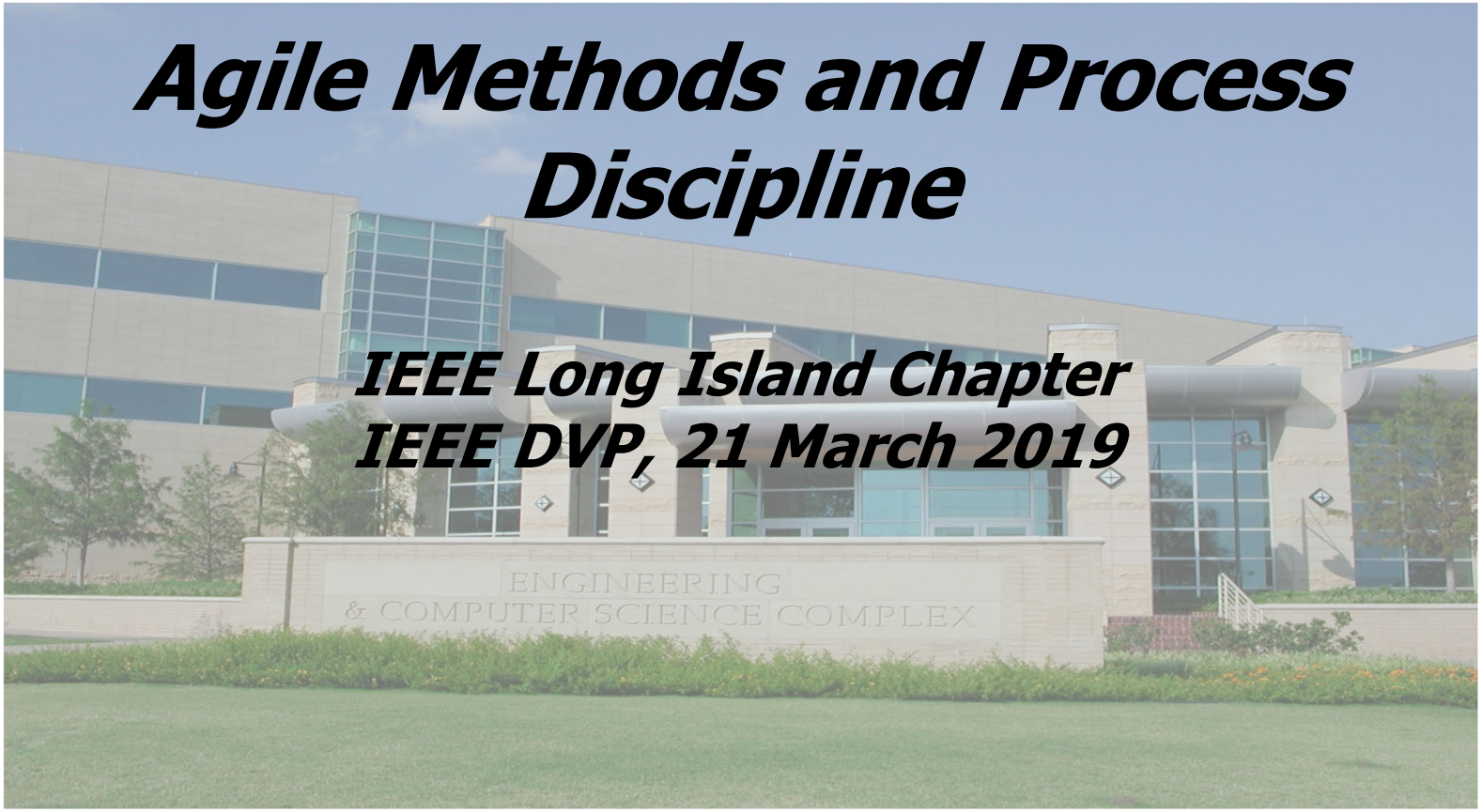


Agile Methods and Process Discipline

IEEE Long Island Chapter
IEEE DVP, 21 March 2019

Dr. Mark C. Paulk
Mark.Paulk@ieee.org, http://utdallas.edu/~mcp130030/

Topics

A Context for Process Discipline

A Context for Agile Methods

Agile and Disciplined Processes

Closing Out…

The State of the Practice?

"I'd rather have it wrong than have it late. We can always fix it later."

- **A senior software manager (industry)**

"The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost."

- **A program manager (government)**

Standish Group – the Chaos Report

- **19% of software projects failures (2015)**
  - **from 31% failures (1994)**
- **52% of software projects challenged (2015)**
  - **from 53% challenged (1994)**

"By regularly putting the development process under extreme time pressure and then accepting poor-quality products, the software user community has shown its true quality standard."

- **DeMarco and Lister (*Peopleware*)**

FAA/TSO Certification

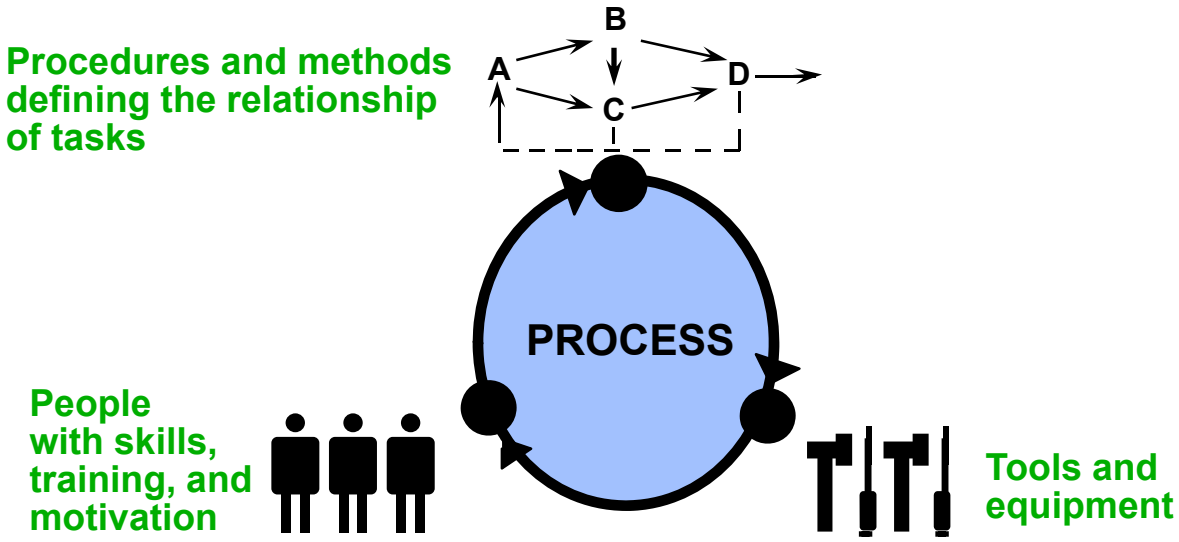
A Technical Standard Order (TSO) is a minimum performance standard issued by the United States Federal Aviation Administration for specified materials, parts, processes, and appliances used on civil aircraft.

- **When authorized to manufacture a material, part, or appliances to a TSO standard, this is referred to as TSO authorization.**
- **Receiving a TSO authorization is both design and production approval.**

Defining Software Process

Process **- a sequence of steps performed for a given purpose (IEEE)**

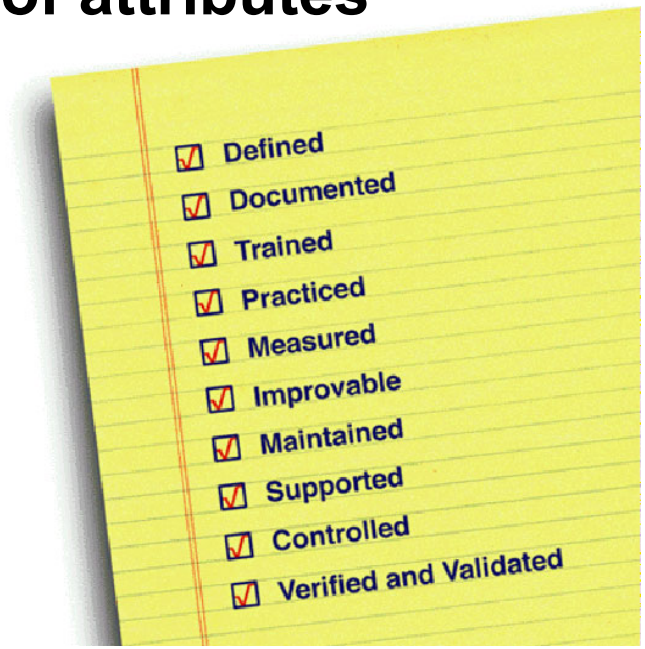
Software process **- a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (Software CMM)**



Qualifiers on Process

Process maturity – the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective

Mature processes share a number of attributes



Other process qualifications:

- **documented process**
- **rigorous process**
- **repeatable (managed) process**
- **defined process**
- **quantitatively managed process**
- **optimizing process**
- **disciplined process**
  - **Say what you're going to do; do what you said you would.**

Process Management Premise

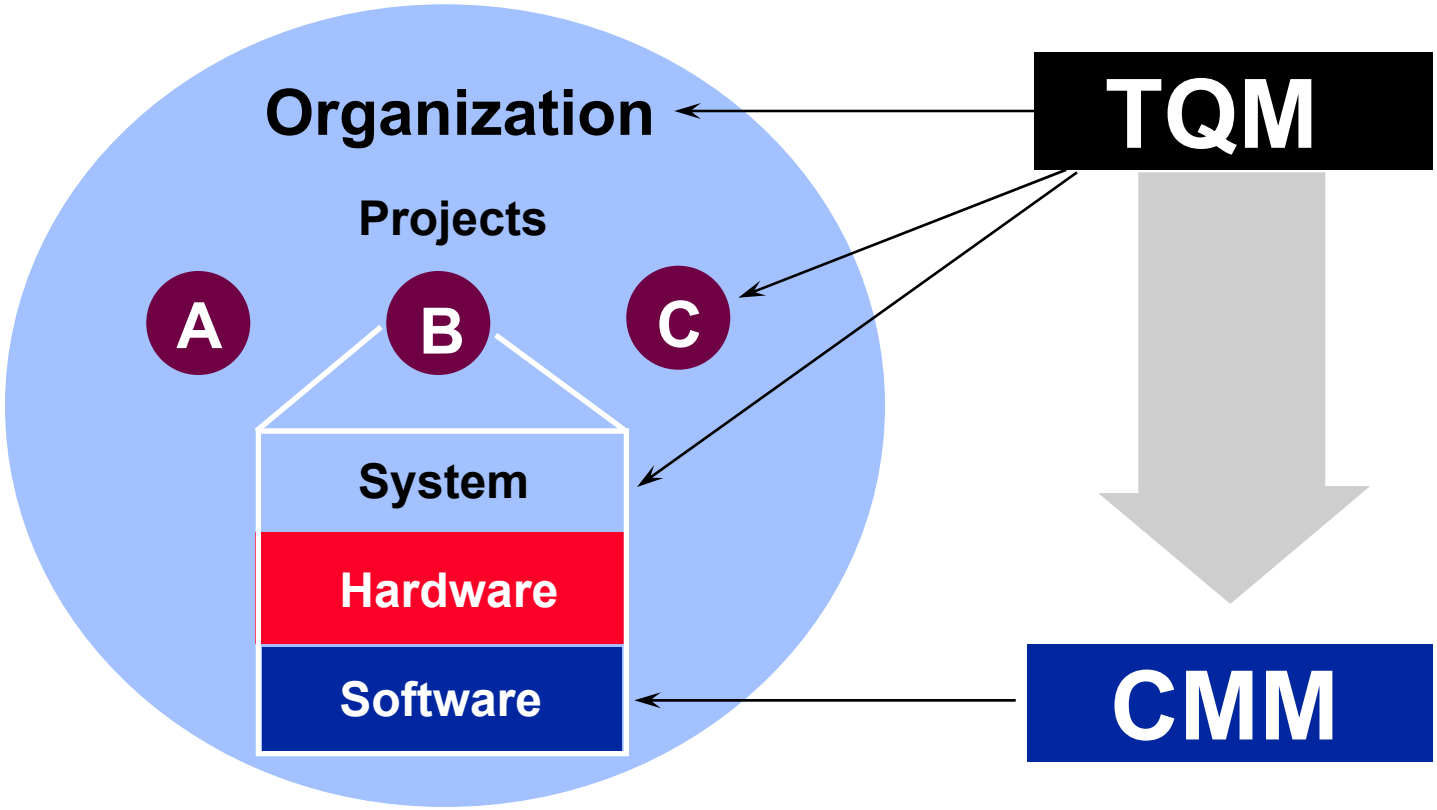
The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it.

This premise implies focus on process as well as product.

The value of this premise is visible world-wide in the Total Quality Management movements in the manufacturing and service industries.

- **performance excellence against business objectives**
- **doing more with less**
- **increasing customer satisfaction (and delight)**
- **improving shareholder equity**

Applying TQM to Software



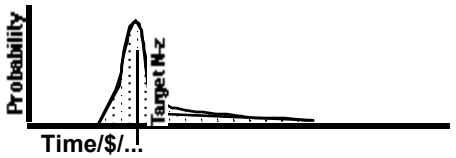
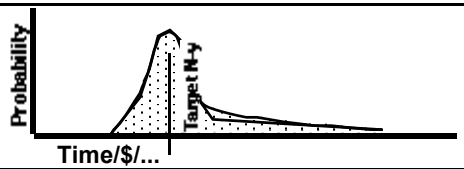
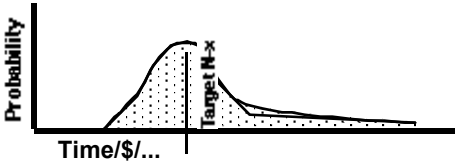
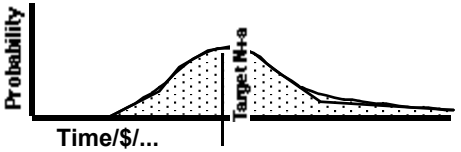
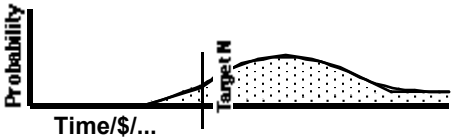
Process improvement fits in an overall business context — CMM applies to software.

Software CMM v1.1 (1993)

Level	Focus	Key Process Areas
5 Optimizing	*Continuous process improvement*	Defect Prevention Technology Change Management Process Change Management
4 Managed	*Product and process quality*	Quantitative Process Management Software Quality Management
3 Defined	*Engineering processes and organizational support*	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
2 Repeatable	*Project management processes*	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
1 Initial	*Competent people (and heroics)*	

Implications of Maturity

Better predictability… Less variability… Improved performance…

	Level	Process Characteristics	Predicted Performance
5	Optimizing	Process improvement is institutionalized	
4	Managed	Product and process are quantitatively controlled	
3	Defined	Software engineering and management processes defined and integrated	
2	Repeatable	Project management system in place; performance is repeatable	
1	Initial	Process is informal and unpredictable	

Prioritizing Business Objectives

Treacy and Wiersma define three values that a company can focus on to drive business success.

Product leadership **– features, innovation**

- **typical focus of commercial shrinkwrap companies**

Customer intimacy **– niche products, relationships**

- **typical focus of IT service providers**

Operational excellence **– promised features, on schedule, on budget**

- **typical focus of custom software development**

M. Treacy and F. Wiersema, The Discipline of Market Leaders, 1997.

Process Framework Objectives

Standards

- **binary qualification that the process or product meets specified requirements**
  - **ISO 9001, ISO/IEC/IEEE 12207, 15288, IEEE 802**

Models focusing on operational excellence

- **Software CMM – software development**
- **CMMI for Development – development of software-intensive systems (software + systems)**
- **CMMI for Acquisition**
- **CMMI for Services**
- **eSCM for Service Providers, eSCM for Clients**
- **ISO 15504 / 330xx (Process Assessment)**

Other Business Objectives

Many other CMM variants created, plus other improvement frameworks

- **creativity and innovation**
  - **R.L. Glass and T. DeMarco, Software Creativity 2.0**
- **human resources (people, competence)**
  - **B. Curtis, W.E. Hefley, and S.A. Miller, People CMM, Second Edition**
  - **J. Pfeffer, The Human Equation**
- **general management**
  - **J. Pfeffer and R.I. Sutton, Hard Facts, Dangerous Half-Truths, & Total Nonsense**
- **knowledge management**
  - **J. Pfeffer and R.I. Sutton, The Knowing-Doing Gap**

CMMI-DEV v1.3 (2011)

Level	Process Characteristics	Process Areas
5 Optimizing	*Focus is on quantitative continuous process improvement*	Causal Analysis & Resolution Organizational Performance Management
4 Quantitatively Managed	*Process is measured and controlled*	Organizational Process Performance Quantitative Project Management
3 Defined	*Process is characterized for the organization and is proactive*	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis & Resolution
2 Managed	*Process is characterized for projects and is often reactive*	Requirements Management Project Planning Project Monitoring & Control Supplier Agreement Management Product & Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	*Process is unpredictable, poorly controlled, and reactive*	

CMMI® V2.0

Transform your organization's performance by improving capabilities to drive meaningful business results.

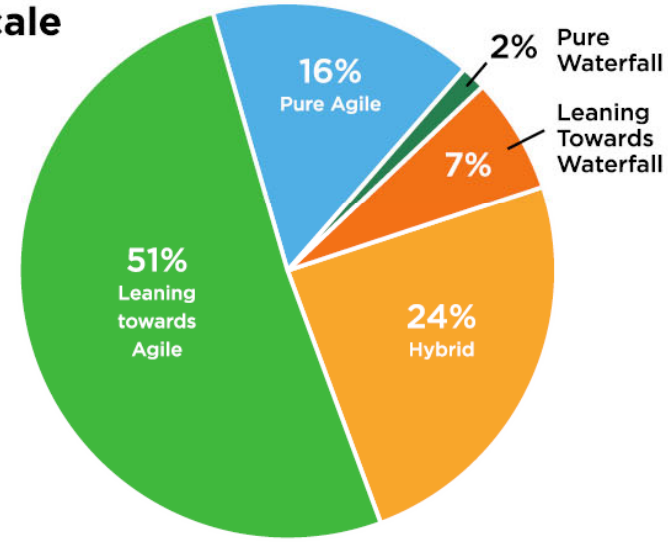
Designed to optimize business performance in an ever-changing global landscape, the CMMI V2.0 model is a proven set of global best practices that enables organizations to build and benchmark the key capabilities that address the most common business challenges, including:

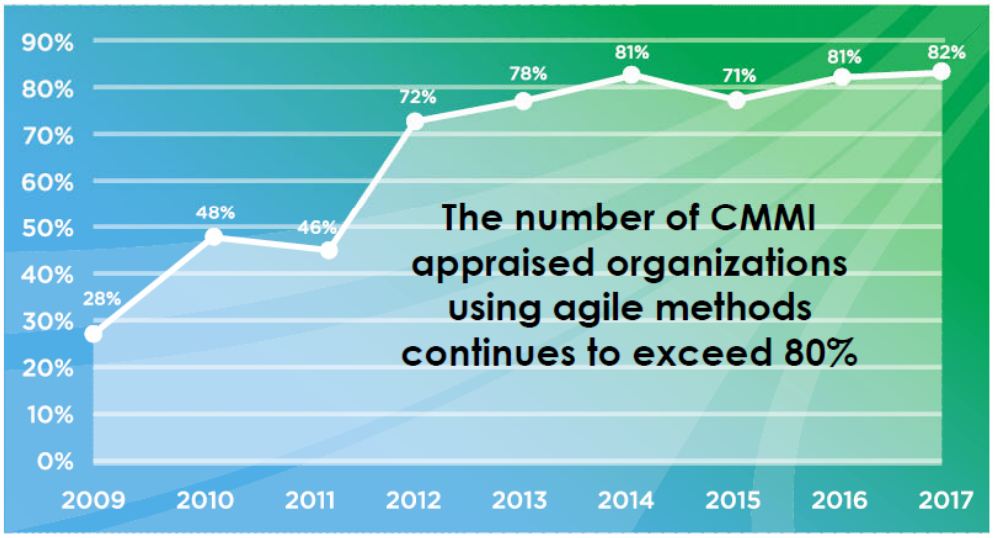
CMMI® Institute
AN ISACA ENTERPRISE

03

Build Agile Resiliency and Scale

CMMI V2.0 includes specific guidance to help organizations that use agile methods for development to strengthen their processes and scale their agile practices with a focus on performance.





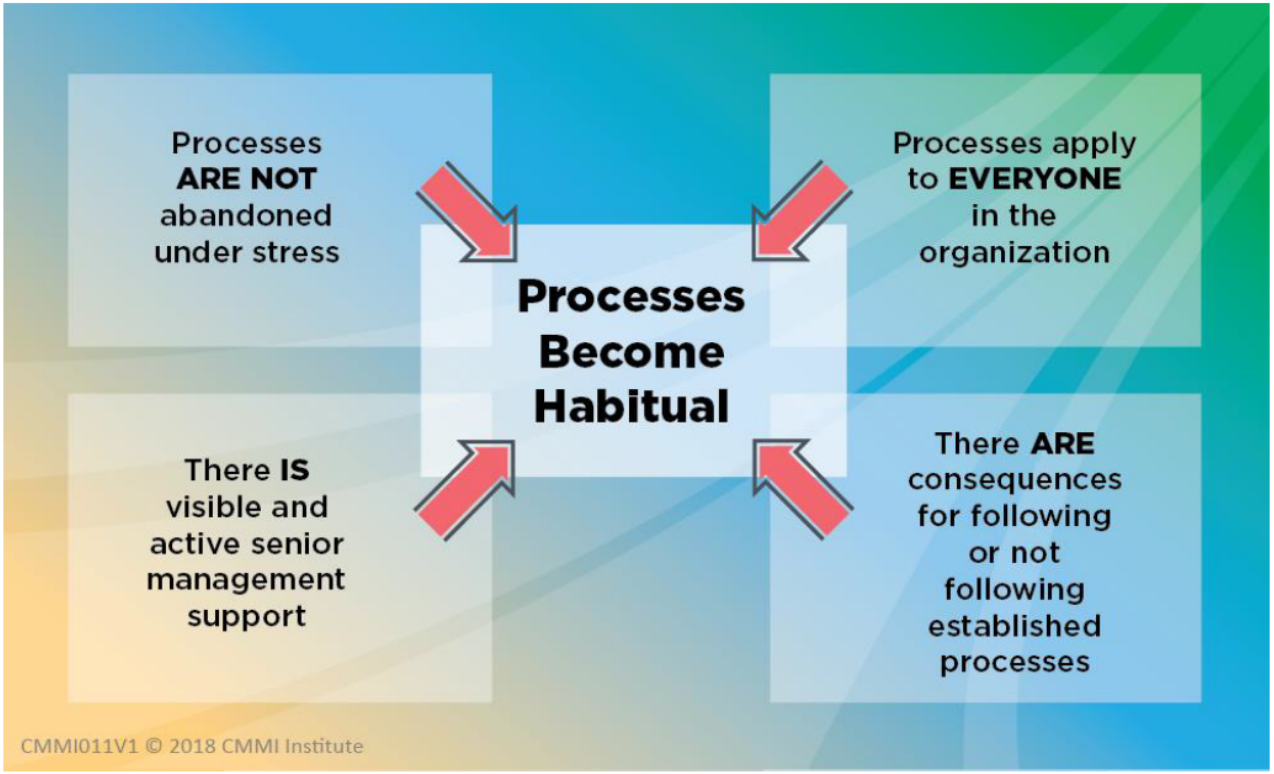
Sources:
HP online survey, 2015
CMMI Institute, Maturity Profile, 2017

CMMI® Institute
AN ISACA ENTERPRISE


Category
Capability Area
Practice Area
Doing
Managing
Enabling
Improving
Ensuring Quality
ENQ
Requirements Development & Management
RDM
Process Quality Assurance
PQA
Verification & Validation
VV
Peer Reviews
PR
Engineering & Developing Products
EDP
Technical Solution
TS
Product Integration
PI
Delivering & Managing Services
DMS
Service Delivery Management
SDM
Strategic Service Management
STSM
Selecting & Managing Suppliers
SMS
Supplier Source Selection
SSS
Supplier Agreement Management
SAM
Planning & Managing Work
PMW
Estimating
EST
Planning
PLAN
Monitor & Control
MC
Managing Business Resilience
MBR
Risk & Opportunity Management
RSK
Incident Resolution & Prevention
IRP
Continuity
CONT
Managing the Workforce
MWF
Organizational Training
OT
Supporting Implementation
SI
Causal Analysis & Resolution
CAR
Decision Analysis & Resolution
DAR
Configuration Management
CM
Sustaining Habit and Persistence
SHP
Governance
GOV
Implementation Infrastructure
II
Improving Performance
IMP
Process Management
PCM
Process Asset Development
PAD
Managing Performance & Measurement
MPM
CMMI009CV2 © 2018 CMMI Institute

SUSTAINING PERFORMANCE PERSISTENCE & HABIT

Persistence:
Firm or obstinate continuance in a course of action despite difficulty or opposition



Habit:
A settled or regular tendency or practice, especially one that is hard to give up

Process → Methodology

CMMI tells "what" to do but not necessarily "how"

- **remember that Process Areas and Goals are normative, Specific Practices are only expected**

Agile methods, e.g., Scrum or Extreme Programming, move much farther into "how" to build software.

- **oriented towards programming and software projects**

The judgment issues lie on whether, and how comprehensively, the agile practices implement the CMMI requirements…

Topics

A Context for Process Discipline

A Context for Agile Methods

Agile and Disciplined Processes

Closing Out…

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Principles Behind the Agile Manifesto

1) **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
2) **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
3) **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
4) **Business people and developers must work together daily throughout the project.**
5) **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**
6) **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
7) **Working software is the primary measure of progress.**
8) **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
9) **Continuous attention to technical excellence and good design enhances agility.**
10) **Simplicity – the art of maximizing the amount of work not done – is essential.**
11) **The best architectures, requirements, and designs emerge from self-organizing teams.**
12) **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**

What Is An "Agile Method"?

A software engineering "methodology" that follows the Agile Manifesto?

A method that supports responding rapidly to changing requirements?

- **Mark Paulk**

Does an agile method necessarily imply

- **Evolutionary / iterative / incremental development?**
- **Empowerment / participation of the development team?**
- **Active collaboration with the customer?**
- **…**

Agile Software Development

A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

"Standard Glossary of Terms Used in Software Testing, Version 2.4," ISTQB, 2014.

Sweet Spots for Agile

Dedicated developers
- 100% of time to project

Experienced developers
- 5 years / 10,000 hours

Small, co-located team
- less than 10
- visible, audible, isolated

Automated regression tests
- essential for continuous integration

Easy access to users
- talk to customer / users on day-to-day basis

Short increments and frequent delivery to real users
- two week iteration is most common agile iteration

Important Agile Methods

Scrum

Extreme Programming (XP)

Crystal Light Methods, specifically Crystal Clear

Lean Development

Kanban

Feature Driven Development (FDD)

and so forth…

Setting the Context

What defines an agile method such as Extreme Programming?

- **Do you have to do all 12 practices all the time?**
- **How much can you tailor the 12 practices and still call what you're doing "XP"?**

What is a "disciplined" process?

- **Communicate what you're going to do…**
- **Do what you said… consistently…**
  - **Even when there are contrary external pressures…**
  - **While keeping flexibility if you really need to change course…**
  - **And leaving a place for continual improvement…**

Scrum

A process for incrementally building software in complex environments.

Backlog – all outstanding work for a product area

Sprints – 30-day increments of work that produce a deliverable

Scrums – daily status check meetings

K. Schwaber, Agile Project Management with Scrum, 2004.
http://www.controlchaos.com

Scrum Roles, Ceremonies, and Artifacts

Scrum roles
- **Product Owner**
- **ScrumMaster**
- **Development Team**

Scrum ceremonies
- **Sprint Planning Meeting**
- **Daily Scrum Meeting**
- **Sprint Review Meeting**
- **Sprint Retrospective**

Scrum artifacts
- **Product Backlog**
- **Sprint Backlog**
- **Burndown Chart (Sprint and Release)**

No "Best Practice"

Cohn: There is not and never will be a list of "Scrum Best Practices" because team and project context trump all other considerations. Instead of best practices, what we need to know are good practices and the contexts in which they are successful.

Schwaber: Scrum is not a methodology, it is a framework.

The strength and pain of Scrum is that you are forced to adapt it to your specific situation.

H. Kniberg, Scrum and XP from the Trenches, 2007.

Extreme Programming Practices

Planning game

Small releases

Metaphor

Simple design

Testing
- **(test-driven development)**
- **(customer tests)**

Refactoring
- **(design improvement)**

Pair programming

Collective (code) ownership

Continuous integration

40-hour week
- **(sustainable pace)**

On-site customer
- **(whole team)**

Coding standard

Kent Beck, Extreme Programming Explained: Embrace Change, 1999.
(http://www.xprogramming.com/xpmag/whatisxp.htm)

The 2004 XP Practices

Primary Practices

- Sit together
- Whole team
- Informative workspace
- Energized work
- Pair programming
- Stories
- Weekly cycle
- Quarterly cycle
- Slack
- Ten-minute build
- Continuous integration
- Test-first programming
- Incremental design

Corollary Practices

- Real customer involvement
- Incremental deployment
- Team continuity
- Shrinking teams
- Root-cause analysis
- Shared code
- Code and tests
- Single code base
- Daily deployment
- Negotiated scope contract
- Pay-per-use

K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd Edition, 2004.

Why Embrace Agility?

Agility helps us manage change and uncertainty…

Change does not yield to rational analysis; responding to change requires a leap of faith…

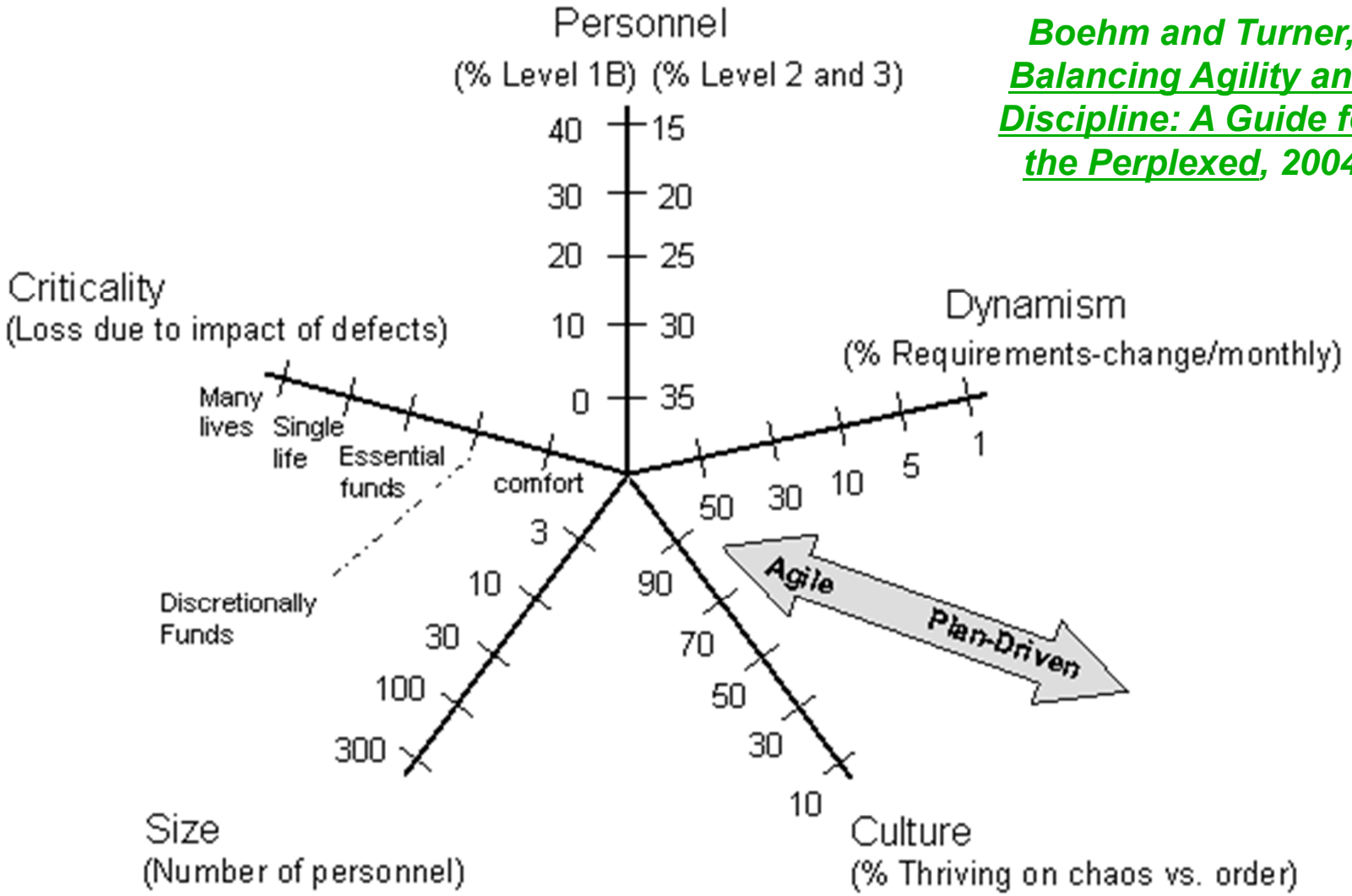
There is never one obvious option, but a multitude of options that seem reasonable.

There is never enough information, and the information in hand is often contradictory.

Change creates ambiguity, uncertainty, doubt, and indecision that lead to floundering.

J. Highsmith, "Agile Leadership: Focus and Clarity," Agile Product & Project Management E-Mail Advisor, 3 June 2010.

Agile vs Plan-Driven Tradeoffs



Boehm and Turner, Balancing Agility and Discipline: A Guide for the Perplexed, 2004.

No Silver Bullet

Scrum will not solve your problems.

Scrum will make your problems visible.

You will have to solve your problems.

Topics

A Context for Process Discipline

A Context for Agile Methods

Agile and Disciplined Processes

Closing Out…

Aligning Agility and Discipline


Early and continuous delivery


Welcome changing requirements ***(RM)***


Deliver working software


Work with business people ***(CMMI / IPPD)***


Motivated individuals with support they need ***(People CMM)***


Face-to-face conversation


Working software is the primary measure of progress


Sustainable development, constant pace ***(People CMM)***


Continuous attention to technical excellence and good design **(CMM)**


Simplicity


Self-organizing teams ***(CMMI / IPPD)***


Reflect on how to become more effective, adjust behavior **(CMM)**

Integrating Agile Into CMMI

CMMI provides a…

- **systematic organizational approach to improvement**
- **framework for managing change across an organization**
- **commitment to disciplined "engineering"**

Agile methods provide

- **specific strategies, methods, and techniques for building software – an implementation**

To obtain the benefits of an agile approach, you have to adopt and consistently implement the method!

→ a disciplined process for doing agile

CMMI Challenges to Agility

CMMI-DEV v1.3 specifies several engineering goals and practices that may be challenging in an agile context.

RD SG2 (develop product requirements)
RD SG3 (analyze and validate requirements)
TS SG1 (select product-component solutions)
TS SG2 (develop the design)

… but the problem may be more in aligning the culture of the customer with agile principles…

Organization-level processes are out-of-scope for agile methods, thus the value of CMMI and similar organizational transformation frameworks.

Cultural Misfits
(Using the DoD as an Example…)

Regulatory requirements for a level playing field raise challenges for evolutionary and incremental development…

The need by the contracts officer for a requirements specification…

Progress payments defined from a waterfall mentality…

Barriers – regulatory and cultural – to a collaborative customer relationship…

Protests from competitors…

Addressing the Cultural Mismatch

M.C. Paulk, "Agile Methodologies and Process Discipline," *Crosstalk: the Journal of Defense Software Engineering*, Vol. 15, No. 10, October 2002.

M.A. Lapham, R. Williams, C. Hammons, D. Burton, and A. Schenker, "Considerations for Using Agile in DoD Acquisition," CMU/SEI-2010-TN-002, April 2010.

M.A. Lapham, S. Miller, L. Adams, N Brown, B. Hackemack, C. Hammons, L. Levine, A. Schenker, "Agile Methods: Selected DoD Management and Acquisition Concerns," CMU/SEI-2011-TN-002, October 2011.

M.S. Palmquist, M.A. Lapham, S. Miller, T. Chick, and I. Ozkaya, "Parallel Worlds: Agile and Waterfall Differences and Similarities," CMU/SEI-2013-TN-021, October 2013.

E. Wrubel, S. Miller, M.A. Lapham, and T.A. Chick, "Agile Software Teams: How They Engage with Systems Engineering on DoD Acquisition Programs," CMU/SEI-2014-TN-013, July 2014.

E. Wrubel and J. Gross, "Contracting for Agile Software Development in the Department of Defense: An Introduction," CMU/SEI-2015-TN-006, August 2015.

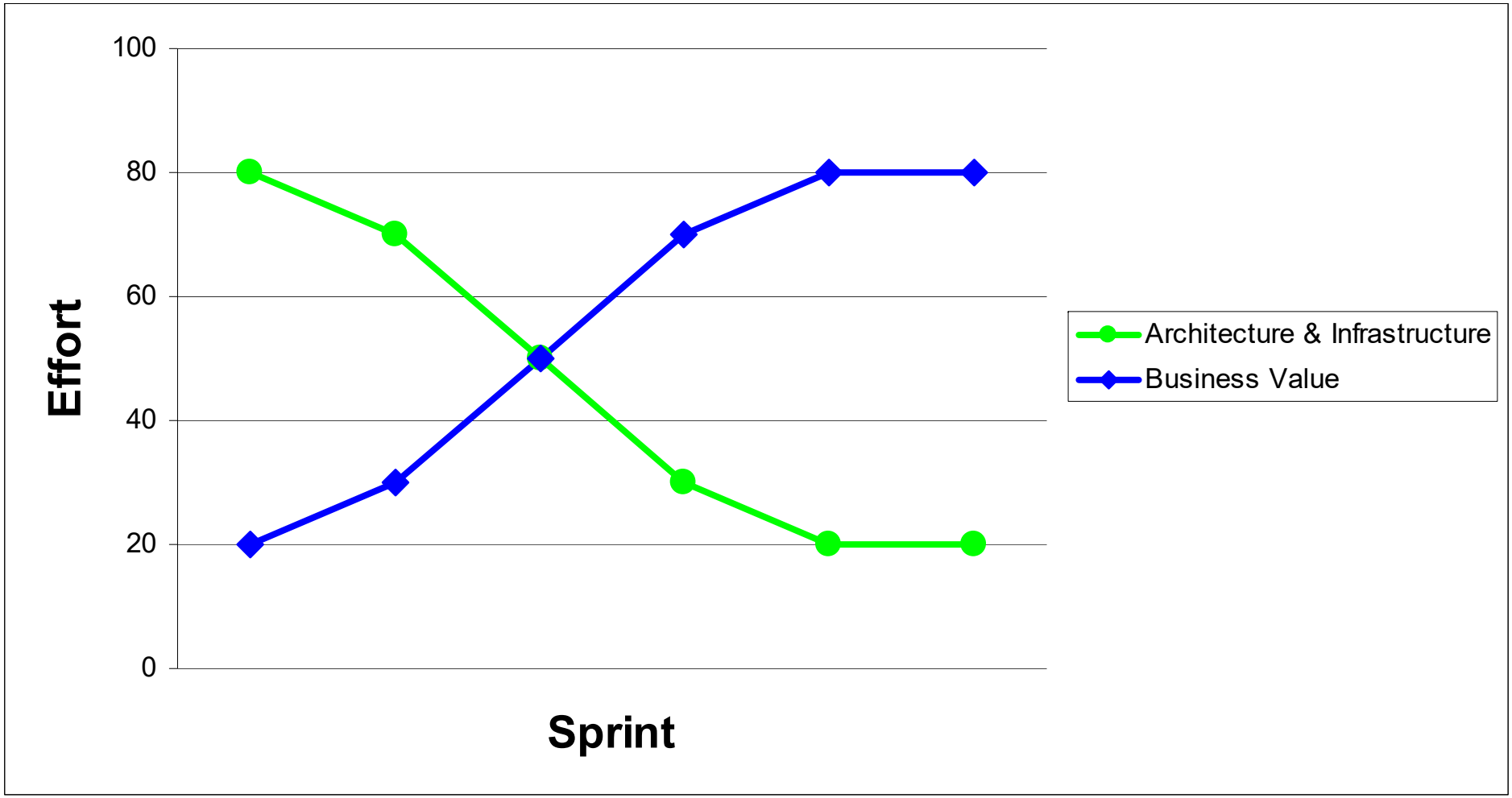
Architecture Breakers

For very large systems, our Pareto analysis of rework costs at TRW indicated that the 20% of the problems causing 80% of the rework came largely from "architecture-breakers."

Over-focus on early results in large systems can lead to major rework when the architecture doesn't scale up.

B. Boehm, "Get Ready for Agile Methods, With Care," IEEE Computer, January 2002.

Building the Foundation (Schwaber)



Barriers to the Success of Agile

A customer who insists on the big specification…

A culture that requires long hours to prove commitment…

Projects that are too big (more than about ten programmers)…

An environment with a long time to gain feedback (e.g., realistically test the software)…

The wrong physical environment (e.g., team members on different floors, not co-located)…

We do "agile," just not most (any) of the practices…

Topics

A Context for Process Discipline

A Context for Agile Methods

Agile and Disciplined Processes

Closing Out…

Process Management and the Known

Management must deal with both the *known* and the *unknown*.

***Process management* focuses on the known, on controlling repeatable (if not repetitive) processes.**

- **mechanisms for managing the known include quality assurance, configuration management, peer reviews, etc.**

***Risk management* focuses on the unknown.**

- **mechanisms for managing the unknown include evolutionary and incremental life cycles, on-site customers, prototyping, etc.**

Agile methods emphasize dealing with the unknown and volatility.

Disciplined processes avoiding known problems.

Caveats

Agile methodologies should not be used without tailoring

- **for life-critical or high-reliability systems**
- **by large and/or distributed, virtual teams**

Agile methodologies can be tailored and "improved" for different environments…

- **consider extensions to deal with "-ilities" as needed for particular domains; for example, safety, reliability, and security**

… but will the emergent properties that provide value in the "agile" context still emerge?

Questions and Answers