Some Management Method Measures

20 minute talk
GilbFest 2013, London
Tom Gilb

Held 24 June 2013, First Slot 10-11

Some Questions

- Do people think completely and logically about management methods?
- Is there a reasonable set of facts to support a position for the methods?
- Is there a generic set of attributes to evaluate a method on?
- Or to what degree do we need to tailor a set of attributes to judge methods on?

Some ‘Methods’

- Software Development Methods
  - Evo
  - Scrum
  - CMM
- Management Evaluation Methods
  - Balanced Scorecard BSC
  - Quality Function Deployment QFD
  - Impact Estimation Tables IET
  - Planguage PL
- Software Architecture Methods
  - TOGAF
  - Planguage

Here are my 10 Principles of Management Methods:

1. Methods have multiple effects.

2. The best of *qualified* methods is one that delivers the best overall set of effects, in relation to the needed set of effects.

3. The needed set of effects is a subjective perception, of a set of stakeholders; but it is subject to negotiation, agreement, and proof of concept.

4. If the environment varies in any subtle way, over time, the needed set of effects will probably vary; and consequently the exact mix of management methods will need to change, to meet necessary objectives of the new environments, and new or different stakeholders.

5. There is no one 'right' management method: there never will be; due to many rapid changes in environment, knowledge, stakeholder perception, and power.

6. The best optimization of management methods is, at best, only *best* for a fleeting moment, and then we need to tune them for inevitable change.

7. A failure of methods' experts to know the numeric effects of their methods, for any defined environment, and any aspect of cost or effectiveness of interest, means their advice can fail us totally.

8. It will be impossible to collect all desired knowledge about management methods for all environments, and effects, in advance of deciding to use the methods.

9. The best we can expect, is to use available knowledge of methods', to pick *likely* methods candidates, as a reasonable starting point.

10. We must expect, at all times to measure early and frequently the actual effects of management methods, and to consequently tune them to meet our current management objectives in practice.

Some assumptions

- All methods have many dimensions of interesting characteristics
- Some dimensions are more interesting or critical than others for given stakeholders points of view
- All dimensions can be expressed quantitatively
- Order of magnitude judgement can be made on any critical dimension of evaluation
- An Impact Estimation table will be a pretty good way to model the methods and to make rough judgements on them

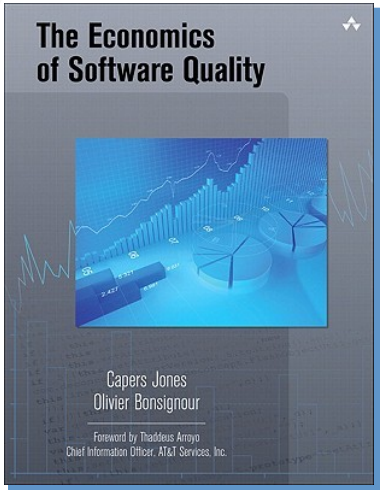
Some Methods Attributes that probably are of major interest

Benefits

- Return on Investment ROI
- Adaptability
  - Problem Type Adaptability
    - Quality Assurance
    - Cost/Resource Management
    - Systems Type
    - Localization
    - Human Language
  - Scalability
- Completeness of Modelling

Costs

- Acquisition/License Costs
- Individual Learning/Training Costs
- Production Costs
- Maintenance Costs
- Auditing Costs
- Certification Costs



Capers Jones Data 2013

Capers Jones Summary

	Best				Median				Worst
Method	>85% Reuse	Hybrid	TSP	RUP	Agile	CMMI 3	Lean 6 Sigma	Offshore	CMMI 1
Language	c/c++	c/c++	c/c++	c/c++	c/c++	c/c++	c/c++	c/c++	c/c++
Size (Function Points)	1000	1000	1000	1000	1000	1000	1000	1000	1000
Defect Potential	1.20	2.25	2.70	3.90	4.80	4.50	4.61	5.40	6.00
Total Defects	1200	2250	2700	3900	4800	4500	4613	5400	6000
% pre-test removal	90.00%	84.38%	81.00%	78.30%	69.75%	67.50%	66.49%	62.33%	45.00%
% test removal	98.00%	93.10%	87.50%	84.00%	81.90%	80.50%	80.10%	78.75%	70.00%
Total Defects Delivered	2	25	67	145	276	302	326	467	1079
Customer Satisfaction at delivery	10.00	9.96	9.89	9.74	9.45	9.40	9.35	8.90	7.30
Development schedule	6.68	11.22	12.02	13.11	11.82	13.34	13.57	13.52	15.85
Staff Size	4	6	7	8	7	8	8	12	10
Dev Effort Months	27	68	86	101	84	107	109	159	158
TCO Effort Dev %	69.74%	67.75%	62.73%	55.57%	43.12%	45.77%	44.77%	44.97%	30.14%
TCO Effort Maint %	4.46%	7.18%	14.05%	23.88%	40.92%	38.67%	38.67%	38.40%	58.71%
TCO Effort Enhance %	25.80%	25.07%	23.21%	20.56%	15.96%	18.25%	18.25%	16.64%	11.15%

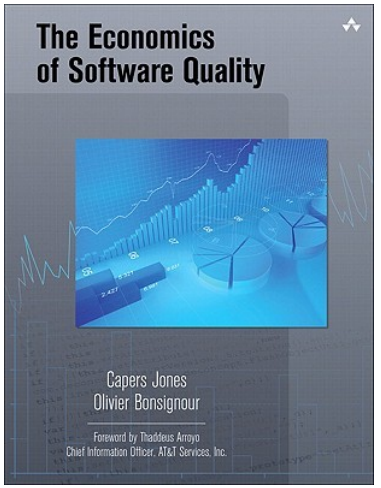
What are Capers' Tables Discussing?

Capers IS trying to quantify and classify

- Defect Injection
- Defect Detection
- Defect Removal Effectiveness
- Entire SW Development Process
- Major methods and interesting combinations
- Size of system (scaling)

Capers is NOT trying to quantify or classify

- Value delivered to stakeholders
- Speed of Value delivered
- Cost of mastering the methods themselves



And Capers deals with some more marginal attributes

-
  - (Req + Des + Code + Docs + Bad Fixes)
-
  Effectiveness
  - (Inspections, static analysis, testing) , C Jones
- Cumulative Defect Removal
  - C Jones
- Total Cost of Ownership (TCO, C Jones)

More Capers methods attributes

- A few more attributes reported by Capers Jones, 2013
  - **Maximum Cyclomatic Complexity**
  - **Security Vulnerabilities Delivered** (>10 indicates security problems)
  - **MTTF at Delivery (Hours)** (24 hours is maximum value)
  - **Customer Satisfaction (at delivery)** (10 is maximum value)
  - **Probability of Contract Litigation** (For poor quality or major overruns)
  - **Stabilization Period (months)** (Months to approach zero defects)
  - **Error-Prone Modules** (>5 indicates quality malpractice)

David Rico Comparisons 2013

Lean & Agile
Project Management

Leading Large Programs & Projects

Dr. David F. Rico, PMP, ACP, CSM

Twitter: @dr_david_f_rico
Website: http://www.davidfrico.com
LinkedIn: http://www.linkedin.com/in/davidfrico
Facebook: http://www.facebook.com/profile.php?id=1540017424
Dave's Agile Articles: http://davidfrico.com/agile-message.doc

Agile Recap

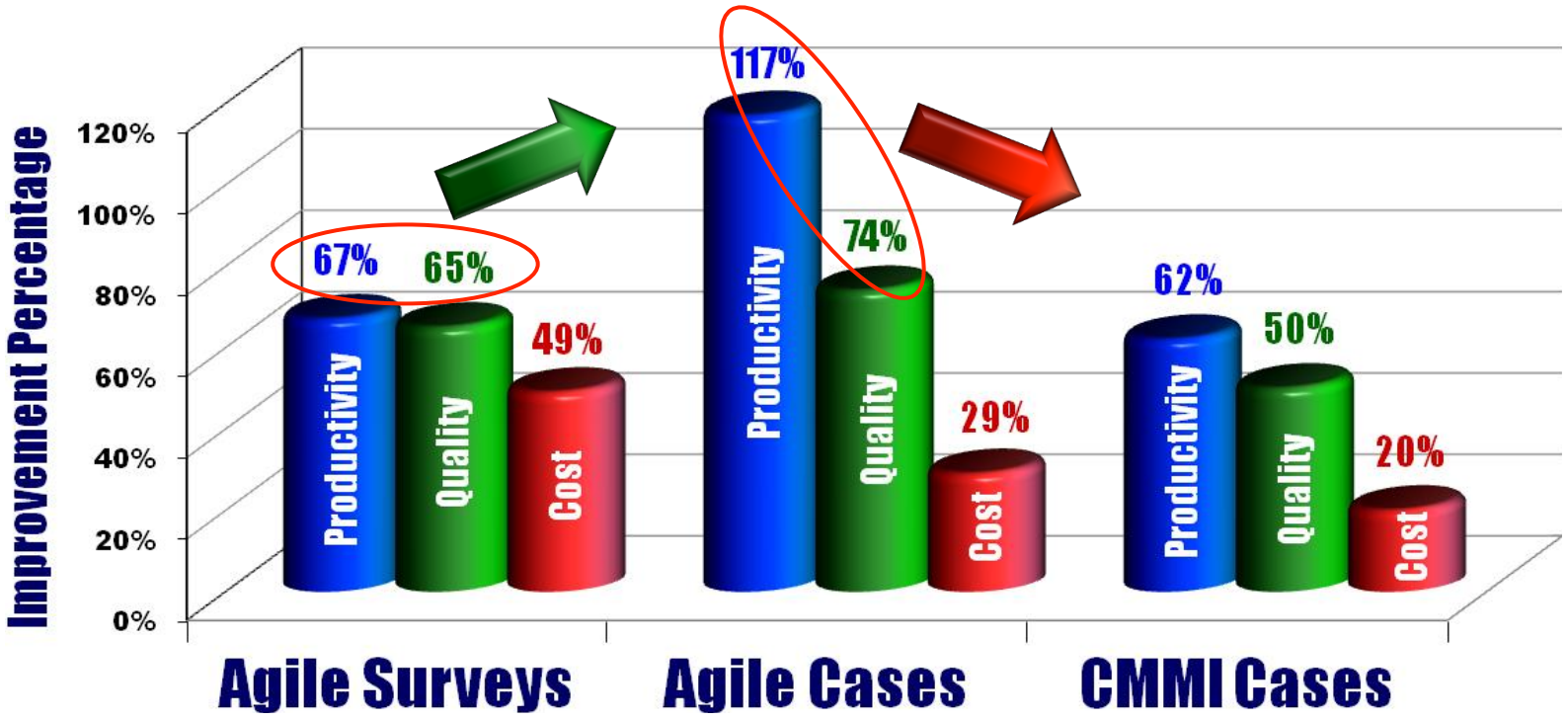
- Agile methods DON'T mean deliver it now & fix it later
- Lightweight, yet disciplined approach to development
- Reduced cost, risk, & waste while improving quality

What	How	Result
Flexibility	Use lightweight, yet disciplined processes and artifacts	Low work-in-process
Customer	Involve customers early and often throughout development	Early feedback
Prioritize	Identify highest-priority, value-adding business needs	Focus resources
Descope	Descope complex programs by an order of magnitude	Simplify problem
Decompose	Divide the remaining scope into smaller batches	Manageable pieces
Iterate	Implement pieces one at a time over long periods of time	Diffuse risk
Leanness	Architect and design the system one iteration at a time	JIT waste-free design
Swarm	Implement each component in small cross-functional teams	Knowledge transfer
Collaborate	Use frequent informal communications as often as possible	Efficient data transfer
Test Early	Incrementally test each component as it is developed	Early verification
Test Often	Perform system-level regression testing every few minutes	Early validation
Adapt	Frequently identify optimal process and product solutions	Improve performance

Rico, D. F. (2012). *What's really happening in agile methods: Its principles revisited*? Retrieved June 6, 2012, from http://davidfrico.com/agile-principles.pdf
Rico, D. F. (2012). *The promises and pitfalls of agile methods*. Retrieved February 6, 2013 from, http://davidfrico.com/agile-pros-cons.pdf
Rico, D. F. (2012). *How do lean & agile intersect*? Retrieved February 6, 2013, from http://davidfrico.com/agile-concept-model-3.pdf

Studies of Agile Methods

- Dozens of surveys of agile methods since 2003
- 100s of Agile and CMMI case studies documented
- Agile productivity, quality, and cost better than CMMI

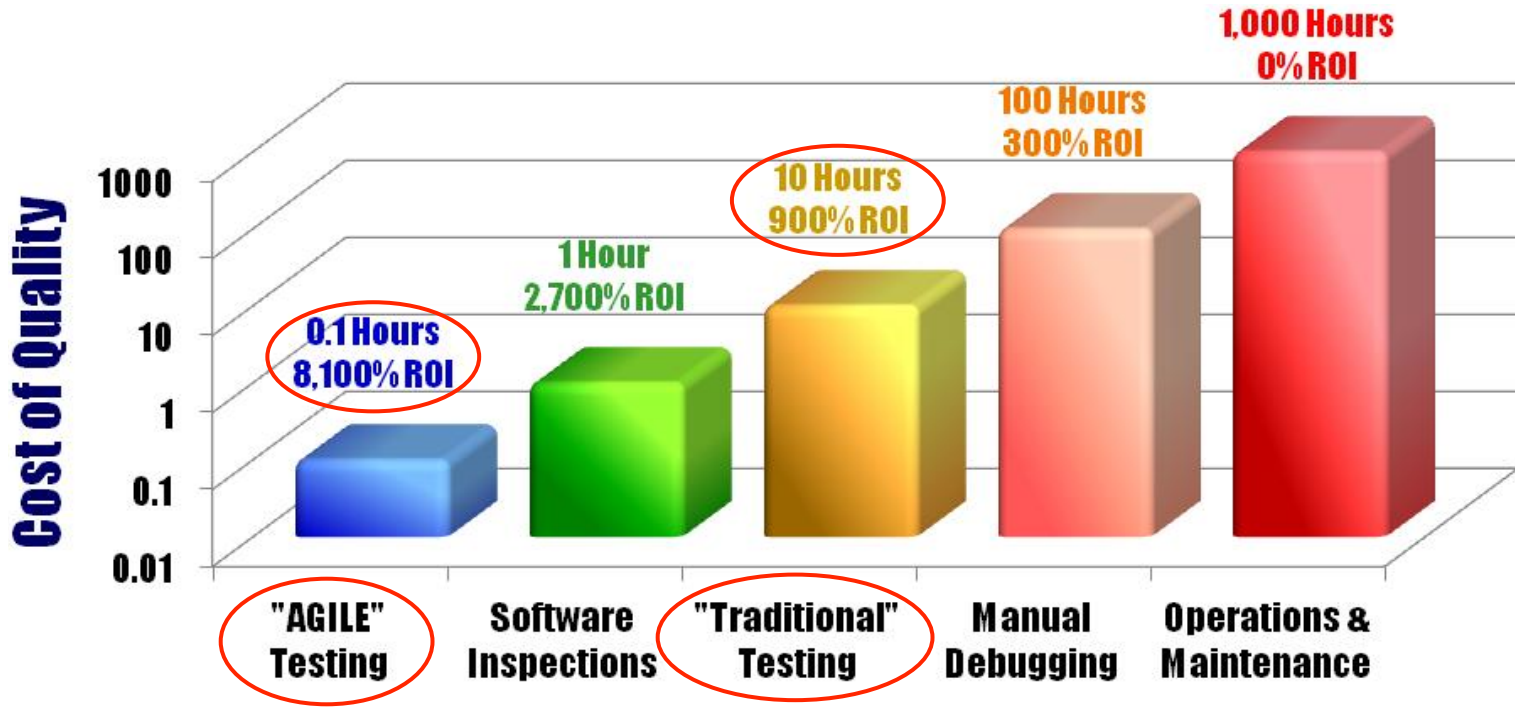


Rico, D. F. (2008). *What is the return-on-investment of agile methods*? Retrieved February 3, 2009, from http://davidfrico.com/rico08a.pdf

Rico, D. F. (2008). What is the ROI of agile vs. traditional methods? *TickIT International, 10*(4), 9-18.

Agile Cost of Quality (CoQ)

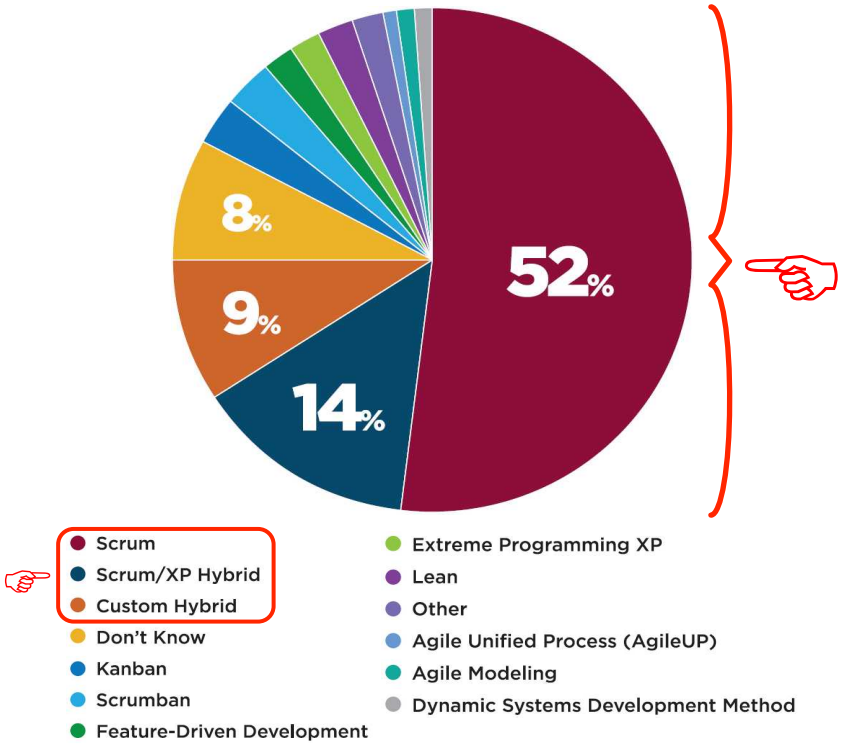
- Agile testing is 10x better than code inspections
- Agile testing is 100x better than traditional testing
- Agile testing is done earlier "and" 1,000x more often

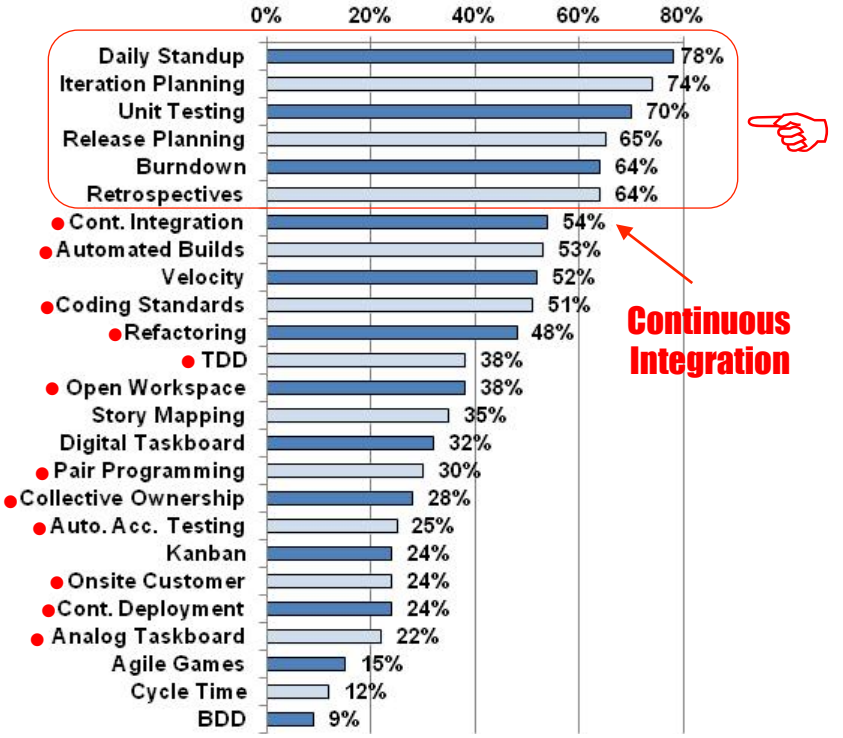


Rico, D. F. (2012). *The Cost of Quality (CoQ) for Agile vs. Traditional Project Management*. Fairfax, VA: Gantthead.Com.

Agile Adoption

- VersionOne found 80% using agile methods today
- Most are using Scrum with several key XP practices
- Lean-Kanban is a rising practice with a 24% adoption

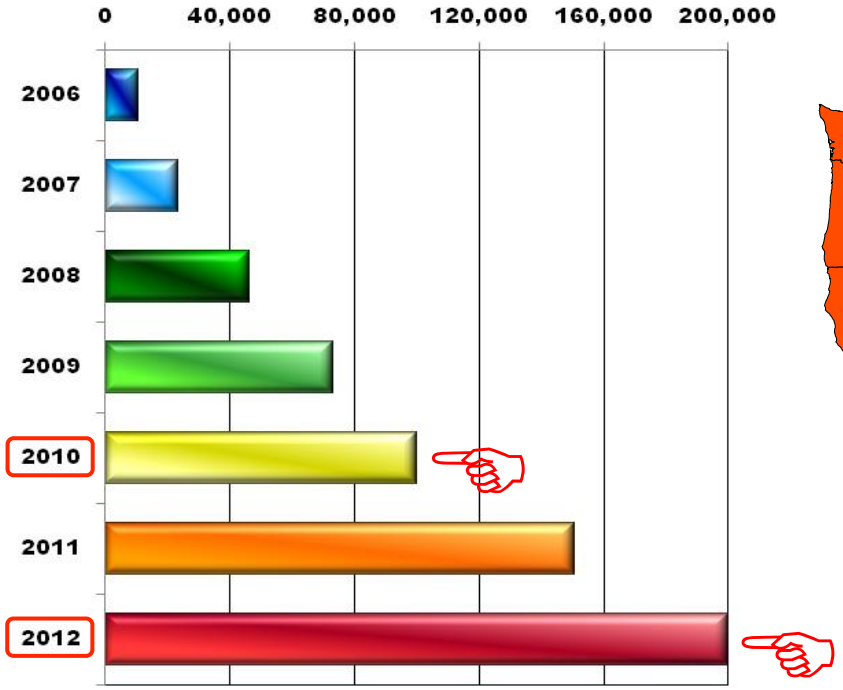


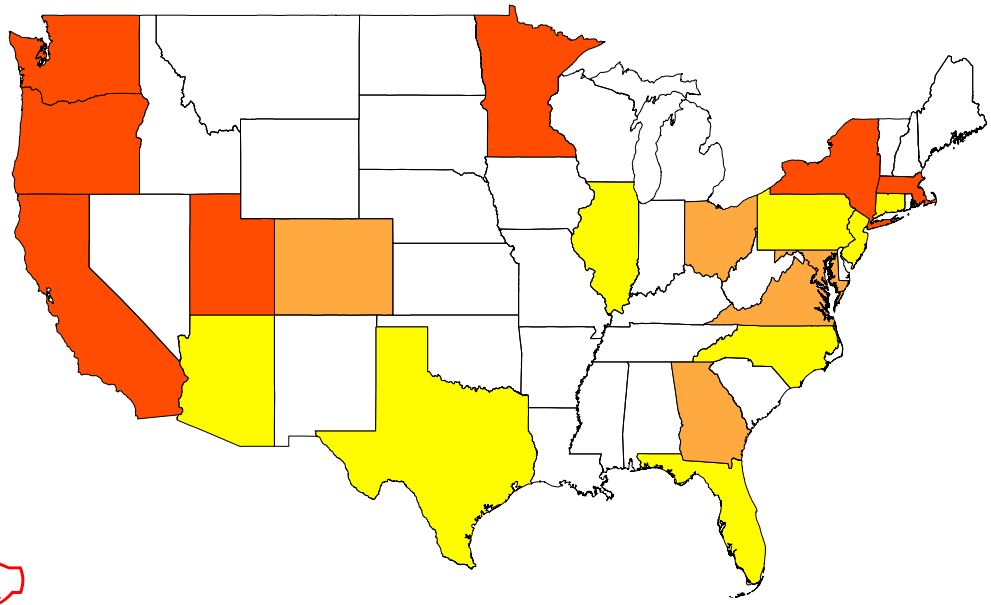


House, D. (2012). *Sixth annual state of agile survey: State of agile development*. Atlanta, GA: VersionOne.

Agile Proliferation

- Number of CSMs have doubled to 200,000 in 2 years
- 558,918 agile jobs for only 121,876 qualified people
- 4.59 jobs available for every agile candidate (5:1)





Scrum Alliance. (2012). *Scrum certification statistics*. Retrieved February 6, 2013, from http://www.scrumalliance.org/resource_download/2505

Taft, D. K. (2012). *Agile developers needed: Demand outpaces supply*. Foster City, CA: eWeek.

Agile Industry Case Studies

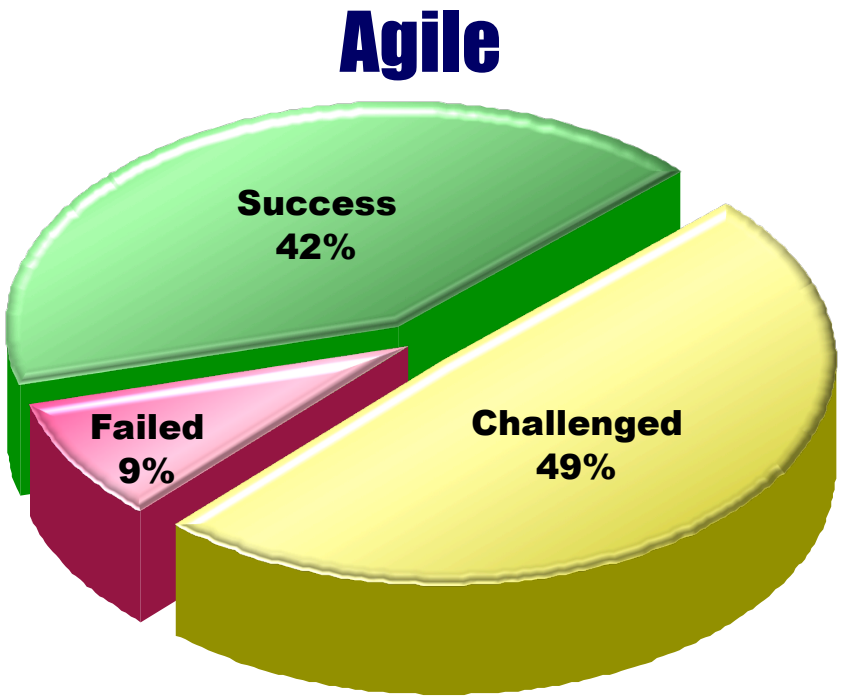
- 80% of worldwide IT projects use agile methods
- Includes regulated industries, i.e., DoD, FDA, etc.
- Agile now used for safety critical systems, FBI, etc.

Industry	Org	Project	Purpose	Size	Metrics
Electronic Commerce	Google	Adwords	Advertising	• 20 teams • 140 people • 5 countries	• 1,838 User Stories • 6,250 Function Points • 500,000 Lines of Code
Shrink Wrapped	**Primavera**	Primavera	Project Management	• 15 teams • 90 people • Collocated	• 26,809 User Stories • 91,146 Function Points • 7,291,666 Lines of Code
Health Care	**FDA**	m2000	Blood Analysis	• 4 teams • 20 people • Collocated	• 1,659 User Stories • 5,640 Function Points • 451,235 Lines of Code
Law Enforcement	**FBI**	Sentinel	Case File Workflow	• 10 teams • 50 people • Collocated	• 3,947 User Stories • 13,419 Function Points • 1,073,529 Lines of Code
U.S. DoD	Stratcom	SKIweb	Knowledge Management	• 3 teams • 12 people • Collocated	• 390 User Stories • 1,324 Function Points • 105,958 Lines of Code

Rico, D. F. (2010). Lean and agile project management: For large programs and projects. *Proceedings of the First International Conference on Lean Enterprise Software and Systems, Helsinki, Finland*, 37-43.

Agile vs. Traditional Success

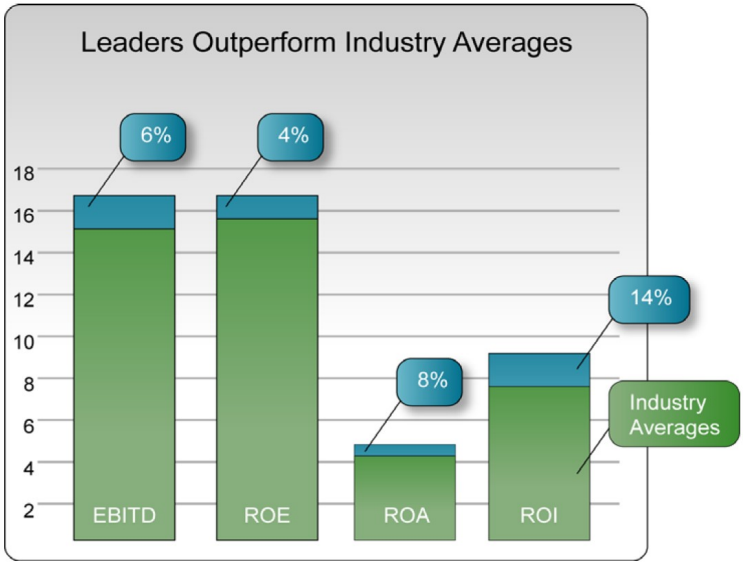
- Traditional projects succeed at 50% industry avg.
- Traditional projects are challenged 20% more often
- Agile projects succeed 3x more and fail 3x less often



Standish Group. (2012). *Chaos manifesto*. Boston, MA: Author.

Benefits of Organizational Agility

- Study of 15 agile vs. non-agile Fortune 500 firms
- Based on models to measure organizational agility
- Agile firms out perform non agile firms by up to 36%



Hoque, F., et al. (2007). *Business technology convergence*. The role of business technology convergence in innovation and adaptability and its effect on financial performance. Stamford, CT: BTM Institute.

Example of Top Management Recommendation Regarding all management strategies

Glasgow, June 10 2013

And then there is the problem of 'Activities', strategies, and all other 'means' for reaching our top level critical business objectives

- Policy Suggestion Basis
  - All 'means' are only as good as their **potential** contribution to our goals,
  - And ultimately only as good as **their real measured** contribution to our goals
- Policy
  - All proposed means to reach our numeric Goals
    - Will be **estimated** for expected effect (IET Method, in CE)
    - Will be prioritised for implementation based on value for money (theoretical, and then real)
    - All 'means' will be decomposed into small practical implementations in SLC
      - So we can see early if they work in practice

1. Objective: the Activities "Tagged"

CCSR: Complete the Core Systems Replacement (CSR) procurement process through to contract award

NDSHE: Complete delivery of the new digital services for HE part-time and Further Education (FE) 24+ Advanced Learning Loan

BIF: Establish the business improvement framework within the Transformation Programme

ODW: Establish the organisational design workstream to design a fit- for-purpose future business model

MAASS: Prepare the organisation for migration of agreed activities to shared services

Estimate the effect of our activities on our Goal by2015?

	CCSR	NDSHE	BIF	ODW	MAASS	∑			
Growth. Platfor m									
0 <-> 40	0% ?	50%	20%	-10%	35%±5	95%			
Cost	100	50	10	35	65	260K£			
Value/£	0/100	50/50	20/10	-10/35	35/65				
Priority	0	2	1	0	3				

Impact Estimation

IE In more detail

- Is based on documented evidence
- Documented source
- ± uncertainty (spread of experience)
- Defined Credibility (0.0 to 1.0)

IE Results

- Forces us to think
- Forces us to plan better
- Allows us to present ideas and strategies better
- Imposes result responsibility better
- Gives an initial Result Budget
- Helps us see priorities better
- Helps us see risks better
- Gives basis for project progress accounting in relation to the 'budgeted' results and costs

Suggested Policy for 'Activities'

(the Heart of Agile is these 'small steps of results)

Policy outline

- *Decompose into activities that give impact on our Goals*
  - *Not just do big projects that we HOPE will give unspecified impacts*
- *Specify the 'Activities' so well (clear, detailed) that they are 'guaranteed' to have the estimated effects*

Consequences

- We need to teach and coach decomposition into value delivery packets
- We need to teach and coach specification to the required level of clarity and effectiveness
- We need to teach impact estimation, based on credible evidence
- Management needs to demand that these standards be met
  - Using Spec QC with numeric Exit would help management make sure that this was done in practice (see CE book, Simple SQC)
  - Example McDonnell Douglas, and Primark, and Intel experience cases

Architecture Frameworks

A Paper on comparing Architecture Frameworks

Does it matter which Architecture Framework we use?

WP0080 | June 2013

Roger Evernden

Roger Evernden has been an Enterprise Architect since 1984, specializing in the highly practical use of EA to manage enterprise transformation.

Architecture Frameworks lie at the heart of EA work, but there remains a great deal of confusion about the subject. While different architects prefer one framework over another they don't often make their selection criteria explicit.

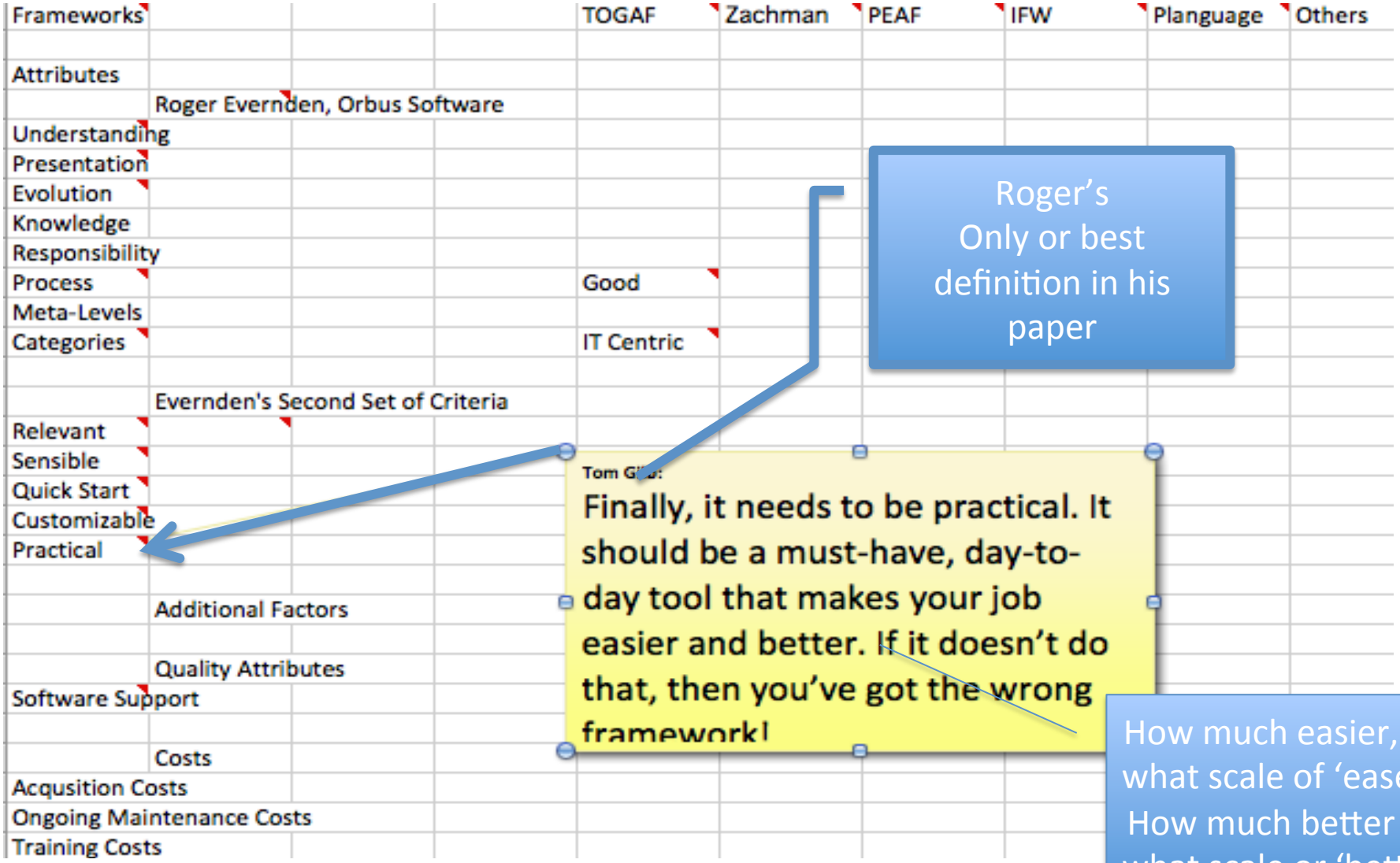
Why and how do we choose an architecture framework? And does it matter which one we choose and use? In this paper I will explain why frameworks are important, show that the framework we choose does indeed have an impact on our success, provide some criteria to make a selection, and suggest that we need to use more than one at a time!

Zachman or TOGAF®, or ...

Should Enterprise Architects use the Zachman Framework, or The Open Group Architecture Framework (TOGAF), or the Pragmatic Enterprise Architecture Framework (PEAF), or Information FrameWork (IFW), or .... Well this list could go on and on! If you search on the Internet you

http://www.orbussoftware.com/downloads/white-papers/does-it-matter-which-architecture-framework-we-use/?_$ja=tsid:50922|cgn:Guest+EA+White+Paper:+Does+it+matter+which+Architecture+Framework+we+use?

A framework for judging AE Frameworks



Trying to define his suggested Architectural Framework Attributes

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Frameworks				TOGAF	Zachman	PEAF	IFW	Planguage	Others			
2													
3	Attributes	Scale											
4		Roger Evernden, Orbus Software											
5	Understanding	% capability of mapping all interesting attributes of any system in realistic detail											
6	Presentation	% capability of presenting measures and **metrics** for all interesting variable attributes											
7	Evolution	% Capability of representing **time** dimensons on all system attributes											
8	Knowledge	% capability of representing useful knowledge about any component of a system											
9	Responsibility	% capability of specifing responsibility for any system artifact											
10	Process	% of potentially interesting relevant architecture pro			Good								
11	Meta-Levels	% capability of defining Meta Concepts, global concepts											
12	Categories	% Capability of dealing with architectual categories			IT Centric								
13													
14		Evernden's Second Set of Criteria											
15	Relevant	Number of stated purposes of the framework											
16	Sensible	Clarity of communicating ideas for governance and stakeholder communi											
17	Quick Start	Relative Speed of learning to do what you need to do, TOGAF = 5 on 0 to											
18	Customizable	Relative Ease of customization and change, TOGAF = 5 on 0 to 10											
19	Practical	Relative ability to make Architect job easier and better. Togaf = 5			Bad								
20													
21		Additional Factors											
22													
23		Quality Attributes											

Tom's First cut at a scale

Roger's best definition of his concepts

"Finally, it needs to be practical. It should be a must-have, day-to-day tool that makes your job easier and better. If it doesn't do that, then you've got the wrong framework!" R E

Trying to define his suggested Architectural Framework Attributes

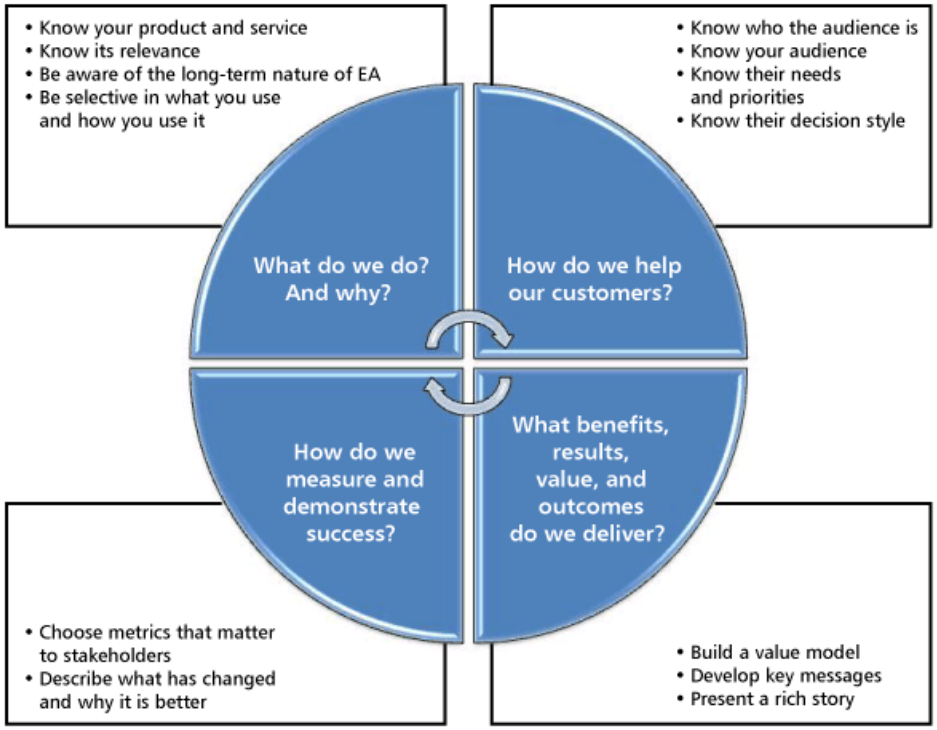
	A	B	C	D	E	F	G
1	Frameworks				TOGAF	Zachman	PEAF
2							
3	Attributes	Scale					
4		Roger Evernden, Orbus Software					
5	Understanding	% capability of mapping all interesting attributes of any system in realistic detail					
6	Presentation	% capability of presenting measures and **metrics** for all interesting variable attributes					
7	Evolution	% Capability of representing **time** dimensons on all system attributes					
8	Knowledge	% capability of representing useful knowledge about any component of a system					
9	Responsibility	% capability of specifing responsibility for any system artifact					
10	Process	% of potentially interesting relevant architecture pro			Good		
11	Meta-Levels	% capability of defining Meta Concepts, global concepts					
12	Categories	% Capability of dealing with architectual categories			IT Centric		
13							
14		Evernden's Second Set of Criteria					
15	Relevant	Number of stated purposes of the framework					
16	Sensible	Clarity of communicating ideas for governance and stakeholder communi					
17	Quick Start	Relative Speed of learning to do what you need to do, TOGAF = 5 on 0 to					
18	Customizable	Relative Ease of customization and change, TOGAF = 5 on 0 to 10					
19	Practical	Relative ability to make Architect job easier and better. Togaf = 5			Bad		
20							
21		Additional Factors					
22							
23		Quality Attributes					

"Finally, day-to-d doesn't

- My reflections
  - This is 'normal' level of language of too many IT Architects
  - No clear definitions of critical attributes
    - Certainly no measures at all
    - He has some checklists, not detailed in this paper, to judge these
      - http://www.evernden.net/enterprise-architecture-foundation-course/
  - No logical (quantified) evaluation of the different architecture frameworks
  - No clear reason why any one framework might be preferred
  - My first draft scales of measure are not very useful, as a consequence
  - I personally would have difficulty explaining to students or clients exactly what these terms mean, unambiguously, let alone measurably

- “Above all, a vital and unique characteristic of EA is the ability to synthesize multiple diverse views.
- **We need to be clear about how we help our customers.** All marketecture must be directed to its needs and priorities. Architects must explicitly show how EA supports decision making and promotes effective investment.
- **Marketecture must focus on the delivered benefits, results, value, and outcomes.** To do this, we need to build a value model that recognizes the value propositions that matter to our stakeholders.
- **We need to measure and demonstrate our success.** We need to succinctly describe the changes we have produced, using metrics that prove things are better.”
- http://blog.cutter.com/2013/04/09/marketects-delivering-good-enterprise-architecture/
- By Roger Evernden

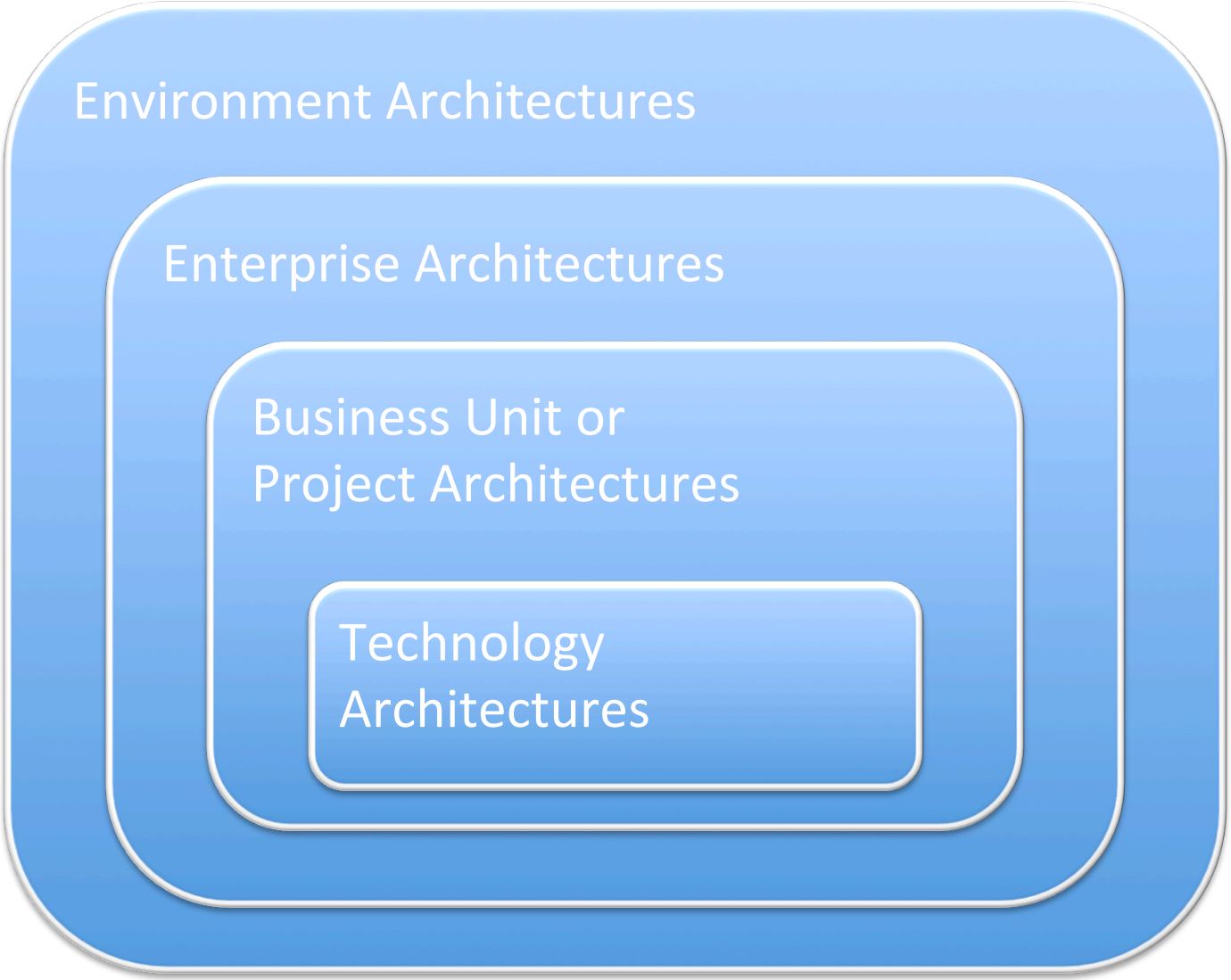
Yet !



A Skype Call with Roger Revealed An interest and ability in Metrics

- June 13 2013

An Integrative Perspective of the Relationship between Architectures - Constraints and Context



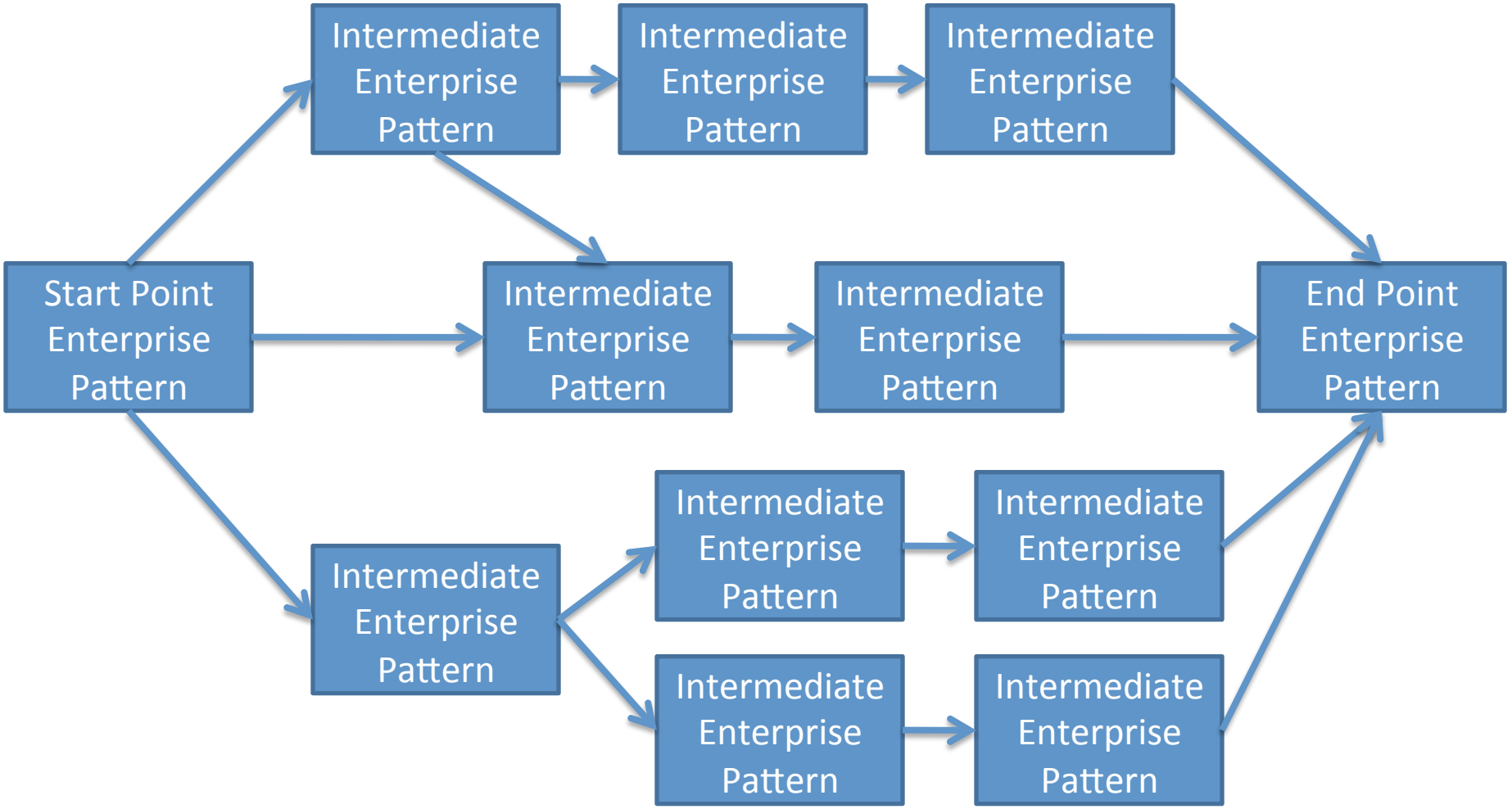
Slide courtesy of roger@evernden.net, June 13 2013, in personal communication to TsG, in Enterprise Patterns Report

Business Units, Business Divisions or Projects are impacted by the possibilities offered and the constraints imposed by Technical Architectures.

In turn, each Business Unit, Division or Project forms part of the Enterprise Architecture, and operates in the context of that Enterprise Architecture.

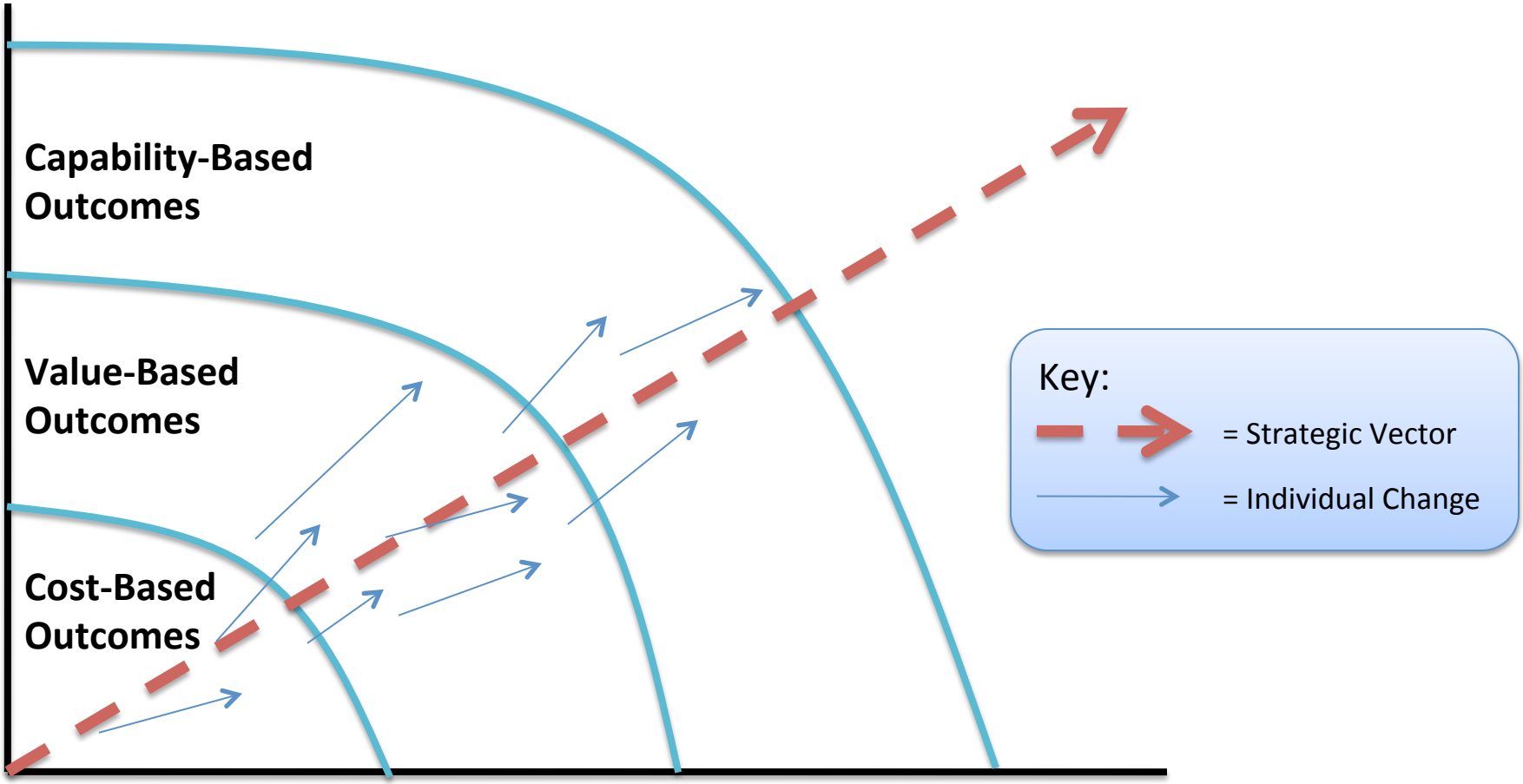
In a similar way, the Enterprise is impacted by the Business Unit or Project architectures, and operates in the context of the Environmental Architectures.

4. High-Level Overview of Potential Roadmaps and Alternative Routes



Slide courtesy of roger@evernden.net, June 13 2013, in personal communication to TsG, in Enterprise Patterns Report

5. Each Strategic Vector is an Aggregation of Individual Change Efforts that Accumulate to deliver a new Enterprise Pattern



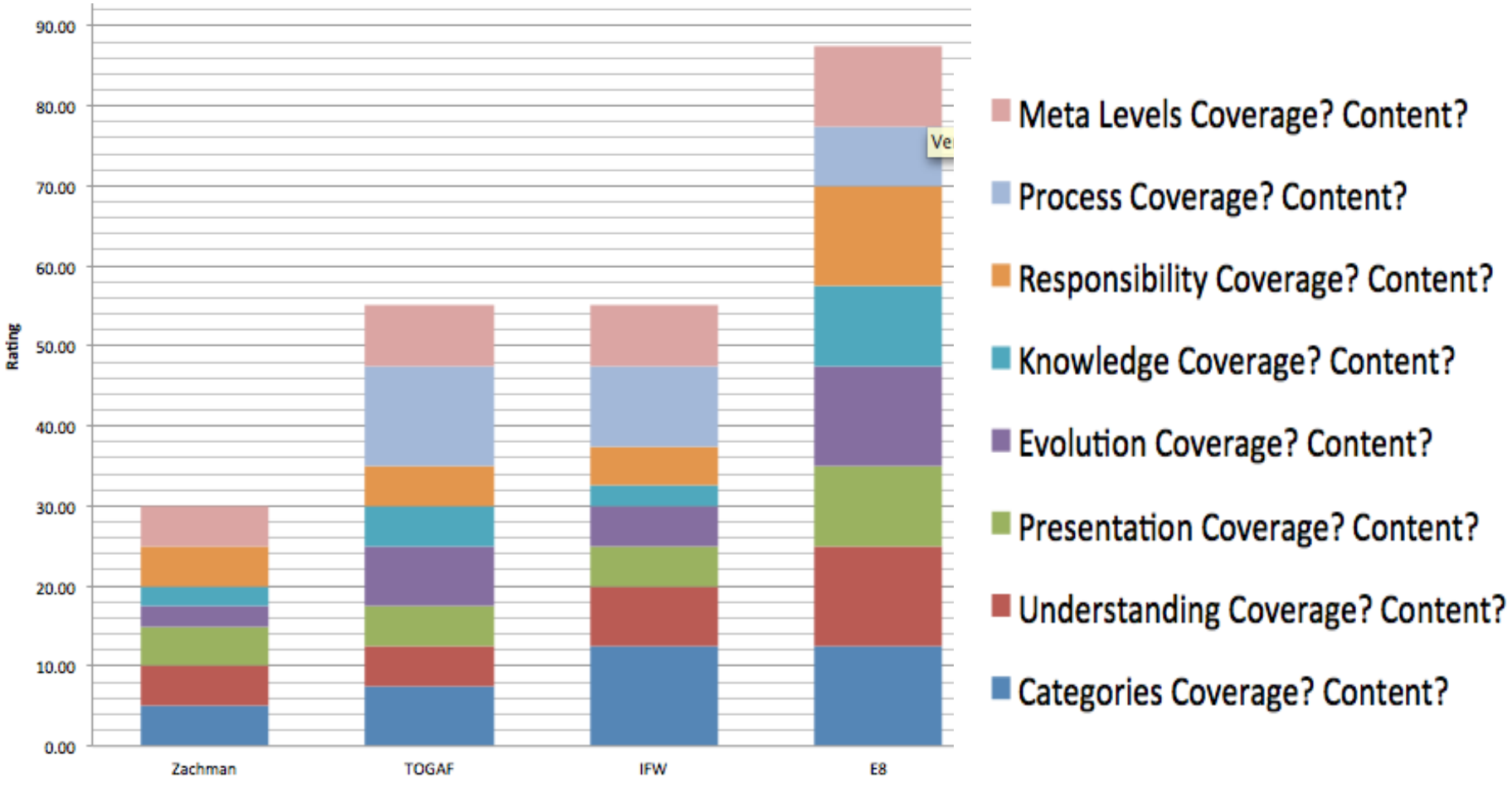
Slide courtesy of roger@evernden.net, June 13 2013, in personal communication to TsG, in Enterprise Patterns Report

Roger Evernden's Rating of EA Frameworks

A	B	C	D	E	F	G	H	I
Rating (0 to 5)								
Area	Categories	Understanding	Presentation	Evolution	Knowledge	Responsibility	Process	Meta Leve
Descriptio	Coverage?	Coverage?	Coverage?	Coverage?	Coverage?	Coverage?	Coverage?	Coverage?
Zachman	2	2	2	1	1	2	0	2
TOGAF	3	2	2	3	2	2	5	3
IFW	5	3	2	2	1	2	4	3
E8	5	5	4	5	4	5	3	4
-								
-								
Raw Rating Scaled to 100								
Area	Categories	Understanding	Presentation	Evolution	Knowledge	Responsibility	Process	Meta Leve
Descriptio	Coverage? Content?	Coverage? Content?	Coverage? Content?	Coverage? Content?	Coverage? Content?	Coverage? Content?	Coverage? Content?	Coverage? Cor
Zachman	5.00	5.00	5.00	2.50	2.50	5.00	0.00	5.00
TOGAF	7.50	5.00	5.00	7.50	5.00	5.00	12.50	7.50
IFW	12.50	7.50	5.00	5.00	2.50	5.00	10.00	7.50
E8	12.50	12.50	10.00	12.50	10.00	12.50	7.50	10.00
-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Slide courtesy of roger@evernden.net, June 13 2013, in personal communication to TsG, "R. E. Framework Comparison V 2" E8 is the Evernden 8 Attributes here, and listed in earlier slides here.

Evernden's Ratings Graph



Slide courtesy of roger@evernden.net, June 13 2013, in personal communication to TsG, "R. E. Framework Comparison V 2"

How Objective are these numeric ratings?

- My (TsG) Thoughts on Seeing these ratings

1. The definitions of the attributes are still not pinned down, and published
2. The facts or evidence upon which the ratings are made are not documented or published
3. This is a start in the right direction
   a. But we need to define attributes better
   b. And provide an auditable set of evidence to justify the ratings given
   c. I hope and assume Roger is the first to agree to these observations

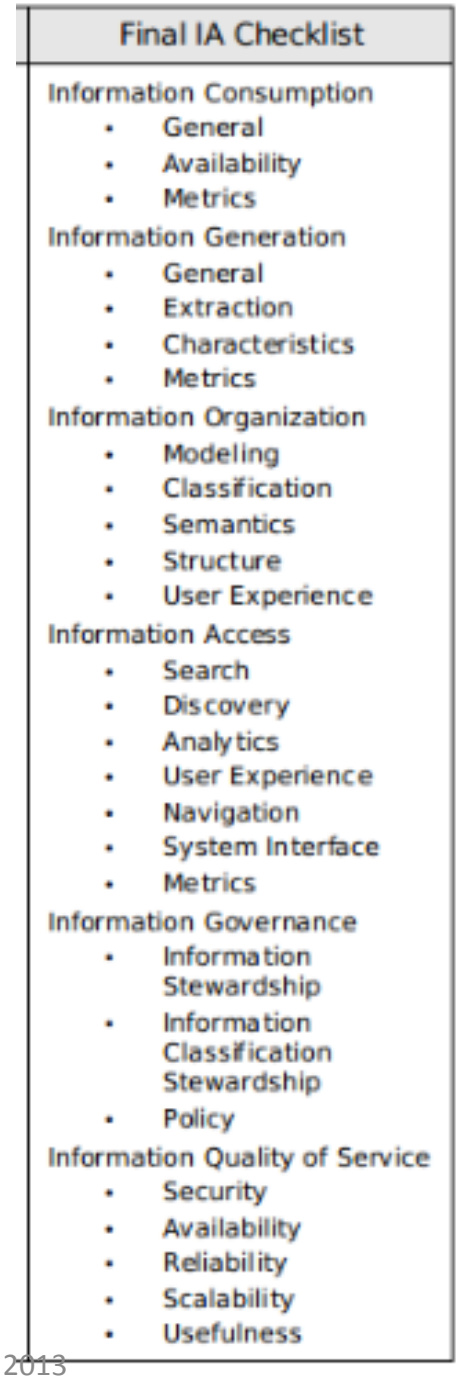
- My earlier (slide) reflections
  - This is 'normal' level of language of most IT Architects
  - No clear definitions of critical attributes
    - Certainly no measures at all
    - He has some checklists, not detailed in this paper, to judge these
      - http://www.evernden.net/enterprise-architecture-foundation-course/
  - No logical (quantified) evaluation of the different architecture frameworks
  - No clear reason why any one framework might be preferred
  - My first draft scales of measure are not very useful, as a consequence
  - I personally would have difficulty explaining to students or clients exactly what these terms mean, unambiguously, let alone measurably

Downey's Checklist

"Building an Information Architecture Checklist."

- http://journalofia.org/volume2/issue2/03-downey/

- Main point is the
  - Acknowledgement of many interesting Architecture Framework attributes
  - Even if they are not quantified
    - In spite of the word 'metrics' appearing



Overview of Downey's Checklist

"Building an Information Architecture Checklist."

- http://journalofia.org/volume2/issue2/03-downey/

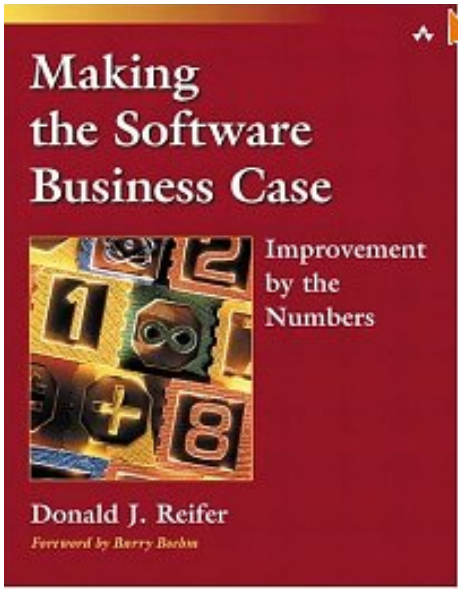
- Refers to Roger Evernden's 2003 Checklist
  - http://www.evernden.net/articles/the-8-factor-approach-to-enterprise-architecture/
- Might this checklist be a basis for evaluating Architecture Frameworks?
- Notice what is missing
  - All cost elements

Sample of Downey's Checklist Detail

6. Information quality of service	
Security	Have the security needs for accessing the information been determined?
	Is the Information access compliant with FOIA and Privacy Act?
	Has the information been classified for the right security level?
	Is there a need for an authorization scheme to protect tagging based on user roles?
Availability	Are the availability requirements for the information determined?
Reliability	Are the reliability requirements for the information determined?
Scalability	Has the volume analysis been performed for the information that will be generated?
	Has the usage analysis been performed for the information that will be generated?
Usefulness	Has the time span for the use of this information been analyzed?
	Has the time for updating the information been scheduled?

- . Building an Information Architecture Checklist.
  - http://journalofia.org/volume2/issue2/03-downey/
- Notice what is missing
  - Any attempt to quantify things that are normally quantified in engineering
  - Gilb, SM, CE
    - http://www.gilb.com/tiki-download_file.php?fileId=26

Don Reifer's 2013 Agile Survey "QUANTITATIVE ANALYSIS OF AGILE METHODS" report July 1 2013 full report acquisition; info@reifer.com : www.reifer.co



Product Type: Sale, Internal Use

Applications Domain	Type of Product			Size Metrics (see Legend for meaning)
	Prototype	For Sale	For Internal Use	
Automation	5		13	SP, UCP, FP, eSLOC
Command & Control	5	3	13	SP, UCP, FP, eSLOC
Financial	5	5	15	SP, UCP, FP
Defense	8	13+		SP, UCP, FP, eSLOC
Info Technology	7		20	SP, UCP, FP, FeP, eSLOC
Medical	6		11	SP, UCP, FP, eSLOC
Mobile Apps	6	12		SP, UCP, FP, eSLOC
Software Tools	6	21	6	SP, UCP, FP, FeP, eSLOC
Telecommunications	6	3	20	SP, UCP, FP, FeP, eSLOC
Web Business	11	25	5	SP, UCP, FP, FeP, WO, eSLOC
TOTALS	65	82	103	

Table 4 - Number of Projects by Applications Domain and Life Cycle Phase and Type of Product

Legend

SP - Story Points13 UCP - Use Case Points14 FeP - Feature Points16 WO - Web Objects17

3. Agile Methods

FP - Function Points15 eSLOC - Equivalent SLOC18

Metrics 1 of 2

Goal	Question	Metric/Measure
Satisfy users' needs with working product	How to specify goals?	***User stories* - *number of stories expressed in story points.*** **Use cases** - number of use cases expressed in use case points. **Functional/Feature specifications or lists** - number of function or features expressed in function points, eSLOCs or another measure.
	How do you measure satisfaction of user needs?	***Sprint Backlog* - *a prioritized list of functions and/or features planned vs. actual for a sprint expressed in user stories and measured in terms of % complete or rate of progress.***
		Function/Feature Set Backlog - the list of functions and/or features planned vs. actual as a function of sprint or release as measured in terms of % complete or rate of progress.
Deliver quality product on-time and within budget	How do you track overall progress?	**Goal Accomplishment Rate** - goal completion measured in terms of stories, use cases and/or functions/features planned vs. actual as measured in terms of % complete or rate of progress.
		Defect Removal Rate - the number of defects remaining vs. expected as measured in terms of % defects open vs. defect model predictions. This metric is useful in deciding whether or not to release product.
		Iteration Flow Completion Rate* - *measures the % of work "defined," "in progress," "completed" and "accepted" in an iteration, sprint or release as a function of time.
		Refactoring Rate - the number of times releases were refactored vs. the number expected as measured in terms of % refactoring vs. model predictions for the same.
		Rework Rate - the number of times releases were reworked vs. expected as measured in terms of % rework vs. model predictions.
	How do you track schedule?	***Release Burndown* - *the work remaining for a release measured in terms of planned vs. actual story points, team days, days and/or functions/features as measured in terms of % complete or rate of progress.***
		Team Velocity* - *the amount of work the team has accomplished in terms of stories or features/functions during a sprint as measured as a function of time. Velocity achieved on one sprint is used to predict the velocity for future sprints.

Metrics for Progress 1 of 2 *Enlarged*
What is missing?

Goal	Question	Metric/Measure
		Goal Accomplishment Rate - goal completion measured in terms of stories, use cases and/or functions/features planned vs. actual as measured in terms of % complete or rate of progress.
		Defect Removal Rate - the number of defects remaining vs. expected as measured in terms of % defects open vs. defect model predictions. This metric is useful in deciding whether or not to release product.
		Iteration Flow Completion Rate - measures the % of work "defined," "in progress," "completed" and "accepted" in an iteration, sprint or release as a function of time.
		Refactoring Rate - the number of times releases were refactored vs. the number expected as measured in terms of % refactoring vs. model predictions for the same.
		Rework Rate - the number of times releases were reworked vs. expected as measured in terms of % rework vs. model predictions.
		terms of stories or features/functions during a sprint as measured as a function of time. Velocity achieved on one sprint is used to predict the velocity for future sprints.

Metrics 2 of 2

		Schedule Progress - major milestones achieved as a function of time measured in terms of planned vs. actuals and expressed in terms of % complete or as a rate of progress.
	How do you track budget?	***Budget Progress - expenditures expressed in head counts, staff hours and/or currency ($) measured in terms of planned versus actuals and expressed in terms of % complete or rate of expenditure.***
		Staffing Rate - planned vs. actual rate team is staffed expressed in terms of rate of progress. Keeping track of staffing is important because staffing up is one of the key factors that leads to success.
	How do you measure quality during development?	**Defect Density** - the number of defects planned vs. actual as a function of time and size per component, sprint, and/or release (defects/KSLOC for release 5.6 at two months prior to release).
		Defect Backlog - the number of open vs. closed defects by criticality as a function of time for a component, sprint or release (5 Category 1 defects open/release 1 at two months prior to release).
		Defect Find vs. Fix Rate - the number of defects found and fixed as a function of time by sprint or release expressed in terms of defects remaining when plotted vs. expectations.
		Escape Rate - the number of defects found and fixed out of phase versus the total number of defects.
		Test Coverage - the number of tests completed vs. the number planned as a % complete or rate of progress.
	How do you measure quality	**Defect Rate** - the number of defects found and fixed post-release during operations as a function of product or release and time in service (3 defects/KSLOC during first year of operations).
Provide value to the customers?	How do you measure value?	***Cost Avoidance - the amount of expenditures that can avoided in the future measured in terms of either reduced staff or increased efficiencies (doing more with less).***
		Return on Investment (ROI) - the payback on an investment measured in terms of the percentage yield from an initiative[27]. Determining ROI requires that you have a handle on both the costs involved and the benefits accrued.
		Reduced Time to Market - the amount of schedule time gained measured in terms of the time it takes from the start of the project to get the product into the hands of the customer or user.
		Improved Competitive Advantage - the increased capacity and capability gained as measured in terms of productivity improvements relative to norms within your industry (as measured by benchmarks)

Metrics 2 of 2 Enlarged

		Schedule Progress - major milestones achieved as a function of time measured in terms of planned vs. actuals and expressed in terms of % complete or as a rate of progress.
		Cost Avoidance* - *the amount of expenditures that can avoided in the future measured in terms of either reduced staff or increased efficiencies (doing more with less).
		Return on Investment (ROI) - the payback on an investment measured in terms of the percentage yield from an initiative[27]. Determining ROI requires that you have a handle on both the costs involved and the benefits accrued.
		Reduced Time to Market - the amount of schedule time gained measured in terms of the time it takes from the start of the project to get the product into the hands of the customer or user.
		Improved Competitive Advantage - the increased capacity and capability gained as measured in terms of productivity improvements relative to norms within your industry (as measured by benchmarks).

Table 5 - Sample Agile Metrics Develop Via Goal-Question Metric Paradigm

What Does the Data Say?

- We looked at 800 projects from 60 firms over a ten year timeline
  - 10 applications domains
  - 250 agile projects
  - 550 traditional projects
  - All less than 10 years old
- Productivity is better
  - Many caveats with the statement
  - Many critics argue results due to Hawthorne effect
  - Data supports conclusions
- Cost is cheaper
  - Again many caveats
  - Data localized to USA
  - Data again supports the conclusions
- Quality is slightly worse
  - Some controversy
  - Several databases in addition to ours support this conclusion
  - Some argue that advocates put too much attention on test and not enough on engineering quality into the product

Number of Projects Surveyed by Agile Method

Applications Domain	Agile Methodology								% of Projects
	Crystal[2]	DSDM[3]	Lean[4]	Kanban[5]	Scrum[6]	XP[7]	Hybrid[8]	Total[+]	
Automation			2	3	8	1	4	18	14%
Command & Control	2	2	3		6	3	5	21	16%
Financial	3	4	3		8	4	3	25	14%
Defense			3	1	8		9	21	7%
Info Technology	4	2			10	6	5	27	18%
Medical	1	2	2		8		4	17	12%
Mobile Apps					4	6	8	18	32%
Software Tools	4	4	2	2	10	5	6	33	22%
Telecomm	1	1	4	4	6	3	10	29	15%
Web Business		2			12	13	14	41	24%
TOTALS	15	17	19	10	80	41	68	250	

Number of Projects by Agile Experience by Method

Applications Domain	No. of Firms	No. of Agile Software Projects	Average Experience with Adopted Agile Method			Example Products Generated by Firm
			1 - 2 Years	3 to 5 Years	5+ Years	
Automation	4	18	44%	50%	6%	Pipeline automation
Command & Control	5	21	38%	43%	19%	Network control console
Financial	5	25	28%	36%	36%	Day trading system
Defense	8	21	57%	33%	10%	Weapons system
Info Technology	7	27	33%	41%	26%	ERP applications
Medical	5	17	41%	41%	18%	Pharmacy
Mobile Apps	6	18	72%	28%	-	GPS location service
Software Tools	6	33	33%	42%	25%	Compiler system
Telecommunications	5	29	45%	36%	19%	Switching system
Web Business	9	41	54%	37%	9%	Web site for travel
TOTALS/AVERAGES	60	250	44%	39%	17%	Weighted Averages

Projects by Size & Life Cycle Phase

Applications Domain	Average Size (eSLOC)	Size Range (eSLOC)	Life Cycle Phase		
			Prototype	Development	Maintenance
Automation	155	48 to 326	5	11	2
Command & Control	175	41 to 543	5	11	5
Financial	255	68 to 755	5	17	3
Defense	78	35 to 265	8	10	3
Info Technology	212	65 to 685	7	8	12
Medical	178	55 to 578	6	9	2
Mobile Apps	55	22 to 172	6	12	
Software Tools	185	92 to 788	6	16	11
Telecommunications	276	87 to 558	6	14	9
Web Business	87	45 to 345	11	30	
TOTALS/AVERAGES	168	22 to 755	65	138	47

Legend

eSLOC (Equivalent SLOC) – is a measure of software size developed to reflect how many new source lines of code are in the application so it can be compared to traditional benchmarks

Projects by Type of Product and Size

Applications Domain	Type of Product			Size Metrics (see Legend for meaning)
	Prototype	For Sale	For Internal Use	
Automation	5		13	SP, UCP, FP, eSLOC
Command & Control	5	3	13	SP, UCP, FP, eSLOC
Financial	5	5	15	SP, UCP, FP
Defense	8	13+		SP, UCP, FP, eSLOC
Info Technology	7		20	SP, UCP, FP, FeP, eSLOC
Medical	6		11	SP, UCP, FP, eSLOC
Mobile Apps	6	12		SP, UCP, FP, eSLOC
Software Tools	6	21	6	SP, UCP, FP, FeP, eSLOC
Telecommunications	6	3	20	SP, UCP, FP, FeP, eSLOC
Web Business	11	25	5	SP, UCP, FP, FeP, WO, eSLOC
TOTALS	65	82	103	

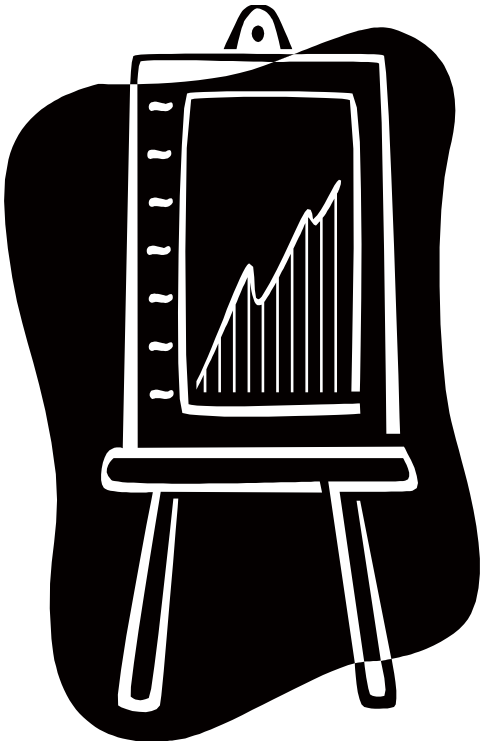
Legend

SP - Story Points UCP - Use Case Points FP - Function Points

FeP - Feature Points WO - Web Objects eSLOC - Equivalent SLOC

Agile Productivity

- Software productivity measure used is eSLOC/SM
  - Selected so we could compare against traditional project performance
  - Hard to convert other measures to eSLOC when estimating
  - Easy to count lines of code in the repository using code counters
- Major influence factors include:
  - Number of agile projects completed (negates Hawthorne effect)
  - Maturity and scalability of method used



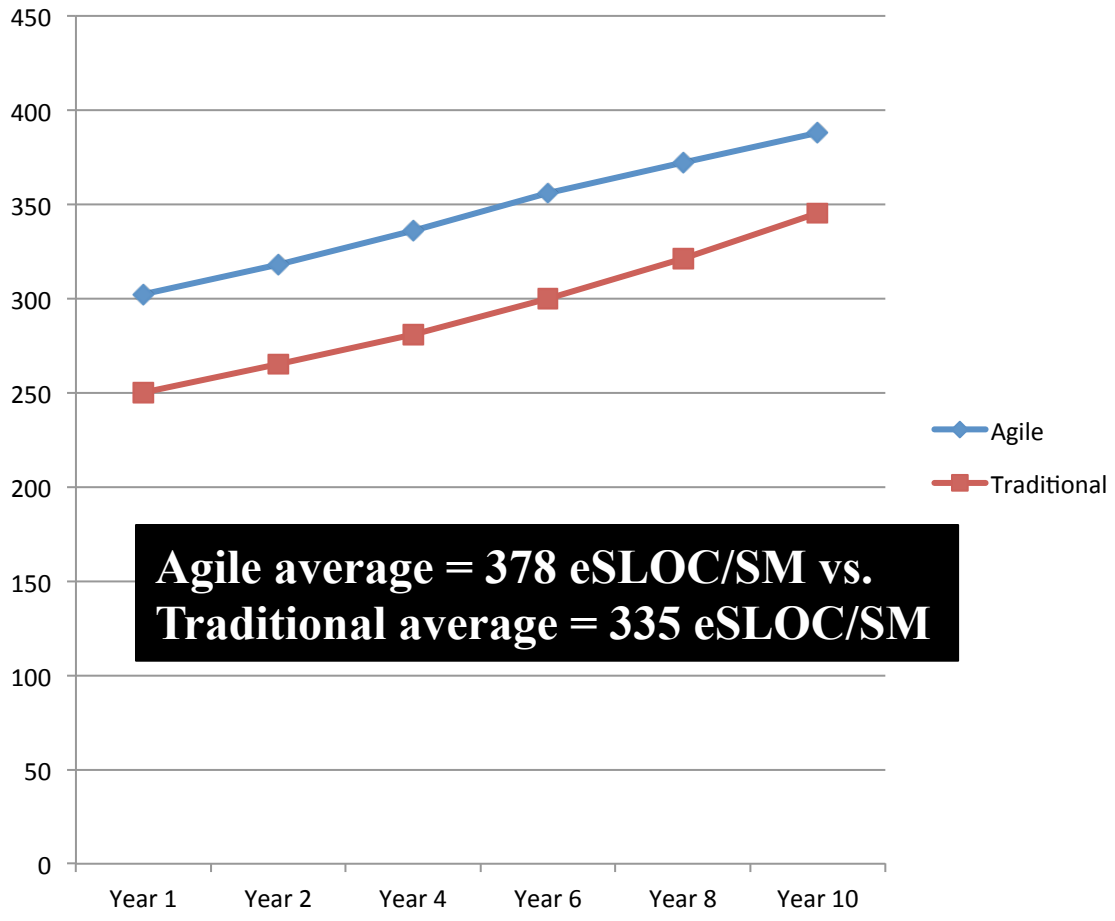
Would you Believe 100%?

Agile Vs. Traditional Productivity

Applications Domain	No. of Firms	No. of Agile Projects	Agile Productivity# (eSLOC/SM)	No. of Non-Agile Projects	Non-Agile Productivity (eSLOC/SM)	Major Contributing Factors
Automation	4	18		30		+, platform constraints
Command & Control	5	21				+, *, degree of reuse
Financial	5	25	443	28	405	+, governance degree
Defense (includes both IT & weapons)	8	21		115		+, precedentedness, required reliability
Info Technology	7	27		35		+, *
Medical	5	17		28		+, governance degree
Mobile Apps	6	18		44		+, *, requirements volatility
Software Tools	6	33		77		+, *, degree of reuse
Telecommunications	5	29		64		+, *, degree of reuse
Web Business	9	41		75		+, *, requirements volatility
TOTALS	60	250	378	550	335	

+ Common factors across all domains include size, complexity and teamwork
* Personnel capabilities and experience (platform, language, application, etc.)
# Agile processes are mature, staff trained and adoption is well under way after 3 to 4 years of effort

Agile Vs. Traditional Productivity

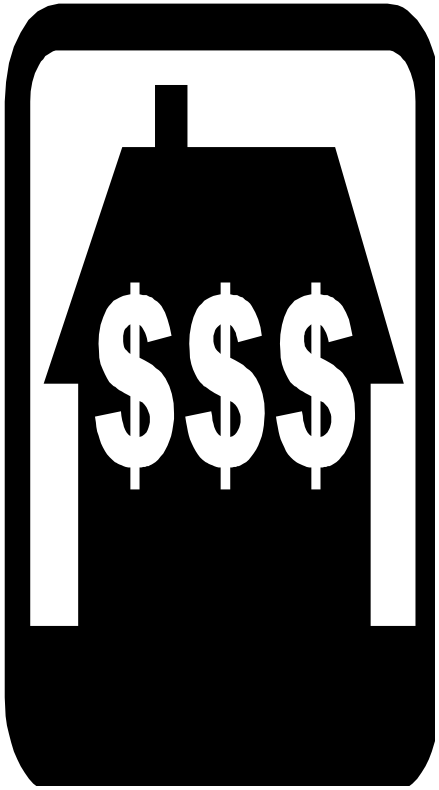


Agile vs. Traditional Software Productivity Trends

- Agile productivity seems higher by factors of 10 to 40% depending on domain
- True even after adoption has taken place and buzz dissipates
- Time to market cannot be measured due to frequent deliveries

Agile Cost

- Agile cost measured in $/eSLOC
  - Easily converted using factors we developed to $/story point, $/UML point, $/function/feature point, etc. for estimating purposes
  - Measure sensitive to labor rates and how they were calculated
  - The following major costs not factored into the computation as they were funded separately
    - Process reengineering (especially those related to CM/DM, metrics and SQA)
    - Facilities costs (agile tools, war room, etc.)
    - Change management (education, etc.)



How Much Did you Say?

Agile Vs. Traditional Cost

Applications Domain	No. of Firms	No. of Agile Projects	Agile Cost# ($/ eSLOC)	No. of Non-Agile Projects	Non-Agile Cost ($/ eSLOC)	Major Contributing Factors
Automation	4	18		30		+, CM, RM, RTM
Command & Control	5	21		54		+, PI, RM, RTM
Financial	5	25		28		+, CM, OS, PI, RM
Defense (includes both IT & weapons)	8	21		115		+, CM, IC, PI, SC, RM, RTM
Info Technology	7	27	22	35	23	+, CM, OS, PI, RM, RTM
Medical	5	17		28		+, CM, PI, RM, RTM
Mobile Apps	6	18		44		+, CM, RM, RTM
Software Tools	6	33		77		+, CM, PI, SC, RM, RTM
Telecommunications	5	29		64		+, CM, IC, PI, SC, RM, RTM
Web Business	9	41		75		+, CM, RM, RTM
TOTALS	60	250	34	550	47	

+ Common factors across all domains include agile technology maturity, % learning curve and prevalent labor rates

Agile processes are mature, staff trained and adoption is well under way after 3 to 4 years of effort

Legend

$/eSLOC - dollars (2012) per equivalent Source Lines of Code

CM - configuration management

IC - Innovative contracting

OS - out-source

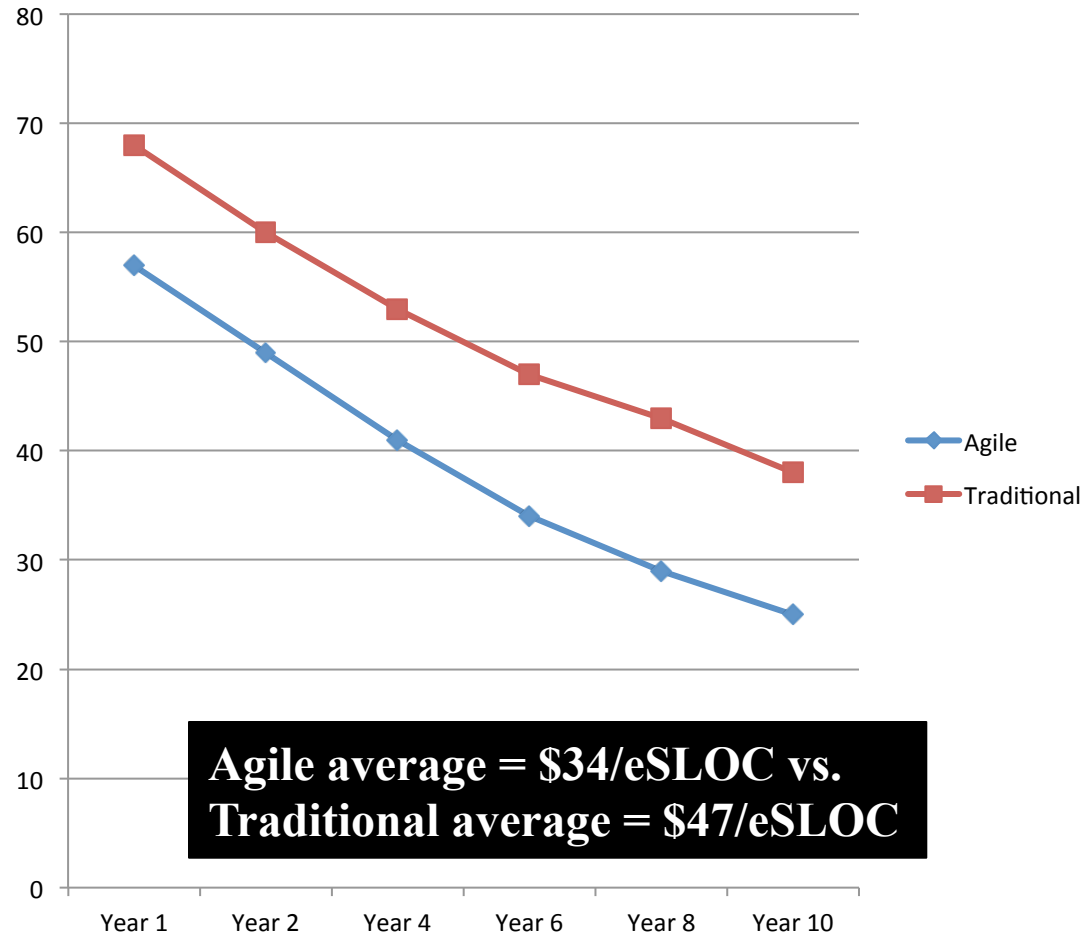
PI - process improvement

SC – subcontract

RM - requirements management

RTM- regression test management

Agile Vs. Traditional Cost

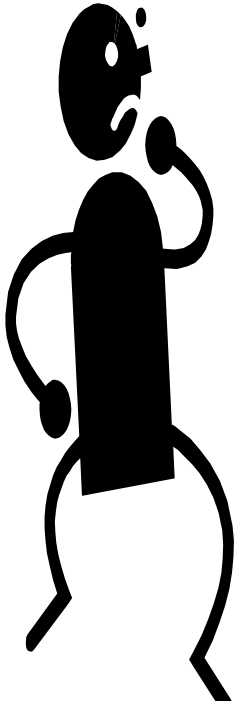


Agile vs. Traditional Software Cost Trends

- Agile seems less costly even when labor rates are normalized across domains
- As noted, many costs associated with change are not accounted for
- Seem to be many issues related to agile supplier management

Agile Quality

- Quality measured in defects/KeSLOC computed post-delivery
  - Again, selected so we could compare against traditional project performance
  - Different measures are used to compute quality during development
  - Quality measures should include more of the "ilities," but hard to quantify softer factors like usability
- Major influence factors include:
  - Degree to which release was tested (latent defects)
  - Whether quality was designed into the product from the start



Bugs, what Bugs?

More on Quality

- Quality involves many more "ilities" than reliability
  - Usability, availability, portability, etc.
- However, most of these are very hard to measure
  - Some are soft measures whose "goodness" is often measured in eyes of viewer
- There are also issues revolving around performance
  - These measures enter into the determination of quality
- We took the simple way out by focusing on the one hard measure for which we had data, defect density
  - Will update our approach as data are available

And the Winner Is?

Agile Vs. Traditional Quality

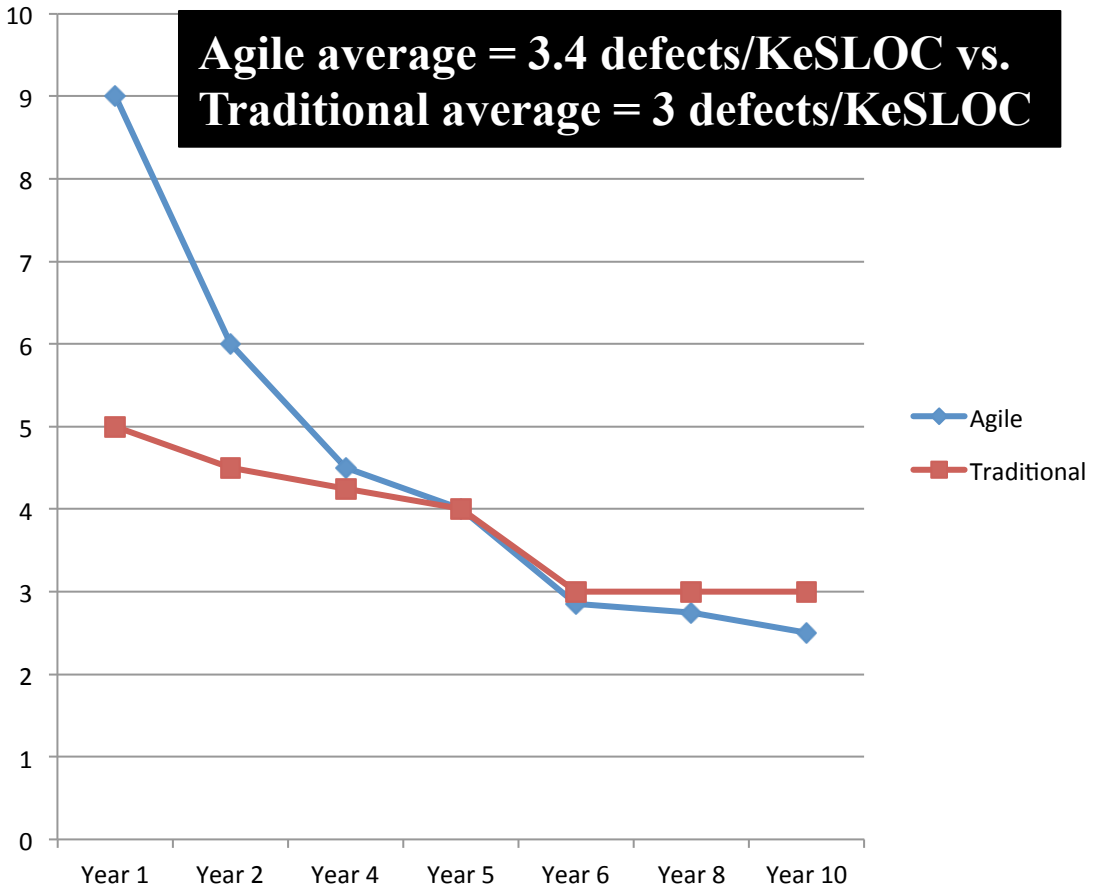
Applications Domain	No. of Firms	No. of Agile Projects	Agile Quality# (d/ KeSLOC)	No. of Non-Agile Projects	Non-Agile Quality (d/ KeSLOC)	Major Contributing Factors
Automation	4	18		30		+, CDW, RAM, RTM, SA
Command & Control	5	21		54		+, CDW, KN, RAM, RTM, SA
Financial	5	25		28		+, CDW, KN, RTM, SA
Defense (includes IT and weapons)	8	21		115		+, CDW, KN, RAM, RTM, SA
Info Technology	7	27		35		+, CDW, KN, RTM, SA
Medical	5	17		28		+, CDW, KN, RTM, SA
Mobile Apps	6	18		44		+, KN, TFM, RTM
Software Tools	6	33	3.5	77	4	+, CDW, KN, RTM, SA
Telecommunication	5	29		64		+, CDW, KN, RAM, RTM
Web Business	9	41		75		+, KN, TFM, RTM
TOTALS	60	250	3.14	550	3	

+ Common factors across all domains include automated testing, defect classification, defect modeling and tracking, defect prevention, nightly builds, root cause analysis, statistical process controls and version control.
# Agile processes are mature, staff trained and adoption is well under way after 3 to 4 years of effort

Legend

d/KeSLOC - defects per thousand equivalent Source Lines of Code
CDW - code and design walkthroughs
RAM - reliability analysis & modeling
KN - Kanban
RTM - regression test management
SA - static analysis (including coverage)
TFM - test first management

Agile Vs. Traditional Quality



Agile Adoption & Quality as Measured by Defect Rates

- Agile quality is not as good as that computed for traditional projects
- Root cause seems to be that too much attention placed on testing and not enough on engineering quality into the product

Agile Strengths & Weaknesses

- Other strengths include:
  - Ability to react quickly to changing priorities
  - Less documentation (+ & -)
  - Increased and continual dialog with customers/users
  - Emphasis on teams, team building and teamwork
  - Increased openness as working code is made available to everyone
  - Emphasis on continuous integration and automated regression testing
  - Increased willingness to try something new

- Weaknesses include:
  - Agile QA needs to be viewed as much more than just testing
  - The jury is still out on agile maintenance
  - Contracting/subcontracting issues abound for agile
    - How do you deal with liability issues?
    - What are good contract terms and conditions?
    - What are the deliverables?
  - Scalability
    - Metrics for large projects?
    - Addressing global projects
  - Project level risk management practices for agile

Addressing Weaknesses

- Very active agile public forums on social media
  - LinkedIn has several discussion groups
  - Lots of questions posted and all sorts of answers
  - Lots of evangelists responding with misinformation
- Many vendors with self-serving information
  - Most surveys reviewed were based on opinions and not hard data

- Some solid information out there
  - Capers Jones at www.namcook.com
  - David Rico at www.davidfrico.com
  - Donald Reifer at www.reifer.com
- Lots of books and advice on change management
- Little advice on what to do after change has occurred

Summary - Beware the Hype

- We shared with you the findings of our analysis of agile "hard" data
  - Based on completed project performance
- The agile pluses are:
  - Higher productivity
  - Lower costs
  - Lots of softer factors that lead to higher morale and motivation
- The negatives
  - Lower quality (not much though)
  - Scaling, contracting, risk management and maintenance issues

Want to Buy a Used Car?

Conclusions – Be Successful

Hi Five!

- We believe that use of agile methods has merit and is worth pursuing
  - Tie adoption to your business goals and a solid business case
- When making the move, do so with both your eyes and ears wide open
  - Do what makes sense in your firm and applications domain
  - Listen to the practitioners, not the zealots
  - Capture data, look at the metrics and be able to change on the fly
  - Do the right things for the right reasons
- Aim at making a positive contribution

Brodie & Woodman Middlesex U on Prioritisation Methods

Prioritization of Stakeholder Value Using Metrics

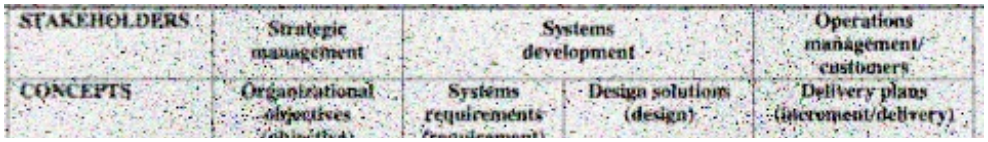
Priortisation vs Data Structuring

Table 3. How prioritization techniques cope with a selection of data structuring issues

Prioritization Technique >	**Grouping**	**Ranking**	**Weighting**	**Metrics**
Example of prioritization data	"High", "Medium" or "Low"	1, 2, 3, ... N	30/100	Time to carry out task to be reduced from 1 day to 5 minutes
Data Structuring Issue				
Stakeholder value	Implicit; value is say, "Medium"	Implicit; value is say, ranked as "2"	Implicit; value is 30% of whatever 100% equates to	Depends on metric. For this metric, an estimate of value is able to be derived if say, monetary rate of pay is known.
Multiple stakeholder viewpoints (Note assuming 2 stakeholders)	N Would be represented as say, "High" and "Medium"	N Would be represented as say, "2" and "20"	N Would be represented as say, 30/100 and 2/100	Y Time to carry out task to be reduced from 1 day down to say, 5 minutes and to 2 hours
Requirements Abstraction	N	N	Y Create hierarchy	Y Create hierarchy
Interdependencies	(Y) Would have to work by selecting an item and then seeing if there were any prior dependencies that would override	(Y) Ditto	(Y) Ditto	(Y) Ditto
Dynamic prioritization	Y Add any new data to an existing data grouping. No extra effort (unless something has changed)	Y Would need to re-examine existing ranks	(Y) Considerable effort needed by stakeholders	Y Add to existing data and reprocess
Scaling up	N Too many in a group	N Difficult to keep track of numerous rankings	N Considerable effort to carry out all the additional pair-wise comparisons	(Y) Would use high-level hierarchical data to reduce numbers

Table 1. Prioritization factors by stakeholder viewpoint and software engineering concept

STAKEHOLDERS	Strategic management	Systems development		Operations management/ customers
CONCEPTS	Organizational objectives (objective)	Systems requirements (requirement)	Design solutions (design)	Delivery plans (increment/delivery)
PRIORITIZATION FACTORS				
OPINION	Vision/intuition/gut feeling/preference/ bias	Preferences/ bias/ importance	Intuition/ preferences/ bias	Preferences/ bias
STRATEGY	Strategic alignment/ business objectives/ product strategy		Long-term Strategy for systems architecture	
	Competition	Quality		
	Customer demand	Originator of requirement		End user value
	New business potential			
TIME	Urgency/time to market/lead time		Time schedule/ time constraints	
	Long term versus short term		Long term versus short term	
LEGAL	Legal mandate/ regulations	Legal mandate/ regulations		
	Contracts in place			
FINANCIAL BENEFIT	Market value/price			
	Financial benefits			
	Financial penalties			
	Benefit/cost ratio			
	Cost of not implementing			
COST	Development costs/ implementation costs/ support costs		Development costs/ support costs	Implementation costs/ support costs
	Operational costs			Operational costs
FIT	Fit with operational context: . business processes . skills/training . delivery timing		Staff competence Balanced workload	Fit with operational context: . business processes . skills/training . delivery timing
			Resource availability/ effort constraints	Resource availability/ effort constraints
	Fit with other products		Change impact/ base code dependencies	Change impact
			Logical implementation order	
			Reuse potential	
EXTERNAL DEPENDENCY	Intermediary channels	External dependencies	External dependencies	
RISK	Business risk Sales barriers	Volatility of requirements	Technical risk in: . current system . proposed system . implementation process	
			Difficulty of implementation/	Difficulty of implementation/



STAKEHOLDERS	**Strategic management**	**Systems development**		**Operations management/ customers**
CONCEPTS	**Organizational objectives (objective)**	**Systems requirements (requirement)**	**Design solutions (design)**	**Delivery plans (increment/delivery)**
PRIORITIZATION FACTORS				
OPINION	Vision/intuition/gut feeling/preference/ bias	Preferences/ bias/ importance	Intuition/ preferences/ bias	Preferences/ bias
STRATEGY	Strategic alignment/ business objectives/ product strategy		Long-term Strategy for systems architecture	
	Competition	Quality		
	Customer demand	Originator of requirement		End user value
	New business potential			
TIME	Urgency/time to market/lead time		Time schedule/ time constraints	
	Long term versus short term		Long term versus short term	
LEGAL	Legal mandate/ regulations	Legal mandate/ regulations		

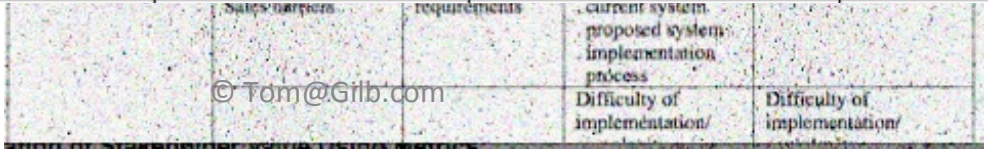


Table 2. Mapping of prioritization technique(s) and scale type(s) to prioritization methods

Prioritization Method	Prioritization Technique(s)	Scale Type(s)
QFD	Weighting and Grouping	Ratio Ordinal
AHP	Weighting (Pair-wise comparison)	Ratio
IE	Metrics	Absolute
Cost-Value Approach	Weighting (Pair-wise comparison)	Ratio
MoSCoW	Grouping	Ordinal
Planning Game	Grouping	Ordinal
Requirements Triage	Grouping and Weighting	Ordinal Ratio

Table 3. How prioritization techniques cope with a selection of data structuring issues

Prioritization Technique >	Grouping	Ranking	Weighting	Metrics
Example of prioritization data	"High", "Medium" or "Low"	1, 2, 3, ... N	30/100	Time to carry out task to be reduced from 1 day to 5 minutes
Data Structuring Issue				
Stakeholder value	Implicit; value is say, "Medium"	Implicit; value is say, ranked as "2"	Implicit; value is 30% of whatever 100% equates to	Depends on metric. For this metric, an estimate of value is able to be derived if say, monetary rate of pay is known.
Multiple stakeholder viewpoints (Note assuming 2 stakeholders)	N Would be represented as say, "High" and "Medium"	N Would be represented as say, "2" and "20"	N Would be represented as say, 30/100 and 2/100	Y Time to carry out task to be reduced from 1 day down to say, 5 minutes and to 2 hours
Requirements Abstraction	N	N	Y Create hierarchy	Y Create hierarchy
Interdependencies	(Y) Would have to work by selecting an item and then seeing if there were any prior dependencies that would override	(Y) Ditto	(Y) Ditto	(Y) Ditto
Dynamic prioritization	Y Add any new data to an existing data grouping. No extra effort (unless something has changed)	Y Would need to re-examine existing ranks	(Y) Considerable effort needed by stakeholders	Y Add to existing data and reprocess
Scaling up	N Too many in a group	N Difficult to keep track of numerous rankings	N Considerable effort to carry out all the additional pair-wise comparisons	(Y) Would use high-level hierarchical data to reduce numbers

Fig. 2. VIE table for bank case study. The shaded area represents the extensions to IE.

Key:
s = seconds
m = minutes
d = days
w = week

Bank System

By End Date: dd/mm/yyyy

Stakeholder Value							Requirements	Designs by expected Increment with design dependencies			
Regulator	IT Dept.	Customer	Rule Admin.	Business Unit	Back Office	Total Value / Benefit		1 D1: Automate Rules + Manual Testing	2 D2: Back Office Loan Decisioning	3 D3: Web Self-Service	4 D4: Automate Rules + Automate Testing
		4				4	R1: Time for customer to submit request 30 min <-> 10 min	-	-	10 m 100%	-
					3	3	R2: Time for Back Office to enter request 30 min <-> 10 min	-	-	0 m 150%	-
		9		9		18	R3: Time to respond to customer request 5 days <-> 20 seconds	-	1 d 80%	20 s 100%	-
					1	1	R4: No of Back Office complaints 10 per week <-> 0	5 50%	<1 90%	0 100%	(2) (80%)
		1			5	6	R5: No of customer complaints 25 per week <-> 5	-	15 50%	5 100%	-
1			5	4	8	18	R6: Time to update business rules 1 month <-> 1 day	2 w 50%	-	-	1 d 100%
1			3	4	6	14	R7: Time to distribute business rules 2 weeks <-> 1 day	1 d 100%	-	20 s 103%	-
2		14	8	17	23	64	Cumulative Total for Performance Requirements	200%	170%	280%	50%
							Development Budget 2.5M <-> 300K	2.3	2.0	1.0	0.5
							Development Cost for Design	0.2	0.3	1.0	0.5
							Cumulative Performance to Devt. Cost Ratio	1000	567	280	100
							Cumulative Stakeholder Value to Development Cost Ratio	23.5/0.2 =117.5	17.8/0.3 =59.3	13.7/1.0 =13.7	9/0.5 =18

- end

Prioritisation Methods

- Brodie and Woodward

Prioritization & Data Structuring

Table 3. How prioritization techniques cope with a selection of data structuring issues

Prioritization Technique >	**Grouping**	**Ranking**	**Weighting**	**Metrics**
Example of prioritization data	"High", "Medium" or "Low"	1, 2, 3, ... N	30/100	Time to carry out task to be reduced from 1 day to 5 minutes
Data Structuring Issue				
Stakeholder value	Implicit; value is say, "Medium"	Implicit; value is say, ranked as "2"	Implicit; value is 30% of whatever 100% equates to	Depends on metric. For this metric, an estimate of value is able to be derived if say, monetary rate of pay is known.
Multiple stakeholder viewpoints (Note assuming 2 stakeholders)	N Would be represented as say, "High" and "Medium"	N Would be represented as say, "2" and "20"	N Would be represented as say, 30/100 and 2/100	Y Time to carry out task to be reduced from 1 day down to say, 5 minutes and to 2 hours
Requirements Abstraction	N	N	Y Create hierarchy	Y Create hierarchy
Interdependencies	(Y) Would have to work by selecting an item and then seeing if there were any prior dependencies that would override	(Y) Ditto	(Y) Ditto	(Y) Ditto
Dynamic prioritization	Y Add any new data to an existing data grouping. No extra effort (unless something has changed)	Y Would need to re-examine existing ranks	(Y) Considerable effort needed by stakeholders	Y Add to existing data and reprocess
Scaling up	N Too many in a group	N Difficult to keep track of numerous rankings	N Considerable effort to carry out all the additional pair-wise comparisons	(Y) Would use high-level hierarchical data to reduce numbers

Impact Estimation Table with Stakeholder Value

Key:
s = seconds
m = minutes
d = days
w = week

Stakeholder Value							Bank System By End Date: dd/mm/yyyy Requirements	Designs by expected Increment with design dependencies			
								1	2	3	4
Regulator	IT Dept.	Customer	Rule Admin.	Business Unit	Back Office	Total Value / Benefit		D1: Automate Rules + Manual Testing	D2: Back Office Loan Decisioning	D3: Web Self-Service	D4: Automate Rules + Automate Testing
		4				4	R1: Time for customer to submit request 30 min <-> 10 min	-	-	10 m 100%	-
					3	3	R2: Time for Back Office to enter request 30 min <-> 10 min	-	-	0 m 150%	-
		9		9		18	R3: Time to respond to customer request 5 days <-> 20 seconds	-	1 d 80%	20 s 100%	-
					1	1	R4: No of Back Office complaints 10 per week <-> 0	5 50%	<1 90%	0 100%	(2) (80%)
		1			5	6	R5: No of customer complaints 25 per week <-> 5	-	15 50%	5 100%	-
1			5	4	8	18	R6: Time to update business rules 1 month <-> 1 day	2 w 50%	-	-	1 d 100%
1			3	4	6	14	R7: Time to distribute business rules 2 weeks <-> 1 day	1 d 100%	-	20 s 103%	-
2		14	8	17	23	64	Cumulative Total for Performance Requirements	200%	170%	280%	50%
							Development Budget 2.5M <-> 300K	2.3	2.0	1.0	0.5
							Development Cost for Design	0.2	0.3	1.0	0.5
							Cumulative Performance to Devt. Cost Ratio	1000	567	280	100
							Cumulative Stakeholder Value to Development Cost Ratio	23.5/0.2 =117.5	17.8/0.3 =59.3	13.7/1.0 =13.7	9/0.5 =18

Prioritization factors by stakeholder viewpoint and software engineering concept

STAKEHOLDERS	Strategic management	Systems development		Operations management/ customers
CONCEPTS	**Organizational objectives (objective)**	**Systems requirements (requirement)**	**Design solutions (design)**	**Delivery plans (increment/delivery)**
PRIORITIZATION FACTORS				
OPINION	Vision/intuition/gut feeling/preference/ bias	Preferences/ bias/ importance	Intuition/ preferences/ bias	Preferences/ bias
STRATEGY	Strategic alignment/ business objectives/ product strategy		Long-term Strategy for systems architecture	
	Competition	Quality		
	Customer demand	Originator of requirement		End user value
	New business potential			
TIME	Urgency/time to market/lead time		Time schedule/ time constraints	
	Long term versus short term		Long term versus short term	
LEGAL	Legal mandate/ regulations	Legal mandate/ regulations		
	Contracts in place			
FINANCIAL BENEFIT	Market value/price			
	Financial benefits			
	Financial penalties			
	Benefit/cost ratio			
	Cost of not implementing			
COST	Development costs/ implementation costs/ support costs		Development costs/ support costs	Implementation costs/ support costs
	Operational costs			Operational costs
FIT	Fit with operational context: . business processes . skills/training . delivery timing		Staff competence Balanced workload	Fit with operational context: . business processes . skills/training . delivery timing
			Resource availability/ effort constraints	Resource availability/ effort constraints
	Fit with other products		Change impact/ base code dependencies	Change impact
			Logical implementation order	
			Reuse potential	
EXTERNAL DEPENDENCY	Intermediary channels	External dependencies	External dependencies	
RISK	Business risk Sales barriers	Volatility of requirements	Technical risk in: . current system . proposed system . implementation process	
			Difficulty of implementation/ complexity	Difficulty of implementation/ complexity

Prioritization factors by stakeholder viewpoint and software engineering concept

STAKEHOLDERS	**Strategic management**	**Systems development**		**Operations management/ customers**
CONCEPTS	**Organizational objectives (objective)**	**Systems requirements (requirement)**	**Design solutions (design)**	**Delivery plans (increment/delivery)**
PRIORITIZATION FACTORS				
OPINION	Vision/intuition/gut feeling/preference/ bias	Preferences/ bias/ importance	Intuition/ preferences/ bias	Preferences/ bias
STRATEGY	Strategic alignment/ business objectives/ product strategy		Long-term Strategy for systems architecture	
	Competition	Quality		
	Customer demand	Originator of requirement		End user value
	New business potential			
TIME	Urgency/time to market/lead time		Time schedule/ time constraints	
	Long term versus short term		Long term versus short term	
LEGAL	Legal mandate/ regulations	Legal mandate/ regulations		
	Contracts in place			

Prioritization factors by stakeholder viewpoint and software engineering concept

	Contracts in place			
FINANCIAL BENEFIT	Market value/price			
	Financial benefits			
	Financial penalties			
	Benefit/cost ratio			
	Cost of not implementing			
COST	Development costs/ implementation costs/ support costs		Development costs/ support costs	Implementation costs/ support costs
	Operational costs			Operational costs
FIT	Fit with operational context: . business processes . skills/training . delivery timing		Staff competence Balanced workload	Fit with operational context: . business processes . skills/training . delivery timing
			Resource availability/ effort constraints	Resource availability/ effort constraints
	Fit with other products		Change impact/ base code dependencies	Change impact
			Logical implementation order	
			Reuse potential	
EXTERNAL DEPENDENCY	Intermediary channels	External dependencies	External dependencies	
RISK	Business risk Sales barriers	Volatility of requirements	Technical risk in: . current system . proposed system . implementation process	
			Difficulty of implementation/ complexity	Difficulty of implementation/ complexity

Slides to be added here

- Management Evaluation Methods
  - –Balanced Scorecard BSC
  - –Quality Function Deployment QFD
  - –Impact Estimation Tables IET
  - –Planguage PL

	BSC	QFD	IET	PL	
Attributes					

	BSC	QFD	IET	PL	
Costs					