

# Taking Advantage of Agile while Minimizing Risk: User Stories and Other Fables

Presenter: Dr. Dave Zubrow  
Deputy Chief Scientist  
Software Solution Division

AFEI Agile Summit  
November 20, 2013



# Report Documentation Page

Form Approved  
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

|   |                                    |                                     |                            |   |                                 |
|---|------------------------------------|-------------------------------------|----------------------------|---|---------------------------------|
| 1. REPORT DATE<br><b>20 NOV 2013</b>  |                                    | 2. REPORT TYPE                      |                            | 3. DATES COVERED<br><b>00-00-2013 to 00-00-2013</b> |                                 |
| 4. TITLE AND SUBTITLE<br><b>Taking Advantage of Agile while Minimizing Risk: User Stories and Other Fables</b>                              |                                    |                                     |                            | 5a. CONTRACT NUMBER                                 |                                 |
|   |                                    |                                     |                            | 5b. GRANT NUMBER                                    |                                 |
|   |                                    |                                     |                            | 5c. PROGRAM ELEMENT NUMBER                          |                                 |
| 6. AUTHOR(S)  |                                    |                                     |                            | 5d. PROJECT NUMBER                                  |                                 |
|   |                                    |                                     |                            | 5e. TASK NUMBER                                     |                                 |
|   |                                    |                                     |                            | 5f. WORK UNIT NUMBER                                |                                 |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br><b>Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213</b> |                                    |                                     |                            | 8. PERFORMING ORGANIZATION REPORT NUMBER            |                                 |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)   |                                    |                                     |                            | 10. SPONSOR/MONITOR'S ACRONYM(S)                    |                                 |
|   |                                    |                                     |                            | 11. SPONSOR/MONITOR'S REPORT NUMBER(S)              |                                 |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT<br><b>Approved for public release; distribution unlimited</b>                                       |                                    |                                     |                            |   |                                 |
| 13. SUPPLEMENTARY NOTES   |                                    |                                     |                            |   |                                 |
| 14. ABSTRACT  |                                    |                                     |                            |   |                                 |
| 15. SUBJECT TERMS   |                                    |                                     |                            |   |                                 |
| 16. SECURITY CLASSIFICATION OF:   |                                    |                                     | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES                                 | 19a. NAME OF RESPONSIBLE PERSON |
| a. REPORT<br><b>unclassified</b>  | b. ABSTRACT<br><b>unclassified</b> | c. THIS PAGE<br><b>unclassified</b> |                            |   |                                 |

**Copyright 2013 Carnegie Mellon University**

**This material is based upon work funded and supported by ODE under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.**

**Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of ODE or the United States Department of Defense.**

**NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.**

**This material has been approved for public release and unlimited distribution except as restricted below.**

**This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).**

**DM-0000763**



# Outline

**Just how good is Agile?**

- **The good, the bad, and the ok....**

**Is it right for all circumstances? If not, when?**

- **Not a silver bullet**

**Must it be done in a “pure” form? If not, what is gained and what is lost?**

- **Maybe so, maybe not**

**What must I do to be successful**

- **Some take aways**



# JUST HOW GOOD IS AGILE



# Empirical Studies on Agile

## Studies by research method\*

| Research method | Number | Percent |
|-----------------|--------|---------|
| – Single-case   | 13     | 39      |
| – Multiple-case | 11     | 33      |
| – Survey        | 4      | 12      |
| – Experiment    | 3      | 9       |
| – Mixed         | 2      | 6       |
| Total           | 33     | 100     |

## Company studies and case studies

- **Microsoft Research**
- **Virginia Polytechnical Institute PhD (2013)**

## Workshops and Cross company interviews

- **SEI Agile Collaborators Working Group**
- **NDIA/AFEI ADAPT**

## Industry-performed quantitative studies

- VersionOne surveys
- Rally Software Quantitative Analysis
- CAST CRASH Report

\*T. Dyba°, T. Dingsøyr. Empirical studies of agile software development: A systematic review, Information and Software Technology 50 (2008) 833–859



# It's a Journey.....

Patriot Excalibur started in 2003 and continues today....



**The Agile Journey of Patriot Excalibur (PEX)**

Acase of "Fortunate Serendipity"  
→ Reimbursable funding model  
→ Beneath the acquisition radar in the test wing

**Adopted XP in 2003**

- more SMEs
- more releases
- more functionality delivered

Went from 20 to 200 adapters in ONE YEAR!

**2007 Agile Struggles**

- More demand led to a bigger team
- 20 on a team was too big!
- "traditional" smells crept in

THAT doesn't smell like the near future!

**2008 Agile ReOrg 1**

- Quasi-Scrum model
- Independent teams
- More SMEs
- Add in SCM

★ MISSION REASSIGNED TO AFLCMC "Agile Evangelists"

- Accredited as a "software program" (like MS Word) instead of a system

**2009/10 Expansion**

- Demand increased to 600 adapters
- Increased # of teams ~ 100 people
- Architecture ⇒ SOA
- Geographically distributed team
- Added automated testing team

**2010 Difficulty Scaling**

- 7 week test event @ end of iterations
- teams still very independent
- introduced technical debt that built up

**2011 Agile ReOrg 2**

- Adopted Scrum "by the book"
- Product Owner Team
- Embedded Tester
- More automation
- Integrated Design Teams
- Stabilized architecture
- @ end of iteration

DEFINITION OF DONE

Red light ~~Green light~~

**2012-2013 Scrum**

- Host @ 16 locations
- Every MATCOM uses it
- Moved to GSA TEM contract - SWdev as a service
- Single code base still a key
- Added specialized skill of security
- Multiple technologies side-by-side: Web & Client Server ~ 2 million SLOC
- AFLCMC Support

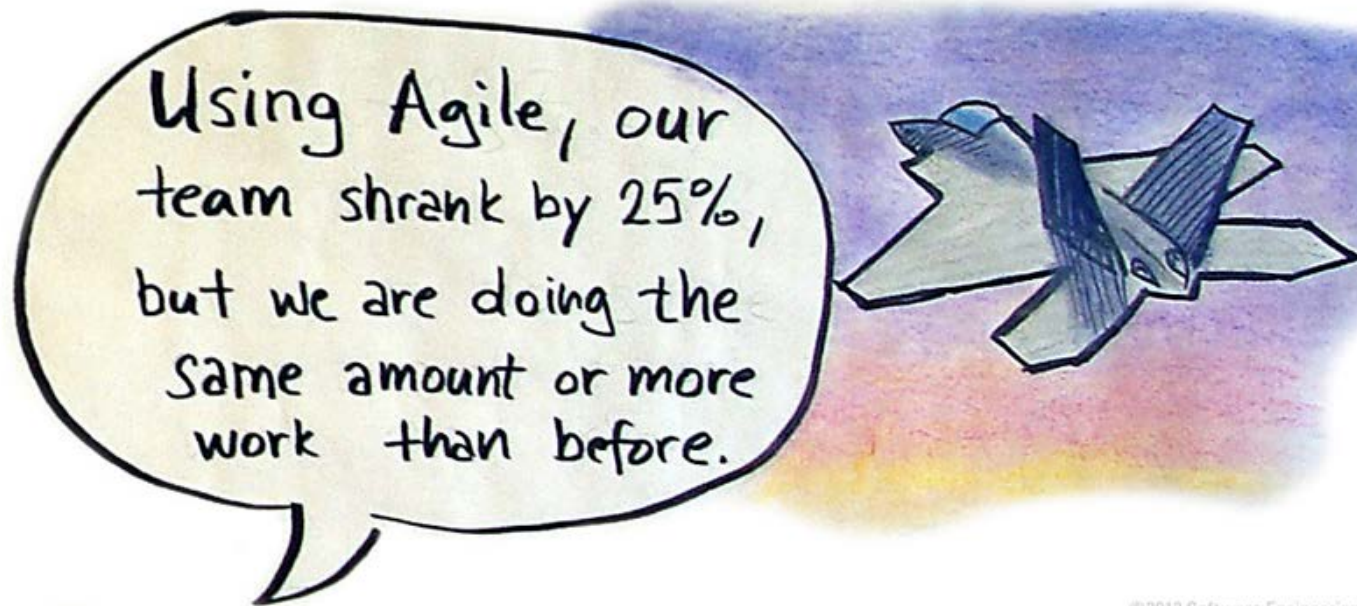
**Open Issues**

- Architecture Committee is inefficient
- Long-range planning
- 7-week system test at the end
- Reliable funding stream
- Code quality & Test Coverage

©2013 Software Engineering Institute



# Agile Delivers Even with Smaller Team

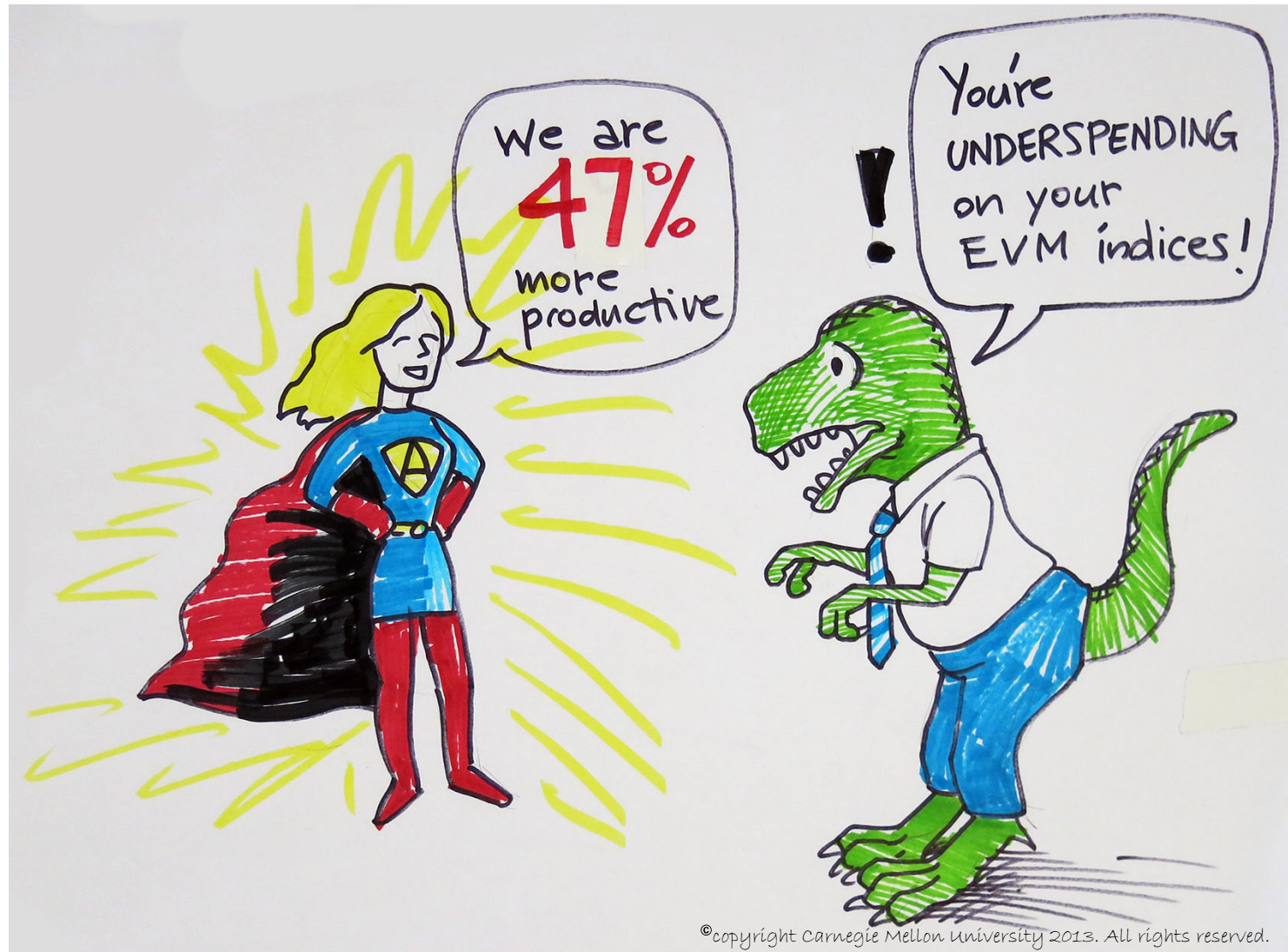


©2012 Software Engineering Institute





# Agile Success Comes with Challenges



# CRASH Report - 2011/12 • Summary of Key Findings

Figure 16b. Transferability Scores by Development Methods

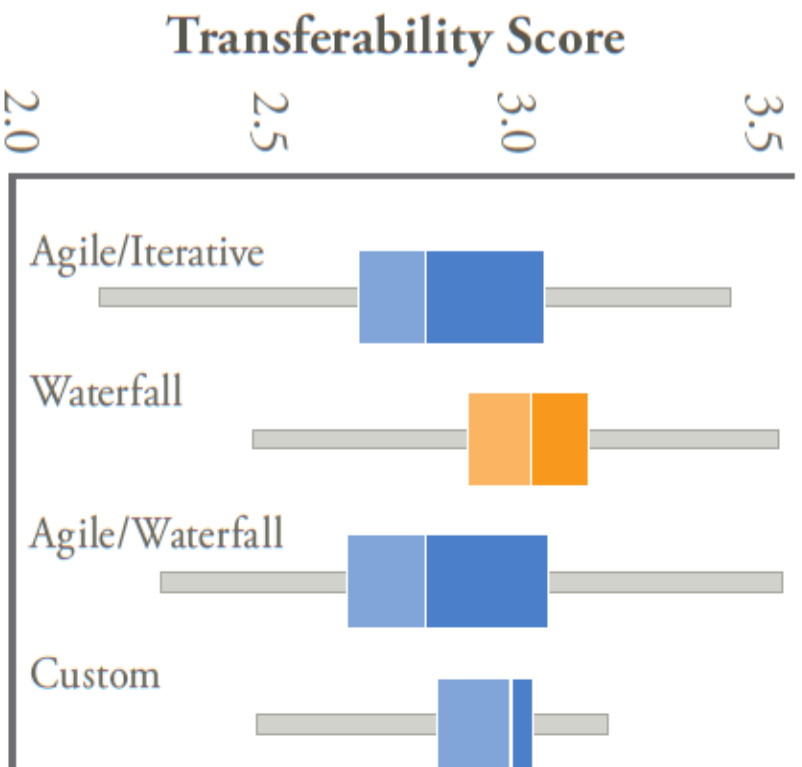
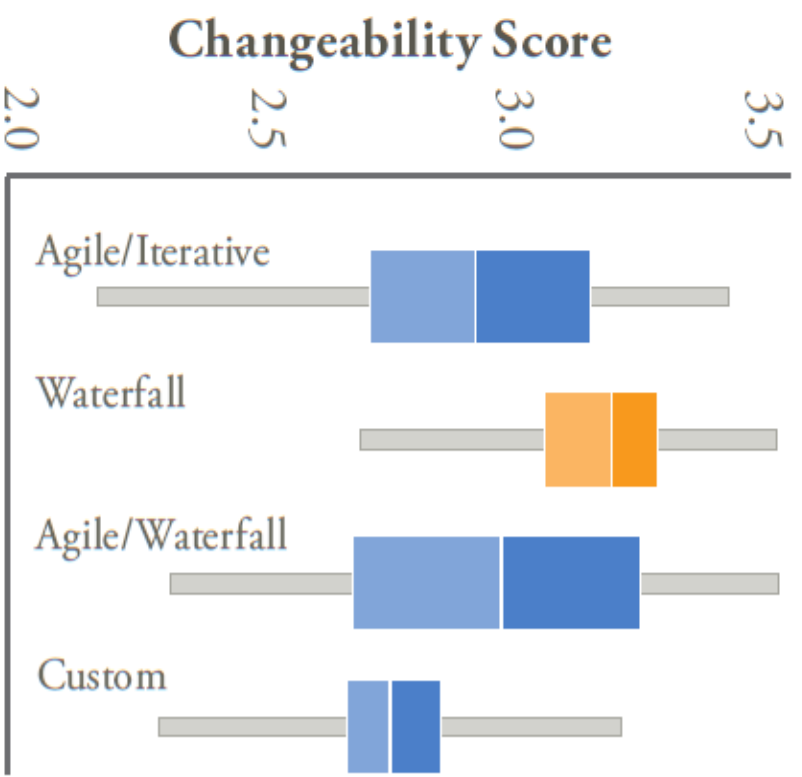


Figure 16c. Changeability Scores by Development Methods

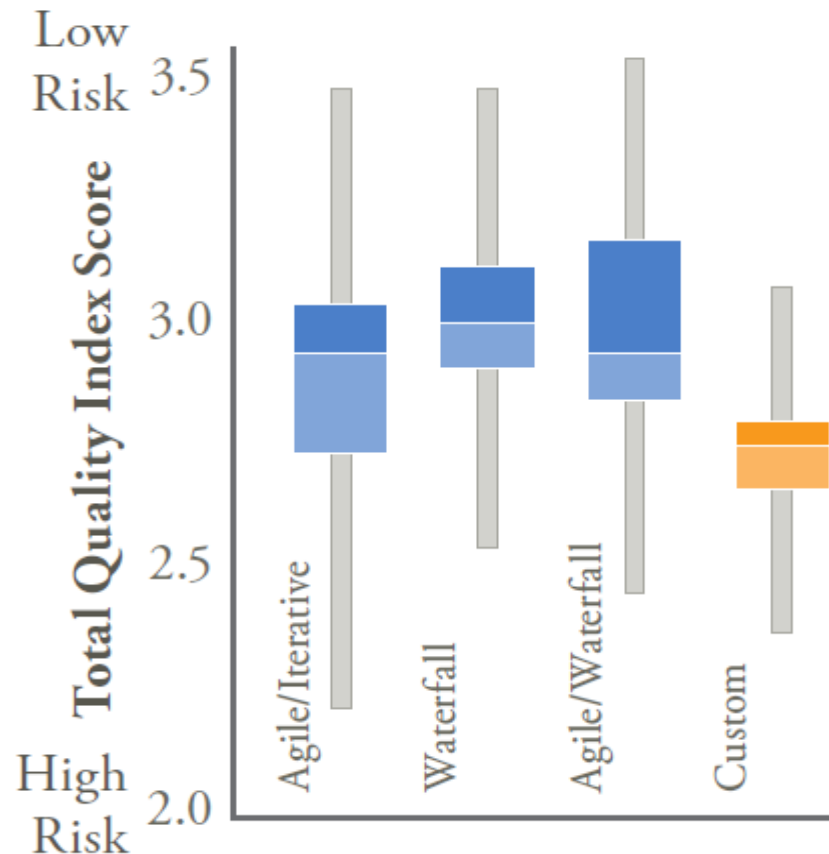


# CRASH Report - 2011/12 • Summary of Key Findings

Figure 16a. Total Quality Index Scores by Development Methods

## Finding 6—Development Methods Affect Structural Quality

- agile methods are nearly as effective as waterfall at managing the structural quality affecting business risk (Robustness, Performance, and Security)
- less so at managing the structural quality factors affecting cost (Transferability and Changeability)



CAST Report on Application Software Health ([research.castsoftware.com](http://research.castsoftware.com))



# Pitfalls of (Agile) Measurement

## The Seven Deadly Sins of Agile Measurement

|   | Deadly Sin  | Heavenly Virtue  |
|---|---|--|
| 1 | Using metrics as levers to change someone else's behavior | Using metrics for feedback to improve your own performance                                     |
| 2 | Unbalanced metrics  | Day-one have one metric from each quadrant   |
| 3 | Believing metrics can replace thinking                    | Use quantitative insight to complement rather than replace qualitative insight                 |
| 4 | Too costly metrics  | Favor automatic metrics from passively acquired data or lightweight surveys                    |
| 5 | Using a convenient metric                                 | Use ODIM to determine metrics that provide critical insight and drive to your desired outcomes |
| 6 | Using bad analysis  | Get your statistics right by consulting experts  |
| 7 | Forecasting without discussing probability                | Use the percentile coverage distribution, the cone of uncertainty, or Monte Carlo simulation   |



#RallyON13

@LMaccherone

[LMaccherone@rallydev.com](mailto:LMaccherone@rallydev.com)

©2013 Rally Software Development Corp



# Some Common Agile Myths

## Mythbusters

| Myth                                    | Responsiveness  | Quality | Predictability          | Productivity |
|---|---|---------|-------------------------|--------------|
| Points + hours better than points alone | Red   | Green   | Red                     | Red          |
| Dedicate to one team                    | Green   | Green   | Green                   | Green        |
| Keep the teams stable                   | Green   | Green   | Green                   | Green        |
| Lower WIP is always better              | Green   | Green   | Red: WIP can be too low |              |
| Ideal team size: 5-9                    | Red: Smaller performs worse but larger about the same |         |                         |              |
| Kanban is better than Scrum             | Grey  | Red     | Grey                    | Grey         |
| ScrumBan is the best of both worlds     | Grey  | Red     | Grey                    | Green        |

|                    |       |
|--------------------|-------|
| Busted             | Red   |
| Confirmed          | Green |
| Minimal difference | Grey  |

 #RallyON13 
  @LMaccherone 
 [LMaccherone@rallydev.com](mailto:LMaccherone@rallydev.com) 
 ©2013 Rally Software Development Corp



# Federal Challenges in Applying Agile

GAO 2012 report of experiences in 5 agencies

- 32 Agile practices identified for consideration
- 10 practices were used and deemed effective
- 14 challenges were identified reflecting on the need to transition
  - Team transition issues
  - Guidance and adoption of tools were difficult
  - Agency commitment of staff
  - Customer trust of iterative solutions
  - Adapting to iteration time frames was difficult
  - Federal reporting and reviews not aligned with Agile



# IS AGILE RIGHT FOR ALL?



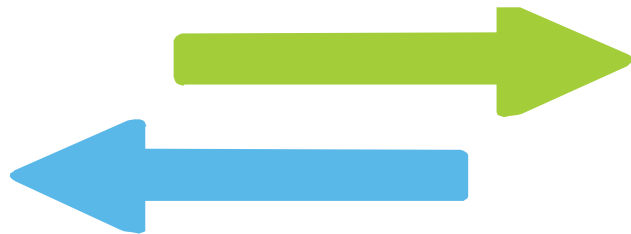
# Dynamic Environments - Traditional versus Agile Worlds

.....the Traditional World struggles to deliver as it constantly looks back at long-fixed requirements and priorities.

- If requirements are stable, then safer and more prudent to use waterfall

.....the Agile World adapts as it delivers by constantly looking forward at evolving requirements and priorities.

- In settings with significant operational or technology dynamism, the Agile methods are an advantage

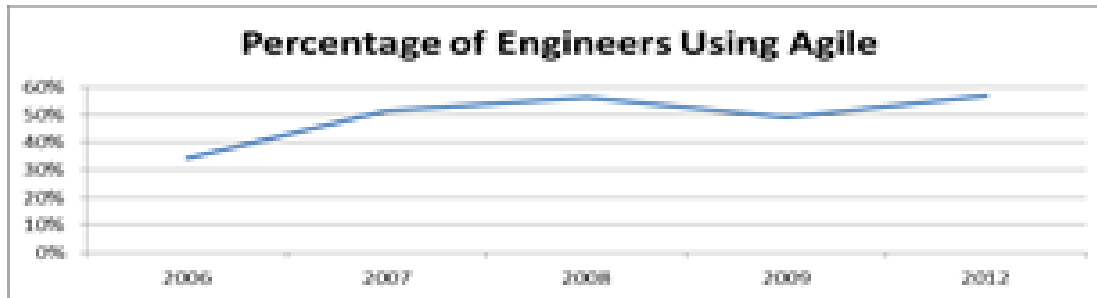


Parallel Worlds: Agile and Waterfall Differences and Similarities (CMU/SEI-2013-TN-021).  
<http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=62901>

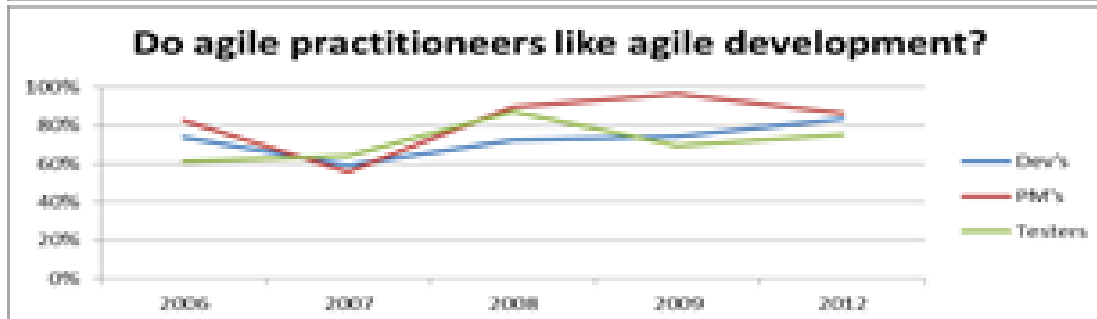




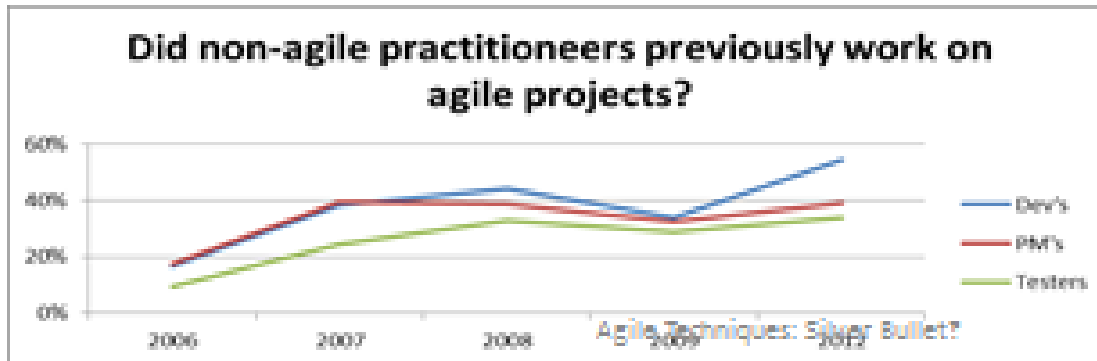
# Microsoft Research Agile Trends (2013)



Usage Increasing



Popular techniques



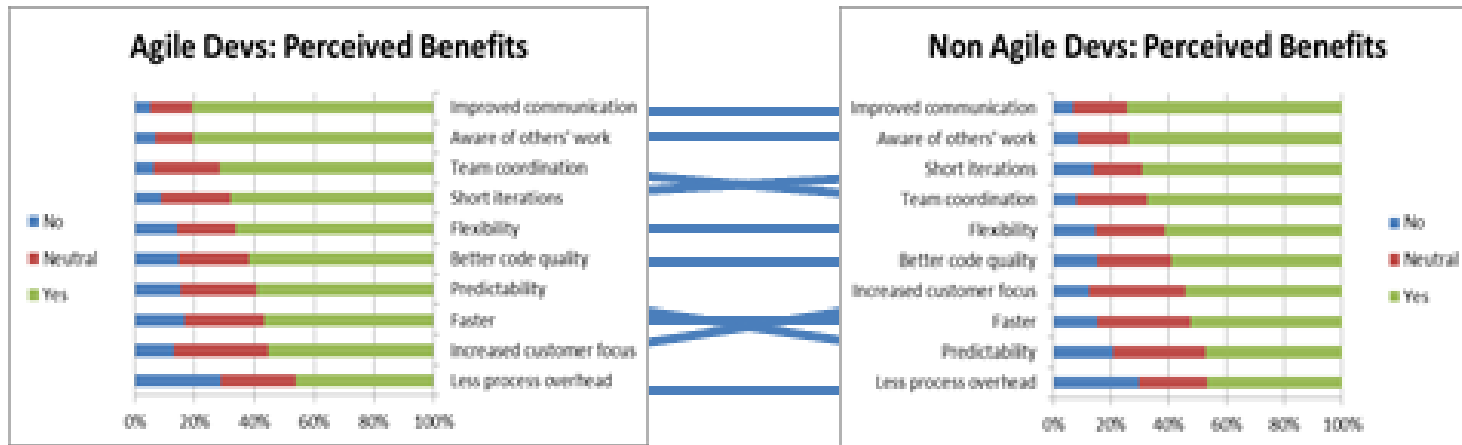
Not Life Changing



# Comparison of Agile Benefits

Microsoft Research

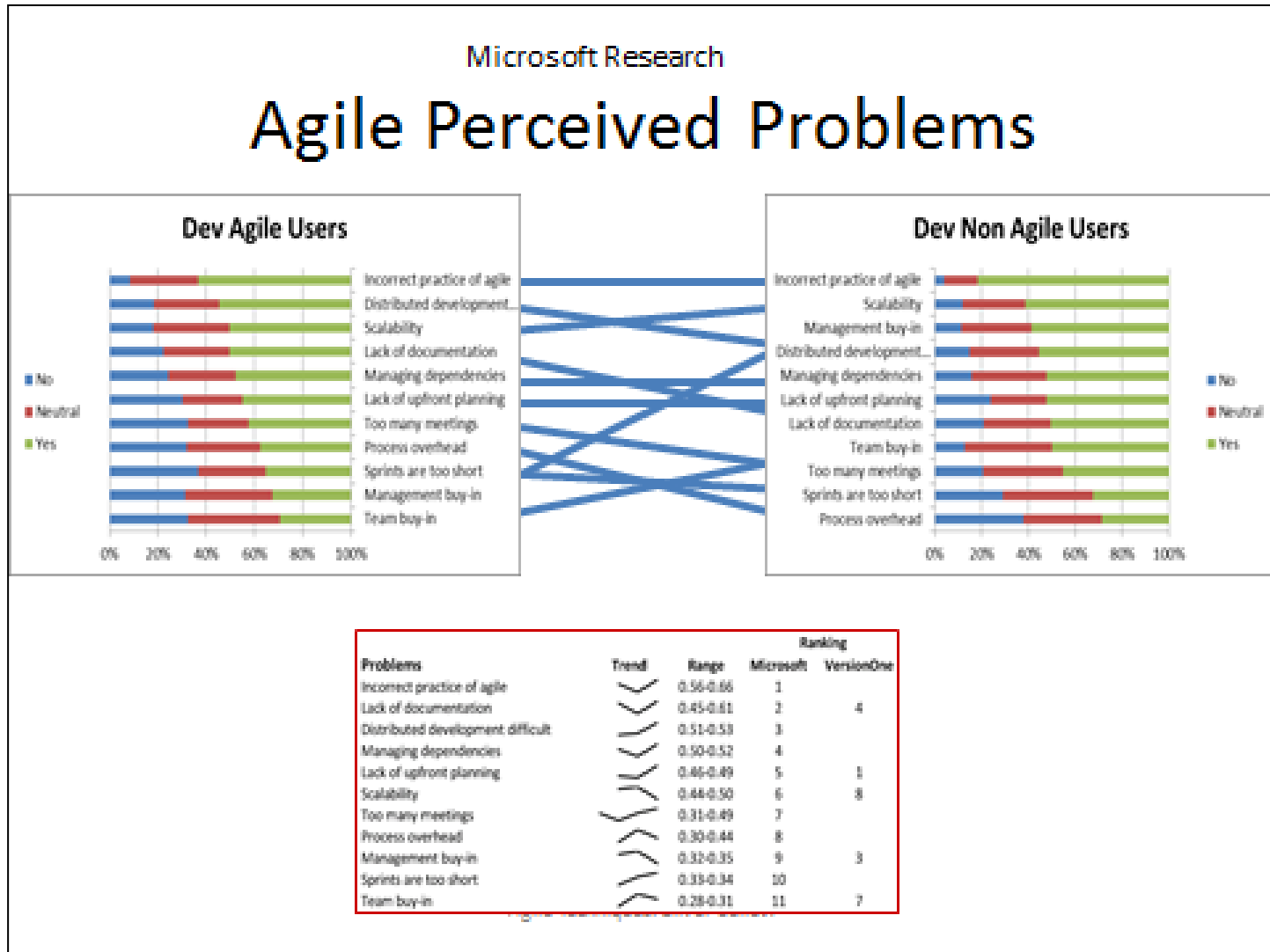
## Agile Benefits



| Benefits                 | Trend | Range     | Ranking   |            |
|--------------------------|-------|-----------|-----------|------------|
|                          |       |           | Microsoft | VersionOne |
| Improved communication   | ↗     | 0.83-0.84 | 1         |            |
| Aware of others' work    | ↗     | 0.74-0.85 | 2         |            |
| Team coordination        | ↗     | 0.73-0.76 | 3         |            |
| Short iterations         | ↗     | 0.67-0.74 | 4         |            |
| Flexibility              | ↗     | 0.63-0.72 | 5         | 2          |
| Better code quality      | ↗     | 0.59-0.62 | 6         | 4          |
| Faster                   | ↗     | 0.54-0.63 | 7         | 1          |
| Predictability           | ↗     | 0.54-0.62 | 8         | 6          |
| Increased customer focus | ↗     | 0.53-0.60 | 9         |            |
| Less process overhead    | ↗     | 0.42-0.52 | 10        |            |



# Some Alleged Agile Problems



# Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

## Results

- the growth of agile adoption at Microsoft is slower than expected
- no individual agile practice exhibited strong growth trends
- both agile and non-agile practitioners agreed on the relative benefits and problem areas of agile techniques

## Conclusions

- no clear trends in practice adoption
- non-agile practitioners are less enamored of the benefits and more strongly in agreement with the problem areas
- the ability for agile practices to be used by large-scale teams generally concerned all respondents



# Traditional vs Agile Approaches Fit<sub>1</sub>

## Traditional approach

- consistent with the acquisition life cycle guidance provided in the DoD Acquisition Deskbook and its supporting documents.
- programs with stable requirements and environment, with known solutions to the requirements
- programs with a homogeneous set of stakeholders who communicate well via documents
- programs for which the technology base is evolving slowly (technology is not expected to be refreshed/replaced within the timeframe of the initial development)



# Traditional vs Agile Approaches Fit<sub>2</sub>

## Agile approach

- programs with volatile requirements and environment
- programs where solutions are sufficiently unknown that significant experimentation is likely to be needed
- programs for which the technology base is evolving rapidly
- programs with stakeholders who can engage with developers in ongoing, close collaboration

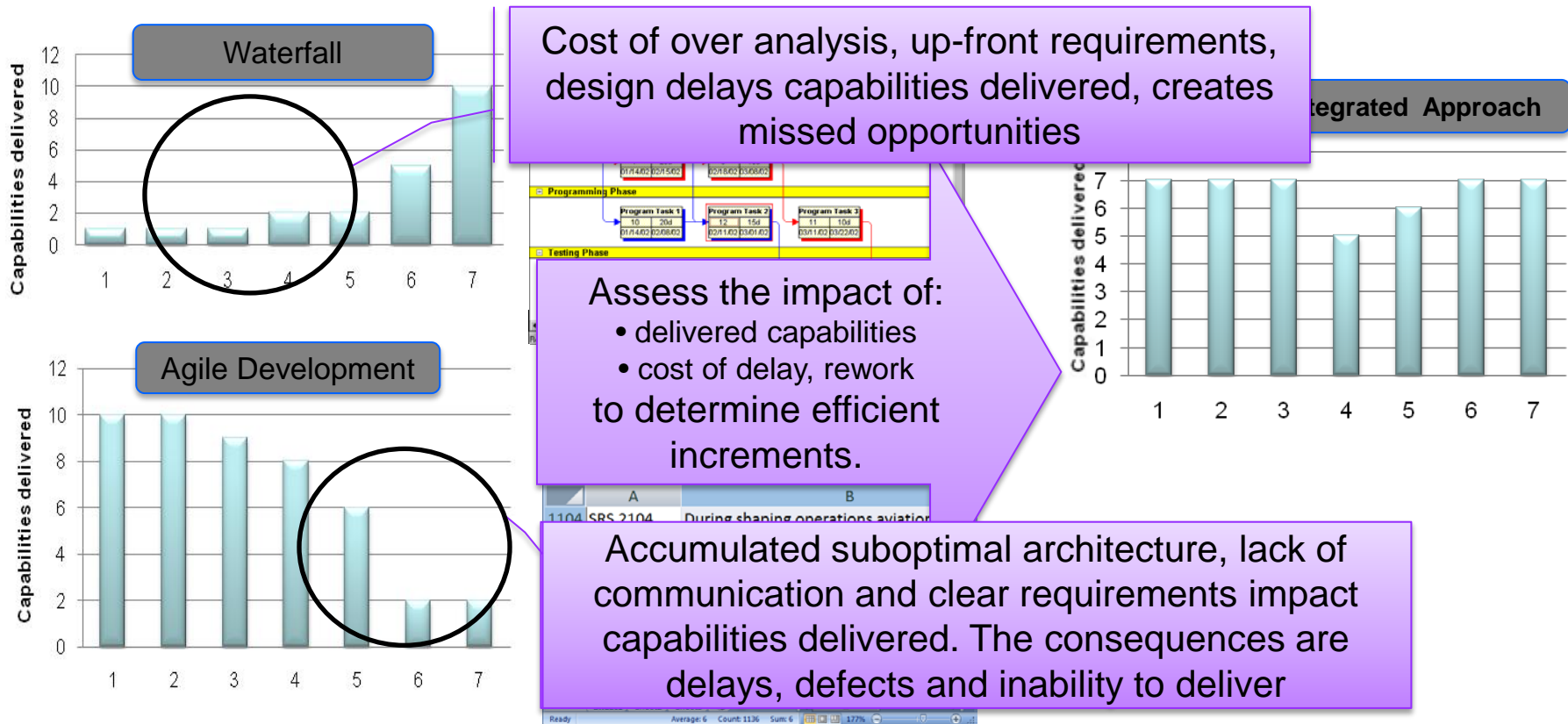
**concluded that, in reality, no acquisition context that we have seen is “ideal” for either the traditional or agile approach.**



**MUST IT BE DONE IN “PURE”  
FORM**



# Both Waterfall and Agile Development Have Risks





# What about modifying SCRUM?

Scrum practices are said to depend on each other and should not be changed

We identified two mismatches between Scrum and the studied organization

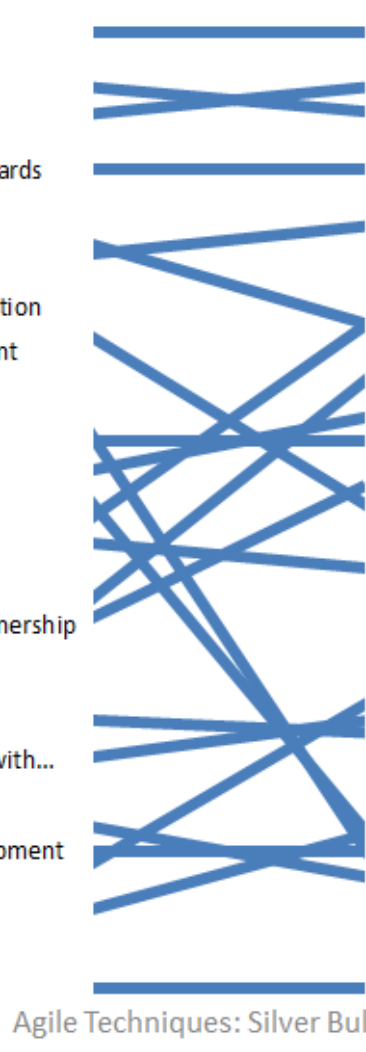
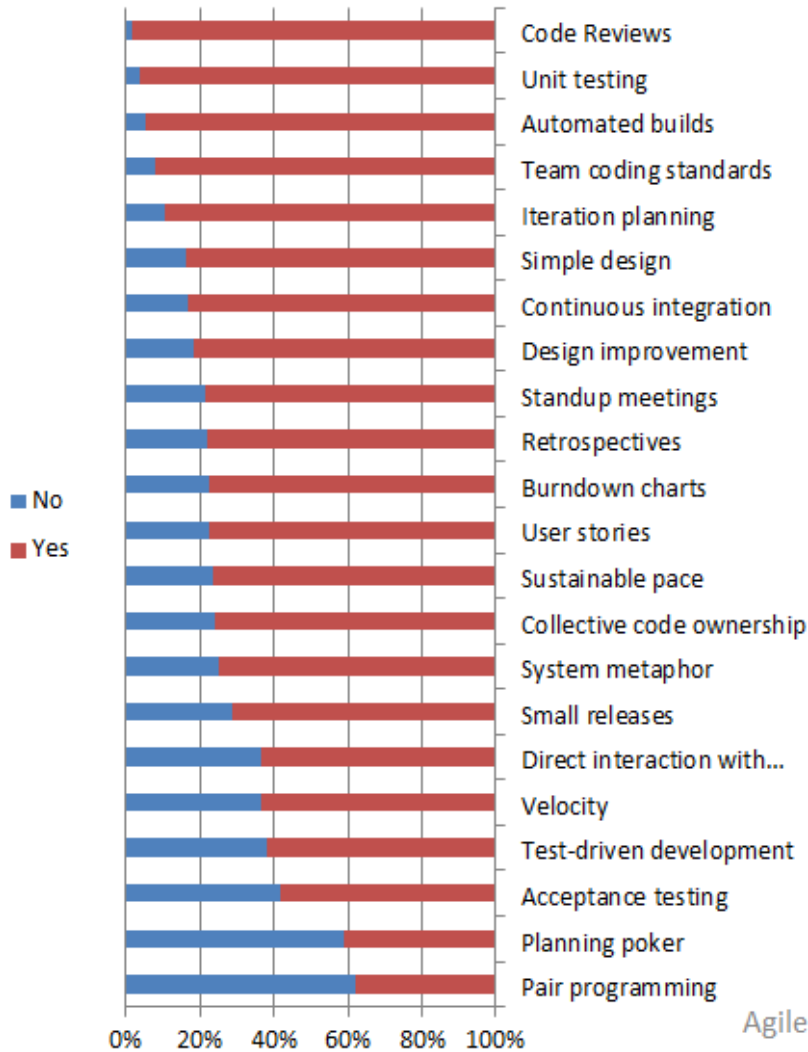
Mismatches we identified were considered necessary or even beneficial

Changes to Scrum cannot categorically be considered detrimental

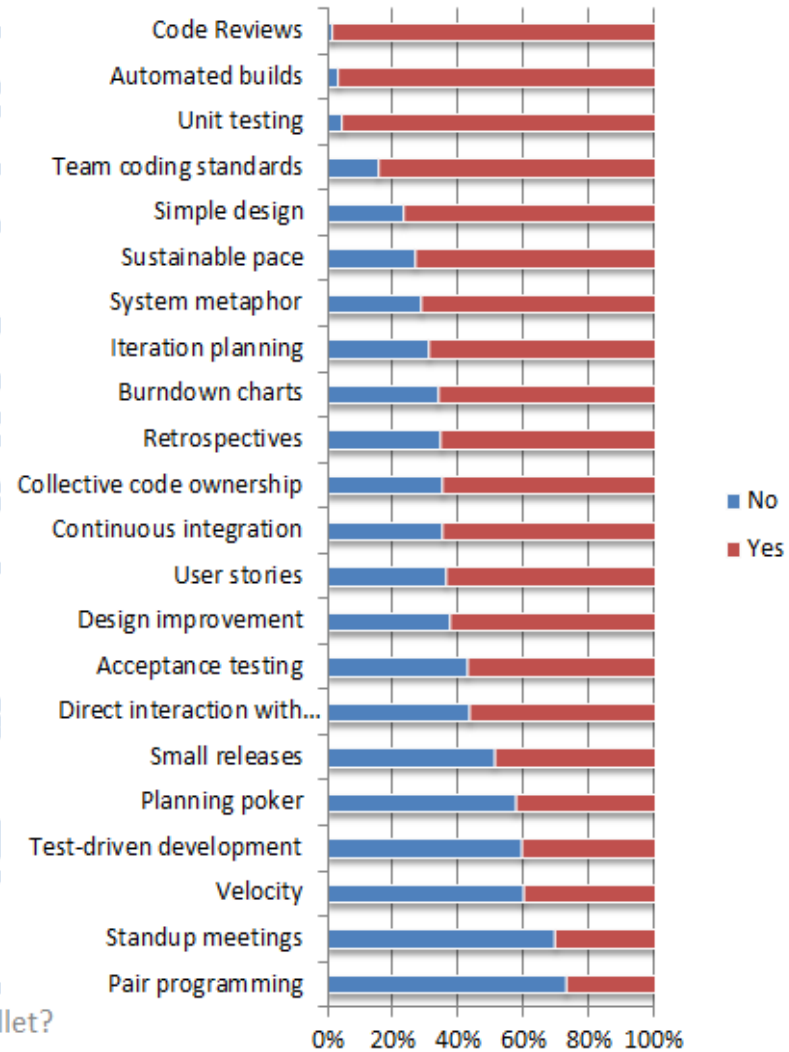


# Microsoft Research Practices

## Agile Devs Team Practices



## Non Agile Devs Team Practices



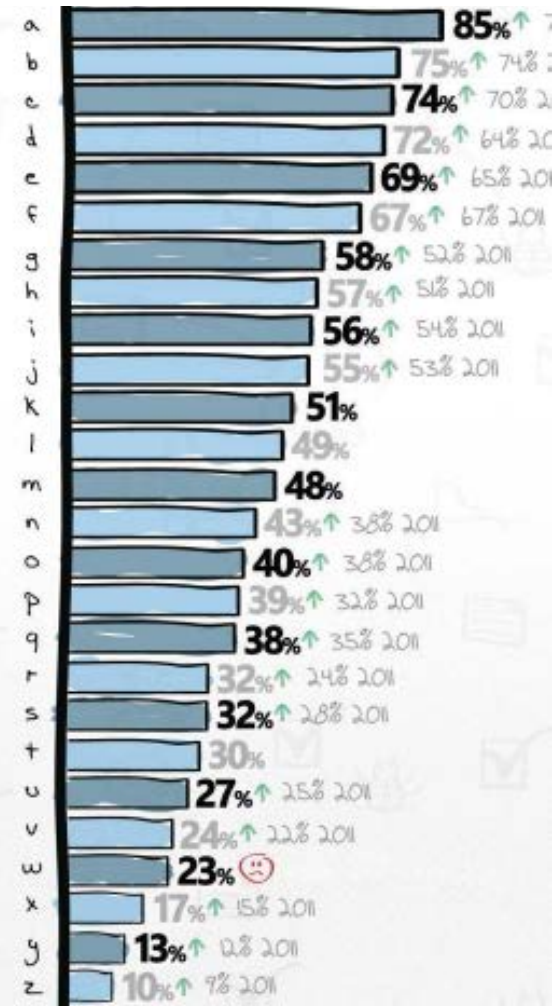
# Use of Agile Techniques

## AGILE TECHNIQUES EMPLOYED

Again this year, core agile tenets currently in use are\* Daily Standup, Iteration Planning and Unit Testing. The two techniques that grew the most in usage from this year to last year were Kanban and Retrospectives; yet, agile techniques increased in every area but one (Continuous Deployment).

\*Respondents were able to select multiple options.

- |                                   |                                |
|-----------------------------------|--------------------------------|
| a Daily Standup                   | n Open Workarea                |
| b Iteration Planning              | o TDD                          |
| c Unit Testing                    | p Digital Taskboard            |
| d Retrospectives                  | q Story Mapping                |
| e Release Planning                | r Kanban                       |
| f Burndown/ Team-Based Estimation | s Collective Code Ownership    |
| g Velocity                        | t Pair Programming             |
| h Coding Standards                | u Automated Acceptance Testing |
| i Continuous Integration          | v Analog Taskboard             |
| j Automated Builds                | w Continuous Deployment        |
| k Dedicated Product Owner         | x Agile Games                  |
| l Integrated Dev/QA               | y Cycle Time                   |
| m Refactoring                     | z BDD                          |



Respondents = 4048

VERSIONONE.COM  
© 2013, VersionOne, Inc. All Rights Reserved



# Scaling Agile Brings in More Variation

## Scaled Agile Framework

- Kanban, SCRUM, Value Stream Mapping

## Disciplined Agile Delivery

- RUP, XP, SCRUM

## DSDM

- Popular scaling approach in Europe

[https://www.theccsiac.com/spruce/resources/ref\\_documents/agile-scale-aas-spruce-sei](https://www.theccsiac.com/spruce/resources/ref_documents/agile-scale-aas-spruce-sei)



# It's not about the practices and methods It's about the principles

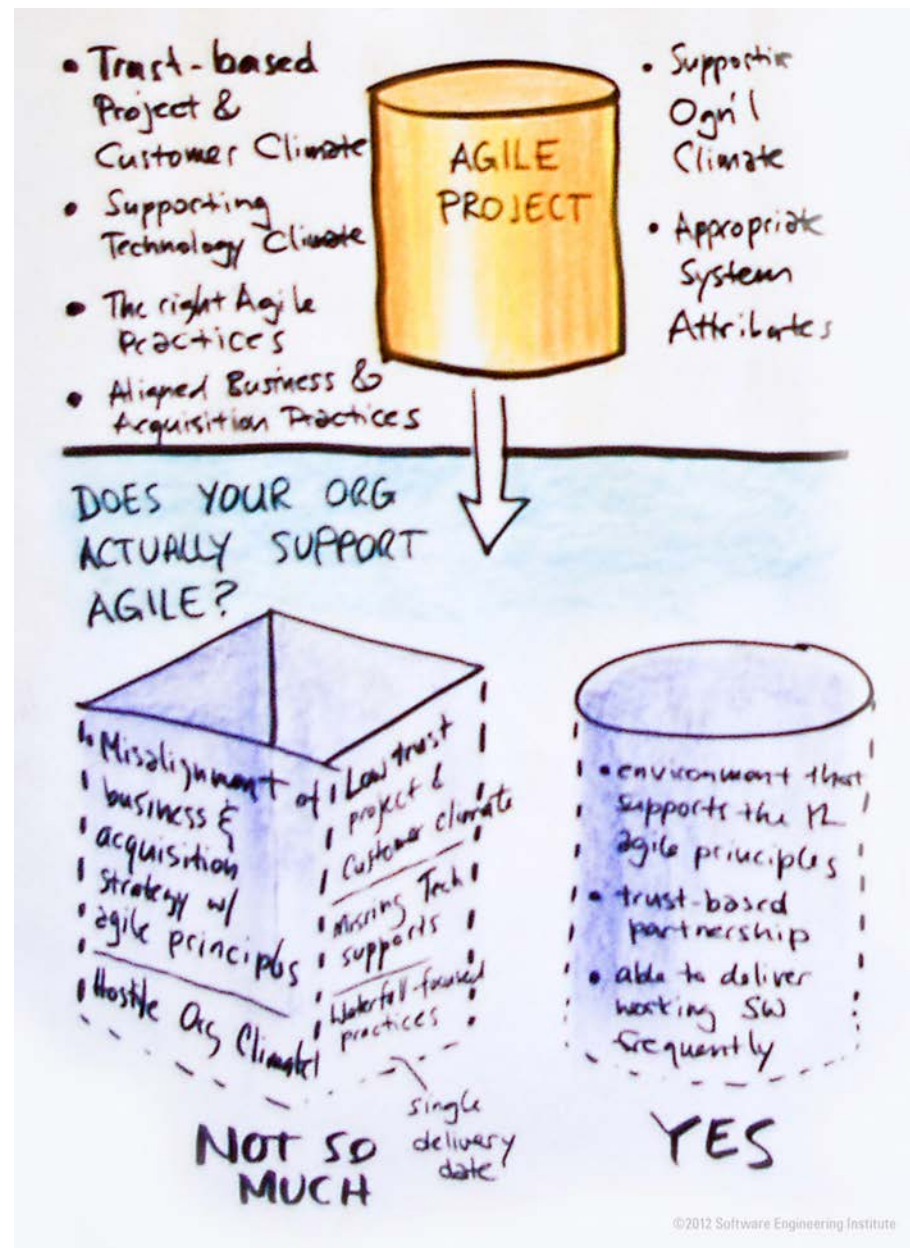
1. Highest priority is satisfy the customer through early and continuous delivery of software.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Provide environment and support they need.
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...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 MUST I DO TO BE SUCCESSFUL



# Understand your organization's alignment with Agile principles and practices



# Traditional approach

Strengths of the traditional approach include:

- enables the comparability and repeatability that standardization provides
- enables a contractually verifiable definition of completed intermediate work products
- reduces risks by means of contractually assured baselines

Weaknesses of the traditional approach include:

- the process drives measurement of compliance with itself as a primary measure of success (i.e., rather than measuring success as deploying a workable solution)
- it depends on documents as the basis to verify and validate the requirements, the architecture, and the detailed design
- most of the requirements are completed before any code is written, thus extending development timelines





# Agile approach

Strengths of this approach include

- early insight by the users into the shape of the solution
- early course correction
- “fail fast” (If the early solution ideas turn out to be flawed, little time or money is spent before that learning occurs.)
- explicit understanding that the requirements are expected to evolve

Weaknesses of this approach (particularly in large acquisition settings) include

- more dependence on tacit knowledge (e.g., lack of explicit documentation) as the basis for decision-making than is comfortable for most acquisition organizations
- dependence on availability of actively engaged user/customers
- difficulty in aligning implementation-driven artifacts and measures with those of the larger traditional acquisition setting.



# Effective Agile Practices in Federal Settings

**Table 1: Practices Used and Found Effective by Five Agencies**

| Practice  |
|---|
| 1. Start with Agile guidance and an Agile adoption strategy.  |
| 2. Enhance migration to Agile concepts using Agile terms and examples.  |
| 3. Continuously improve Agile adoption at both project and organization levels.                                 |
| 4. Seek to identify and address impediments at the organization and project levels.                             |
| 5. Obtain stakeholder/customer feedback frequently and closely.   |
| 6. Empower small, cross-functional teams.   |
| 7. Include requirements related to security and progress monitoring in your queue of unfinished work (backlog). |
| 8. Gain trust by demonstrating value at the end of each iteration.  |
| 9. Track progress using tools and metrics.  |
| 10. Track progress daily and visibly.   |

Source: GAO.



# Successful Management Traits within Agile Teams

## Executing Side (developer)

Leader – more time with team than behind the office desk

Coach – seed team with ideas and allow them to solve the problem

Expeditor – help remove operational impediments

Champion – communicate with upper-level management and stakeholders (translator role)

Ambassador – cultivate relationships with end users and subject matter experts and their management

## Acquiring Side (PMO)

Leader – establish and maintain relationships with executing group

Coach – help existing personnel make transition to fast-tempo, high-interaction environment of Agile

Expeditor – efficiently deploy people interacting with development team.

Champion – maintain buy-in from external funders and stakeholders

Ambassador – ensure appointment of end users or SMEs to work with developers.

[Agile Methods: Selected DoD Management and Acquisition Concerns  
http://www.sei.cmu.edu/library/abstracts/reports/11tn002.cfm?DCSext.  
abstractsource=SearchResults](http://www.sei.cmu.edu/library/abstracts/reports/11tn002.cfm?DCSext.abstractsource=SearchResults)



# Bottom Line Take Aways

Accumulated empirical evidence is scant but increasing. Evidence shows:

- Agile can be effective
- Agile is not a silver bullet
- Agile is not conducive to every situation
- Agile is a different mindset and requires trust
- Agile requires planning and hard work



## **For More Information, or to Join SEI's Agile Collaboration Group, Contact...**

**Mary Ann Lapham**

Principal Engineer

Telephone: 412-268-5498

Email: [mlapham@sei.cmu.edu](mailto:mlapham@sei.cmu.edu)

### **U.S. Mail:**

Software Engineering Institute

Carnegie Mellon University

4500 Fifth Avenue

Pittsburgh, PA 15213-3890

