-3N	CHK2-3P	CHK2-3S	CHK2-3D	CHK2-3B
Test	CHK2-4N	CHK2-4P	CHK2-4S	CHK2-4D	CHK2-4B
ALPHA	CHK3-1N	CHK3-1P	CHK3-1S	CHK3-1D	CHK3-1B
BETA	CHK3-2N	CHK3-2P	CHK3-2N	CHK3-2D	CHK3-2B
GAMMA	CHK3-3N	CHK3-3P	CHK3-3N	CHK3-3D	CHK3-3B
DELTA	CHK3-4N	CHK3-4P	CHK3-4N	CHK3-4D	CHK3-4B
Operate	CHK4-1N	CHK4-1P	CHK4-1N	CHK4-1D	CHK4-1B
Plan	CHK4-2N	CHK4-2P	CHK4-2N	CHK4-2D	CHK4-2B
Remediate	CHK4-3N	CHK4-3P	CHK4-3N	CHK4-3D	CHK4-3B
Evaluate	CHK4-4N	CHK4-4P	CHK4-4N	CHK4-4D	CHK4-4B
Market	CHK5-1C (customer) and CHK5-1V (vendor)				
Advertise	CHK5-2C (customer) and CHK5-2V (vendor)				
Prospect	CHK5-3C (customer) and CHK5-3V (vendor)				
Propose	CHK5-4C (customer) and CHK5-4V (vendor)				

4.15 LINKING EFFORTS TO OUTCOMES, SEPTEMBER 26, 2015



Let's focus on software development for this discussion. As I have mentioned elsewhere there is a profound error in thinking embedded in both Agile Software Manifesto and in DevOps: Volume of working software alone is not a sufficient measure of the operational value in either method or in implementations using those methods. To measure projected outcomes and the operational value of software development you also need **CODE RELEVANCE**.

GigaSLOCs of completely irrelevant code only measure waste.

How do you demonstrate code relevance of your development efforts? How would you measure code relevance? Renewed, redoubled efforts to assure that you are not simply wasting money and development effort are at the heart of the new law FITARA to manage Federal Government spending, and this will probably be followed by the same in other governments at all levels, and in industry. (The concepts for good technology investment management have been around for many years, but a bit of publicity always helps.) If your coding efforts do not produce relevant operational outcomes, they are waste. (Note how well this concept fits with the stated goals of DevOps.)

Outcomes are measured or demonstrated improvements in your business or mission operations (not in your software development productivity, unless the enterprise mission is software development). If you produce cars as an enterprise, you might produce them for less, with better quality, and produce more- these would be outcomes. (A capability, on the other hand, is for example the potential to produce cars by some means, and we will not be discussing capability here. Capability is not relevant to the analysis.)

MEASURING

Outcomes that your enterprise desires should be stated (listed) in your strategic plan, as strategic goals and objectives. To demonstrate linkage of your software, you want to demonstrate how it supports achieving these.

In business and government we measure the performance improvements leading to achieving these strategic goals and objectives by Key Performance Indicators or KPIs.

One widely accepted way to measure a system is by identifying Measures of Effectiveness (MOEs) and Measures of Performance (MOPs). The International Council on Systems Engineering (INCOSE) published a guide on measuring systems that describes these.

How would we link strategic goals and objectives to MOEs and MOPs, through KPIs?

(Note: If you do not like MOEs and MOPs you can use however you characterize software locally. However it is important to make that believable, or your evaluators may not be convinced.)

THE LINE OF SIGHT DIAGRAM

There is a well-known and venerable enterprise architecture artifact used to visually link systems to outcomes. It is quite simple, and you might use it to demonstrate that your code is relevant to operational outcomes, that your software development has worth. (This may help you survive FITARA and similar efforts that make comparison of your efforts value against that of others.)

In the Line of Sight Diagram you have columns. The left column could be MOPs. Column two could be MOEs. (Hint: Put each item, MOP or MOE, in a little box in the appropriate column.) You can draw lines from your system MOPs to your system MOEs indicating which belongs to which, because as INCOSE points out these are linked in a hierarchy.

On the right side of the line of sight diagram would be strategic goals and objectives. Usually objectives are breakouts of goals. (Again you might use boxes.) Have two columns and put goals on the far right, and draw lines to show linkage.

All that is left is to show linkage from measures of effectiveness to strategic objectives. You can use a middle column for the organization's KPIs, if they have sufficient coverage and are related to strategy. Sometimes the KPIs published are garbage, and clarify nothing. Sometimes no one bothered to measure the KPI. In either case you can make up new ones or skip this middle column. Note that making a believable case is harder without good KPIs.

Now draw the remaining lines between MOE and objectives, optionally passing through KPIs.

(The Line of Sight Diagram demonstrated "Alignment". Code Relevance is the projection of the concept of alignment into software development.)

ANNOTATIONS

If you just list the name, and not the actual value of the measurement, you prove nothing. Place the actual measurement in the box, wherever known. If a change is indicated, an improvement, reduction, increase, or whatever place both measurements in the box. Note the percentage change. Where improvements or values are projected, note that with a different color or other indicator.

Describe in text how the links show real support. This is especially relevant in the area between MOE and objective. For example is the goal is higher product quality, and the objective is fewer customer complaints, then describe exactly how the improved help system in the software (MOE) will reduce complaints.

MATH

You can compute the strength and logic of the linkages between elements. Assign a believability to each link (line) in the diagram, from weak to strong, say as 1 to 3. Then multiply all linkages in one path from end to end to get a number. Add all the paths' numbers. This is a measure of relevance.

You can also measure the believability of MOPs, MOEs, KPIs and such the same way, multiplied by the degree of improvement in each. This is an alternative measure of relevance. (If you multiply the two it may be best to take the square toot of that product.)

To demonstrate value of your efforts take your code volume (per the theory in the manifesto) and multiply it by relevance of the effort. Evaluators can compare your number to other numbers from other efforts. You can use this number as a proxy, a stand-in for ROI (Return on Investment) if you divide this by cost. You get relevant effort per dollar spent. This helps when actual ROI is difficult to calculate because of intangibles, or it can be used in addition to calculated ROI as a cross-check. The later is best.

Projects and programs with the lowest ROI, or by proxy the lowest relevant efforts per dollar, should be cut to make way for efforts with better return- if there are such candidates. This is portfolio management. If the portfolio is managed by transformational initiative, all the component projects and programs in the initiative should be summed up before comparison. Some will have software development, others may not.

ANOTHER WAY

Here is another way to demonstrate line of sight.

1. Get your as-is and to-be business process diagrams, and highlight the portions being automated or streamlined by your software.
2. Write a narrative showing which strategic goals and objectives are supported by these process changes.
3. Crete artifacts connecting each process change to a software module, screen or what have you.
4. Calculate or estimate the projected improvement in time, throughput or cost produced and list this as ROI. Compare it to the total costs, software development plus retraining and whatever else, to make that change.

CONCLUSION

Volume of working code is not a sufficient measure of operational value. You can prove linkage of your software development to the goals of the organization, your code relevance. Projects and programs can be compared side by side and which ones to cut can be made obvious.

With a bit of effort you can make investment decisions on quantitative or pseudo-quantitative evidence. You can be objective. Anyone can do it.

Note: This artifact, by itself, does not prove a lack of redundancy in the effort or in the spending. It is also not a measure of risk. It does not demonstrate good project management or project schedule and spending at completion. It does a good job at identifying alignment, code relevance, and delivered value with present data. Other data will be required for your investment decision.

One last note: Agile and DevOps are said to arise from thinking like KaiBan, Six-Sigma, Lean. In those methods there is an analytical effort to manage quality that takes a bit of calculation, and without this you are not performing the method correctly. If you do not perform some calculations on quality in Agile or DevOps, your justification of these methods becomes very tenuous. Quality in this sense is fitness for use in improving operations.

4.16 FITARA: NEW CIO AUTHORITY, SEPTEMBER 19, 2015

I have already heard people misquoting what is in FITARA. Here is the real scoop on the new CIO authorities, with my commentary.

PUBLIC LAW NO: 113-291 (12/19/2014). AS PASSED

"SUBTITLE D--FEDERAL INFORMATION TECHNOLOGY ACQUISITION REFORM"

SEC.831. CHIEF INFORMATION OFFICER AUTHORITY ENHANCEMENTS. (THIS SECTION DISCUSSED BELOW.)
SEC. 832. ENHANCED TRANSPARENCY AND IMPROVED RISK MANAGEMENT IN INFORMATION TECHNOLOGY INVESTMENTS.
SEC. 833. PORTFOLIO REVIEW.
SEC. 834. FEDERAL DATA CENTER CONSOLIDATION INITIATIVE.
SEC. 835. EXPANSION OF TRAINING AND USE OF INFORMATION TECHNOLOGY CADRES.

SECTION 831, CIO AUTHORITY, AS PASSED

Quoting the law, as written, sans BS...

> ``(B) ADDITIONAL AUTHORITIES FOR CHIEF INFORMATION OFFICERS.-- ``(1) PLANNING, PROGRAMMING, BUDGETING, AND EXECUTION AUTHORITIES FOR CIOS.-- ``(A) IN GENERAL.--THE HEAD OF EACH COVERED AGENCY OTHER THAN THE DEPARTMENT OF DEFENSE SHALL ENSURE THAT THE CHIEF INFORMATION OFFICER OF THE AGENCY HAS A SIGNIFICANT ROLE IN-- ``(I) THE DECISION PROCESSES FOR ALL ANNUAL AND MULTI-YEAR PLANNING, PROGRAMMING, BUDGETING, AND EXECUTION DECISIONS, RELATED REPORTING REQUIREMENTS, AND REPORTS RELATED TO INFORMATION TECHNOLOGY; AND ``(II) THE MANAGEMENT, GOVERNANCE, AND OVERSIGHT PROCESSES RELATED TO INFORMATION TECHNOLOGY.

The CIO will have a "sufficient role"- a mushy statement subject to various interpretations. This will have little effect by itself.

> ``(B) BUDGET FORMULATION.--THE DIRECTOR OF THE OFFICE OF MANAGEMENT AND BUDGET SHALL REQUIRE IN THE ANNUAL INFORMATION TECHNOLOGY CAPITAL PLANNING GUIDANCE OF THE OFFICE OF MANAGEMENT AND BUDGET THE FOLLOWING: ``(I) THAT THE CHIEF INFORMATION OFFICER OF EACH COVERED AGENCY

> OTHER THAN THE DEPARTMENT OF DEFENSE APPROVE THE INFORMATION TECHNOLOGY BUDGET REQUEST OF THE COVERED AGENCY, AND THAT THE CHIEF INFORMATION OFFICER OF THE DEPARTMENT OF DEFENSE REVIEW AND PROVIDE RECOMMENDATIONS TO THE SECRETARY OF DEFENSE ON THE INFORMATION TECHNOLOGY BUDGET REQUEST OF THE DEPARTMENT.

The CIO, not somebody else, must approve the IT budget (outside DOD).

> ``(II) THAT THE CHIEF INFORMATION OFFICER OF EACH COVERED AGENCY CERTIFY THAT INFORMATION TECHNOLOGY INVESTMENTS ARE ADEQUATELY IMPLEMENTING INCREMENTAL DEVELOPMENT, AS DEFINED IN CAPITAL PLANNING GUIDANCE ISSUED BY THE OFFICE OF MANAGEMENT AND BUDGET. `

CPIC and investment management are now law, not just policy. This is how money will be allocated for IT, not some other mechanism. Outside DoD its not some Joint Requirements Council from DoD, the law is different outside DoD.

> `(C) REVIEW.-- ``(I) IN GENERAL.--A COVERED AGENCY OTHER THAN THE DEPARTMENT OF DEFENSE-- ``(I) MAY NOT ENTER INTO A CONTRACT OR OTHER AGREEMENT FOR INFORMATION TECHNOLOGY OR INFORMATION TECHNOLOGY SERVICES, UNLESS THE CONTRACT OR OTHER AGREEMENT HAS BEEN REVIEWED AND APPROVED BY THE CHIEF INFORMATION OFFICER OF THE AGENCY;

The CIO must approve any IT contract. The CIO, not some other person.

> ``(II) MAY NOT REQUEST THE REPROGRAMMING OF ANY FUNDS MADE AVAILABLE FOR INFORMATION TECHNOLOGY PROGRAMS, UNLESS THE REQUEST HAS BEEN REVIEWED AND APPROVED BY THE CHIEF INFORMATION OFFICER OF THE AGENCY; AND

You cannot reprogram the CIO's funds.

> ``(III) MAY USE THE GOVERNANCE PROCESSES OF THE AGENCY TO APPROVE SUCH A CONTRACT OR OTHER AGREEMENT IF THE CHIEF INFORMATION OFFICER OF THE AGENCY IS INCLUDED AS A FULL PARTICIPANT IN THE GOVERNANCE PROCESSES.

OOPS, now we have the loophole. Develop some process where the CIO gets boxed in by politics, and then it's all OK. Let's see how this works out.

> ``(II) DELEGATION.-- ``(I) IN GENERAL.--EXCEPT AS PROVIDED IN SUBCLAUSE (II), THE DUTIES OF A CHIEF INFORMATION OFFICER UNDER CLAUSE (I) ARE NOT DELEGABLE. [[PAGE 128 STAT. 3440]]

No CIO delegation of these powers, except as noted below.

> ``(II) NON-MAJOR INFORMATION TECHNOLOGY INVESTMENTS.--FOR A CONTRACT OR AGREEMENT FOR A NON-MAJOR INFORMATION TECHNOLOGY INVESTMENT, AS DEFINED IN THE ANNUAL INFORMATION TECHNOLOGY CAPITAL PLANNING GUIDANCE OF THE OFFICE OF MANAGEMENT AND BUDGET, THE CHIEF INFORMATION OFFICER OF A COVERED AGENCY OTHER THAN THE DEPARTMENT OF DEFENSE MAY DELEGATE THE APPROVAL OF THE CONTRACT OR AGREEMENT UNDER CLAUSE (I) TO AN INDIVIDUAL WHO REPORTS DIRECTLY TO THE CHIEF INFORMATION OFFICER.

For non-IT, the CIO can delegate to a direct report of the CIO. Not to someone else. A direct report.

> ``(2) PERSONNEL-RELATED AUTHORITY.--NOTWITHSTANDING ANY OTHER PROVISION OF LAW, FOR EACH COVERED AGENCY OTHER THAN THE DEPARTMENT OF DEFENSE, THE CHIEF INFORMATION OFFICER OF THE COVERED AGENCY SHALL APPROVE THE APPOINTMENT OF ANY OTHER EMPLOYEE WITH THE TITLE OF CHIEF INFORMATION OFFICER, OR WHO FUNCTIONS IN THE CAPACITY OF A CHIEF INFORMATION OFFICER, FOR ANY COMPONENT ORGANIZATION WITHIN THE COVERED AGENCY.

The CIO of the department approves a the CIO of any component agency. This clause does not entitle any CIO of a component agency to approve contracts or investments. It does not provide the Department CIO a means to fire the component CIO either, perhaps to weak to assure full immediate compliance. It does not ensure that the component CIO report to the Department CIO either, weak.

> ``(C) LIMITATION.--NONE OF THE AUTHORITIES PROVIDED IN THIS SECTION SHALL APPLY TO TELECOMMUNICATIONS OR INFORMATION TECHNOLOGY THAT IS FULLY FUNDED BY AMOUNTS MADE AVAILABLE-- ``(1) UNDER THE NATIONAL INTELLIGENCE PROGRAM, DEFINED BY SECTION 3(6) OF THE NATIONAL SECURITY ACT OF 1947 (50 U.S.C. 3003(6)); ``(2) UNDER THE MILITARY INTELLIGENCE PROGRAM OR ANY SUCCESSOR PROGRAM OR PROGRAMS; OR ``(3) JOINTLY UNDER THE NATIONAL INTELLIGENCE PROGRAM AND THE

MILITARY INTELLIGENCE PROGRAM (OR ANY SUCCESSOR PROGRAM OR PROGRAMS).''

None of this applies to "black programs" aka intelligence activity.

CONCLUSION

FITARA increases the authority of the Federal CIO, but with loopholes. Several critical bits are missing. It will have an impact, but I believe the loopholes will assure that billions are still wasted in Federal IT budgets. We shall have to wait and see.

4.17 FITARA: MANDATORY PORTFOLIO REVIEW, SEPTEMBER 19, 2015

This post continues my -no BS- review of FITARA, with commentary from someone who has participated in the process in small and various ways for many years. Now we will examine the requirement for mandatory portfolio management, Section 833. You can see the actual text of the law, as passed, right next to my comments. You can evaluate for yourself, with my bits of insight as a starting point for your analysis. Take it or leave it. Have fun!

SEC. 833. PORTFOLIO REVIEW, AS PASSED

SECTION 11319 OF TITLE 40, UNITED STATES CODE, AS ADDED BY SECTION 831, IS AMENDED BY ADDING AT THE END THE FOLLOWING NEW SECTION:

Just warming up...

``(C) INFORMATION TECHNOLOGY PORTFOLIO, PROGRAM, AND RESOURCE REVIEWS.--

``(1) PROCESS.--THE DIRECTOR OF THE OFFICE OF MANAGEMENT AND BUDGET, IN CONSULTATION WITH THE CHIEF INFORMATION OFFICERS OF APPROPRIATE AGENCIES, SHALL IMPLEMENT A

PROCESS TO ASSIST COVERED AGENCIES IN REVIEWING THEIR PORTFOLIO OF INFORMATION TECHNOLOGY INVESTMENTS--

IN THE POST ON NEW CIO AUTHORITY UNDER FITARA I NOTED WHERE THE LAW REQUIRES MONITORING OF INVESTMENTS UNDER CPIC, BY LAW (EVEN IF NOT BY NAME). NOW WE THAT CPIC REVIEWS FOR SEVERAL PURPOSES, NOT MENTIONED UNDER AUTHORITY, MUST BE ACCOMPLISHED VIA CPIC (INVESTMENT MANAGEMENT). THIS IS NO LONGER JUST POLICY, IT IS LAW.

FITARA seems to create no new penalties for failure to comply by government officials. Presumably existing fraud, waste and abuse statutes will be sufficient.

``(A) TO IDENTIFY OR DEVELOP WAYS TO INCREASE THE EFFICIENCY AND EFFECTIVENESS OF THE INFORMATION TECHNOLOGY INVESTMENTS OF THE COVERED AGENCY;

Information technology is managed as a set of transformative investments to increase organizational effectiveness. For many years this was policy. Now it is law that the CPIC (portfolio management) process- not something else- will increase that efficiency and effectiveness.

``(B) TO IDENTIFY OR DEVELOP OPPORTUNITIES TO CONSOLIDATE THE ACQUISITION AND MANAGEMENT OF INFORMATION TECHNOLOGY SERVICES, AND INCREASE THE USE OF SHARED-SERVICE DELIVERY MODELS;

Here again the CPIC (portfolio management) process will consolidate acquisitions. This will come as profound news to many in the acquisitions community who have had this backwards for years.

``(C) TO IDENTIFY POTENTIAL DUPLICATION AND WASTE;

Here Congress directs departments to eliminate duplicative systems and remove misaligned or otherwise ineffective systems via portfolio management aka CPIC. This would presumably include eliminating "zombie investments", which are disfavored initiatives defunded to a minimum but kept in operation with no possible chance of achieving their objectives. (Surely all reasonable persons will see zombie investments as a waste of government funds. Now the external portfolio reviews below may bring this wasteful practice into the light.)

``(D) TO IDENTIFY POTENTIAL COST SAVINGS;

Here is another bit of long existing policy moved to law. Analysis of alternatives and cost benefit analysis have been mandatory in policy for years, and are the standard means to identify cost savings. That includes the "non-material solution" where government should change its processes without buying any IT, for example.

> ``(E) TO DEVELOP PLANS FOR ACTIONS TO OPTIMIZE THE INFORMATION TECHNOLOGY PORTFOLIO, PROGRAMS, AND RESOURCES OF THE COVERED AGENCY;

AHA, a bit of Enterprise Architecture here. Again another area of policy, on the books for years, now cast into law. See for example the Sequencing Plan or Roadmap of investments first described under FEAF 1.x. As I have noted for years EA supports CPIC and investment management, and here we see a law describing exactly that.

> ``(F) TO DEVELOP WAYS TO BETTER ALIGN THE INFORMATION TECHNOLOGY PORTFOLIO, PROGRAMS, AND FINANCIAL RESOURCES OF THE COVERED AGENCY TO ANY MULTI- YEAR FUNDING REQUIREMENTS OR STRATEGIC PLANS REQUIRED BY LAW;

Again a blow for Enterprise Architecture, whose purpose is to align those transformation initiatives to the strategic plan(s) required by law, and the budget. I have a post on aligning the budget to EA, and here it is in law. Those who doubted the link between budget and EA can now call a lawyer if they wish to press their case to Congress. I also have links about connecting the strategy to EA. For those unaware, US Government agencies are required both to produce an agency strategy and an IT strategy (called the IRM Strategic Plan for historic reasons). All other documents called strategy in an agency tend to cloud the central authority of these mandatory documents.

> ``(G) TO DEVELOP A MULTI-YEAR STRATEGY TO IDENTIFY AND REDUCE DUPLICATION AND WASTE WITHIN THE INFORMATION TECHNOLOGY PORTFOLIO OF THE COVERED AGENCY, INCLUDING COMPONENT-LEVEL INVESTMENTS AND TO IDENTIFY PROJECTED COST SAVINGS RESULTING FROM SUCH STRATEGY; AND

This again looks like the "Sequencing Plan" or roadmap of FEAF. All the investments and their acquisitions should be put on a timeline showing how and when systems will be upgraded, consolidated, replaced, eliminated- and when strategy is enabled by same.

> ``(H) TO CARRY OUT ANY OTHER GOALS THAT THE DIRECTOR MAY ESTABLISH.

Hey, you can have goals outside the strategic plan! This is a loophole. It is a bit mushy. But section by section this bit of law has put teeth in CPIC and EA, for which see my relevant posts or the eBook. We see one major weakness in a strong bit of law.

What this loophole will allow is continued misapplication of "mission need" outside DoD, in civilian agencies, for a longer time until measurement and review at OMB becomes mature. Goals should be enumerated, a solid list, so no one can play games and add new ones at whim. If a new goal is discovered it should be added to a new revision of the appropriate (and legally required) Strategic Plan (or IRM Strategic Plan) for that agency. Pursuit of other goals, not listed, should be fraud, waste or abuse of funds. Admittedly, this would require an improvement in the quality of strategic planning. It would have to be taken seriously, and used as real guidance, a big change.

Don't get me wrong. All that "mission needs" chain belongs in DoD, where it is appropriate. There "capabilities" to destroy the enemy are created with the hope that they may never be used. However in the civilian world you create things that must then be used. Real outcomes, measured by metrics, replace the potential of capabilities. IT investments, as described by OMB policy and FITARA, should be measured by projected ROI (Return On Investment). ROI should be documented next to cost and risk in a business case. Simply because DoD staff and Veterans who work now in civilian agencies did not learn proper civilian business management practice is no excuse to do it the wrong way.

The result will be continued "misaligned" investments slipping through, and continued waste.

Now another topic in the FITARA law...

> ``(2) METRICS AND PERFORMANCE INDICATORS.--THE DIRECTOR OF THE OFFICE OF MANAGEMENT AND BUDGET, IN CONSULTATION WITH THE CHIEF INFORMATION OFFICERS OF APPROPRIATE AGENCIES, SHALL DEVELOP STANDARDIZED COST SAVINGS AND COST AVOIDANCE METRICS AND PERFORMANCE INDICATORS FOR USE BY AGENCIES FOR THE PROCESS IMPLEMENTED UNDER PARAGRAPH (1).

For many years OMB policy has directed all agencies (including DoD) to use ROI and cost savings and a few other metrics as the basis for selecting transformative investments in the portfolio. Now civilian agencies get to do that by law. You may elaborate the core set, or fail to follow OMB policy with regard to required metrics, but now the law may make that clear and eventually yield some consequences for choosing investments and acquisitions by political whim rather than the formal process, based on metrics. There will be no more back-rooms full of cigar smoke making funding decisions as in the time of the Chicago Political Machine, allocation of funds and selection of investments will now be made by analysis of metrics. The better investment, not the more influential political position, wins.

> ``(3) ANNUAL REVIEW.--THE CHIEF INFORMATION OFFICER OF EACH COVERED AGENCY, IN CONJUNCTION WITH THE CHIEF OPERATING OFFICER OR DEPUTY SECRETARY (OR EQUIVALENT) OF THE COVERED AGENCY AND THE ADMINISTRATOR OF THE OFFICE OF ELECTRONIC GOVERNMENT, SHALL CONDUCT AN ANNUAL REVIEW OF THE INFORMATION TECHNOLOGY PORTFOLIO OF THE COVERED AGENCY.

No, Congress does not trust you, Department, to implement this fair and honest allocation of IT funds by non-political means. OMB and the DepSec will review your CIO's choices, and your process, yearly, together. No funny business.

> ``(4) APPLICABILITY TO THE DEPARTMENT OF DEFENSE.--IN THE CASE OF THE DEPARTMENT OF DEFENSE, PROCESSES ESTABLISHED PURSUANT TO THIS SUBSECTION SHALL APPLY ONLY TO THE BUSINESS SYSTEMS INFORMATION TECHNOLOGY PORTFOLIO OF THE DEPARTMENT OF DEFENSE AND NOT TO NATIONAL SECURITY SYSTEMS AS DEFINED BY SECTION 11103(A) OF THIS TITLE. THE ANNUAL REVIEW REQUIRED BY PARAGRAPH (3) SHALL BE CARRIED OUT BY THE DEPUTY CHIEF MANAGEMENT OFFICER OF THE DEPARTMENT OF DEFENSE (OR ANY SUCCESSOR TO SUCH OFFICER), IN CONSULTATION WITH THE CHIEF INFORMATION OFFICER, THE UNDER SECRETARY OF DEFENSE FOR ACQUISITION, TECHNOLOGY, AND LOGISTICS, AND OTHER APPROPRIATE DEPARTMENT OF DEFENSE OFFICIALS. THE SECRETARY OF DEFENSE MAY DESIGNATE AN EXISTING INVESTMENT OR MANAGEMENT REVIEW PROCESS TO FULFILL THE REQUIREMENT FOR THE ANNUAL REVIEW REQUIRED BY PARAGRAPH (3), IN CONSULTATION WITH THE ADMINISTRATOR OF THE OFFICE OF ELECTRONIC GOVERNMENT.

Ahem- not DoD. They have different laws about Joint this and Joint that that do not apply to civil government. (I know consultants at a big FFRDC who incorrectly advised a large department on this point. Twice, and once after the error was brought to their attention. Hopefully now the error will be crystal clear.)

This has ramifications to the SDLC in some Departments. A DoD style Mission Need Statement is now clearly less appropriate than a business case full of those required (by law) metrics, for example. (Note that an MNS and a business case are redundant documents. The business case has additional information, and is required to some extent as incarnated in OMB Form 800.)

> ``(5) QUARTERLY REPORTS.-- ``(A) IN GENERAL.--THE ADMINISTRATOR OF THE OFFICE OF ELECTRONIC GOVERNMENT SHALL SUBMIT A QUARTERLY REPORT ON THE COST SAVINGS AND REDUCTIONS IN DUPLICATIVE INFORMATION [[PAGE 128 STAT. 3444]] TECHNOLOGY INVESTMENTS IDENTIFIED THROUGH THE REVIEW REQUIRED BY PARAGRAPH (3) TO-- ``(I) THE COMMITTEE ON HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS AND THE COMMITTEE ON APPROPRIATIONS OF THE SENATE; ``(II) THE COMMITTEE ON OVERSIGHT AND GOVERNMENT REFORM AND THE COMMITTEE ON APPROPRIATIONS OF THE HOUSE OF REPRESENTATIVES; AND ``(III) UPON A REQUEST BY ANY COMMITTEE OF CONGRESS, TO THAT COMMITTEE. ``(B) INCLUSION IN OTHER REPORTS.--THE REPORTS REQUIRED UNDER SUBPARAGRAPH (A) MAY BE INCLUDED AS PART OF ANOTHER REPORT SUBMITTED TO THE COMMITTEES OF CONGRESS DESCRIBED IN CLAUSES (I), (II), AND (III) OF SUBPARAGRAPH (A). ``(6) SUNSET.--THIS SUBSECTION SHALL NOT BE IN EFFECT ON AND AFTER THE DATE THAT IS 5 YEARS AFTER THE DATE OF THE ENACTMENT OF THE CARL LEVIN AND HOWARD P. `BUCK' MCKEON NATIONAL DEFENSE AUTHORIZATION ACT FOR FISCAL YEAR 2015.''.

You have until 2020 to begin reporting on this process governing technology investments (with subordinate acquisitions) and its results regularly to just about everybody. Publicly. That is pretty fair seeing as how this has all been policy for fifteen or twenty years. No?

CONCLUSION

If you have been reading my posts, or have read the eBook, you will be very familiar with all the elements that Congress has now made law. You will know exactly how they all fit together to improve alignment and allocation of government funds. There is little new or unexpected here, but it does add great weight to overcome those in government who refuse to follow policy or do the right things.

If this works it will no longer be possible for government executives to allocate tax dollars to projects based on political expedience. Instead documented objective metrics will be required. Imagine how much better your tax dollars will be allocated without trading funds for political gains.

4.18 VENDORS VS CONTROLS, DECEMBER 27, 2014

Vendors seek to sell product. They have marketing and sales processes designed to convince the customer to buy product. Corporate controls are designed to, among other things, prevent you from buying redundant or unneeded products. There is a natural tension.

ENTERPRISE ARCHITECTURE

Enterprise architecture (EA) is a discipline that attempts to assure efforts support strategy. This concept is called "alignment". EA is most associated with the governance process called portfolio management or (in US Federal Government) capital planning and investment control.

These processes are intended to identify and cancel redundant transformational investments, cancel investments that do not support strategy, and assure the investments you selected have the greatest impact. Other supporting processes assist in assuring the suite of products used is narrowed to achieve lower total cost of ownership.

Sun Certified Enterprise Architect

At one time Sun Microsystems (tm) had a certification program designed to test your proficiency as a Java developer. This certification was named for enterprise architecture.

The body of foundational enterprise architecture papers, which described management of systems and expenditures, had almost nothing in common with the Java programming material in the Sun(tm) certification.

The purpose of this certification was to sell software and hardware.

SYSTEMS ENGINEERING

System engineering is a discipline for managing the complexity of large projects. A key system engineering concept, the System Development LifeCycle (SDLC) is closely associated with a governance process called Stage Gate Review. These assure that systems developed provided good functional fit for solving the problem at hand.

MICROSOFT CERTIFIED SYSTEM ENGINEER

Microsoft(tm) at one time had the MCSE certification. It provided information on the administration of desktop and server platforms. An INCOSE (International Council on Systems Engineering) document at the link below describes eleven valid system engineering roles and a twelfth designed to cover the misunderstanding of system engineering as portrayed in classified ads. That last seems to match best with the former MCSE certification from Microsoft.

http://www.incose.org/educationcareers/PDF/12-roles.pdf

The purpose of this certification was to sell software.

VIVEK KUNDRA

The former CIO of all government Vivek Kundra warned of a vast vendor cartel influencing government procurement. However he seemed to blame large defense contractors and was apparently biased towards equally wasteful West Coast (Silicon Valley) giants instead. These large companies have long desired a greater slice of Federal spending, with fewer barriers.

http://fcw.com/articles/2011/09/01/kundra-comments-on-existence-of--it-contracting-cartel-hit-a-nerve.aspx

http://www.infosecisland.com/blogview/15381-Federal-CIO-Vivek-Kundra-Warns-of-IT-Vendor-Cartel.html

http://www.computerworld.com/article/2510487/government-it/outgoing-federal-cio-warns-of--an-it-cartel-.html

IT PROCUREMENT REFORM

Several laws have been passed, or are in waiting, to manage IT procurement. Congress has sought to control growth in IT spending since the 1980s.

http://en.wikipedia.org/wiki/Clinger%E2%80%93Cohen_Act

http://www.nextgov.com/cio-briefing/2014/12/fitara-analysis-will-cios-use-their-new-powers-good/101160/

http://en.wikipedia.org/wiki/Federal_Acquisition_Reform_Act

AGILE

Both enterprise architecture and system engineering have been de-emphasized to make way for Agile Development. The Agile manifesto was produced in Utah by folks who sell software. The Agile signatories and beneficiaries are software developers. Never once have I heard that Agile is meant to stop development of unneeded software desired by end users in support of support obsolete wasteful processes and tasks.

http://martinfowler.com/articles/agileStory.html

http://agilemanifesto.org/authors.html

CONCLUSION

Congress has struggled to manage and control expensive and fruitless software development, including redundancy and excess misdirected efforts. Some have accused vendors of having a vast near cartel to influence government procurement. Vendor efforts have in the past undermined the two major control regimes designed to assure tax dollars are not wasted on meaningless or irrelevant software development.

There are currently ***trillions of dollars*** of unneeded software efforts underway in the US Federal Government. The pendulum has swung too far toward accommodation of random software development to placate the temporary tactical desires of local users. Despite rhetoric from big special interests, the controls designed to eliminate this waste should be properly applied, supported, encouraged and increased.

SECTION 5: FINDING WASTE

5.1 FINDING WASTE, FEB 14, 2016

Enterprise architecture attempts to align efforts to strategy or mission through transformation initiatives, thus improving organizational performance. In doing this, it must also attempt to eliminate waste as contrary to performance and distraction from the mission or strategy. You really cannot do one well without the other.

IDENTIFICATION

In the US Federal Government this job of finding waste for enterprise architecture to eliminate is aided by several things:

- **Inspector General Reports:** The Inspector General (IG) for each government agency produces reports of inefficient and wasteful areas, processes, items, expenditures. They represent the head of the agency in doing so.

- **Government Accountability Office Reports:** The US Government Accountability Office (GAO) also produces reports of things that are ineffective, inefficient and wasteful. Sometimes they also produce reports of progress and compliance across government. They represent Congress in doing so.
- **Office of Management and Budget (OMB) activity:** The OMB represents the President. OMB has reviews like CPIC-Stat that look for proper functioning of agencies.
- **Audits:** OMB Circular A-123 mandates internal audits as part of internal controls of all agencies. If performed effectively these will highlight waste.

To my way of thinking, any government agency that does not hook these sourced of identification of waste to EA is not doing its job. Then hooking EA to portfolio management so things get fixed completes the cycle.

OTHER MEANS

Other means to find waste include:

- **Cost based Accounting:** Checking comparative costs between your organization and others.
- **Benchmarking:** Looking at other organizations who do something well to see how you might do it better.
- **SWOT Analysis:** This is a common analysis method to documenting known organizational strengths, weaknesses, opportunities and threats.
- **Balanced Scorecard:** Another means of documenting known areas of strength and weakness.

KINDS OF PERFORMANCE

In waste we look for a lack of performance. It is critical to know the three kinds of performance involved, and keep them straight in your mind. If you confuse these, or

focus on one without managing the others, you will be doing a poor job of eliminating waste.

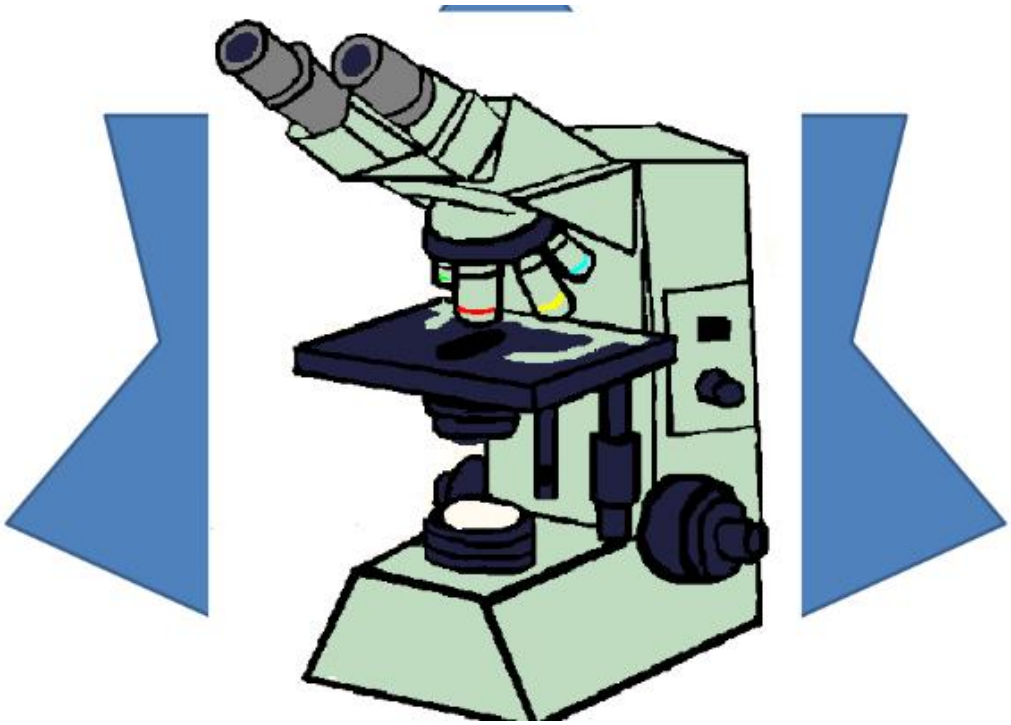
- **Project Performance:** Projects and programs create new capabilities and systems. The creation process, the process of building the thing, can be efficient or inefficient. Tools like EVMS, stage gate reviews and periodic management reviews are designed to measure this performance. These do not measure the system.
- **System Performance:** The thing built will have a performance when operated. After the project or program is over the system will do its job, and it will be efficient or inefficient. There are measures of this effectiveness, including MOEs (Measures of Effectiveness), MOP (Measures of Performance), KPPs (Key Performance Parameters, selected MOPs), and TPMs (Technical Performance Measures). These do not measure the project. They are measured in testing.
- **Organizational Performance:** The organization itself has a performance. While the organization, its processes and people, may use a system the measurement of organizational performance does not focus on the system but rather getting the job done. Measures for this are called KPIs (Key Performance Indicators). Such things are measured in evaluations or exercises.

  To manage the elimination of waste you must look at all of these and apply them correctly.

CONCLUSION

If you have all this information, finding waste is easy. All the guidance and method are provided, you just have to do your job. This identification of waste should be incorporated as key business drivers to your enterprise architecture program. If you are not using these to drive enterprise architecture, you are probably negligent.

5.2 FOCUS ON THE PROBLEM, JUL 13, 2015



I am an engineer and architect. We focus on the problem, and the solution. To us focus is an asset. We exclude all the irrelevant factors and search for the key element or elements to be adjusted to create success. We study the thing, not the packaging or the wrapper.

Focus is concentration. No significant technical advance happens without concentration, unless it falls on your foot and you stub your toe on it. Concentration is hard, hard work. We work hard. Applying science to reality is hard work. We have to study all our lives. We read books, take training and generally have less fun than most.

Engineering has, arguably, brought about more improvement in the living conditions of man than any other discipline. We engineers commonly claim the wheel, fire, heat,

central AC, the modern mattress, the couch, the TV, the computer, the Internet, food distribution and manufacturing. We can face Medicine head to head for first place in the betterment of mankind's condition. Politics, soft sciences, and sales and marketing can all move to the rear of the line.

If you want to be an engineer, you focus on the problem. If you are building an airplane, focus on that airplane or its parts. Ignore all those trying to redirect your attention to people, their emotional baggage, their needy interaction issues like encouragement or whatever. To be an engineer is to find approval in a working part, a prototype that does not catch fire. It is not a discipline devoted to getting along with gadflys. Politics are for the gregarious sorts who produce less value to society and accomplish less. Water coolers are often just for water, unless you need a break.

Focus on the problem, the item under construction or development. Push out those distractions. The following elements of modern life will seek to decrease your technical contribution:

- Cubicles will fail to isolate you so you can work productively. The manager guy stops by to ask why you have not solved the problem yet when you spent all morning hearing about how Sam's dog ate the pet bird and had to go to the Vet.
- Managers will wonder why you are not nice to Suzy when you were barely aware there was a Suzy.
- Slackers will accuse you of arrogance simply because they never worked hard, they studied basket weaving and they feel intimidated. Surely their lack of effort makes the case that you are a smug, superior bastard?

- You have to attend the Stand-Up Meeting. Those who talk and do not work will decide your fate if you do not defend yourself at the Stand Up Meeting.
- Sociologists will insist you need to devote time to emotional intelligence, when that has nothing to do with the part that does not function and the problem at hand. Who is Suzy again?
- At some point when sales is briefed on your progress, some sales guy will begin making statements of the form: "I do not understand it so no one else can understand it"; "I do not understand it so it cannot work"; and my favorite "I do not understand it so you must stop all progress". If you point out the fallacy and incredible egotism of such a position, the sales guy will declare you arrogant.
- Everybody wants to tell you how smart you are. You know darned well that smart is when the part works as designed. They will not shut up. You measure yourself against the problem, and until it is solved you are quite clear that your skill and intelligence are not yet adequate. They tell you how great you are and you will solve the problem by this personal greatness, but this gives you no clue how to solve the problem. AARGH!
- Some turkey who has fifty less IQ points will want to review you every year, just to piss you off. Further the guy cannot tell you anything that helps to make the part function as designed. He adds no value to solving the problem. He keeps wanting to meet about your contribution to the organization, which would be greater if he left you alone to solve the problem!
- Various smooth talkers will insist you know nothing about anything outside your current focus. They will insist that you will drool and stutter if taken out of your present problem domain. Despite the fact that you may have read 50 books on the

subject they refer to, and may be an international expert, they will claim you are incompetent because of your current focus on some other problem. Distraction results.

- Your wife will wonder why you study and work so hard. They should pay you more.
- And then there is Fred. Fred is not focused on the problem, he is focused on competing with you. Fred is barely aware of the problem. Fred spends his time positioning to be more important than you. Fred barely contributes, but has great status reports. Fred routinely tells you that you are an idiot, because the important thing is building relationships with the senior managers that will promote you (meaning himself). Fred knows little about engineering or the problem domain, and tells everyone you are doing the job poorly. You try to ignore Fred, as you are all about solving the problem.
- You finally solve the problem, and it saves the company 10 million per year. You get a patent. They give you $100 for the patent and a $1000 one time bonus. Then the manager guy shows up again to review you and says you are not a good employee because you are mean to Suzy. Who is Suzy? Did this manager guy ever, in his career, save he company $10 Million per year? Did Suzy? Did Fred? No, none of them.

Our society does not support engineers or engineering. It did once, not so long ago. Its a flipping pandemic. This list has many more potential entries.

"Great minds discuss ideas; average minds discuss events; small minds discuss people." Eleanor Roosevelt

5.3 OUTCOMES VS CAPABILITIES, SEP 19, 2014

The term capability is used often in current enterprise architecture discussion. Lets examine its use and meaning relative to real outcomes.

An **outcome** is the result of your activity regarding enterprise transformation, in terms of operational improvement. Are we now following a strategy that engenders success? Has our customer perception improved? Have our sales increased? In so much as EA is tied to performance measures, an outcome should be expressed as a change in those performance measures.

AN OUTCOME IS A MEASURABLE CHANGE IN MEASURED PERFORMANCE INDICATORS.

A **capability** is a potential for the enterprise to do something. If it is possible for the enterprise to do something, then it is capable. Capabilities are important to organizations like DoD, where you may not want to exercise all your capabilities and destroy everything. Sometimes the mere capability is effective as a deterrent.
A CAPABILITY IS SOMETHING THE ENTERPRISE COULD DO, IF EXECUTED.

Outcomes not Outputs: Outcomes are commonly confused with outputs. The thing produced by a production in your business is not an outcome. Outcomes are the improvements produced by transformation. Outcomes should be measurable by performance indicators, metrics, and not described as circumstances. When you list some circumstance it tends to be part of a solution not a measure of the problem, and in the end you can often only prove that you spent money but not that you affected the performance of the organization.

Examples: "Customer complaints will decrease by 30%" is an outcome. "A new help desk will be created" is not an outcome, but may be a capability.

Comparison: In business, and in civil government, it is not enough for some process or method to be possible. You must realize the potential gains. Basing an enterprise architecture on capabilities, potential actions, is not adequate and not the point. Real transformation requires capabilities to be realized and the performance of the organization changed.

Primary Use: The primary use of capabilities is in evaluating alternatives for real implementation. In defense applications, that decision is delayed until needed and potentially unused capabilities are still acquired or developed. In business and civil government unused alternatives are not usually developed or acquired, as this would be cost-ineffective.

Conclusion: The term capability is useful. Lists of capabilities may be useful. The focus of EA is not on what might be possible, but what is actually implemented to

transform organizational performance. Capabilities are often overemphasized in enterprise architecture efforts within business and civil government. You will be measured on real outcomes, not potentials.

5.4 CULTURE VS STRATEGY, AUGUST 22, 2014

For a moment, let's examine the current rhetoric that organizational culture is more important than strategy. We should also examine the notion, proposed by some, that given organizational culture a strategy is not needed. To examine all this we will need an example, and one springs to mind: Robin Hood and his Merry Men.

INTRODUCTION

You may recall the story. In a common version, young Robin of Loxley (or Robin of Sherwood) was off fighting the Crusades. He returns home to find that his property has been seized by the false king John (the usurper). King John had arranged for, or was complicit with, the kidnapping of good King Richard (the Lionhearted, I believe) on his way back from the very same Crusades. Robin's family apparently supports the true king. Robin takes to the woods and attracts some followers.

CULTURE

Robin and the Merry Men have plenty of culture. Not only are they mainly a martial bunch, but they have English culture in abundance. They have Christian culture. They have common feudal values: All power is granted by the king. They even have their own cleric (Friar Tuc) as keeper of Christian culture and values.

In one version the more ancient culture of the land was added to all this. BBC ran a series in the 1980s with music by Clannad. In this series Robin is deeply connected to the forest and its meaning, its ways. He is the "Son of Hearn the Hunter", a role not a lineage as such. He is deeply connected to ancient values, symbols, traditions and rites of various sorts. He and the Merry Men have even more culture.

RESULTS OF CULTURE

Robin and the Merry Men resisted King John and his henchman the Sharif of Nottingham. They did this because of feudal values, God had made Richard the king. They served Richard. In addition they robbed from the rich who passed through Sherwood, and gave to the poor peasants whose wealth had been overtaxed first by the Crusades, and then by King John.

In the end, as enemies of the state, they would have been detained and killed.

STRATEGY

At some point in most variants Robin and the boys develop a strategic goal. They determine that they will take some of the money stolen and pay Richard's ransom. They execute the strategy, and Richard returns. In most versions Robin's lands and estates are restored, and his peasants are better cared for because of it.

LESSONS

What may we learn from this legend, retold precisely because it holds important cultural values and such? Well,

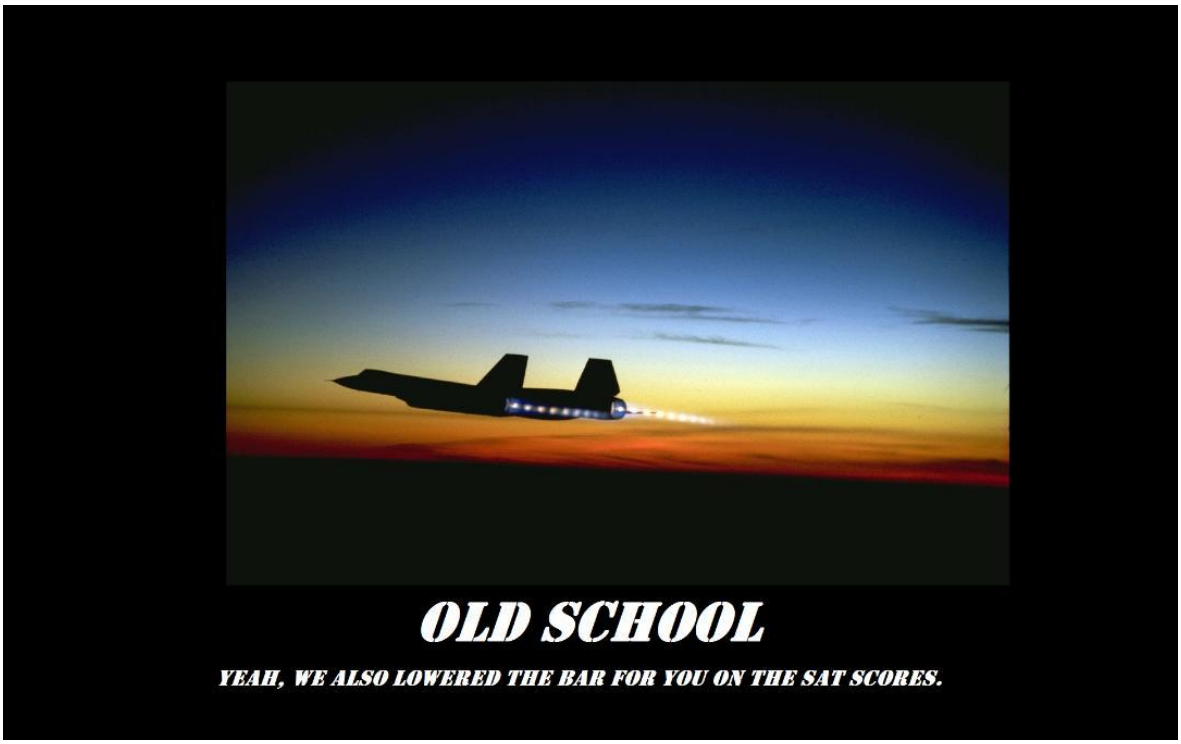
- The land and country have a culture;
- You work together and cooperate because of culture and values;
- Working against cultural values as the Sherrif and King John did is evil;
- The culture and values are partially embedded in law, and law must follow culture;
- When law and culture collide, the culture of the country and land win out;
- Follow the culture and values of your land and country, or be destroyed by them;
- Having a strategy is good, without it evil may win or you may lose.

THOUGHTS

Today we have corporations usurping law and rights of individuals, and attempting to displace the authority of nations and society. Each people should have the freedom to express their cultural values and symbols and myths in law and society. Corporations should not be free to usurp society or culture of a people or nation. However corporations should be free to follow culture and values to engage and motivate employees in accordance with societal norms and laws.

Corporations should have a strategy, have goals and objectives, and achieve them if they are not contrary to culture, society and law. Culture may trump strategy, but that is not corporate culture, rather the culture of the society around it. Without a strategy, without goals and objectives, any band of merry men is unlikely to achieve much.

5.5 THE CULTURE OF MEDIOCRITY, JULY 19, 2014



I recently heard William Eggers speak about his book, and why 60% of government managers polled believed government is less capable of execution today than years ago in the moon mission and Cold War era. Surely I share the sentiment. I do not believe the effect is confined to government.

I have an opinion on why America is performing poorly. I suppose I could go out and research that opinion phrased as a thesis. It's not really my area. Yet the anecdotal evidence is everywhere. It strikes me as common sense. We have adopted a culture of mediocrity.

Let me describe what I mean. I will compare a notional culture of excellence to a culture of mediocrity.

COMPETITION

In a culture of mediocrity people would avoid competition. They would cooperate in circumstances that demand competition. They would not seek competitive advantages and would instead harmonize with their competitors.

Competition, by the way, is often about self-improvement or team improvement, about reaching goals. The point is often to seek perfection or something nearer to that, and not to humiliate or degrade the other competitors. The other competitors may just be the best available measuring stick. At other times competition is about getting the last basket of food, or the only job, so your family eats. Sometimes it is about survival. This really happens, as in the Cold War.

TEAM MEMBERSHIP

In a culture of excellence people would feel membership in their team. This would include all teams they might be members of in addition to any notion of spectator activities. It would include patriotism as members (citizens) of the country, as well as a feeling of membership in company, city, county, state, region, or ethnic group. In contrast a culture of mediocrity would seek to minimize differences, taking sides, membership or belonging. Leadership of the team would represent the team's interests, and not those of the competition.

LOYALTY

In a culture of excellence people would not switch sides easily, because they are invested in and committed to the prosperity or triumph of their team. Leaders would be commuted to followers, followers to leaders, teammate to teammate, all working toward the common goals of the team(s) which they share. People and companies would not move as much, because of loyalties, and those loyalties would be strengthened by staying put.

IDENTIFICATION

In a culture of excellence people would identify with a unifying set of symbols, heroes, myths, values, ideals and icons. In a culture of mediocrity almost anything will do.

COST AND QUALITY

In a culture of mediocrity low cost is king. In a culture of excellence you value things of higher quality, and may pay more for them. I do not refer to the fancy car with the shiny label, but real quality such as longer service life, fewer repairs, higher effectiveness. This includes services, and in a culture of excellence you might pay a bit more for good customer service, or you might hire the American expert rather than the cheap visa applicant. In excellence experience and qualifications are valued, in mediocrity the most experienced cannot find work.

MBA AND PMP

Today we have a notion of a professional management class, who need not deeply understand the business or technology they are managing. Yet they assume the responsibility of making technical decisions far too often. Failure to restrict decisions to acknowledged experts contributes to a culture of mediocrity. These professional managers, feeling unsure of their decisions in a technically complex world, often rely on management by committee of non-expert stakeholders; Exactly how many monkeys with typewriters does it take to reproduce the Magna Carta? In the past era under discussion the term Engineer included the notion of being a manager of things, processes and sometimes people, thus producing expert decisions with more accountability and less consensus.

MEASUREMENT AND COMPARISON

In ta culture of excellence measurement and comparison is welcomed, to gauge your progress towards excellence. In mediocrity managers and individuals shun metrics, measurements and comparisons.

FACT VS FEELINGS

In a culture of excellence the focus is on the facts, not feelings. We sought progress toward goals and objectives not harmonious equilibrium. Objectivity was valued. In the culture of mediocrity company culture must come first rather than accepting the culture of society, and each person must feel comfortable and un-offended.

CONTRIBUTION

In a culture of excellence each team member would seek to contribute to the team and reduce burden to the team (self-reliance). Each team member would seek to build up other members of the team, develop them, make them more capable of contributing to team success. Team triumph would be triumph for team members, and the team would contribute back to its loyal members.

CHEATING

In a culture of excellence cheating would be shunned. In a culture of mediocrity, cheating and corruption or lax values would be tolerated to ensure more harmonious cooperation among competing interests. In excellence the rules are followed and respected, in mediocrity not so much. (In excellence rules might also be reviewed to assure all can follow them, and remove conflicts, redundancies, and excess.)

You just can't get an accurate measure of excellence or improvement if somebody is cheating. The field is just not level, the results are skewed.

UNIVERSAL APPLICABILITY

A culture of excellence or mediocrity would be pervasive, and would apply to all aspects of life. Culture is like that. It would apply to politics as well as corporate leadership as well as company policies as well as city planning as well as team sports. This is what the jocks mean when they emphasize sports as building character.

AMERICA THEN AND TODAY

In WWII, in the Cold War, we had competition and loyalty and identification and membership and contribution as a country, as a whole. We disliked and shunned cheating and lawbreaking. Being an american was a special privilege to those who chose to be one, to follow the rules, to adopt the common identity. to compete on our side. Today we harmonize laws with other countries, treat foreign citizens as above Americans, we cooperate when we should compete. Our companies act as if the cities and states they belong to are temporary parking spots, and helping fellow Americans is nonsense. Our national politicians are more interested in competing within the team than with other countries.

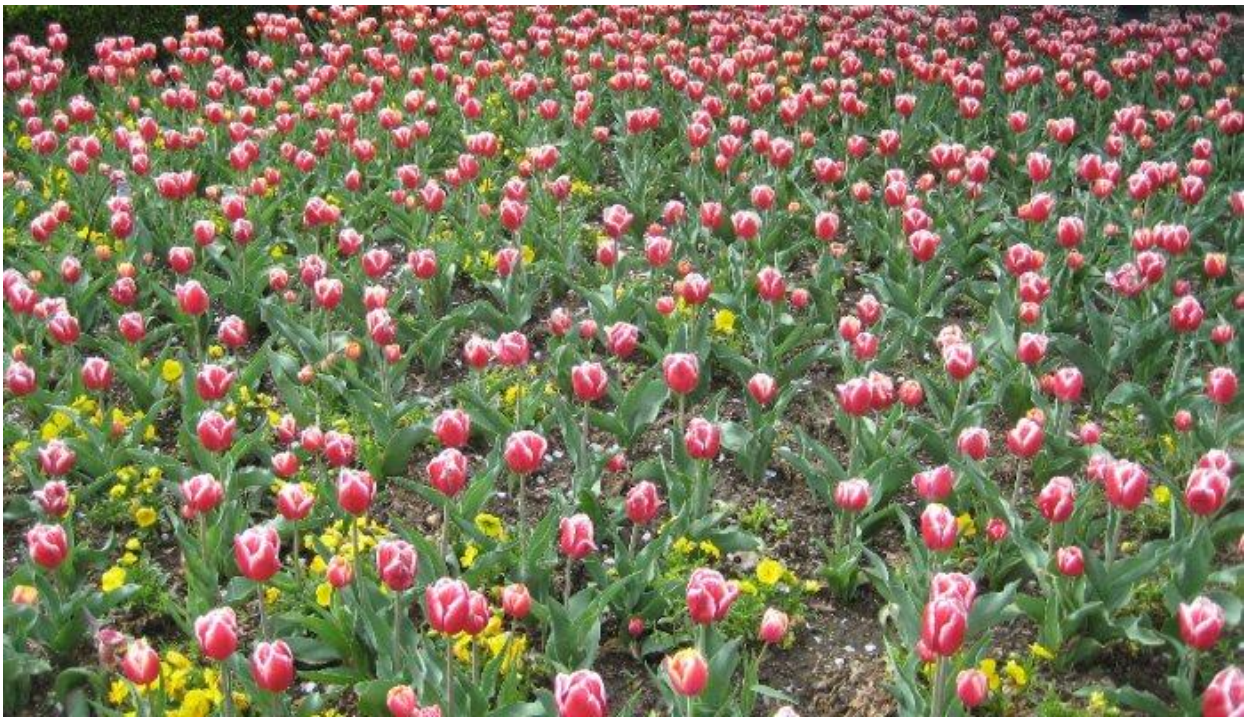
CONCLUSION

If we, as a country, seek mediocrity then we will probably find it. Our culture has been diluted, has changed, to one of mediocrity. Eggers' analysis is not so much wrong as it is just not the root cause. This culture of mediocrity is why our nation falters, why our companies do not prosper, why our government is ineffective, why we do not accomplish great things. We simply do not seek excellence and accomplishment as a country. We do not act as, work as, compete as a team. I suggest we might reverse that, if we try.

ONE LAST THOUGHT

The population continues to grow, but our natural resources and ability to produce do not. Some resources are being exhausted by overpopulation. At some point, perhaps soon, America and Americans may be forced to compete more for far fewer resources per person. The choice may be out of our hands soon, making the culture of mediocrity obsolete if it is not already so.

5.6 AN ARGUMENT OF ARCHITECTS, JUNE 11, 2015



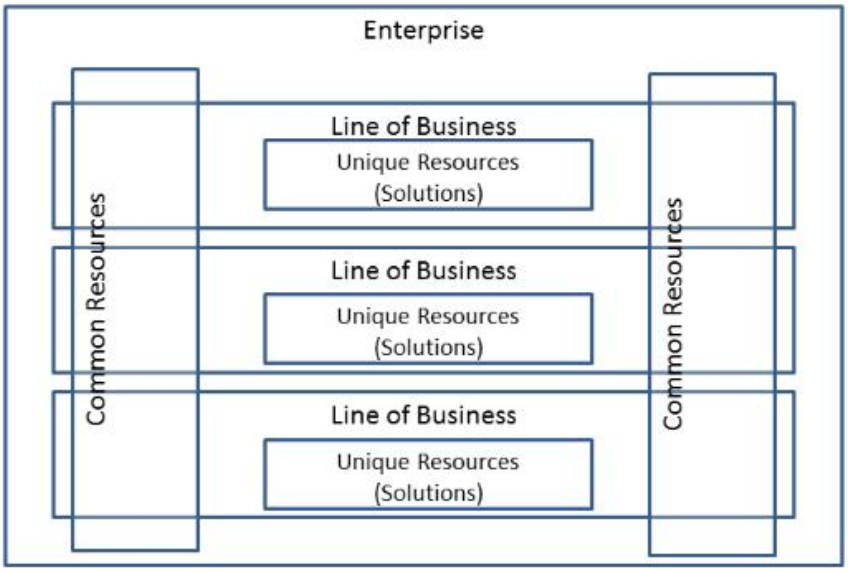
It's a gaggle of geese, a herd of cows, a murder of crows, a pod of whales, a school of fish and... wait for it.... an argument of architects. Architecture is a team sport. Architects love to argue. Given the opportunity to work or debate, architects may well chose debate. They can't help it.

The thing that fixes that is called a framework. Get some architects to agree on a framework, and the arguments subside. With common terminology and approach the arguments end and progress begins.

One architect can have high positive ROI. However architects are like Furbies(tm), put several in a room and they will collaborate if a framework is present. Their ROI will skyrocket through synergy. If no framework is present they may argue and debate. Once an argument of architects goes bad (after a normal adjustment period), just shoot them and get new ones.

If you meet an IT architect who knows no frameworks, or who dislikes them, they probably prefer to argue and debate, or work alone.

5.7 UNPRODUCTIVE EA, SE & PM CONCEPT CONFLICTS, APRIL 10, 2015



There are several sets of concepts at work in enterprise architecture, system engineering and project management. People will, as a matter of course, mix and match these concepts without regard to which work together in a consistent whole. Lets examine that for a moment, as it may remove vast heaps of confusion.

CONCEPT SET ONE

- An enterprise, according to Webster, is either a project/undertaking or a business/company. Either one. An enterprise can have an architecture.
- An enterprise consists of potentially multiple lines of business.
- A line of business may be a product line, or a line of service.
- A line of Business can have an architecture. Some resources may be used in common across multiple lines of business.
- Such common elements may have an architecture. Some resources, including IT or non-IT solutions, may be specific to one line of business. These may have an architecture.

CONCEPT SET TWO

- An organization may consist of multiple organizational units (divisions, departments, whatever).
- Those in turn may consist of multiple organizational sub-units.
- Some operations may be projects and programs which have a beginning, middle and end (these generally transform the organization)
- Others may be operations that are ongoing, and do not have an end.

CONCEPT SET THREE

- An organization exists to perform a mission.
- There may be a gap between the mission and what portion of it you can currently perform.
- That mission-gap requires the organization to purchase or acquire things.
- Acquiring complex things requires system engineering, a means of organizing and constructing complex things so they will operate correctly in support of the mission.
- An acquired thing may have an architecture.

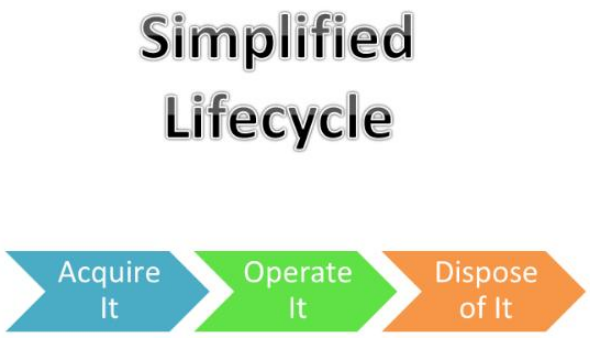
CONFUSION

Confusion sometimes occurs when concepts from different sets are mixed. For example: Enterprise architecture, in organizations that focus on acquisition, may be thought of as only supporting acquisition programs. Yet this need not be true, and the enterprise itself may have an architecture. Yet usually the enterprise architecture effort is restricted only to transformation efforts (the set of projects and programs). Headaches may result.

HARMONIZING

These concepts may be harmonized. It takes a bit of mental discipline to do so.

- In system engineering any entity, any organization or project or program or line of business or shared resource or unique resource (solution) may be a system.
- In system engineering a sequence of activities shows that everything, enterprises and projects and the rest, have both a construction (or transformation) phase and an operational phase.
- Almost any organizational unit has a purpose, function, or mission.
- Even organizational units may be thought of as being acquired. All have a similar lifecycle when seen in a long-term viewpoint (see diagram below).



- Acquiring complex things and making them work requires an organizational unit, project or program.

TERMINOLOGY AND SEMANTICS

Terminology and semantics make discussions of these concepts difficult. Such confusion can be counterproductive and very disruptive. (Efforts slicing everything down to smaller concepts using ever less substantive distinctions do not help much.) If you ignore some border cases though, the situation can be made clear. As the vast majority of disagreements occur due to differences in concept, terminology and semantics creating a simplified view of all this helps enormously. In the diagram below you can see what I use to simplify discussion enough to achieve progress and serve the mission.

There is no rocket science here. The enterprise has a PMO with a portfolio and an enterprise architecture, and the operational view of the effort is the primary vision. A line of business or a common resource is commonly implemented by a program, and a solution within these is commonly a project. Programs become known by some departmental sort of name when they are operated, whereas projects often become known as the system they produced once they begin operations.

While many insist that a large organization has a mission it is more true that it has many missions. Each line of business (service or product line) is more likely to have a unique mission. Organizations are commonly driven by a strategy, lines of business by an operational plan of some sort (sometimes mislablesd strategy). A single solution or tool has a tactical plan to say how it ill be applied, often expressed in a concept of operations.

Here is a typical semantic obstruction example: What if your enterprise is a program or project? Who cares- that is the other definition of enterprise and we did not use it to help clarify

Harmonious Concepts

Organizational Level	Transformational Phase Org. Name	Operational Phase Org. Name	Purpose Document
Enterprise	PMO, Portfolio	Company Name, Organizational Name	Strategic Plan, Mission Statement
Line of Business	Program Name	Departmental (or equivalent) Name	Operational Plan, Mission Statement
Common Resource			
Solution	Project Name	Solution or System Name	Tactical Plan, CONOPS

things. You can still have an architecture, but it will not have an ambiguous category-name that causes problems. Move on.

CONCLUSION

You can spend endless fruitless time arguing terminology and semantics, or adopt a common simplified view such as this. My point is that the later is productive, the former less so.

> Make common terminology a policy if differing terminology disrupts productivity, and put that terminology under a continuous improvement cycle with occasional recurring review. Focus on work, which is not debate except in areas like research and academia. Don't let the discussion of mental tools designed to aid organization and progress bring that progress to a halt.

See also

: http://www.unauthorizedprogress.com/images/EA_as_5_activities_2014.pdf

5.8 UNAUTHORIZED PROGRESS, NOVEMBER 2, 2014

Unauthorized progress is a term referring to taking initiative. In fact it refers to more than initiative, including performing acts that have not been approved. It may refer to performing acts that would not be approved. Any productive effort that is not approved or authorized, including those that would not ever be approved or authorized, fits right in there.

LEADERSHIP FAILURE

One of the most important reasons to engage in unauthorized progress is to bypass ineffective leadership. We have a crisis of leadership in the USA, it is said, so there is a need for some unauthorized progress now and then.

STIFLING BUREAUCRACY

Sometimes you find that the leader cannot get anything done. There may be governance or committees designed to achieve review or consensus. These mechanisms may be dysfunctional. The leader may make a back-room deal with a stealth tiger teal to perform some critical activity. This may require unauthorized progress.

CROSS ORGANIZATIONAL POLITICS

Sometimes an effort requires activity between two organization, but the politics of cooperation preclude quick or decisive action. This may require unauthorized progress.

USMC

In the United States marine Corps there is a formal mechanism to engage in unauthorized progress. It falls under "individual unit initiative", and Line Officers are trained in taking such initiative when required. (I admire the Devil Dogs for that.)

PATRIOTS AND FRIENDS

I happen to know several people who have engaged in unauthorized progress initiatives. They have many stories. As such efforts imply some risk to those performing the work, they were brave in doing what was required without approval.

One, Dr. Geoff Abbott, is writing a book on the phenomenon.

LATITUDE TO OPERATE

In today's society there is less and less latitude to operate based on your own responsibility. Corporations and government organizations have bound people tightly to rules and procedures. Such reduced freedom makes unauthorized progress more important as a check against excessive rigidity in the system.

LEADERSHIP

Any leadership training is incomplete without understanding when to do what is not approved, when that is ethical, when that is moral, and how to form a secret team to accomplish the required work. How do you recognize targets of opportunity? How do you identify when you strategy and mission are in jeopardy for some small element? How do you choose who will lead such a team?

LAST WORD

I usually speak of governance and controls, but sometimes there is a need for a bypass mechanism. If it is overused, that becomes corruption. However, if all is completely controlled, innovation and rapidity of response will suffer.

(Unauthorized Progress is not a LinkedIn skill. If you have done it, add it.)

5.9 IMPROVING GOVERNMENT ACQUISITION, APRIL 4, 2015

I have supported US Federal Government Acquisitions as far back as 1979. I did so first in uniform as an ancillary duty, then later as a system engineer, and more recently as an enterprise architect. I was offered a position supporting a contract to streamline DoD expenditures late last year, to help cut trillions, but turned it down. Instead, during my illness, I chose to contribute on more familiar ground. (I am recovering well, thanks.)

So here is what I would like to say to all those out there desiring government to operate more efficiently, and especially US Government Senior Executives. I will list a very few key process changes you should make to improve allocations of budgeted funds and improve government efficiency. As campaign finance reform would improve Congress regarding spending, these few changes would improve the Executive Branch. You have been willing to pay for my opinion, and here you can have it for free.

Fixing government spending requires process reform, and not a short term hatchet-fest. Rapidly defunding current investments creates huge losses and enormous risk.

Most government departments and agencies use an SDLC or SELC to manage acquisitions. Such a lifecycle process is often described as distinct from Capital Planning and Investment Control (portfolio management) and acquisitions process. ***Merge them.*** Three redundant parallel processes to manage the same thing are not better than one. The topmost decision on spending (investment in improvement) should follow CPIC guidance in the current highest policy on IT management, which should be expanded to all large procurements. Treat almost all major expenditures as investments in organizational performance improvement (except certain operational costs).

Commercial industry selects opportunities for spending based on***comparison of business cases***. This is the current unambiguous policy of OMB regarding IT spending. Business cases are also mandatory for certain large non-IT investments. Do a business case for all proposed expenditures and programs. Do not use something like a JRC, this is non compliant and focuses the decision on politics and not efficient use of funds. Instead use an IRB (Investment Review Board) as described in CPIC.

Eliminate the "Mission Need Statement" or MNS, as it is redundant with the business case. A business case will express the mission, the need (or gap), risks, estimated costs to fill the gap, and estimated returns. (In government costs and returns have more intangibles than in industry BTW). The Strategic Plan and IRM Strategic Plan you are required to write yearly by law will contain all the recognized important gaps anyway. Select investments (expenditures) based on the business case. Analyze alternatives using the baseline alternative presented in the business case. The result will be more transparent to citizens and industry, and will shift the decision from political weight of problems to efficient use of funds.

Secondarily, ***eliminate the "Capability Development Plan"*** or CDP. Shift emphasis from the raw capability to the returns (benefits) provided, especially in terms of organizational performance improvements. The CDP is mostly redundant with the acquisition plan, anyway. It is not needed.

Provide transparency into the business cases not selected compared to those selected. Then the taxpayer can see exactly what kind of fund allocations are occurring with their tax money. This is a direct extension of the current OMB website describing all major (selected) business cases.

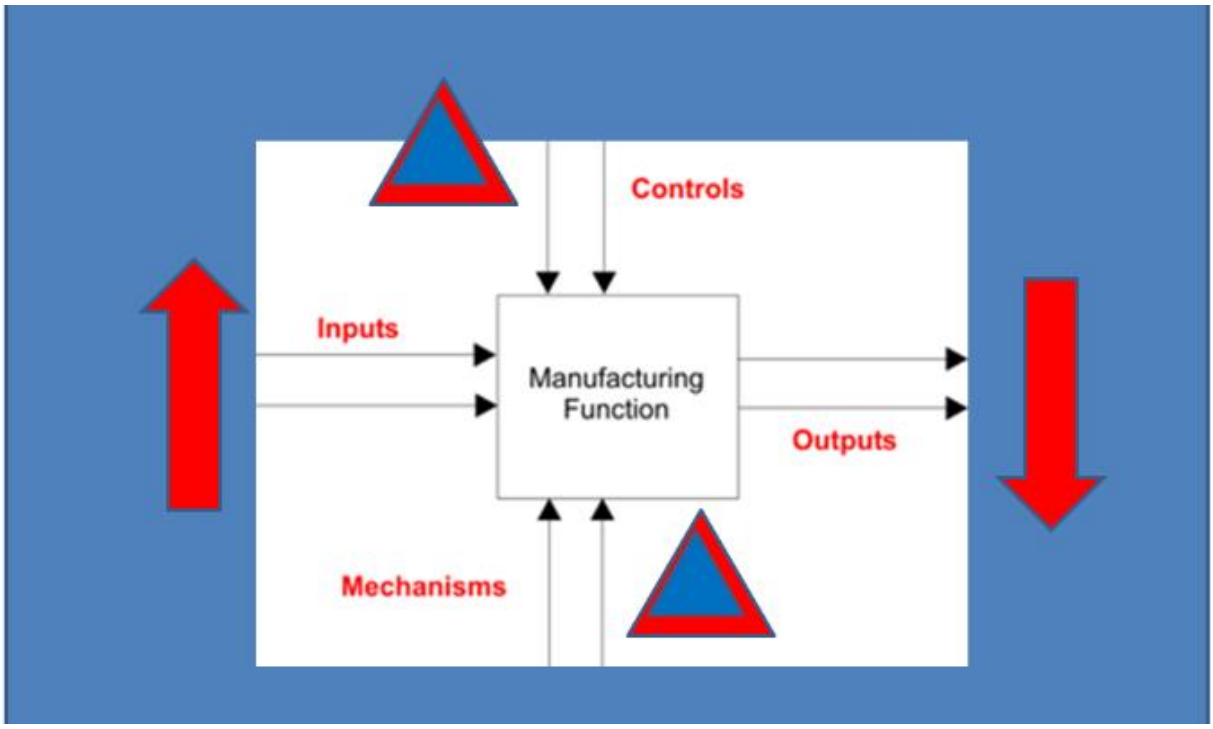
Track all funds in a rigid hierarchy leading to these top level investments (excluding certain operational costs). No sneaky allocating of funds from Peter to pay Paul.

Nearly any MBA can tell you this will improve your allocation of funds and transparency. This is not rocket science. If you want improvement, you can have it,

Conclusion

I suggest basing spending decisions on return, not on the politics of problem or gap priority. In this way money will be more effectively allocated. Gaps will be reflected in strategic plans as goals and objectives, business cases will align to strategy, the most effective business cases will be chosen, and each program or project will be held to the standard of the outcomes listed in their business case.

5.10 DETECTING CORRUPTION VIA ICOM, DECEMBER 26, 2014



The ICOM model is useful in defining performance or efficiency, and in assigning responsibility. Corruption is a type of inefficiency. ICOM can aid in identifying corruption and other inefficiency.

SYSTEMS

In systems engineering and troubleshooting ICOM and block diagrams have long been used to assist in troubleshooting and diagnostics. When output decreases for a given input, something has caused decreased efficiency. Conversely when maintaining output level requires increased input, something is similarly amiss. Troubleshooting ensues.

IDEF-0

ODEF-0 is an ICOM block diagram technique associated with defining or analyzing organizational function. It helps assign responsibility. It helps to define organizational performance. Some of these features can make it unpopular with organizational elements that avoid responsibility or perform badly.

DETECTING CORRUPTION

In the case of corruption a decrease in output occurs, or an increase in required input to maintain levels of output. Input may be labor or materials. This change will be associated with new or changed controls and mechanisms. Some change in standards, policies, tools or budgets will have occurred. The change may be some interference with or mitigation of controls. This technique can work for automated, mechanized and manual processes.

FOLLOW THE MONEY

To determine who changed the controls or mechanisms, follow the money to the party who is now being paid more, or receiving material benefit. It is not science, but common detective work. The approach is often valid despite non-technical basis.

FALSE CORRUPTION

Sometimes intentional inefficiency is introduced to transfer wealth to certain parties, such as the disabled or minorities. Quality increase may also sometimes manifest like corruption, with decreased output and higher costs. This kind of introduced inefficiency is much like corruption, but is intentional and has social benefit.

EXAMPLE 1

Suppose enforcement of environmental laws suddenly became far less effective. Despite many more cases and more obvious violations, convictions decreased. Suppose this corresponded with a defunding (mechanism change) of environmental law enforcement. Further, new unwritten policy was that no environmental laws would be enforced. (2001-2009)

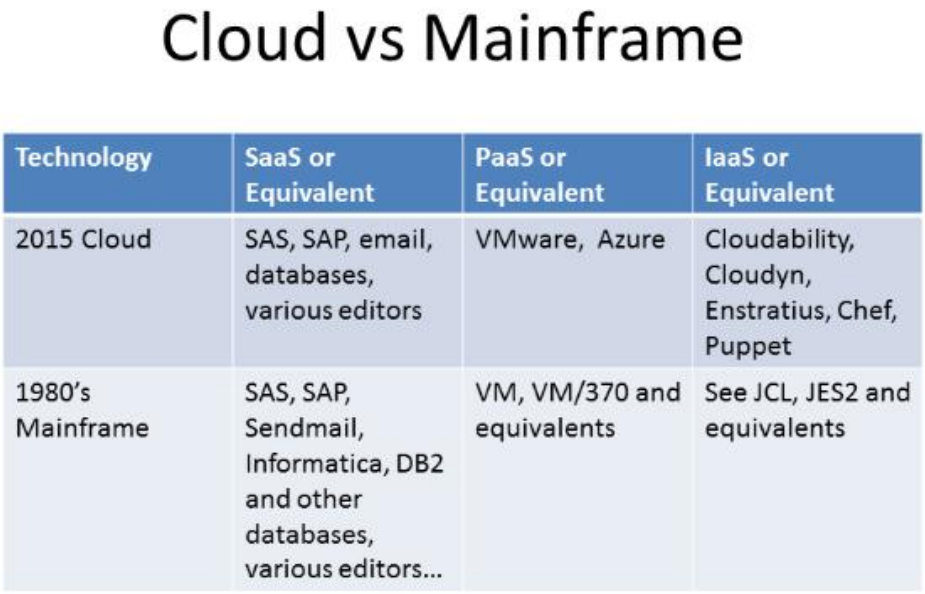
EXAMPLE 2

If production of large, complex information systems suddenly begins to fail more often you might look for a change in controls or methods. Suppose a new method, Agile, had been introduced. Contracts and money were shifted to new vendors from Silicon Valley .Controls, such as EA governance and SE SDLC were simultaneously reduced (defunded, sidelined). (2009-present)

CONCLUSION

Detecting corruption or false-corruption becomes obvious using block diagrams or ICOM, if you know the pattern to look for. You do not need the smoking gun to detect its presence. Determination of the exact means and circumstance of the entity benefiting producing the change can be left to the details, after corruption is identified. If no actual unethical causal link can be found, simply identify it as another category of inefficiency. There is always a cause, things do not happen by chance too often. All you need is a start date, an end date, and some measurements to start the process.

5.11 IS THE CLOUD NEW, JUNE 29, 2015



How does the mainframe environment of the 1980s compare to the current cloud environment? I will present a brief comparison based on current cloud terminology and offerings. This post will be based somewhat on my opinions, as comparison and evaluation will be required.

SAAS

In Software as a Service you buy the application and any supporting computing resources. You are a user of the application and manage nothing, unless you arrange to perform some application level administration. This may include email packages, document editing, statistical analysis, database applications like ERP, or almost any other software application. Such customer usage of the applications run in the data center were the "bread and butter" of the mainframe era.

PAAS

In Platform as a Service you rent a virtual machine, You own and administer that virtual machine. In the mainframe era the virtual machine environment of IBM was called VM, and you could run any IBM operating system (and later UNIX) on top of it. You could "peg up" a virtual machine and leave it running as long as you like, or demand one "on the fly". Today we buy VMware or an equivalent to do similar things.

IAAS

Infrastructure as a Service allows you to request machine resources rapidly and then have them provisioned for you "on the fly". In the mainframe era a sophisticated mechanism called JCL (Job Control Language) automated this. A sophisticated version was JES2 (Job Execution System 2), which provided for all the cloud related benefits like resilience, broad network access, resource

pooling, rapid elasticity and measured service with billing. Mechanisms to do the same in today's cloud are slightly less flexible and sophisticated IMO. These include Chef and Puppet.

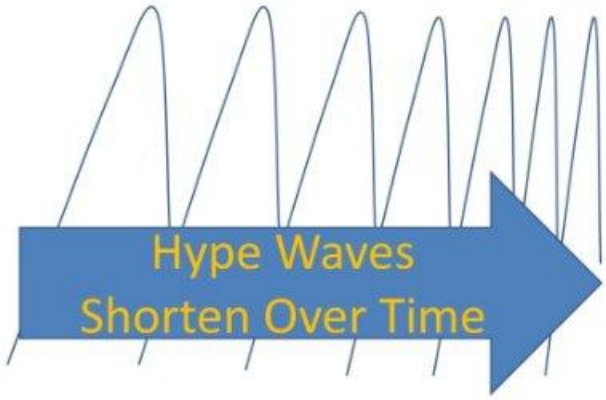
WHAT HAS CHANGED

Of course the operating systems and applications have changed. The Internet has replaced SNA. Process isolation and built-in security in the environment have decreased IMO. Reliability and availability have mostly decreased.

I have to conclude that no part of the cloud concept is particularly new, although the products are.

"The more things change the more they are the same." Alphonse Karr

5.12 TOP 2014 HYPE, AUGUST 9, 2014



I realize I have been doing this technology thing a long time. That has advantages and disadvantages, but it does give you some perspective. One thing that I have noticed is that the latest big thing is more transitory now than in the 1970s. Whatever buzz word is on the table is out there for a shorter period of time before it is replaced. Some will last longer than others, but on average they all get shorter.

Waves of substantive technology also have grown shorter. Before I started, although I repaired some of it, we had mainframes and timesharing. In the 1970s we had departmental computing, minicomputers. Later we had client server. Now we have SOA. Nothing ever completely disappears, it just gets added to the toolbox and the hype goes away, or it goes into the trash and becomes rarely used bits of history (ie COBOL)

Management trends are also subject to hype, substantive or not. TQM, CPI all had their periods of great hype. It all comes and gets oversold, then the actual value of the thing is discovered and it lands in the toolbox or the trash can.

I am a big fan to the Gartner Hype Curve (tm), which I see as a great public service. However they do not throw management trends and languages and what have you into the model.

So here are my top contenders for the biggest hype this year, 2014:

- **Agile:** Is Agile a management trend, a methodology, a technology category for tools? No, it is all things to all people, a panacea! Like many of its hyped brethren, there is a kernel of useful stuff in there with a clear range of applicability. However there is more over-application and excessive claims than there is substantive solid truth. Like the old medicine show cures Agile will make your organization profitable, will revolutionize test and integration, will work for large complex systems, is applicable to planning and architecture, and cures warts. (I added that last one.)
- **NoSQL:** IBM IMS is probably the most successful NoSQL database ever. On the PC there was DB2 and Paradox. B-Trieve was everywhere. Non-relational technology is as old as the computer. Not seeing your favorite latest brand here? Sorry! It has a range of advantageous applicability, like Agile, but it is not a panacea. Specifically staging databases (where input rate is high and data must be received and stored for a time as it is being more slowly categorized, indexed and quality checked) and OLAP applications (Data Mart or Data Warehouse technology replacement). The distributed parallelism thing is a step forward, but they have put a proper SQL front end on it.
- **Big Data:** Here the hype is that there are enough legal ethical uses of massive data, exceeding the capacity of RDBMS technology. Terabyte relational databases are now common using conventional technology, haw many petabyte database applications do you

think exist? How many companies need their own copy? How quickly do those applications violate privacy or reasonable uses?

- T**he Cloud:** What is the cloud? If you combine virtualization and self service provisioning you get a reasonable answer. There is public cloud and private cloud, and most company data ends up too sensitive for the former. In the end "the cloud" is an outsourced or insourced data center contributing to the consolidation of wasteful redundant data centers. All good so far, but with the holes in PKI and open source encryption code the notion of "cloud security" is an oxymoron. Simplified security models are the basis of cloud security certifications, missing all use specific controls. Some cloud offerings only cover file storage, web hosting and email. No panacea here.
- **SOA:** SOA is still subject to hype. The base concept is useful, but in common marketing usage SOA is synonymous with Web Services. Web Services suffer from extreme verbosity, with very low information theoretic efficiency in the way they are commonly used. There is no great advantage of web services over RPCs and the official standards acknowledge that, but common use has it backwards. From a security standpoint putting all your APIs on a web server, the most hacked and often most vulnerable server you have, might be a tad unwise. Ripping apart legacy applications to graft on some standard services your local geek squad made up has proven wasteful and expensive.
- **Leadership**: Leadership is a responsibility, not a privilege of the MBA.While you can teach it to those with the underlying natural talent, teaching the others produces comic results. Everyone will not be a leader. Every manager will not be a leader.
- **Entrepreneurship:** Quick, everyone ditch your jobs and create a company with a brand new product! OOPS, society and markets can only accommodate so much change and there is a limited need for entrepreneurs. Most fail. Think hard before believing.
- **Globalism:** Yes the globalization of lots of things has occurred. In some cases it worked out well but in others it has been destructive. We went too far, the trend is over, and some retraction is either likely or advisable. The end goal of the world is not to create a homogeneous mush of culture and society lacking local color or differentiation of any kind, no local customs, no local laws, no local government by the people, no freedom to be different, no religious differences, etc.
- And here is my list of the under reported substantive trends:
- **Privacy**: There is a big need, a big market for privacy. I know nobody wants us all to have any in a commercial or government sense, but the market is mostly everyone and the desire is very great.
- **Information Security:** Companies and governments are losing trillions. Many are still in denial. This alone trumps all the hype items above.
- **Border Security:** Pretty much every country on the planet wants more of this.

Hey, these are just my opinions. You are free to disagree. But this is what I am seeing on a daily basis. Let me know what you think.

5.13 REAL VS HYPE 2015, JULY 2, 2015

Here is Matt's list of real issues vs hyped issues for mid-2015:

- Agile Software Development: Agile software development per the "manifesto" is impracticable for large, complex or back-end systems. This stuff is being massively redefined by any who must use it for other than simple, small, user facing applications (apps or web pages). Hype.
- DevOps: DevOps emphasizes pushing out more code, faster, regardless of applicability to the problem. It ignores a wide range of corporate controls for real integration into operations, as currently promoted. Hype.
- CyberSecurity: Costs of exploits and identity theft grow daily. Major intrusions are in the news routinely. Real.
- Software Quality: This has become a major issue, including releases of untested versions with vulnerabilities. Yet the issue remains unpopular. Real.
- Hydrogen Fuel Cells: The technology is real, but distribution of hydrogen remains fantasy. Hydrogen explodes, distribution is unwise. Hype.
- Artificial Intelligence: Once this was the technology that was the future, and always would be. Now it is getting capable, and dangerous. Law is far from catching or controlling the danger. Real.
- Genetics: Improving daily. Real.
- Additive Manufacturing: This technology is not yet mature, and far from cost effective yet in many applications. Mostly hype, so far.
- Drones: Real, duck now. The problems are here, now. Now.
- Distributed Manufacturing: Means of production may be spread around, but know-how cannot. This is a threat to the national security of many nations. Hype.

- Mobile: Everybody will want to manage their healthcare via running SAP on their little pocket screen! Apps will replace all enterprise applications! Run document editing on your phone! Part real, part hype.
- Cloud: There is no ROI. Hype.
- Big Data: Hype. Data Analytics: Real. The new platforms are old technologies with niche applications, the old RDBMS can store terabytes, but the need for analysis remains,
- IoT: This is a collection of different approaches, not unified. Most of it should not be placed on the Internet. Big problems remain unsolved for real use. So far, hype, but moving fast.
- Feel free to disagree.

However, everyone interested in this post should also be aware of technology readiness level models. DHS has one. DoD has one. NASA has one. You need to skim one of these if you think about technology trends over time.

Try not to make the success of your architecture dependent on the realization of technologies that are hype (likely to die off) or immature (not ready).

5.14 SEND MONEY NOW, FEBRUARY 22, 2015

Project funding is scarce. Which software projects get funded in your organization? Which projects "deserve" funding? How do you pick?

STATUS-QUO:

People may have different subjective viewpoints:

1. Project Manager: I have a signed charter, The Stakeholders and the sponsor say my project should be funded!
2. Software Developer: We use Agile. The end user tells us how to improve his work. Obviously that should get funded!
3. Project Sponsor: My job is to improve my department. All the other departments have software to simplify work, and I need to get mine improved. Its in my performance plan.
4. There may be several approaches to funding software projects:
5. Pick the one we have money for.
6. Pick the one with the best presentation at the big meeting.
7. Rotate money around to different managers' projects.
8. Give it to the best managers and their pet projects.
9. Fund all of them, or as many as possible.
10. How do you measure the value delivered by the software projects?
11. DevOps: We deliver more code! Much more code!

12. Agile: We focus on delivering working software that responds to customers and their changing priorities!

NONSENSE!

Most software projects offer some convenience to people performing their present jobs. However many of the present jobs being performed are inefficient, wasteful, and maybe not what the company has chosen to do next. Building software to automate wasteful inefficient processes is like casting that ineffectiveness in concrete so it can never be changed!

What is the composition of a typical set of project funding requests? Lets say we have 100 requests for funding:

- Many (maybe 50 of our hundred) will be redundant software to do the same job some other software already does, or another funding request. Share you childish managers!
- Many (maybe 75) are to support obsolete processes or work methods that should be revised. No software should be begun until the new process and methods are selected. Face the future you resistant workers!
- Several (say 60) may have nothing to do with the organization's core business, focusing on support functions or irrelevant operations that will not affect the top line, the bottom line, or anything but the vendor's sales.
- Only a few will merit funding.

EFFECTIVE PROJECT FUNDING

The difference between an organization that learns and improves is how it spends funds on improvement (aka transformation). An organization falls on a spectrum from stagnant, random efforts to directed, targeted efforts at improvement. (An organization that improves is more truly "agile" than those following the mumbo-jumbo of a thousand pundits.) Look for these traits:

- Business cases are used to justify funding.
- Performance is measured.
- The business cases with the highest ROI (in terms of performance improvement) are selected.
- Sponsoring managers are not involved in voting on the decisions. Objective 3rd parties are used to decide, based on the well documented business cases.
- Effectiveness of the projects is measured in terms of performance improvement.
- No partial funding to create "zombie" projects lacking sufficient resources to achieve effectiveness.
- Funding is tied to strategic goals and objectives.
- Core business improvement has a higher priority than support functions, unless there is a strategic goal involved.
- The discipline called "portfolio management" (aka CPIC in US Fed. Gov.) is present, supported and used for these decisions.

ORGANIZATIONAL EVOLUTION:

Is your organization a dinosaur? Is it known and criticized for widespread ineffectiveness? Does it fund ineffective projects that have little effect and do not address your future direction or

improvements? Is funding driven at random by the debate and politics between the line managers involved? Do end users and ordinary labor set your future path? Have you no vision of the future? Then stop reading this stuff and go prepare to be fired.

The organization must have a vision, a strategy, a direction, a mission, or it wanders randomly like an amoeba across the microscope stage of investors or oversight organizations. If you want to become a more effective and better performing organization then some work and some management commitment will be required.

Enterprise architecture is about organizational transformation and performance improvement, which is all important stuff that is respected and recognized. You don't have to use the term to do EA.

5.15 PICK PRODUCTS FIRST, MARCH 25, 2015

Why is there always some useless technophile trying to pick products first, and then guess what requirements can justify their frivolous decision? Is this sheer ignorance? Have they no interest in the needs of the organization? Are they stupid, insane, so jaded they refuse to make an effort, too lazy? Do they simply have a genetic lack of integrity? Where do these people com from?

> EVERY FEW MONTHS YOU RUN IN TO ONE OF THESE FOLKS AS AN ARCHITECT. "GIVE ME AN ARCHITECTURE WITH PRODUCTS X, Y AND Z IN IT."

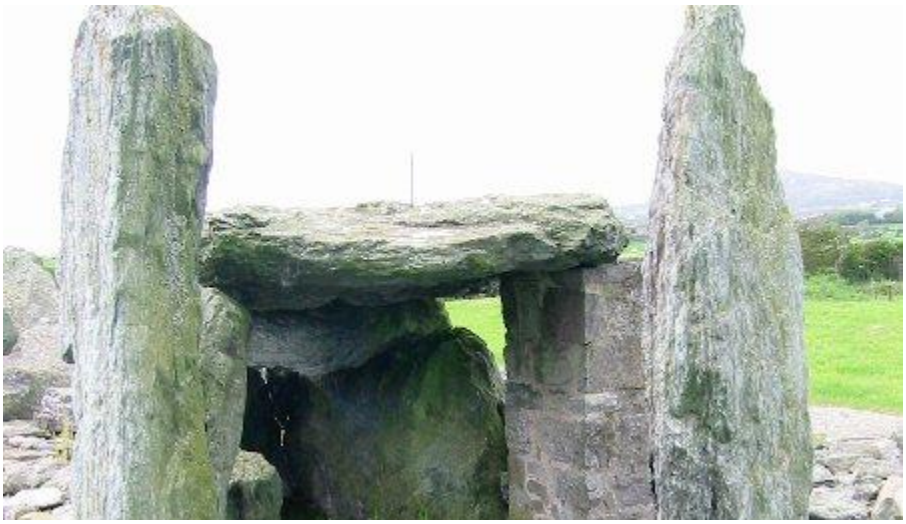
Listen you spoiled child, these IT systems are not your toys:

1. If you think architecture has only technical elements, you are hopeless. Architecture artifacts should include those that link to business need and functional use.
2. Just because some sales chick in fishnets bought you lunch and promised you a seat in the box at the ball game does NOT mean your organization needs this stuff.
3. If you just want some fool to draw diagrams any way you tell them, you are wasting the time of your architects and your organization or customer. The value of their time may be more than the value of yours you useless dunsel.
4. They have a name for this kind of behaviour. They call it fraud, waste, abuse of funds.
5. I met one of these useless sorts not long ago. The joker would screw up and blame others for it. He would demand that professionals ignore the customer and write up some other drivel. He had no particular skill, and the only way he had achieved power was through sleazy dealings as far as I could tell. I guess it all fits in a nice package.

They are like potholes in the highway of life. Put a big red X on them so the next architect knows to avoid the job. Walk away! Walk away!

OK, end of rant. You can go back to whatever. ‘Sorry to interrupt.

5.16 ENTERPRISE SOFTWARE REUSE, FEBRUARY 18, 2015



Software reuse is the holy grail of software developers. Theories and models and patterns have been developed to make it happen. The entirety of FEA cost savings in US enterprise architecture was once based on a notion of reuse. However such notions of cost savings via reuse almost never become reality in enterprise computing.

VBX

The VBX component was perhaps the greatest software reuse success ever. In the heyday of Visual Basic 6 (tm) there were hundreds of VBX components for sale to build the majority of your application. You wrote a bit of glue between them and POOF, success. The release of VB.net (tm) heralded the end of the VBX, although the market remains like a sort of zombie in the .net era.

JAVA BEANS

JAVA Beans are to JAVA as the VBX was to VB (tm). Only that never really happened to the same ubiquitous level. JAVA Beans remain as a sort of JAVA oddity, used occasionally.

3 TIER ARCHITECTURE

In three tier architecture you wrap all your data objects. You leave all the logic out of your presentation objects, All your business logic goes in to your business objects, which will be reused. But business objects are not reused, they are specific to some application which is usually tied to some implementation. Implementing the same applications multiple ways, in the same language and environment or close enough for reuse, is redundant. It wastes money. Cost savings are almost never achieved in 3 Tier Architecture.

Note: an apologist for the 3 tier approach (in the comments) has admitted that it has higher initial costs (much higher) but says that the maintenance costs will be lower. Yet in my experience that are not, with so much additional code to maintain. Further the lifetime of such systems may be foreshortened. Lastly, entire pages may be rewritten based on different business process, no problem with the rapid development in 2-tier, negating conventional maintenance of the prior page designs. In total maintenance costs are lover for the same functionality in 2-tier unless some suite

of applications changes the balance by business object reuse- which almost never occurs in enterprise environments in my experience.

COTS PACKAGES

We still have the ubiquitous Commercial Off The Shelf Software, the software product. Via the product many customers can share the same code, reuse it in a gross sense, and save big money on implementation and maintenance of that code. Unfortunately even here enterprise buyers take the COTS package and re-code parts of it, destroying the economics of the shared code-base. Customers refuse to color between the lines. Just look at your typical ERP or CRM implementation.

FEA COMPONENTS

The US government's EA program was fueled in part by a notion of software reuse where major applications would be vended and delivered with and in their hardware package. When the data center needed a new CRM system or ERP system or database, one would be wheeled in within a crate, unpacked and deployed. That never worked, of course.

COMMERCIAL PRODUCTS

The only place software reuse seems to work is in commercial products unrelated to enterprise software. Apps and games and such without databases and roles and customization seems to agree with "inheritance" methods. Hot stuff. But those inherited classes do not seem to transfer benefits to some completely other developer from some other company customizing ERP or what have you.

WHY THE DIFFERENCE?

In a company producing software there may be one or few coding environments. In the enterprise there are many products from many vendors. Many languages and development environments may be used. The opportunity for reuse is limited, and any effort to systematize reuse is far more complex.

PROGRESS

Also, as Jason Holt points out, there is always progress, newer code based on other newer code or a newer approach limits reuse everywhere.

WHAT WORKS

For enterprise software cost savings really occur in those environments that implement 2 tier logic. All the extra wrapping of data in 3 tier development is wasted anyway. The code to implement the application is built in to the front end, without some beautiful conceptual separation between business objects and presentation. These environments just will not die, such as Cold Fusion (tm), PhP, and now HTML5. They ignore software reuse and the data is reused in the back-end, available to multiple applications and apps.

The VBX approach also worked, but required a supporting environment that was not transferred to JAVA in beans.

CONCLUSION

Give up your academic dreams of software reuse in custom enterprise coding. Face the empirical truth. When the development team promises reuse, they are probably wrong. Limit them to the reuse projections within a single project for cost comparisons and alternative comparisons.

5.17 SYSTEM ENGINEERING INEFFICIENCY, NOVEMBER 27, 2014

System engineering has been criticized as a large and inefficient effort, sometimes ineffective. There is a good deal of inefficiency in the practice of system engineering. Here are some tips on eliminating inefficiency and increasing effectiveness.

Multiple SDLCs: Everyone has their own System Development Life-Cycle. This causes retraining each time a system engineer moves to a new organization. Further, because each organization has limited resources the SDLC of each is less effectively managed, improved and optimized. Fewer projects form the experience base.
In the US Federal Government for example the SDLC should be added to OMB policy, and a single standard SDLC should exist across government.

Systems are Not Software: Some insist that an SDLC, and systems engineering, is about software. The view of the total system is lost in myopia. Systems may include

hardware and humans. and more. Software centric SE practices should be discouraged in favor of greater generality.

Mixed Paradigms: Some SDLCs include mixed paradigms or redundant conceptual threads. For example a business case and a mission needs statement are redundant. Both are evaluated to determine the need for an investment. Both express the need. A business case also includes the cost estimate, and the ROI, as well as risks- redundant with other often required documents.

In one important SDLC both a system design document, a logical design document and a physical design document are required. Surely these are redundant.

Poor Document Focus: Examine for example the CONOPS document, the concept of operations. Through years of poor OJT (On the Job Training) the clean, simple purpose of this document has been lost. I have seen huge templates for a CONOPS including all material except that core purpose.

A CONOPS SHOULD CLEARLY EXPRESS HOW THE ITEM TO BE ACQUIRED OR BUILT WILL IMPROVE THE OPERATIONAL PERFORMANCE OF THE ORGANIZATION. IT NEED NOT BE FILLED WITH OTHER GARBAGE.

Too Many Documents: A typical contemporary SDLC may contain dozens of documents to be sorted through. By splitting the material between so many documents page counts will grow. The front material explaining the basic concept and context will be repeated. The back material with appendices full of glossaries, other documents, events and whatever will be repeated. In the middle a synopsis of the immediately preceding documents will be required to set the context for the new material.

Adding to this there is an editorial tendency to make each document better and more inclusive, adding substance. This detracts from brevity and pointed progress.

One Approach: Suppose there are only four volumes (barring completion letters and forms for compliance checks) in your SDLC effort:

- The Project Plans Document (including financials, acquisition approaches, estimates and baselines)
- The System Manuals
- The Requirements Document (including MNS or CONOPS and all items in that train of thinking)
- The Testing Document (including TEMP and test plans and test results and test completion letters)

So what, you say? Because the volumes are consolidated appendices and front matter need only appear once. The story is more complete due to proximity, leading to less need for additional clutter. You can cut page volume by half or more.

For any chapter in the volume not yet required in the SDLC sequence, just put in a page indicating that the material is not yet required and will be procured when the system project enters the ---fill in--- stage.

Effectiveness: Proximity and clarity through brevity will reverse not only bloating but ineffectiveness. By clearly and succinctly stating your concepts and goals, your metrics and measures, the system is less likely to wander from those goals and measures.

In this era of Lean and Agile, this kind of simple approach trumps much of the hype and may be more effective than most of the silly language and trivialized process being so widely advocated.

5.18 WHAT IS SUBOPTIMIZATION?, AUGUST 26, 2014

When you focus on optimizing the parts of a system, thinking you are optimizing the whole system by doing so, you produce suboptimization.

Here are three well known cases of suboptimization, widely discussed, and a caboose.

FIRING THE BOTTOM GUY AT GE:

A big name manager was famous for a draconian policy in which each department had to rank all its employees in order of effectiveness, and then fire the bottom few percent, every year.

Just suppose a department of 50 had only persons that were in the top 100 of all employees. You fire the bottom 2, say. That gets rid of 2 of the top 100 people in the company. You have suboptimized!

If there is no mechanism to move people around it will be very common to have some departments full of all winners, and some full of all low performance dead-weight. All departments will not be average. The world does not work that way.

What if the policy had been to try bottom performers in another department, and can them only after 2 or 3 visits to the least effective category? Pools of high and low performing departments would have been eliminated. Biases of individual managers regarding age, race, gender or religion would have been mitigated. Overall organizational performance would have increased. Training and recruiting costs would have been reduced with lower turnover. I would not be citing suboptimization.

MULTIPLE PORTFOLIOS IN GOVERNMENT

The US Government has a policy that requires portfolio management. All the spending on transforming the organization is divided up into different investments (containing programs, projects). These would all be put into a portfolio and compared. Each agency was to fund only those with the highest return on investment, the most improvement for your tax dollars.

Instead many government agencies broke the money first into several portfolios by subject area. They allocated money to each portfolio. They allocate more money to problem areas. The result was that important programs that could improve government were not funded, where as marginal programs that would barely improve problem areas were funded. Suboptimization!

This was exacerbated by "zombie programs", poor performing programs that were never killed but just reduced in funding to keep them moving... although so underfunded they would never produce significant results. Worse, vast chunks of transformation effort have been left out of comparison, hidden as tiny investments or mis-characterized as special categories of expense. Sometimes massive fraud occurs in unreported programs. More suboptimization!

THE SQUEAKY WHEEL

In business and industry top managers routinely make some pet manager their showcase. They grow that area, fund that area, lavish extra help on that area. Insufficient attention to other areas results, and overall reduced performance. Personal biases, cronyism and similar issues hinder the performance of many organizations. Suboptimization!

COMPUTERS

Suboptimization was not originally a management term, although organizations are systems. Management learned the concept from system engineering. Today we have computer hardware and operating systems that overemphasize performance at the cost of security. Security losses to users of those computers are massive, but no change is evident.

CONCLUSION

Suboptimization is a hallmark of poor management, Try to avoid it. If you do this kind of thing, don't get mad at me for posting. I can't really help you. Go study some system engineering or some management. It's not my fault if you produce poor results. Honest!

5.19 WHY FEDERAL IT PROJECTS FAIL

I will try my hand at a better top ten list for why Federal Government IT projects fail. I saw one recently, and I think I can do better. I will do it free-style, without a net, for this pass. I believe I may have a better perspective than most vendors, having worked for the government customer for many years. Let me know how I did.

1. **Vendor Hype.** Often vendors oversell some solution or method, making excess claims. They do this in a formal proposals, with great credibility. They may do it with the backing of pundits as an industry trend. Unfortunately, in practice, the solution does not deliver.
2. **No Real Organizational Benefit.** Often the acquisition was never thought through, and the performance of the organization (not the software) after implementation is no better. After spending a ton of money, the government folks who do the real work see that the change has been worthless. This happens when the stage gate system defining how the system will improve the organization is bypassed, and test or evaluation to check that that the operational link is working may also be bypassed. In many cases no business plan may have ever been written or reviewed. The system may succeed but the application may fail.
3. **Not Understanding Agile Limits.** Agile is great for small projects focused on human interface. Agile often fails for complex projects of large size focused on back-end function. This is true even with Agile extensions. Complex system development requires additional systems engineering, architecture, and testing/evaluation efforts. Applying Agile to the wrong systems has resulted in widespread and catastrophic failures.
4. **Confused COTS, GOTS, MOTS, Custom tradeoffs.** (aka Poor Buy vs Build Decision, or poor alternative analysis). You can but an off the shelf system and

change the business processes to fit it, or you can build a custom solution to fit your custom processes. Custom software is more expensive than off the shelf software, but costs of changing internal processes are often ignored. It is far more expensive and difficult to modify your off the shelf software to fit custom processes, more so than either custom or off the shelf alternatives. Risk skyrockets. This is poorly understood in many government agencies. Failures result.

5. **No Distinctions between Mission Systems and Support Systems.** Mission critical and mission central systems may require custom development, as often there is no one else doing that mission and no software to automate it. These systems, of direct import to the mission, have the most likelihood of increasing organizational performance. Support systems may as well be COTS, as they will not substantively improve organizational performance. By putting too much effort into support systems money required for real benefit is squandered, leaving insufficient funds to implement mission systems.
6. **No Logistics Train.** Once the solution is delivered, there is often no plan to support or maintain it. Costs skyrocket versus estimates. Funds run out. Failure occurs.
7. **Ignoring Non-Material solution.** Often the government agency could simply change the way it operates and buy noting to increase performance. Vendor and political pressure may force a purchase anyway, so as to appear to be doing something. The results may prove immaterial. (This is a sub-case of item 2)
8. **Poor Project Management.** Often agencies ignore standard practice from PMI or decades of project management wisdom in favor of simplified methods. Stakeholders may be ignored. Scope may increase. Requirements may change too much. Risk may not be managed.
9. **Lousy Requirements.** This is a subset of item 7. Customers often get what they specify, even if they specified the wrong thing. Sometimes customers specify too little, and get random results or missing functions.

10. **Integration in a Can.** Many organizations think they can just leave integration as an afterthought, trusting SOA or EAI or ESB to fix everything. Integration takes work and analysis, specification of interfaces and testing. Improper effort may cause failure.
11. **Zombie Investments.** The Federal Government is not very good at canceling programs. Sometimes when a program goes awry, it remains active with a minimum of funds. At such funding levels it can never achieve objectives, and it fails against the original plan.

    OOPS, I got 11 instead of ten. It was a quick try. What do you think?

5.20 TOO MUCH MODELING, TOO LITTLE TESTING

The US Coast Guard (USCG) is one of my favorite government agencies. When I was a boy in Erie Pennsylvania our local Civil Air Patrol squadron cooperated with the USCG on a few SAR (Search And Rescue) missions. The USCG is full of fine people, some of whom are my friends. We all share some common values and experiences. A few years ago, the USCG made a mistake.

The USCG fleet was aging, obsolete, and desperately needed refurbishing. Instead of doing this piece by piece they did it in one big contract. That is understandable, as issuing contracts is difficult and time consuming. However the contract they issued trusted the contractor a bit too much. With USCG identifying the need, the contractor was free to design, engineer, modify, and deliver among other things, 123 foot cutters.

The USCG needed more cutters. The contractor had the bright idea of slicing through the center of patrol boats to be modified and inserting a new section, creating 123 foot cutters. Thirteen feet would be added. That is not a bad idea- so far.

Now here is the core of it. When sent to sea trials the cutters in question would shake and vibrate in strange and unnatural ways. They could not reach or even approach top speed. They would shudder, vibrate, and threaten to fall apart. The new hull sections were not strong enough to keep from moving about under stress at speed. This caused major problems. The contract was halted. The ships were docked (several had been modified). Stress cracks had developed, and leaking holes had occurred.

None of this was unexpected to the engineers at USCG. They had complained that the structural strength of the hull section inserts was inadequate. Even several engineers at the contracted company had predicted failure. No one listened. All testing on the contract was performed by the vendor, none by the customer. Testing of one unit by the customer, before modifying the other units, might have avoided various problems, reduced waste and saved money.

" The Coast Guard and ICGS had not identified the cause of the problem or why their computer models did not predict the problems that are occurring. "

One brave engineer with high integrity became a whistleblower. All 8 modified patrol boats (now cutters) were scrapped in 2006. The contractor was accused of supplying false data, and this false data may have been the source of the modeling error.
Now in the field of Enterprise Architecture we model a good deal. However we do not test against reality as often as we might. Software development too no longer tests

as much as they might. We seem to be failing a good deal, especially in regard to security. Perhaps we could all do a bit more of that testing stuff. Without empiricism we might as well be theologians and philosophers of the dark ages counting angels dancing on the head of a pin. This is not to say we should not model, but only to say that modeling does not completely replace testing. There is a limit.

(Edited for accuracy 23 Dec 2015. The "mixed metal" issue occurred on a different class of boat in the same program.)

5.21 GOOGLE™ VS ENTERPRISE ARCHITECTURE

Recently pundits have written that Google™ has advanced Agile enterprise architecture. Those who understand the subject will note that the enterprise level of architecture supports portfolio management, and the deliberate choice of transformative investments. This is not solution architecture, and distant from it in methods and practices. For example software development is not much involved. Between the two, segment architecture improves a service or product, but also has little to do with software development directly. We are speaking here only of the architecture of enterprise transformation, supporting the portfolio, whose purpose is to select investments in transformation of the enterprise.

An architecture is a set of component parts, some relationships between them, and some operating principles for a thing (paraphrasing ISO and IEEE). At this (enterprise) level the components are entire systems, entire databases, datacenters and offices, strategic goals and business drivers such as new laws and policies, not smaller

details. Think big. Relate these to big pots of money, investments in the future of the company or organization.

So, let's examine if Google has advanced enterprise architecture, or not. Google currently spends 2.7 billion on R&D. These R&D efforts should be listed somewhere, managed, and controlled in a portfolio. Now enterprise level architecture, restating from above, supports portfolio management. If EA exists the portfolio would have transformative investments, like R&D, that "align" to the corporate strategy to improve operations. Enterprise architecture is all about "alignment" and improving internal processes and such to improve performance.

http://ycharts.com/companies/GOOG/r_and_d_expense

Instead Google invests in spaceflight, bicycle powered monorails, immortality, beekeepers, and windmills. These do not seem to "align" with a sane strategic plan and do not seem to be targeted at improved internal operations and improved search services, improved web services, or any identified line of business. Some other transformative investments, such as buying Motorola and Android or buying YouTube do seem to align and might produce improved operations. The investments are sometimes aligned, but many are way out there on the fringe.

http://www.businessinsider.com/google-weird-uses-of-money-2011-5

I think this broad investment push may be inevitable when you have so much money you have difficulty employing it well. But enterprise architecture and portfolio management are all about employing that capital effectively and efficiently.

Based on the helter-skelter nature of transformative investments at Google, there seems to be no architecture behind its transformation. Google seems to employ a shotgun pointed at a wall full of potential investments, hoping some will succeed. I am not sure such a haphazard approach, lacking alignment, could itself be called Agile in any way.

You can say that Google has Agile solution architecture, and I may believe you. However, knowing the purpose and definition of enterprise level architecture- well- I am hard pressed to agree that that is occurring. Presumably when investors reign in the money, and money becomes scarce, Google will become like all other organizations and seek to apply that capital only to transformative investments wht will improve the top line, the bottom line, and only existing products or services. Wild bets on possible new products and services will presumably end.

For the time being, Google seems to have too much money to bother with portfolio management, enterprise architecture or alignment of any kind. It also seems to have fans who do not understand the term "enterprise architecture".

COUNTER-EXAMPLES

Here are some examples of companies who made investments that did transform their business:

For a contrast to Google look at Amazon's™ investment in robots for stock picking. You could convince me they have an enterprise architecture.

https://www.youtube.com/watch?v=UtBa9yVZBJM

Here is a slide deck describing transformative investments at Fed-Ex™ that made them a market leader and icon. You could say they had an enterprise architecture, although the coin and methods had not been formalized yet, and that the investments they made aligned to the mission.

http://www.slideshare.net/HarishkumarC/fdx1

Governance & Maturity Management

SECTION 6, AGILE SOFTWARE DEVELOPMENT & DEV OPS

6.1 CODE RELEVANCE.,MAY 17, 2015

Some software will enhance top line growth, bottom line growth, mission performance, quality of delivery and/or execution of strategy. Other software will not. All software is not equal in this regard. The software that does enhance these operational imperatives is relevant.

"Code Relevance" is the measure of how software (especially enterprise software) impacts operational imperatives. I have coined this term to make the issue clear.

Code relevance can be seen as the portion of "alignment" (an enterprise architecture term) that applies to software. Not all of EA applies to software, and much of it does not apply to software, but code relevance is specifically about software and its alignment to operational goals.

If you produce software with high relevance, it will have a high positive ROI and it will improve operations. If it has lower relevance, a lower ROI will result. If the software has no relevance, the ROI will be zero or negative, cost without return. The enterprise has limited funds for software development or procurement, so average code relevance should be kept high.

Code relevance for direct mission support software can be higher than that for support functions. The software at FedEx that allows overnight sorting and routing by barcode, for example, will have higher code relevance than the HR recruiting portal.

Code that directly implements the strategic intent for organizational transformation of senior executives can have higher relevance than code that makes some end user's job a bit easier. This is

especially true if the end user's processes are obsolete and the code will simply enshrine that process in permanence.

When enterprise architecture is applied to software development, the intent is usually, and most appropriately, to increase code relevance.

Agile Methods and DevOps often ignore code relevance. GigaSLOCs (individual lines of code measured in billions of lines) of working but irrelevant code just increase cost of implementation, maintenance and support without much ROI. Code relevance outweighs mere software production, and one could argue that there is no software productivity without code relevance.

I am not saying code is sometimes slightly irrelevant in some cases, in some few organizations. I am saying that I have seen organizations where the majority of the code is highly irrelevant. In those organizations money is being wasted by the bucketfull. Without organizational controls, the irrelevant code may dwarf valid relevant coding or software acquisition efforts.

6.2 AGILE LIMITS, JULY 4, 2014

Agile Development Sweet Spot	Small, Simple	Large, Complex
Front End (GUI) Development	Applicable	Modified Application
Back End Development	Modified Application	Not Applicable

Now that the hype has begun to subside, and the era of labeling everything "agile" has begun to diminish, I will try to state a few true things. Agile software development as a management technique faces two important limits.

GUI VS BACK END DEVELOPMENT

Agile is based on the notion of user stories, epics, etc. This is applicable to the development of user interfaces. There are types of software development where the end user is not involved. The database and storage structures of a large enterprise application is one example, where the data model should be analyzed and developed for perhaps months. The user is involved, but the structures undergo extensive analysis without the user. Another case is SOA, ESB, EAI, or middleware development, in which internal data structures between large applications are exchanged and merged and adapters are developed. End users most often have no direct role.

SMALL, SIMPLE APPLICATIONS VS LARGE COMPLEX APPLICATIONS

The simplified methods of Agile, keeping the effort on the wall on small cards, are well suited to smaller applications. As the amount of detail to be managed begins to exceed wall space, the utility of Agile fades and more formal management methods begin to provide superior results.

WORK-AROUNDS

For both limits there are means to work around Agile limits. In larger systems you may apply "scrum of scrums". Back end development may be treated as special sprints with analysts or architects as the user. The first sprint may be designated as where the database model is developed, with extended length. In all these cases the clean vision and definition of Agile is stretched. A long list of other fixes is not presented here. At some point, calling it agile is a semantic artifice, and subject to debate.

SCRUM VS AGILE

I believe it is important to separate the notion of iterative construction from the notion of a manifesto of extra constraints. Responsible engineers that I have spoken with do not often indicate that construction of whatever should not begin with an example (prototype, pilot, sample) that will subsequently be improved after examination or testing. The use of the scrum in software development is often cited as predating agile by decades. Observations here regarding the limits of agile software development are not intended to argue against iteration, prototypes, pilots, or scrum.

OTHER METHODS

Luckily there are other technology management methods specifically designed to function best where Agile Software Development does not. Systems engineering is intended to manage complexity, and has a proven history of success back to at least WWII. Relational design, object oriented design and analysis, master data modeling, and other methodologies have been proven highly effective in managing back end data structures and canonical data structures for back-end development.

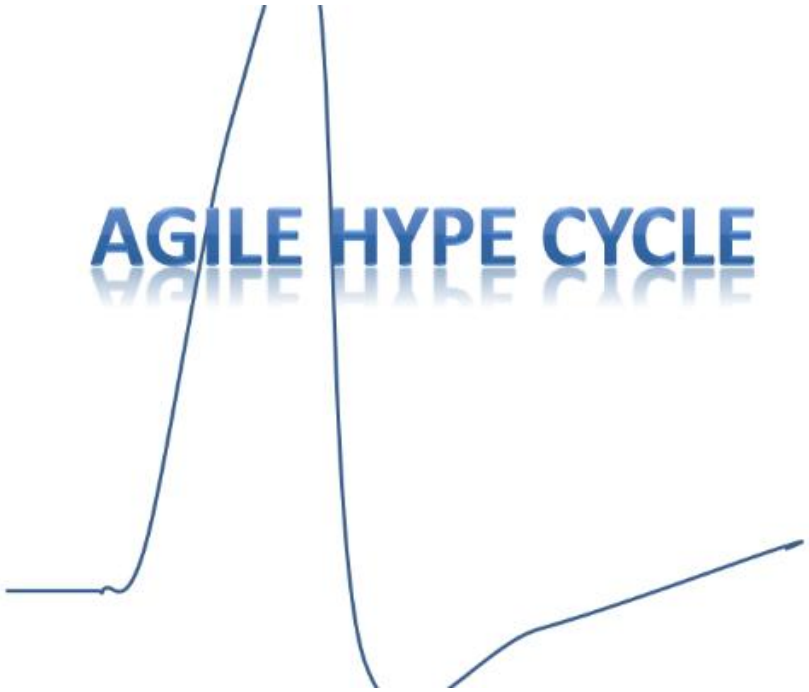
CONCLUSION

In truth a range of approaches, a hybrid mix, of management methods is required to succeed in today's enterprise IT environment. That customer enterprise environment never was like the simplified product development environment where Agile software development was conceived. In real use on successful efforts various approaches are merged together in pragmatic ways that cause debates to become semantic efforts.

AFTERWORD

Hype and buzzwords pass through the industry constantly. They may do significant damage as they pass, causing misunderstandings and failures of critical efforts due to purist idealism. Beware hucksters bearing buzzwords. Beware manifestos and true believers. As engineers we are committed to the application of science and measurement of real results, not religious fads.

6.3 THE AGILE HYPE CYCLE, MARCH 15, 2015



Have you heard of the Gartner (tm) Hype Cycle? Most managers in technology have. What if there was a Gartner (tm) hype cycle for Agile Software Development? This is not a new concept, several have asked the question.

> So where would Agile Software Development be in the Gartner (tm) Hype Cycle right now, in March 2015?

I complained about Agile Hype last year:

- https://www.linkedin.com/pulse/20140809133718-86002769-2014-hype?trk=mp-reader-card

Well, Gartner did place Agile on their curve in 2014. It was at the peak, maximum hype.

- https://www.gartner.com/doc/2810920/hype-cycle-application-development-

After the peak comes the "Trough of Disillusionment". This guy asks if Agile made it through to the proven part of the curve. His post seems premature:

- http://www.infoq.com/news/2011/03/agile-trough-of-disillusionment

This guy declared Agile dead last year...

- http://effectivesoftwaredesign.com/2014/03/17/the-end-of-agile-death-by-over-simplification/

Here is Brian complaining last fall...

- https://plus.google.com/108396230666464824427/posts/2SwcgTC5SuV

These guys here are focused on the downside of Agile, and debate it routinely:

- https://www.linkedin.com/groups/Dark-Side-Agile-4688911

Actually Dave Thomas (Manifesto guy, not the hamburger guy) declared Agile dead some time ago and the debate has raged ever since. This is old news.

- http://pragdave.me/blog/2014/03/04/time-to-kill-agile/

Not to be left out, I even wrote a post saying Agile has limits (Duh):

- https://www.linkedin.com/pulse/20140704132728-86002769-agile-limits?trk=mp-reader-card

...And another pointing out that agility is the opposite of maturity scale, there is a spectrum (Do I remind you of Capt. Obvious now?) (The benefits of maturity, BTW, a proven approach, may significantly outweigh those of Agile.):

- https://www.linkedin.com/pulse/20141120095201-86002769-agility-vs-maturity?trk=mp-reader-card

RECOMMENDATION:

If you are still in the middle of moving your organization wholesale toward Agile software development methods, ***STOP***! I think this panacea is about to hit the through, and has not moved through it. The depth of criticism and blame that could be applied to Agile has barely been scratched.

- http://darkagilemanifesto.org/dark-side-of-agile-janes-succi-splash-2012.pdf

The damage caused by misapplication of Agile has not been dully tallied. It has played a significant part in billions of dollars of big-integration failures. Distance yourself now! At best the wave is over. At worst you are about to be painted as part of the failures. Apply this stuff only in its range of applicability! Stop the madness!

NEXT WAVE:

Dev-Ops is the next big hype-wave in this space. Gartner had it still heading to the peak of hype in 2014. I wrote a bit on Dev-Ops:

- https://www.linkedin.com/pulse/20140809223124-86002769-beyond-devops-combined-ops?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141101092654-86002769-devops-or-devops?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141111154421-86002769-bi-modal-it-combined-ops?trk=mp-reader-card

Mostly I have said not to let this goat rope screw up your SDLC and corporate controls. Did I really have to say that? Clearly mania has been rampant.

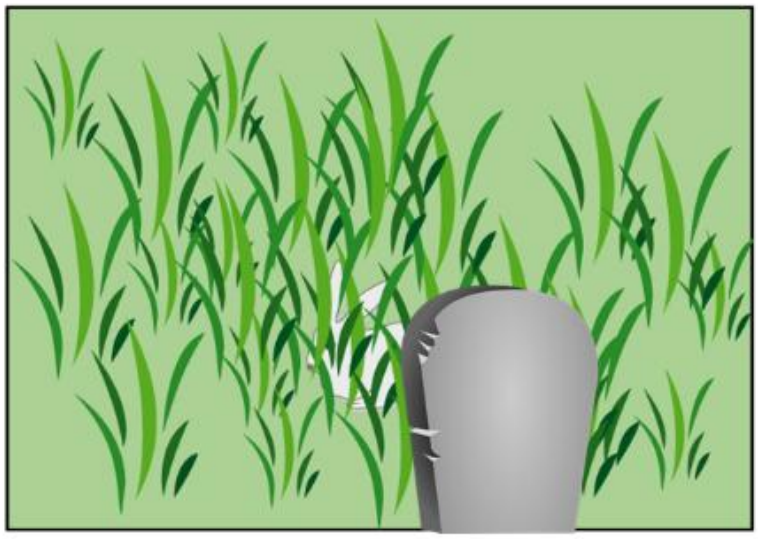
WATCH OUT:

There is a whole branch of the consulting industry selling you ideas just to make money. I have written about that too:

- https://www.linkedin.com/pulse/20140926111337-86002769-false-innovation?trk=mp-reader-card
- https://www.linkedin.com/pulse/20140821111841-86002769-buying-ideas?trk=mp-reader-card

Buyer beware. All management fads end.

6.4 AN AGILE EPITAPH, MARCH 17, 2015



An alternate reading of the holy gospels according to the Agile Manifesto:

- "Individuals and interactions over processes and tools" This phrase implies immature processes and process management, and a low CMMI rating. The tools portion will not be addressed as nearly all Agile projects are highly focused on a specialized set of oddball tools.
- "Working software over comprehensive documentation" This implies slipshod documentation processes and software that may not match specification or manuals.
- "Customer collaboration over contract negotiation" This phrase implies shoddy contract management and low reliability in achieving contracted objectives.
- "Responding to change over following a plan" This phrase implies poor project management and execution, poor change management and poor adherence to prior plans or specifications. Commonly either sufficient system engineering/architecture will be performed before programmers show up, or such work will be ignored in Agile projects.

Perhaps most important, the Agile movement was a form of suboptimization, a system engineering term denoting optimization of a small component that reduces overall system performance. In this case software development attempted to optimize itself without due consideration for corporate controls and the role the business case in setting ROI by establishing firm objectives for the effort. Portfolio management would be used to eliminate the vast majority of software boondoggles that provide no real operational value, but not in Agile as commonly implemented. Mostly Agile just implements the user's desires for any group that can squirrel away a budget, and those particular users or their current methods and operations may not feature in the new processes that have ROI and enhance organizational performance. Automation in so many of these cases is an Agile waste of money and effort, regardless of how many lines of working operationally irrelevant code you might produce.

While the original intent of the Agile folks was very constructive, the actual reality has been far from the visualized ideal. The hype wave is ending.

'Next fad please.

6.5 AINO (AGILE IN PRACTICE), MAY 22, 2015

Agile as first envisioned has its place, its range of applicability. But Agile software development, as described in the "manifesto", mostly does not work for large or complex projects or backend development. Many large organizations have adopted the name Agile and put the important processes back in. You might recognize the list. These important processes may include:

Decision Analysis

Risk Management

Interface Management

- Technical Planning
- Requirements Management
- Configuration Management
- Technical Assessment
- Technical Data Management
- Architecture & Design
- Contract Management
- Stage Gates and SDLC

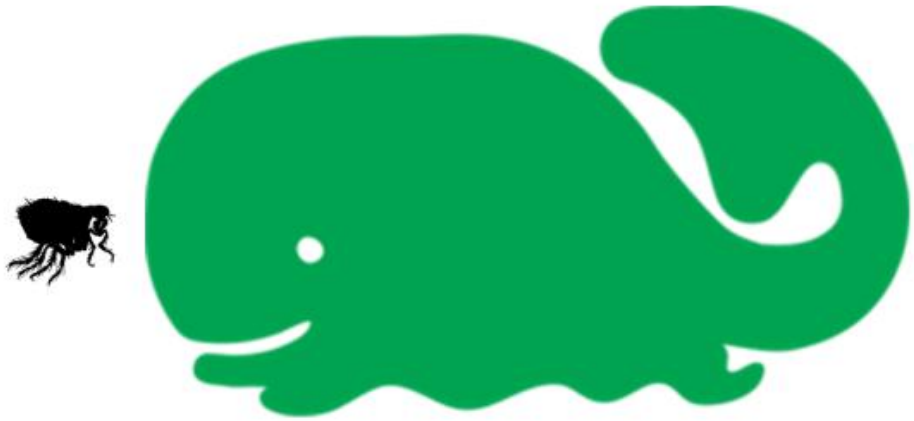
Most of these things were demoted to the sidelines in the manifesto, but they are back in full force, in real use, right now, some here, some there, more over there. Many current Agile practitioners really practice **AINO**, a highly modified form of the concept grafting on much of what was too hastily discarded.

I CALL IT AINO: AGILE IN NAME ONLY

Like Al Einstein said, you should make it as simple as possible but not one bit simpler. It breaks.

So when we all meet in private, we can all talk about the true next generation beyond Agile, **AINO**. I just call them like I see them.

6.6 SCRUM SANS THE AGILE BAGGAGE, JUNE 6, 2015



I have written several posts about how Agile Software Development may not be appropriate to your ***enterprise software*** efforts. This post describes how those observations do not extend to **SCRUM** methods. (For links to those past posts concerning Agile or DevOps please see the list at the bottom of this post.)

In 2005 I was privileged to work with a small group of real programming geniuses developing a commercial product We had to modify an existing police/fire/ambulance dispatch legacy codebase to add situation awareness display for sensor data. This code had been migrated at least twice with major platform changes, and was not modular. It was, in a word, inflexible.

Onto the legacy software we had to add an external real-time in-memory hierarchical database for sensor status and properties. Modular replicable drivers for sensor systems would attach to that. A bit of the cutting edge was in there, and some challenging development. I was the solution architect on this team of brilliant programmers who far exceeded my talents in coding, probably beyond anything I might achieve in this life (but hopefully not perhaps in system architecture).

We used Scrum. Now this was 2005, and scrum was not yet very popular. Yet I and some of the others had used it or something like it before, before Agile Software Development became routine, before the hype, before the manifesto became popular. Here we would use it again, something scrum like, with daily and weekly meetings and a whiteboard with a few architecture documents. I would have to say what we achieved was pretty miraculous in the time allotted. It impressed me anyway.

This was definitely modified scrum. No existing user base had ever seen anything quite like this new software. The internal details were often in the back end and very complex, as were the issues. No product manager outside our group understood the vision and detail sufficiently to

lead. We were not doing anything quite like Agile, it was something different than that, but we used scrum anyway. To this day some become emotional and even argue that this is not possible. I wonder what they would think of several other systems we implemented in the early 1990s, around 1990 to 1992, using scrum but not Agile?

So how did we do this, transcending the complexity limits of what Agile can handle, bypassing the reliance on end users having a vision of something they had never seen anything like, something that had never existed? On simple rule covers most of it:

> ***PRODUCT OWNER = ARCHITECT***

Yup, the architect has that vision of the thing that has never been built. The architect has the understanding of how this sprint must be integrated with others into a whole working system. The architect can coordinate with other architects, governance, customers, outside factors and incorporate all that into the whole. Simple. This was the primary difference between what we were doing and what became famous as "scrum".

You can try to scale Agile Software Development beyond its limits of product complexity, beyond its limits of project size, beyond its focus on visible interfaces product managers and end-users can see. You can try to use SAFe or Scrum of Scrums, but those are not always effective, ineffective concerning certain issues, Many problems remain, such as corporate controls and code relevance issues. However this simple expedient beats them all (in my experience), all those hype driven methods and means. Simply make the architect, or chief architect on bigger items, or subsystem architect on even bigger items, the product manager.

How do I know it scales? Well in the mid 1990s I ran a very large team building the world's second largest imaging backfile conversion system, at the time. We had more than a dozen teams running in parallel, each with a specific user role as the customer. We used scrum, combined with a distinct conventional RDBMS analysis and modeling team to coordinate the common DB model. There were 35 people, and up to 75 at one time. few were using the terms "scrum" or even "agile" back then. But it was scrum. We used a bulletin board. We had real requirements though. It worked great.

Ken Schwaber and Jeff Sutherland were not the only ones doing this kind of thing. I never read their paper. Yes maybe what we were doing was somewhat different, but it had a good deal in common too. Many were experimenting with how to make these software projects more efficient back then. If you want to call it something else, I think you may be stuck on semantics. To me, the main thing is that it works.

This is what I know, and I learned it on real jobs:

> **Using "Scrum" does not require "Agile Software Development", nor the goofy religious manifesto, nor must it bypass corporate controls and regular old architecture, nor need it bypass**

contractual requirements or requirements management, and so on. Yet it is still lean and efficient.

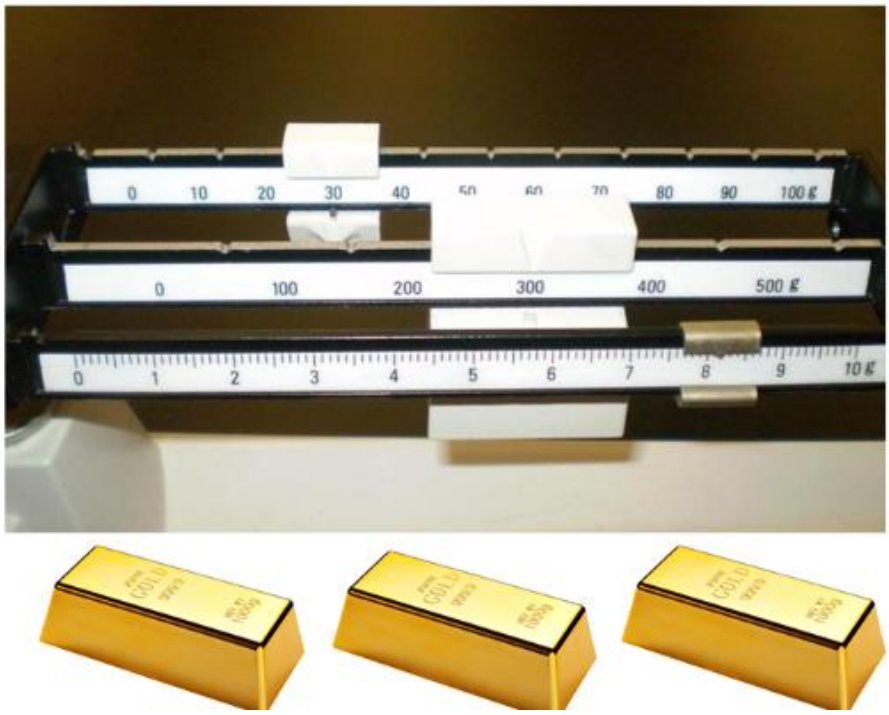
I was there. It really happened. Real, working software resulted. Yup. I'll bet this post will really annoy some manifesto purists and Agile Nazis. Too bad. Its the truth. Go ahead and try it, it works just fine.

... Here is a list of some related Posts concerning Agile limitations, fixing those in DevOps, and measuring the ROI of your software efforts:

- https://www.linkedin.com/pulse/aino-agile-name-only-matthew-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/anti-agile-manifesto-matthew-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/interpreting-agile-manifesto-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/agile-hype-cycle-matthew-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141120095201-86002769-agility-vs-maturity?trk=mp-reader-card
- https://www.linkedin.com/pulse/agility-maturity-matthew-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141111154421-86002769-bi-modal-it-combined-ops?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141101092654-86002769-devops-or-devops?trk=mp-reader-card
- https://www.linkedin.com/pulse/20140809223124-86002769-beyond-devops-combined-ops?trk=mp-reader-card
- https://www.linkedin.com/pulse/20140704132728-86002769-agile-limits?trk=mp-reader-card
- https://www.linkedin.com/pulse/20141116170109-86002769-enterprise-architecture-vs-safe?trk=mp-reader-card
- https://www.linkedin.com/pulse/enterprise-level-anti-patterns-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card
- https://www.linkedin.com/pulse/20140826222008-86002769-what-is-suboptimization?trk=mp-reader-card
- https://www.linkedin.com/pulse/20140726190101-86002769-performance-measures-in-the-enterprise?trk=mp-reader-card

- https://www.linkedin.com/pulse/20141120000844-86002769-alignment?trk=mp-reader-card
- https://www.linkedin.com/pulse/code-relevance-matthew-kern-msea-cea-pmp-itil-cissp-issap?trk=mp-reader-card

6.7 ENTERPRISE SOFTWARE, CAPEX & OPEX, FEBRUARY 28, 2015



I was recently on the Wikipedia page for cloud computing. It proposes that the Cloud moves enterprise software expense from a CAPEX (Capital Expenditure) item to OPEX (Operational Expenditure). This is self serving marketing BS. Here is why.

1. There is COTS (Commercial Off The Shelf) and GOTS (Government Off The Shelf) and Open Source, and then there is MOTS (Modified Off the Shelf). When you take the off the shelf software and rewrite it to fit your need that is MOTS. It involves significant expenditure, front loaded. Almost all ERP and CRM ends up this way.
2. The public cloud hides CAPEX and turns it to OPEX, but most sensitive information belongs on the private cloud. In the private cloud you have some designated machines or capacity, and crypto connection to them. The most secure way to get the private cloud (for large organizations) is to build it internally, in you own data center. He who can touch the machine can take the data. Good luck running internal background checks on every administrator and janitor at your 3rd party cloud provider.
3. The infrastructure in your offices in total often vastly exceeds that in your data center. oops. IaaS is not significant here.
4. All transformation should be examined as CAPEX anyway, including business process changes and personnel and training and such. IT servers are an ever decreasing part of the costs. Most system expenses are not justified (CAPEX or OPEX) and should be examined in a business case.

Sure the vendors want you to buy COTS products without regard to needed modifications, and without internal controls review using the OPEX model. Then slap those in an external cloud where the expenditure and operations are hidden from review like a nice little stovepipe. That moves the most product and lines their pockets fastest.

In the end the enterprise has a limited budget to be expended on operational improvement facilitated by technology. There are more opportunities to spend then there are business cases with some ROI. Choose wisely via a formal central process for review.

6.8 BEYOND DEVOPS: COMBINED OPS, AUGUST 9, 2014

Agile is over, some say. The next wave is DevOps. DevOps integrates development processes with IT operations processes. You can find examples and introductions on YouTube.

But so far DevOps descriptions have left out some of the most essential IT Operations functions, driven by the limited perspectives of the founding communities, which mostly involve vendors of software. In large customer organizations producing or acquiring custom systems or complex integration for internal use, the metrics are not as described so far by the DevOps community. DevOps promises quality software, but this is not enough. It hints at "external metrics" but these processes too are integrated with IT Operations. The future is more integrated than the current vision being marketed.

OPERATIONS PERFORMANCE

Examine for a minute why a large customer organization (not a software company) produces software. The performance of the software is important, but the software is part of a system including external interfaces and platform and network access. Perfectly running software is worthless when the platform it runs on fails to operate, or the network no longer connects, or that key interface to another system is down. Software is one component, or perhaps a few, of a larger system. The software performance merely contributes to, and is subordinate to the system performance. This is the first level of Operational performance outside the scope of the computer scientists and the programmers.

Beyond that, IT Operations has a customer. The customer desires to use the system to perform work. The fact that the screens are pretty and the algorithms are fast is not lost on them, but it is

not the point. The customer desires to improve their operations by use of the software, which will automate or expedite one or more steps in their processes. The customer is measured by performance measures unrelated to the system or its smaller software components. For the software to produce a *Return On Investment* it must positively affect the customer's business processes. It must automate the right process steps, expedite the right operations, perhaps even those in a new process not to be executed until the new software enables its use.

Beyond that, not all customer operations contribute equally to the performance of the overall organization. Some supporting organizations or functions have processes that no matter how much you improve them, do not change the organizational performance much. Others are central to execution of business strategy. It is important to invest in improving the right processes to show an organizational improvement.

Both of these last two concerns are central to the IT operations in architecture and planning functions. The full set of external performance measures that DevOps will be judged by include those discussed here. This is critical to real implementation and success of DevOps in real non-vendor organizations.

INTERNAL CONTROLS

In most organizations development funds are released based on the satisfaction of internal controls proving that development efforts are focused on the real problem and not the whims of developers or working level users. These internal controls such as SDLC and governance are highly integrated into real IT operations.

LIGHTWEIGHT METHODS

Lightweight methods with short iterations and reduced planning are the starting point for DevOps. These include Agile and Lean development, extreme programming, and other related software management. These have succeeded in producing quality software, but have not succeeded in addressing the needs of the systems, process or organizational level improvements required. They are rapid and flexible, however.

HEAVYWEIGHT METHODS

The set of methodologies shoved into a lump by recent hype including system engineering, spiral development with full test and evaluation are proven to produce ROI. Architecture is critical to providing results. These take time and effort. They also produce superior security results.

EXPERIMENTS IN COMBINATION

Experiments to combine lightweight and heavyweight methods are underway at many organizations. Many different approaches are under evaluation.

COMBINED OPS

Combined Ops is a term coined for the combination of these methods. The most obvious approach *(as an example)* is to simply have two different release schedules. One heavyweight release schedule runs in a yearly cycle. Several lightweight iterations occur between. The two types

of cycle proceed in parallel, simultaneously, on the very same large complex system. Lets examine this direct approach.

PROOF OF ROI

Each year a full test cycle proves the new functions analyzed to be critical in improving those processes relevant to organizational performance improvement. Test is followed by evaluation, and yearly evaluation proves the reality of improved processes and operations. The problem of pretty and cool software produced at significant expense that does nothing for the bottom line is eliminated.

HEAVYWEIGHT FOCUS

Heavyweight process is focused on the core functions, the external interfaces requiring months long security negotiations, the RDBMS data model expansion and other areas that ensure ROI and complete function. Lightweight processes improve core function, add ancillary function and improve usability over time.

THE INITIAL CYCLE

Jump starting software development before initial requirements analysis is the holy grail of Combined Ops. This would allow work to start earlier, key functions to be prototyped earlier. It would eliminate the long frustrating wait while analysis occurs. Discussions and experiments are occurring now in this area as well. Methods will be found.

CONCLUSION

Combined Ops is a coined phrase to describe means to execute full systems engineering and lightweight iterative processes in parallel to enhance DevOps type methodology. An example is provided of perhaps the most straightforward approach under examination, perhaps the most easily described. By combining these internal controls for assuring ROI and eliminating waste can be satisfied, and security requirements may be addressed more fully.

6.9 DEVOPS OR DEVOPS, NOVEMBER 1, 2014

"DevOps" is the next software management wave replacing the now declining "Agile". Most current discussion of DevOps is dominated by development considerations. Operations are reduced to a means to deploy developed code.. Little consideration is given to certain critical mechanisms within operations. I call this **DEV**ops. Operations is a secondary consideration, an afterthought.

To implement balanced, sustainable, responsible DevOps you must embrace the full range of requirements for IT operations. Primary components of IT operations involve planning what is needed, and then checking to assure that need was (or continues to be) met. This happens using a wide variety of corporate or operational controls. These controls have a vast literature, widespread adoption and are widely accepted as critical to operational success.

You would not have your organization pay invoices for equipment not delivered. Why would you allow your organization to pay for operational improvements not delivered?

The prior wave of development innovation was "Agile" development. While the definition of Agile varies, and its implementation varies, and nearly everything about Agile varies, we can talk about a certain large subset of Agile implementation. In this subset much of the gains shown from adoption of Agile come from bypassing corporate or operational controls. Since corporate controls are what assure money and efforts are being spent on actual improvements and not the whims of

programmers or end users, such approaches are somewhat irresponsible and do not meet operational needs. I call advocates of this irresponsible and unprofessional approach the "Agile Cowboys". (I do not include those actively involved in addressing Agile shortfalls and limitations in a responsible fashion, they are NOT the cowboys I speak of.)

Let's examine the corporate or operational controls we are addressing, and why they are critical to your operational success:

CONTROLS FOR OPERATIONAL NEED

Long before spending is authorized, long before a project is chartered, long before the programmers are engaged, an operational need is identified. No spending, no activity is warranted without an operational need. The purpose of the activity is to fill the operational need- not to produce more or better code. Once the needed tools have been placed in operational hands the improved operations must be evaluated, usually repeatedly, often yearly. The operations are tested to assure they meet the organizational needs. The technological tools are not important here, not directly tested, except as a small part of the full operation.

Note that so-called agile testing has nothing whatsoever to do with this kind of organizational evaluation- it is unrelated, too narrow and detailed to be relevant. This evaluation is measured against the mission, the strategic objectives, the operational needs. We are not speaking of IT operations here, but organizational operations that IT operations supports and serves. Some mission relevant line of business will be the object of evaluation. Only for software vendors are these the same thing.

CONTROLS FOR BUSINESS PROCESS IMPROVEMENT

The main reason new enterprise software is commissioned or authorized is to implement new and improved business process. Processes may be improved in terms of cost, quality or throughput/speed of response. This makes the business run better. Surely you would not buy software to cause the business to run less effectively?

Most often processes may be improved by automation or incorporation of some bit of technology properly applied to a business problem. To do this operational improvements are examined. specified, designed. Improvements are checked for presence in operational tests.

CONTROLS FOR SYSTEM PERFORMANCE:

Programmers sometimes forget that the performance of their software, as implemented, is dependent on many things that are not their delivered software. There is a wider scope, and if the network capacity is lacking or the server capacity or any of a dozen other things- the system has not performed to meet needs.

When you pay for a whole system, you test that the whole system was delivered: Configured correctly, with backup implemented, with the hot spare site included, producing the correct response times, etc. The software, even though it is of great emotional importance to the ego of the developer, is only a subsystem.

When you buy the whole system as a unit, you test that system at delivery to assure you have not been cheated. System specific performance indicators are constructed to characterize what is

needed, and then system tests are performed to test the whole system (not the software in isolation).

Such testing is also sometimes called integration testing, as the components have been brought together. Note that the term "Continuous Integration" refers only to rapid production of software releases, and ignores these issues of site, net or platform.

CONTROLS FOR SOFTWARE PERFORMANCE:

Here is where "Agile" methods can be applied, There is still a gap, however. In some implementations of "Agile" functional requirements are not specified. Instead the end-user is taken as authoritative, and software is built to satisfy that end-user. How can the end-user possibly know what the needs of their new role, not executed anywhere to date, are? When an organization and business process are transformed the way they did things is obsolete, and so is their thinking. This is doubly true when significant new technology is applied, especially "disruptive" technology. The end-user's viewpoint is obsolete.

Only by analysis of the new, intended business process or rules can the requirements to meet the needs of the new role, the new functions be achieved. Only by testing that the new functions were implemented can the software support the new processes, new enabling technologies, and new operational concept.

CONTROLS FOR IT SECURITY

IT security is full of controls. There are controls on software production methods like code reviews. There are controls on testing of administrative system configuration. There are penetration tests of live systems. Some Agile implementation discards much or all of this. If you have not noticed, this is a big operational problem. IT Security is a big operational problem. Ignoring security controls will not work for operations.

CONTROLS FOR PROJECT AND PROGRAM PERFORMANCE

A project or program is defined by a business case. Cost, return, risk and scope are specified. The manager and his implementation must be checked and measured to assure costs are controlled and scope is satisfied but not exceeded. Controls here include Earned Value Management.

Many Agile implementations have sloppy scope, and do not provide measures of the percentage of scope completed, nor perhaps clear indication that the scope is completed and work is over. Perhaps programmers want to keep pumping out code, but the reason to pay them may be over.

CONTROL FRAMEWORKS

Sets of these controls are often described in an SDLC (System Development LifeCycle). Between stages are stage-gates, designed to test if the money spent has been well spent and if more money spent on further work might result in more progress toward defined business need. Many agile implementations bypass all or some such checks.

Controls are also often synonymous with governance. Have a look at the momentum and effort in COBIT for example. Unlike Agile, governance is not a fad and not ending soon.

HIDDEN RISK

Agile Methods not only evaded controls, but important analysis. Without analysis key risks were not known, and were accepted without being identified. The problem is worse in large and complex systems, where the risks are also large and complex.

REIGNING IN THE AGILE COWBOYS

Bypassing all these controls must end. While greater efficiency in producing software may be nice, many of the approaches in the now obsolete "Agile" wave do not meet the needs of operations in customer organizations. Agile was developed for product vendors, a much simplified operational environment. Agile use in customer organizations, where the mission is not software development, must grow up and become responsible.

Massive failures in the US Federal Government, UK Government and elsewhere have occurred in the past few years dominated by Agile approach. Can this be attributed to this bypassing of controls? The GAO said it was a factor in the ObamaCare debacle. Similar reports exist for other cases.

A vast slow decadent decrease in software quality has occurred over the last decade, which has been dominated by the previous wave of Agile thinking. Can this be attributed to bypassing controls? "You-bettcha."

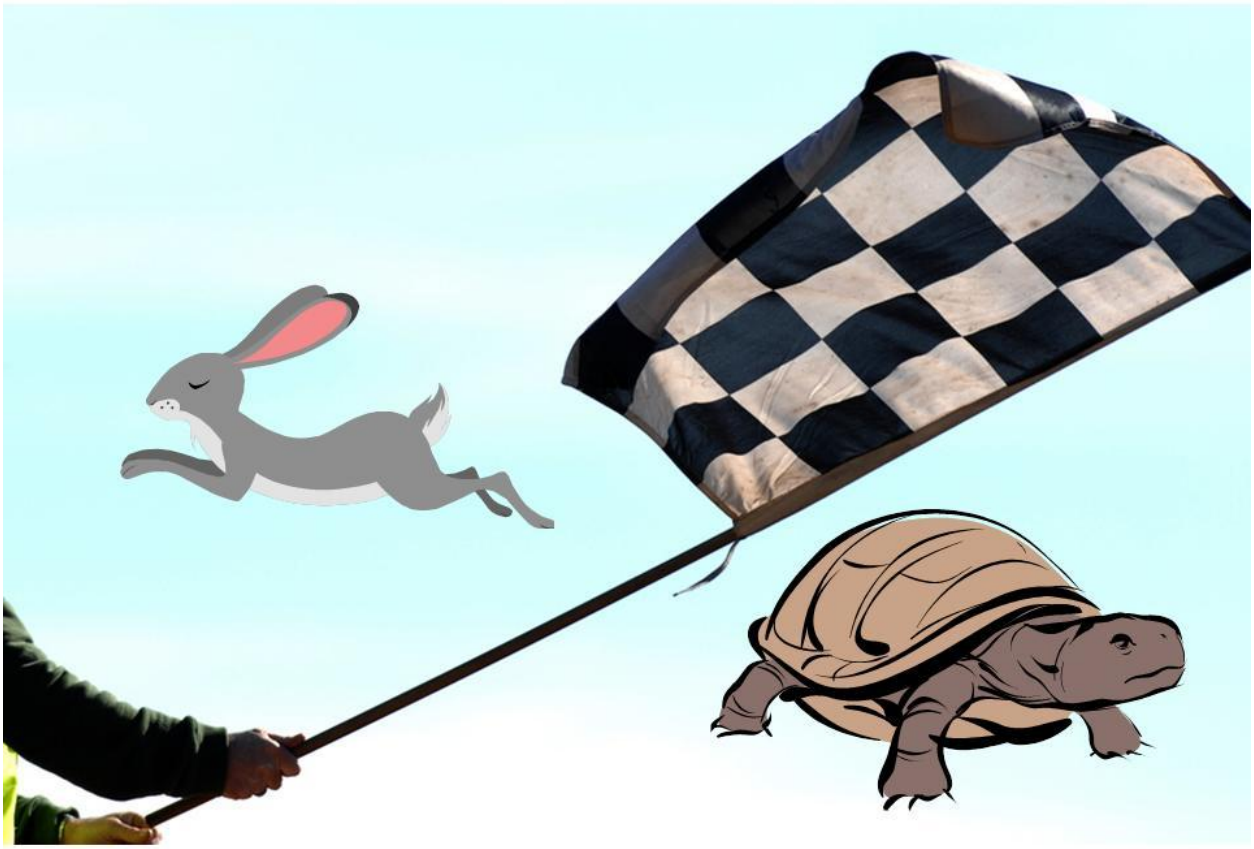
All over town here in DC the CIOs and the staff are working to strike a better balance between "Agile" and operationally required controls. No one wants to be that next failure. DevOps will need to span this gap. <update: DHS is about to release its formal SELC for Agile. Census has an SDLC for Agile already. IBM, Sapient and other companies have released SDLC versions incorporating Agile.>

DEVOPS NOT DEVOPS

To date discussion of **DEV**ops has focused on the needs of the subordinate function, the operationally less important element, the developers, and not on the more important needs of the organization addressed by operations. To produce a more balanced DevOps, this must change. Controls covering the full range of operational concerns must be integral to DevOps.

The future is here. Giddy-up you Agile Cowboys, have a blast. Your rodeo may end soon. Every management fad has its day, and its day of reckoning. Yeee-Haw!

6.10 BI-MODAL IT & COMBINED OPS, NOVEMBER 11, 2014



Gartner has released a news bulletin stating that every CIO shop needs bi-modal operations. Good job guys. http://www.gartner.com/newsroom/id/2903717

Related to this, I have written before about Combined-Ops, a coined name for merging DevOps with organizational controls for finance, waste, security etc. https://www.linkedin.com/pulse/article/20140809223124-86002769-beyond-devops-combined-ops?trk=mp-reader-card

I have written about the need for this imperative: https://www.linkedin.com/pulse/article/20141101092654-86002769-devops-or-devops?trk=mp-reader-card

You probably know you should act on this, and empirical evidence shows that you probably are working on it now. Many organizations are already moving toward this model, in some fashion,or variation, today. The question is one of what to do and how to do it. I am just putting a description around what I see people starting to do. So here let me further describe the three phases of a simplified Combined-Ops approach. (Customize this for local use.)

The basic idea is that you control enough to assure that you are producing what is needed to support organizational results, and then ease up and let local processes dominate. You will be moving from discontinuous transformation methods to continuous methods as the project progresses. Here I present an approach, chosen as simple to explain and more sensible than some.

https://www.linkedin.com/pulse/article/20140723111655-86002769-enterprise-transformation-methods?trk=mp-reader-card

PHASE 1: BEFORE IOC:

Before Initial Operational Capability is achieved, during conceptualization, planning and justification for a major expenditure, use the standard SDLC in the standard way (avoiding redundant and excessive documents, I will write about that later). Try to include explicit analysis of new to-be (future) operations, as no one has performed these yet. Assure full and proper test activities and controls are not bypassed.

PHASE 2: IOC TO FOC

Having achieved the Initial Operational Capability, and having assured that the key elements of ROI have been included, move to leaner, more agile methods. However continue to use initial planning (long completed) as a guide. To the extent it exists, use it. Reduce documentation. Document deviations. Test based on the gap and deviations.

PHASE 3: AFTER FOC

Having achieved Full Operational Capabillity move fully to lean and agile methods in maintenance and operations, and for subsequent releases.. Combine this with continuous improvement methods for operational processes, such as Lean, Six Sigma, etc. The large system problem has now been reduced to smaller problems suitable for such methods to perform well. Yearly evaluation of operations as improved should continue to measure effectiveness.

ALTERNATIVE APPROACH

Another approach I have heard discussed is to differentiate methods based on release size. A minor release would be very lean or agile. A major release would use more thorough planning, testing and evaluation.

APPLICABILITY OF COMBINED-OPS

The Combined-Ops model is intended for use in complex integration or large enterprise system development. It is not intended for small scope projects or product development. Automation of release and deployment may be used in all phases, but controls must occur when required. This method is designed to avoid another Healthcare.Gov incident.

Again, I am not suggesting every small, low cost, low impact, non enterprise app or application or project use this method. It has a range of applicability. Use responsible engineering judgement.

(Note: claims of universal applicability and panacea like magic bean descriptions are probably what has done much of the damage in adopting Agile and related methods to date. Modifications are required to apply this stuff in different circumstances.. Pay attention to the range of applicability of any technology or method.)

STARTING PROJECTS FAST

There is nothing stopping you from starting a pilot or prototype using Lean or Agile methods, now, today, in parallel with early SDLC work. 'Nothing.

CONCLUSION

Gains of lean or agile methods based in the bypassing of controls and governance are illusory. You have achieved little. Developing mountains of additional but irrelevant code is not progress. Combine SDLC with DevOps to achieve targeted software that directly supports real organizational transformation (aka Digital Transformation or Enterprise Architecture).

DHS, Censius, IBM and Sapient among many others have now released hybrid Agile/SDLC lifecycle descriptions. That is the gist. Go forth and do good. Call me if you get stuck. As always, I hope it helps.

6.11 BIG SYSTEM MANIFESTO, MARCH 28, 2015

An Alternative Manifesto for more Responsible Enterprise System Development...

(Including Lessons Learned from Agile Software Development)

We have repeatedly found that agile does not scale without special additional methods. Those methods do increase the size and complexity of Agile software development, but do not address a whole range of important associated issues. This is what works,

WE BELIEVE IN:

Measurable results over unsubstantiated claims.

1. Science and rigorous engineering over hype, marketing, manifestos and mysticism.
2. Quality improvement via process maturity (as in CMMI, ISO 9000 and Six Sigma) rather than ad-hoc (Agile) methods.
3. Accurate and sufficient documentation of all salient aspects of a product delivered to a customer.

4. Satisfying contractual obligations wherever possible unless and until such obligations are mutually removed from the contract by formal agreement (i.e. no fraud or default).
5. Formal change control and configuration management with review rather than rushing to change functionality any time one stakeholder finds something she dislikes.
6. Planning and achieving vetted and approved goals and objectives, favoring strategic or operational advantage over the potentially parochial tactical viewpoints of local users wherever conflict may occur. (Performance improvement and competitive advantage are not often found without analysis and forethought.)
7. Formal requirements management as the effective, proven means to increase software quality, decrease defects, and reduce software project failures.
8. Objective professionalism, with polite human interactions engendering respect rather than petty politics, manipulation or excessive pandering to personal emotional issues.
9. Compliance with applicable law and policy in software development such as FAR 37-104, FISMA, corporate controls such as "stage gates" and accessibility via Section 508.

PRINCIPLES:

1. The most effective means of technical communication may require formal engineering drawings, formal "schedules" (lists) or formal specifications.
2. The best architecture comes from educated and trained architects exercising the methods of their profession with forethought, analysis and planning.
3. Business people and software developers should work together using proven methods such as enterprise architecture and business process reengineering.
4. Mature processes with formal documentation and continuous improvement produce repeatable, high quality results.
5. Deliver working software when it has been tested and poses no security threat or risk of operational failure to the organization.
6. Build projects around professionals with integrity, sufficient education or training and devotion to the mission.
7. Organizational performance improvement as the measure of return on targeted investment in software development, not heaps of operationally irrelevant code.
8. Excellence comes from mature processes, reviews, transparency and testing.
9. Formal, documented continuous process improvement causes teams to become more effective.
10. Correct and comprehensive requirements have been proven to be more effective than other means in improving software quality and decreasing risk of software project failure.
11. Formal definition of subsystems and interfaces reduces complexity and risk.

(This document may be read as a point by point rebuttal of the Agile Manifesto, which is mostly wrong in its specifics. Most who use the word Agile do not mean that, but just want to use the name Agile for the market momentum. Few believe in the nonsense printed in the Agile Manifesto. If you manage to do all of the above and still call it Agile, there is probably no remaining issue. Heck, you can call it Suzy then.)

Governance & Maturity Management