Agile Development

Karen Anderson Cianci
Vice President
Deputy Operations Manager

21 April 2009

IISBU
Excellent People Excellent Solutions

SAIC
From Science to Solutions

Agenda

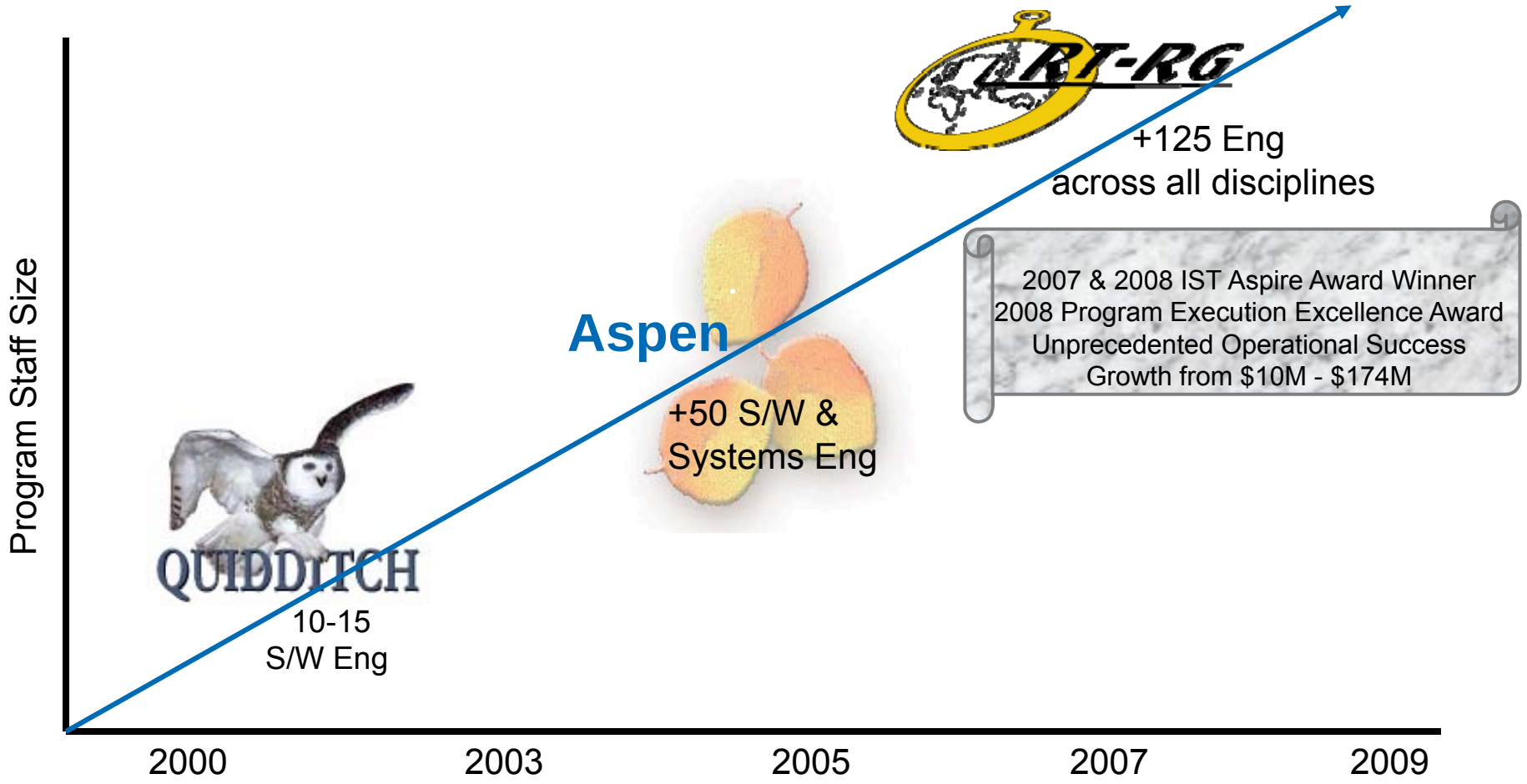
- Agile Evolution on engineering projects
- Agile Defined
- Project Complexities
- Agile Leadership
- Agile Success
- Agile's New Frontier

> ***Don't tell people how to do things, tell them what to do and let them surprise you with their results.***
>
> *- George S. Patton*

Key Take Away

Agile is the most effective way to quickly develop innovative software/products in complex, uncertain, and changing environments.

Agile Evolution in IISBU

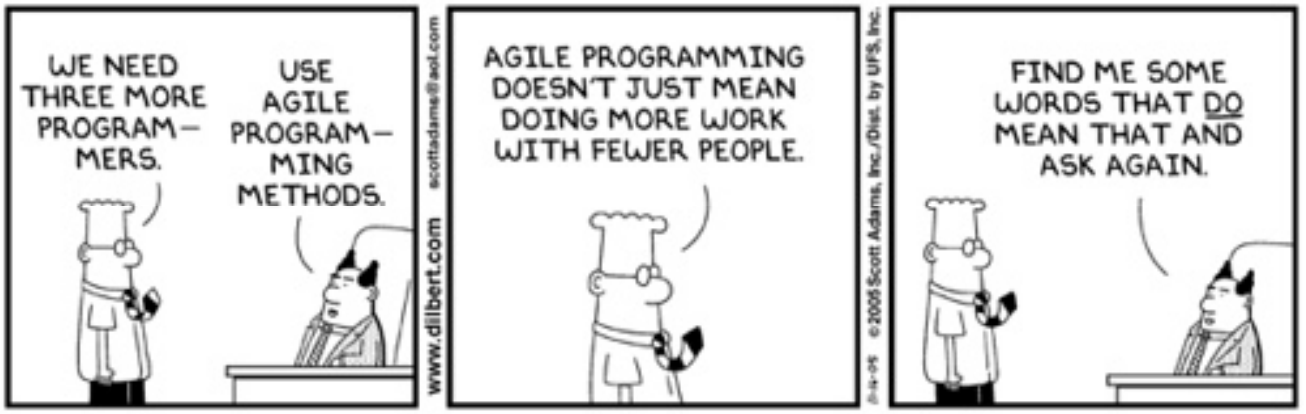


Agile Defined

We are what we repeatedly do.
Excellence, then, is not an act, but a habit.
-Aristotle

Ready, fire, aim (the agile approach to software development).
Ready, aim, aim, aim, aim, ... (the alternative).
-Anonymous

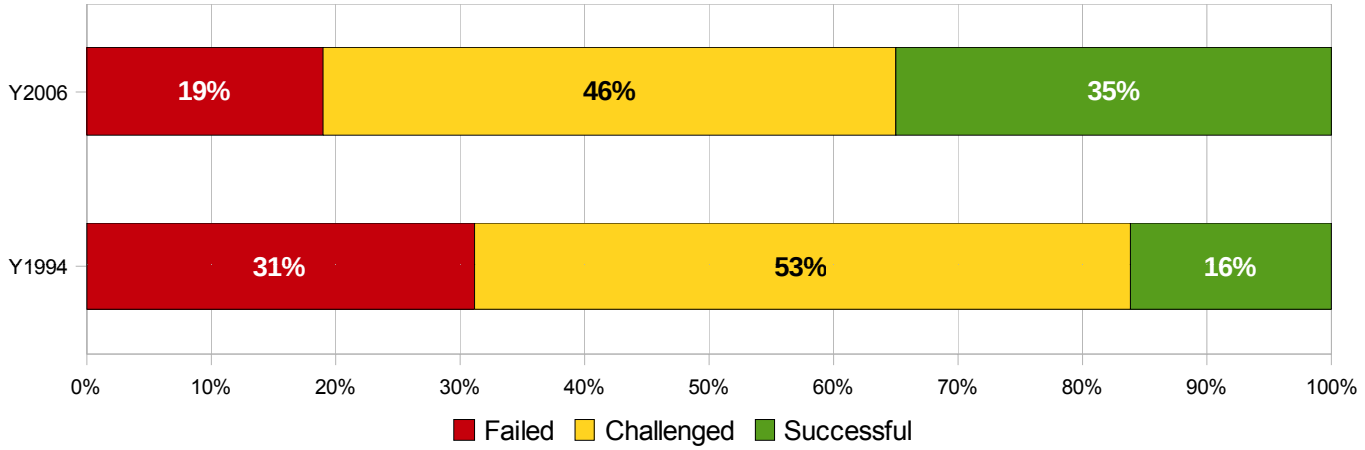
© Scott Adams, Inc./Dist. by UFS, Inc.



© Scott Adams, Inc./Dist. by UFS, Inc.

Who Uses Agile?

The CHAOS Chronicles



- 64% of features built rarely or never used
- 20-40% of COTS products never deployed
- Software value per $1 spent: 59 cents

"The Chaos Chronicles" 1994, 2006 The Standish Group

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

While there is value in the items on the right, we value the items on the left more."

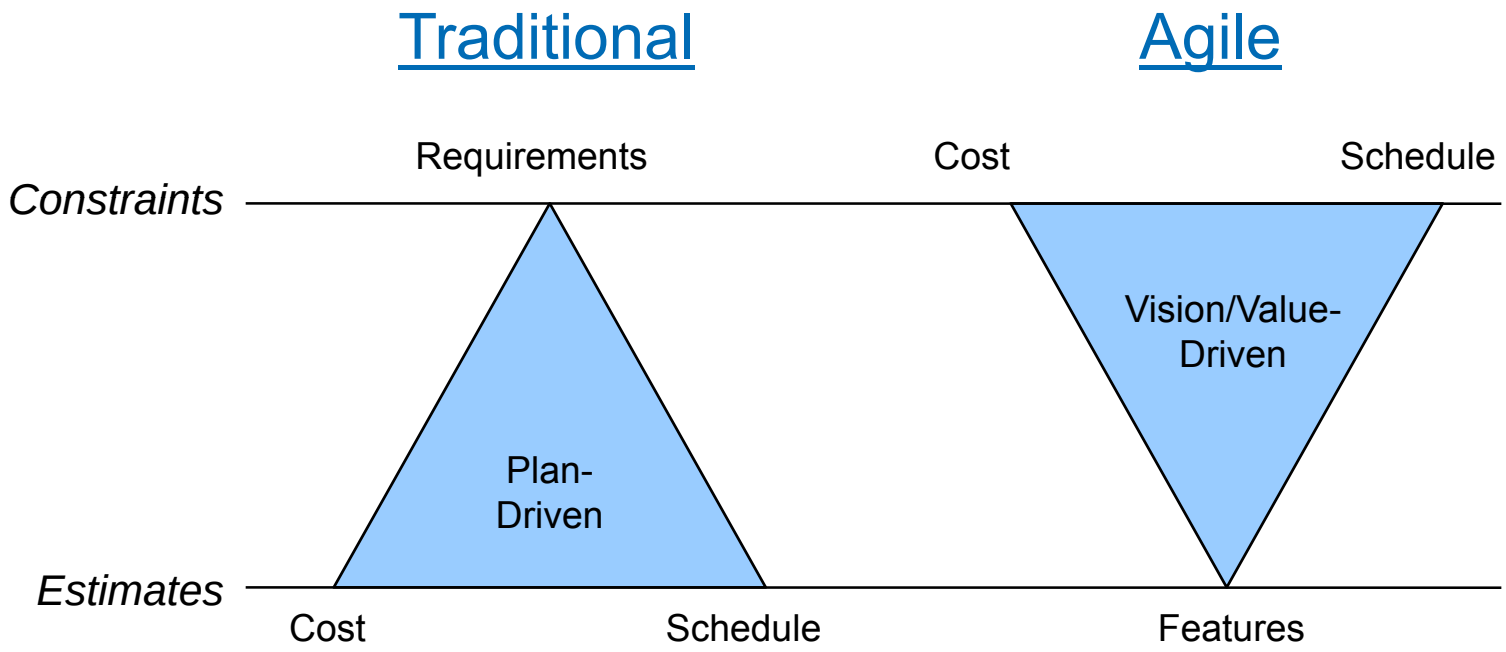
Signed in 2001 by the original 17 members of the "Agile Alliance"

Accepting Change as Inevitable

"Agility is the ability to both create and respond to change in order to profit in a turbulent business environment."-Jim Highsmith, Agile Software Development Ecosystems

- Nimbleness and flexibility **balanced with structure**
- Agile ≠ undisciplined
- Identify a few key practices (rules); let them evolve to meet specific problems
  - Frequent inspection, adaptation, and reprioritization
  - Emergence of requirements, technology, and team capabilities
  - Team empowerment - Team commitment and autonomy to self organize to succeed
  - Open workspaces, close contact, face to face communication, white boards
  - Good engineering practice -Development tools and environment, standards, feature or test-driven development
  - Documentation – necessary and sufficient
- Emphasize simplicity and the "***art of the possible***"

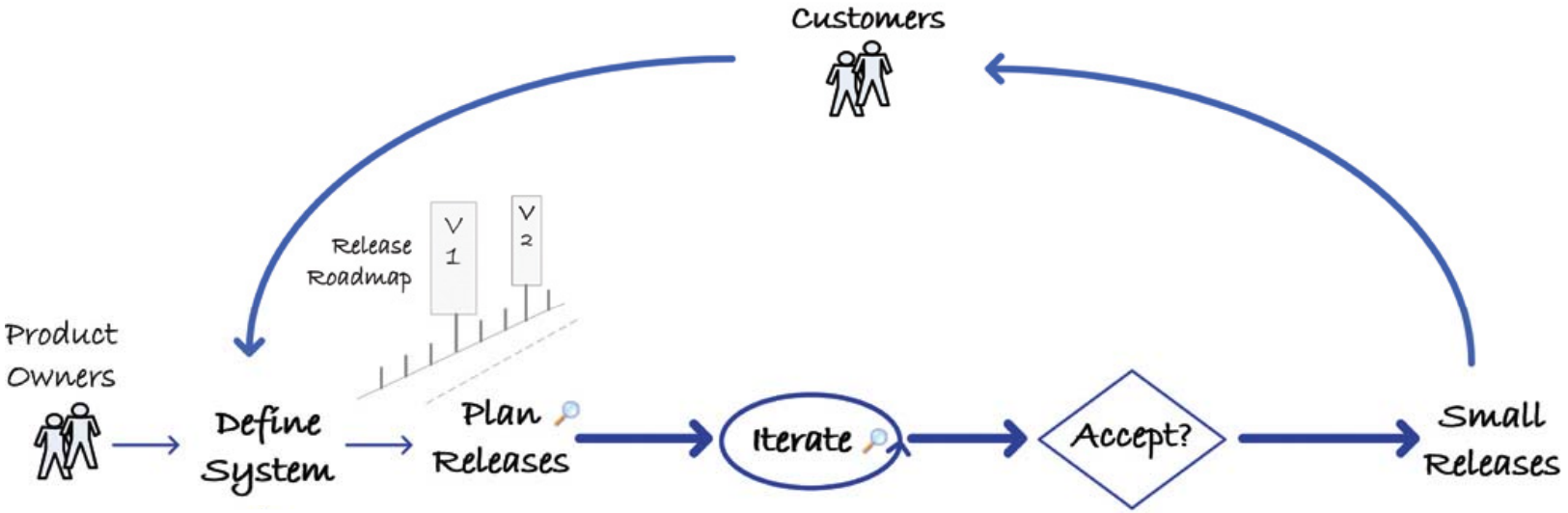
The Agile Paradigm Shift



Agile Methods

- Dynamic System Development Model – Faulkner
- Adaptive Software Development – Highsmith
- Crystal Methods – Cockburn
- SCRUM – Schwaber/Sutherland
- XP - Beck
- Lean Software Development – Poppendieck
- Feature-Driven Development – Code/DeLuca

SCRUM Rhythm



Release N: Theme

- **Feature 1**
- **Feature 2**
- **Feature 3**

R

Release

Backlog	Iteration 1	Iteration 2	Iteration 3	Iteration …	Backlog
• Feature 1 • Feature 2 • Feature 3 • Feature …	• Feature 1 • Feature 2	• Feature 3 • Feature 4	• Feature 5 • Feature 6 • Feature 7	• Feature 8 • Feature 9 • Feature 10	• Feature 11 • Feature 12 • Feature …

IISBU Excellent People Excellent Solutions

SAIC From Science to Solutions

Agile Leadership

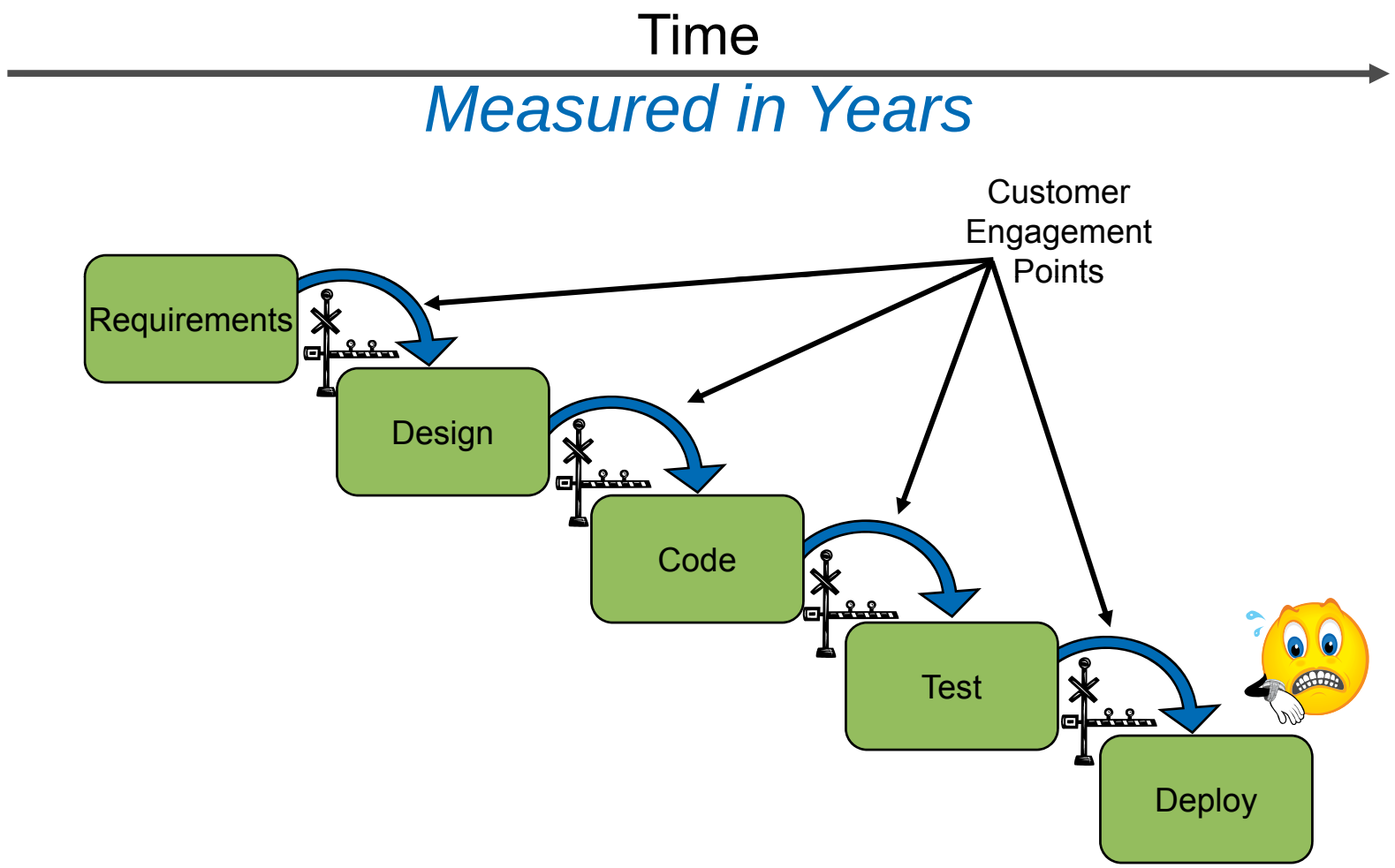
Traditional:

- **Plan what you expect to happen**
- **Enforce that what happens is the same as what is planned**
  - Directive management
  - Control, control, control
- **Use change control to manage change**
  - Change Control Board
  - Defect Management
- **100% requirement satisfaction**
- **Slow Decision Velocity**

Agile:

- **Plan what you expect to happen with detail appropriate to the horizon**
- **Control is through inspection and adaptation**
- **Use Agile practices to manage change:**
  - Continuous feedback loops
  - Iterative and incremental development
  - Prioritized backlogs
- **Willing to accept 80% functionality**
- **Rapid Decision Velocity**

Traditional Timeline

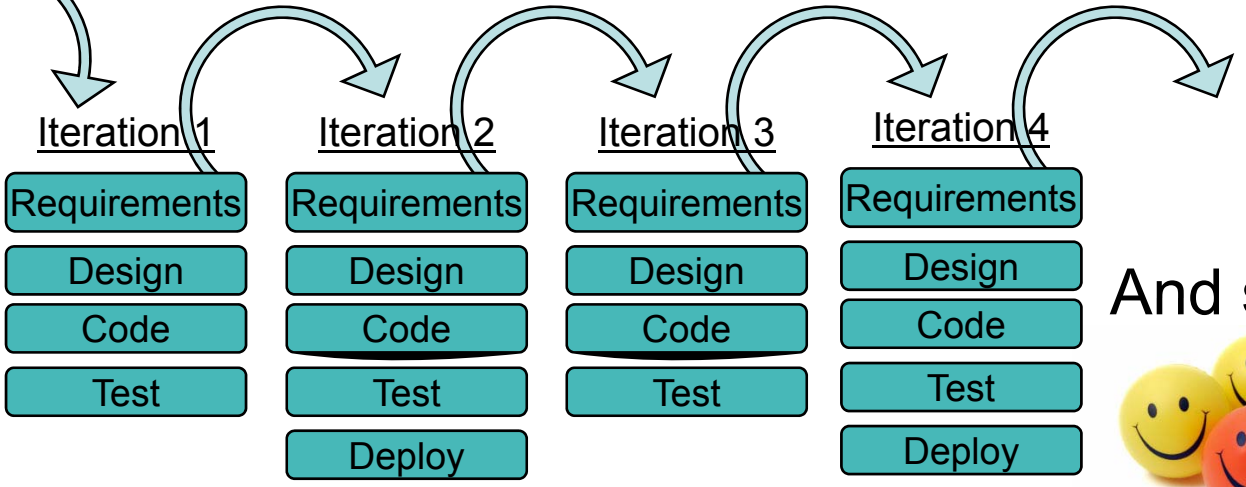


Agile Timeline

Time

Measured in Months

Continuous Customer Engagement



And so on ...

Project Complexities

Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction.

-A. Einstein

In theory, there is no difference between theory and practice, but not in practice.

-Anonymous

Defined vs. Complex Processes

- *Defined Process* - conforms to rules, linear, repeatable, predictable
  - Strict command and control can be used (e.g., Traditional approaches)
  - Example: Mowing the grass
- *Complex Process* - Non-linear, not-repeatable, defies prediction
  - Empirical controls are used instead (e.g., Agile approaches)
  - Example: Product development

"It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice."

- Process Dynamics, Modeling, & Control, Ogunnaike & Ray, Oxford University Press, 1992

Complexity of Requirements

- Capturing requirements all-at-once is very difficult:
  - The time and effort required to build what traditional methods demand is rarely available, or even possible for new systems
  - Disagreements within the customer may be hard to resolve
  - Committees: everybody gets in his “pet” requirement
- Incomprehensible requirements specifications:
  - Difficulties in communication - written requirements are often ambiguous, contradictory, and incomplete
- Requirements volatility:
  - Because understanding grows over time
  - Because priorities change

Complexity of Technology

- Today's systems are built on layers and layers of interacting technologies:
  - Incompatibilities may exist that must be overcome
  - Unexpected interactions are hard to find and fix
  - Bugs that are encountered in components you are using
- Competitive pressures keep us at the bleeding edge
- Uncertainty about the applicability of a technology:
  - How do you *know* the technology will actually work and help solve the problem until you get it working?
- Technological volatility:
  - Varying levels of familiarity by developers
  - Varying levels of product maturity

Examples of Technology Challenges (RT-RG)

- Innovative solutions through iterations:
  - Near real-time scalable database ingest
  - Real-time alert delivery to a web client
  - EITC Hardware Platform
- Unfamiliarity: Oracle RAC
- Hard problems: Blade Lockups

Complexity of People

- Guess what? It's people that build and use software!
  - This obvious fact is generally ignored - there is little literature about software development and people
  - But, the "people factor" that is the strongest indicator of project success or failure
- Customers:
  - Collaborative, Representative, Authorized, Committed, and Knowledgeable (CRACK)
- Developers:
  - Vary tremendously in ability, training, experience, interests and motivation
  - Team Dynamics: "Jelled" teams work best, but often hard to form, especially in difficult projects

Complexity of Communication

- Project requires continuous communication of dense, volatile content:
  - BUT, people communicate imprecisely
  - BUT, computers demand absolute precision
  - AND, it gets exponentially harder for larger teams
- Agile ways of improving communication:
  - Use the most appropriate mode of communication
  - Increase chances for information exchange
  - Repetition is good
  - Keep teams as small as possible, but no smaller
  - Use “information radiators” in work areas

Agile Communication (RT-RG)

- Sprint Prioritization & Planning
  - Determine the prioritized tasks for the next Sprint
  - Requires customer or customer proxy involvement
- Sprint Reviews
  - Demonstrate what has been built
  - Reflect on reasons why certain tasks were not completed
- Sprint Retrospectives
  - Provides meaningful feedback to leaders on what needs to change
  - Make sure to follow through on commitments
- Daily SCRUMS & SCRUM of SCRUMs
  - The 4 questions
- Weekly Customer Meetings
- Unplanned Team collaboration

Agile Leadership

Some Managers are More Than Bosses - They're Leaders, Too

- Horowitz

You cannot manage men into battle. You manage things, you lead people.

- Grace Murray Hopper

Own the Process

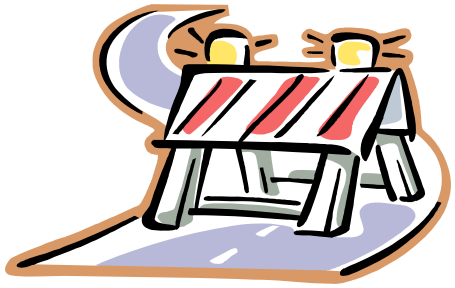
- **You must be the Agile knowledge repository so you can answer these questions from your team:**
  - "Why are we doing this again?"
  - "What's the point of having a <Stand-up, Demo, Release Planning Meeting, etc>?"
  - "Why can't I just do things the way I used to?"
  - "We're having a problem here. What does <Scrum, XP, etc.> say we're supposed to do now?"
  - "Why don't you just tell me what to do?"

Facilitate Team Discussions

- **Guide the team in planning and decision-making**
  - Keep the team focused
  - Make sure all voices are heard
- **Foster effective working relationships on your team**
  - Know how to untangle communication issues
  - Guide your team in navigating conflict and negotiation
    - Conflict is inevitable

Remove Roadblocks

- Maximize team productivity
- Represent the team to management
- Represent management to the team
- Improve the productivity of development team
  - in any way possible (and legal)
- Help the team establish working agreements
- Understand and work the collaboration continuum



Shift the Culture

- Redefine how you measure individual performance in an Agile cross-functional team
- Redefine a career path for your team members
- Redefine how individuals and teams are recognized and rewarded
- Create an environment that encourages experimentation and discovery
- Allow challenge and eliminate fear
- Be a living example

Empowerment Strategies

- Avoid competition for power, status, recognition
- Delegate
- Create and communicate a vision
- Insist that others diligently work to achieve meaningful goals
- Help others believe in their own worth and potential
- Create a culture in which fear and intimidation are replaced by trust
- Demonstrate a willingness to be supportive of others
- Places responsibility for spotting/solving problems on employees
- Requires leaders to ask for suggestions AND lets employees make decisions

Agile Success

The person who says it cannot be done should not interrupt the person doing it.

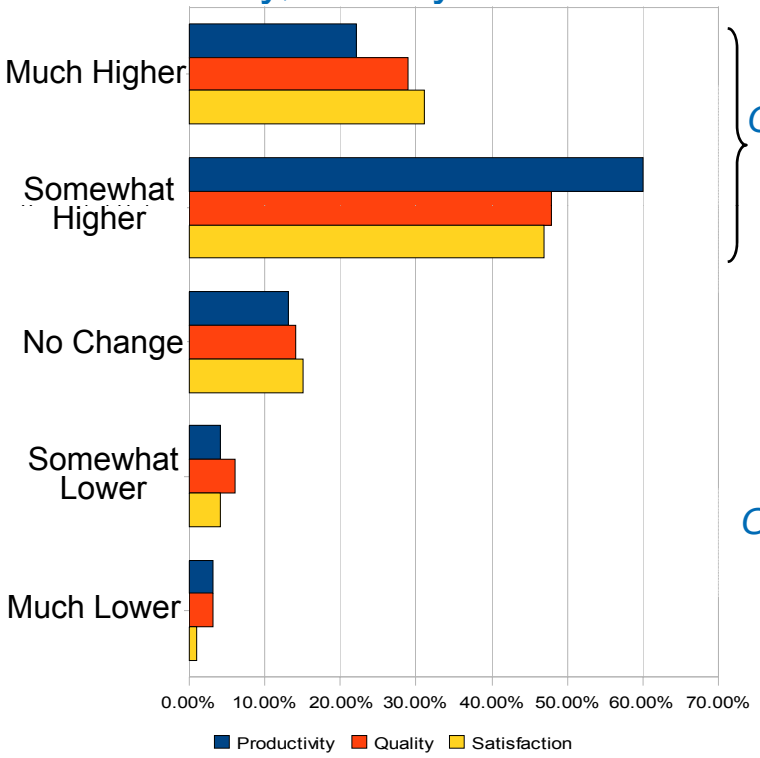
-Chinese Proverb

If at first you do succeed, try not to look too surprised.

-Anonymous

Success of Agile

Productivity, Quality & Satisfaction

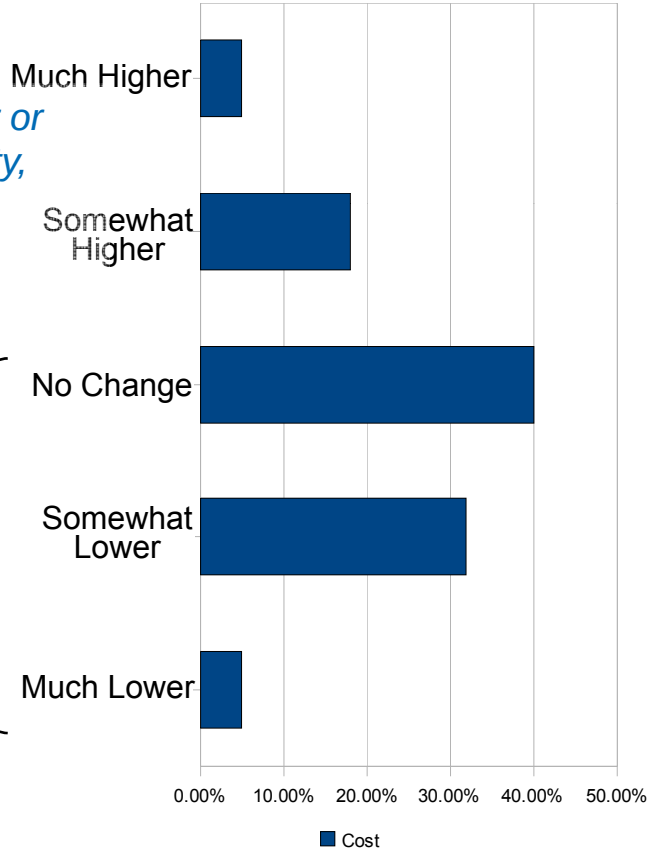


Over 70% report Much Higher or Somewhat Higher Productivity, Quality & Satisfaction (Hyper Productivity)

AND

Over 70% had no change or somewhat lower cost

Cost



Source: Scott W. Ambler* Surveys
www.ambysoft.com/surveys
**Original Signator of Agile Manifesto*

The Future of Agile

When it comes to the future, there are three kinds of people: those who let it happen, those who make it happen, and those who wonder what happened.

John M. Richardson

New Frontiers in Agile

- Large Scale Programs
- Distributed Teams
- Traditional Acquisition
- Traditional Engineering Disciplines (Systems Engineering, Test, Operations)
- Difficult/Critical Systems (DoD, FDA, CMMi, SOX, Banking)

Scaling Challenges

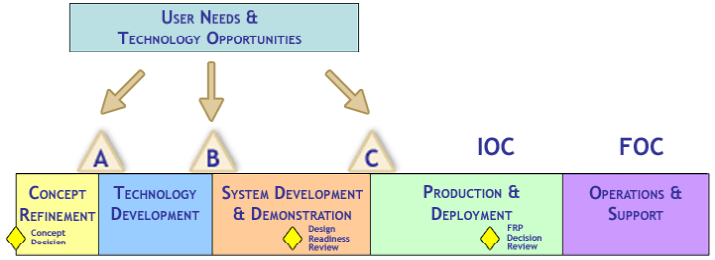
- Staffing
  - Only pick the best
  - Make sure they will fit with the philosophy – not everyone can adapt
  - Teams tend to self organize

- Communications Challenges
  - Small teams – Daily SCRUM will suffice
  - Large teams require more formal structure (SCRUM of SCRUMS, Domain-specific SCRUMs, All Hands, Retrospective, Roadmaps)
  - Multiple Forms of Communication – E-mail, Sharepoint, WIKI
  - Geographically Distributed Stakeholders
  - Adapt to the needs of the team

There is a loss of efficiency when scaling

Agile's Growing Influence

- PMI is offering Agile training, mappings exist for PMBOK areas
- Intelligence Community is beginning to embrace agile methodology due to proven success – Agile Acquisition Strategy
- DoD is starting to look at it as well, still have to resolve tension with traditional acquisition methodologies
- DoD 5000 Milestone C can be attained with Agile, it's all in how you sell it



- SEI starting to embrace Agile (SEI Report published November 2008: "CMMI® or Agile: Why Not Embrace Both!")

Suggested Resources

- www.agilealliance.org
- www.agilemanifesto.org
- www.controlchaos.com
- Agile Software Development With Scrum, Ken Schwaber
- Implementing Lean Software Development: From Concept to Cash, Mary Poppendieck
- Agile Project Management: Creating Innovative Products, Jim Highsmith
- Scaling Software Agility: Best Practices for Large Enterprises, Dean Leffingwell
- *"CMMI® or Agile: Why Not Embrace Both!"* Hillel Glazer, Jeff Dalton, David Anderson, Mike Konrad & Sandy Shrum
- *"Toward Agile Systems Engineering Processes"* Dr. Richard Turner, Systems and Software Consortium

Contact Information

- Bob McGinnis, VP, Senior Program Manager
  mcginnisro@saic.com 410.312.3741
- Karen Anderson Cianci, VP, Division Manager
  ciancik@saic.com 410.312.2066

Questions