

Scrum: A disciplined approach to product quality and project success.

CQAA

February 23, 2011



Patricia Rotman
Siemens Industry, Inc.

Introductions



Alternate Titles Considered

Scrum: Just do it!

Scrum: It only works if you do it.

Scrum: It is harder than it looks.

Scrum: It is not a silver bullet.

Scrum: Misunderstood as a dream for developers when in reality it is a dream for management and QA.

Agenda

- Goals for this presentation
- Background
- Definitions
- Discipline in Scrum
- Conclusions
- Q&A

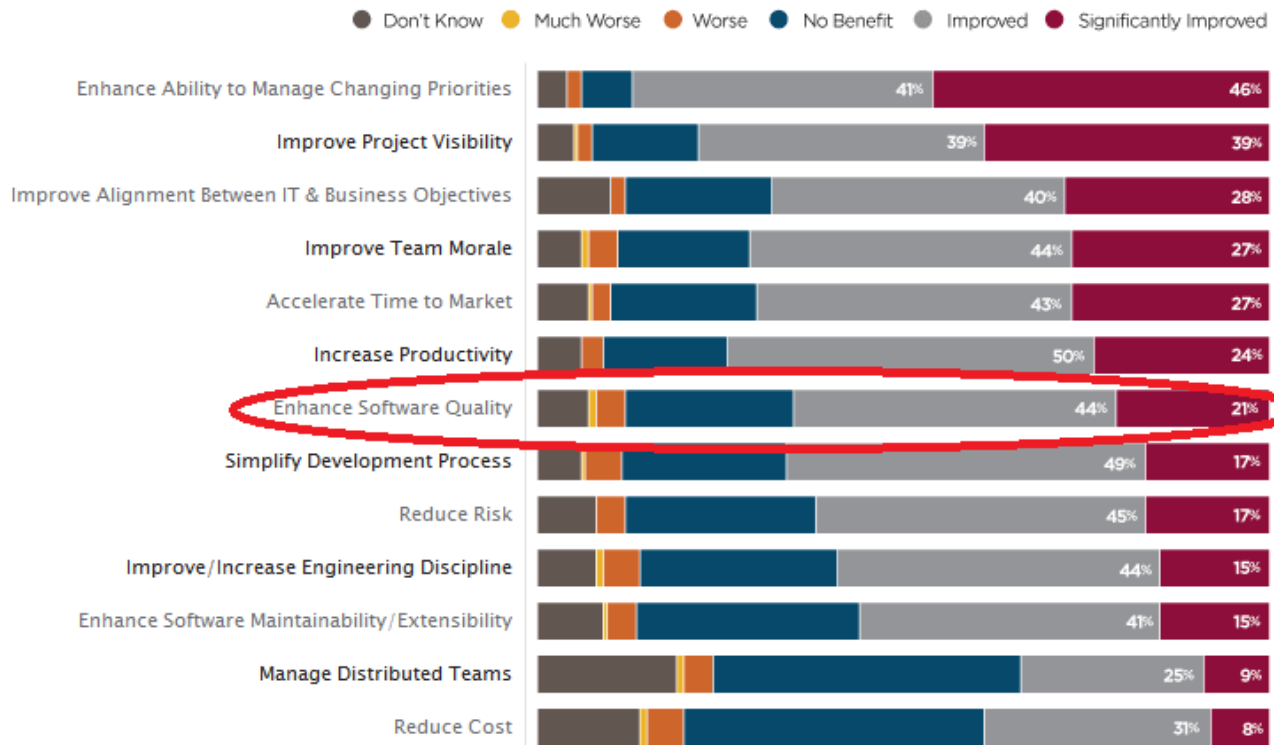
Goals for Presentation

- Be inspired
- Understand key areas of discipline in Scrum that will move you closer to quality and success
- Take away several practical ways to improve your Scrum practice

State of Agile Development

BENEFITS OBTAINED FROM IMPLEMENTING AGILE

While the most common reason for adopting agile was improved productivity and time to market, respondents cited that they experienced improved managerial abilities as the biggest benefit of implementing agile.



* State of Agile Development Survey Results

Sponsored by VersionOne, The fifth annual "State of Agile Development" survey was conducted between August 11 and October 31, 2010.

Background

Siemens Industry Scrum Project Attributes

- 5+ years of Scrum
- Global development teams, Global customers
- Shrink wrapped products
- FDA regulated users
- CMMi3
- 4 legacy products, one is ~25 Million lines of code
- 2 major new development efforts
- 150+ people in SBT involved in Agile projects
- Prior to Scrum we were better than average at hitting our deadlines, but not good enough

Background: Why Scrum?

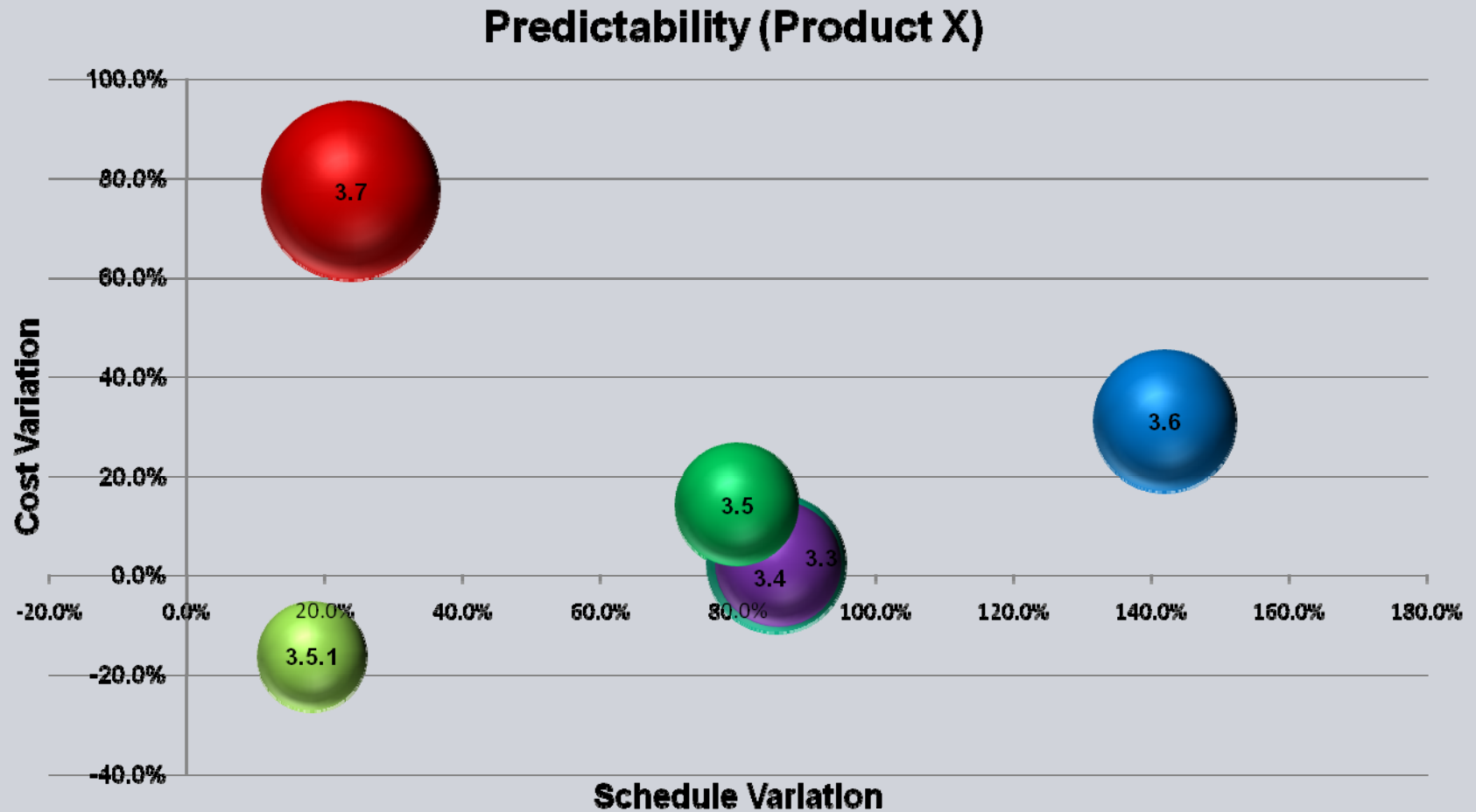
Improve Predictability



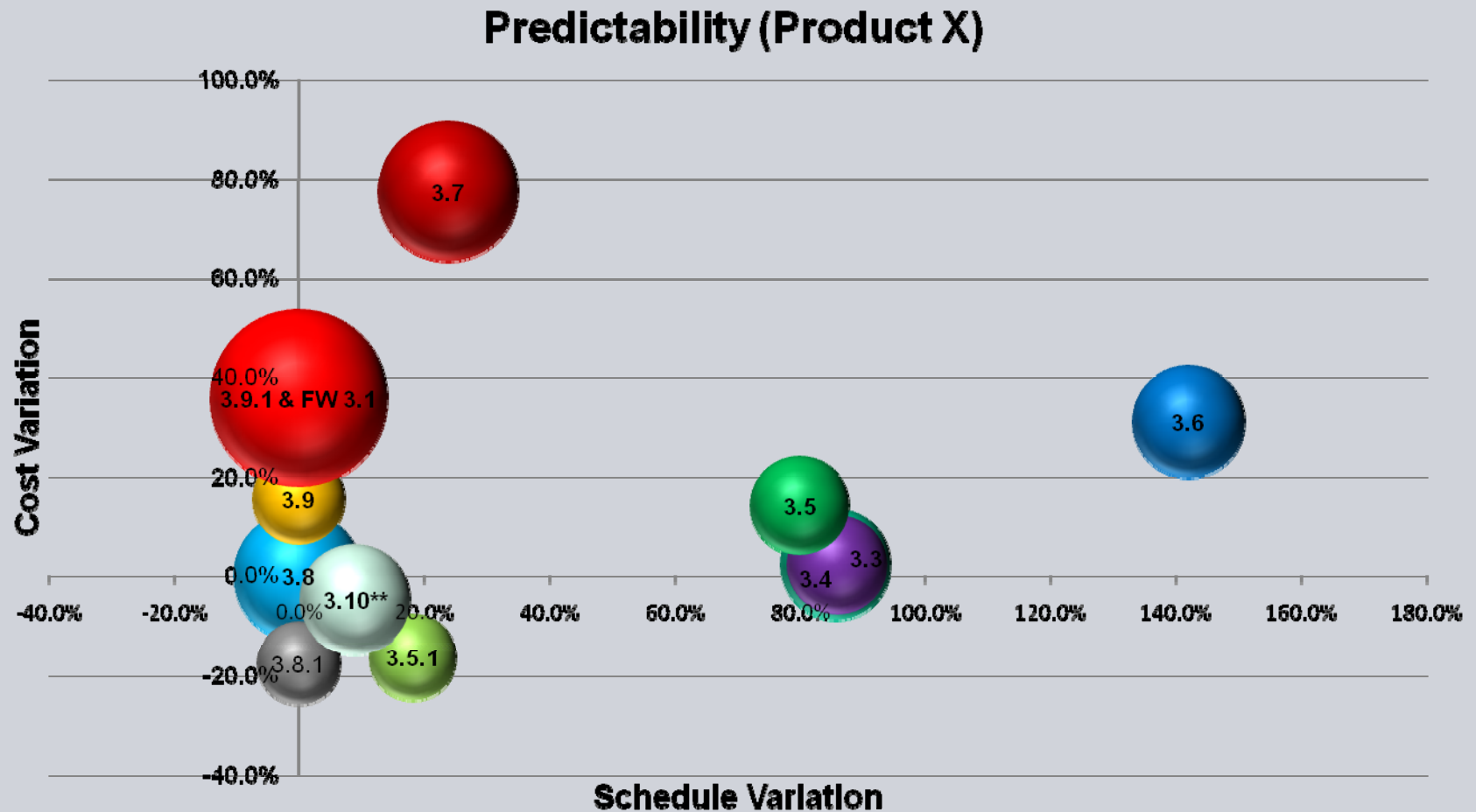
Make Offshoring Work



Background: Predictability before Scrum



Predictability After Scrum



Background: Success Stories

- Significant decrease in defects found by our customers
- Significant decrease in patches
- Success in off-shoring
- Positive change in QA role

Definitions

Scrum: A disciplined approach to product quality and project success.

Definitions: Scrum

- A set of practices and principles
- An Agile project management framework
- Transparency / Visibility

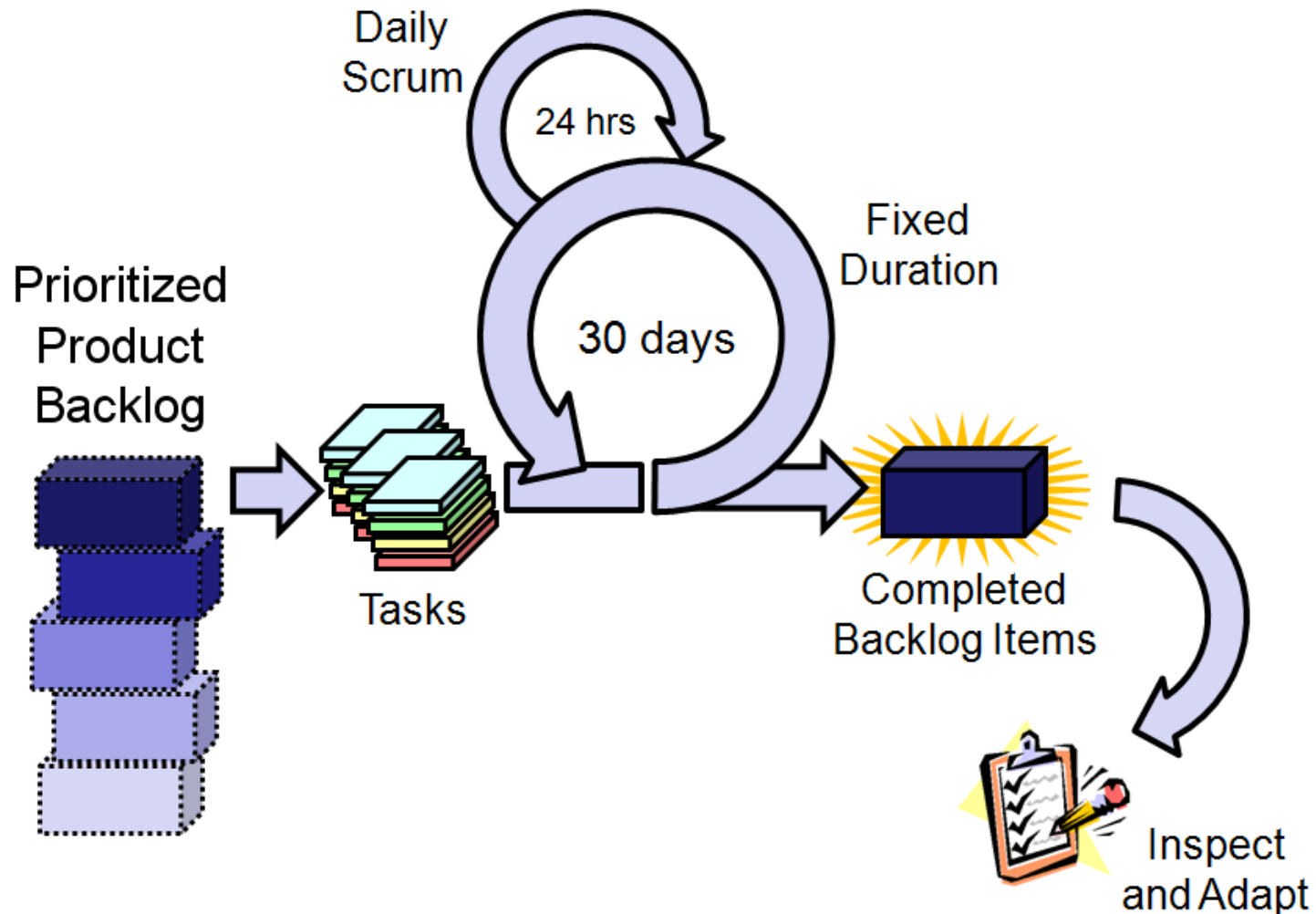


Scrum Principles

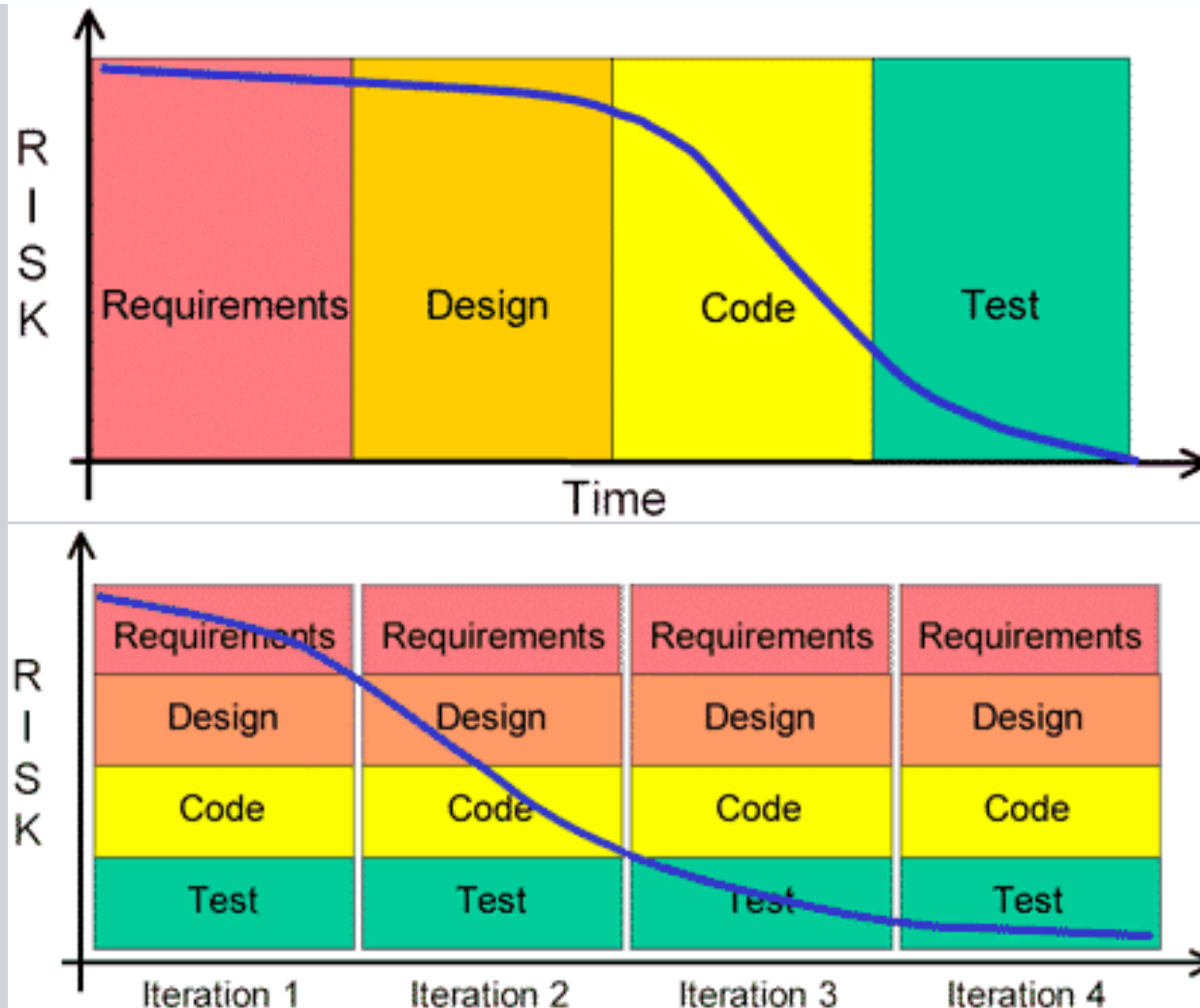
- Working software every iteration
- Delivering the “right” product
- Rooted in “The Toyota Way” principles
- A holistic (vs. sequential) approach to product development
- Simple but Hard



Scrum Overview



How is Scrum different than sequential approaches?



Definitions: Discipline

adj. Possessing or indicative of discipline: a dancer's disciplined body; a disciplined set of work habits.

1. disciplined - obeying the rules; controlled - restrained or managed or kept within certain bounds;

2. disciplined - trained mentally or physically by instruction or exercise; trained - shaped or conditioned or disciplined by training;

Discipline

What do you picture when you think of discipline?



Definitions: Quality

adj. Having a high degree of excellence



- What the customer wants
- Works as expected
- “free from bugs and flaws”

Definitions: Success

n. The achievement of something desired, planned, or attempted.

- Schedule
- Cost
- Features
- Quality

The Standish Group's "CHAOS Summary 2009"

- marked decrease in project success rates
- downtick in the success rates from the previous study
- significant increase in the number of failures
- low point in the last five study periods
- highest failure rate in over a decade



Project Success

The Standish Group International, Inc. -Project Success Factors

1. User Involvement 15.9%
 2. Executive Management Support 13.9%
 3. Clear Statement of Requirements 13.0%
 4. Proper Planning 9.6%
 5. Realistic Expectations 8.2%
 6. Smaller Project Milestones 7.7%
 7. Competent Staff 7.2%
 8. Ownership 5.3%
 9. Clear Vision & Objectives 2.9%
 10. Hard-Working, Focused Staff 2.4%
- Other 13.9%

Discipline in Scrum

Ways
to
improve



Topics

User Stories
What is Done
Technical Debt
Collaboration
Product Backlog Management
Continuous Integration
Scope
Inspect & Adapt
Measurements

QA
Role



User Stories

Quality starts at the beginning.

- How do you know if you have a quality User Story?
- Is your process for creating User Stories working?
- Is QA involved in User Story definition?

How do you know if you have a quality User Story?

Do your stories pass the INVEST test?



Independent
Negotiable
Valuable
Estimable
Small
Testable

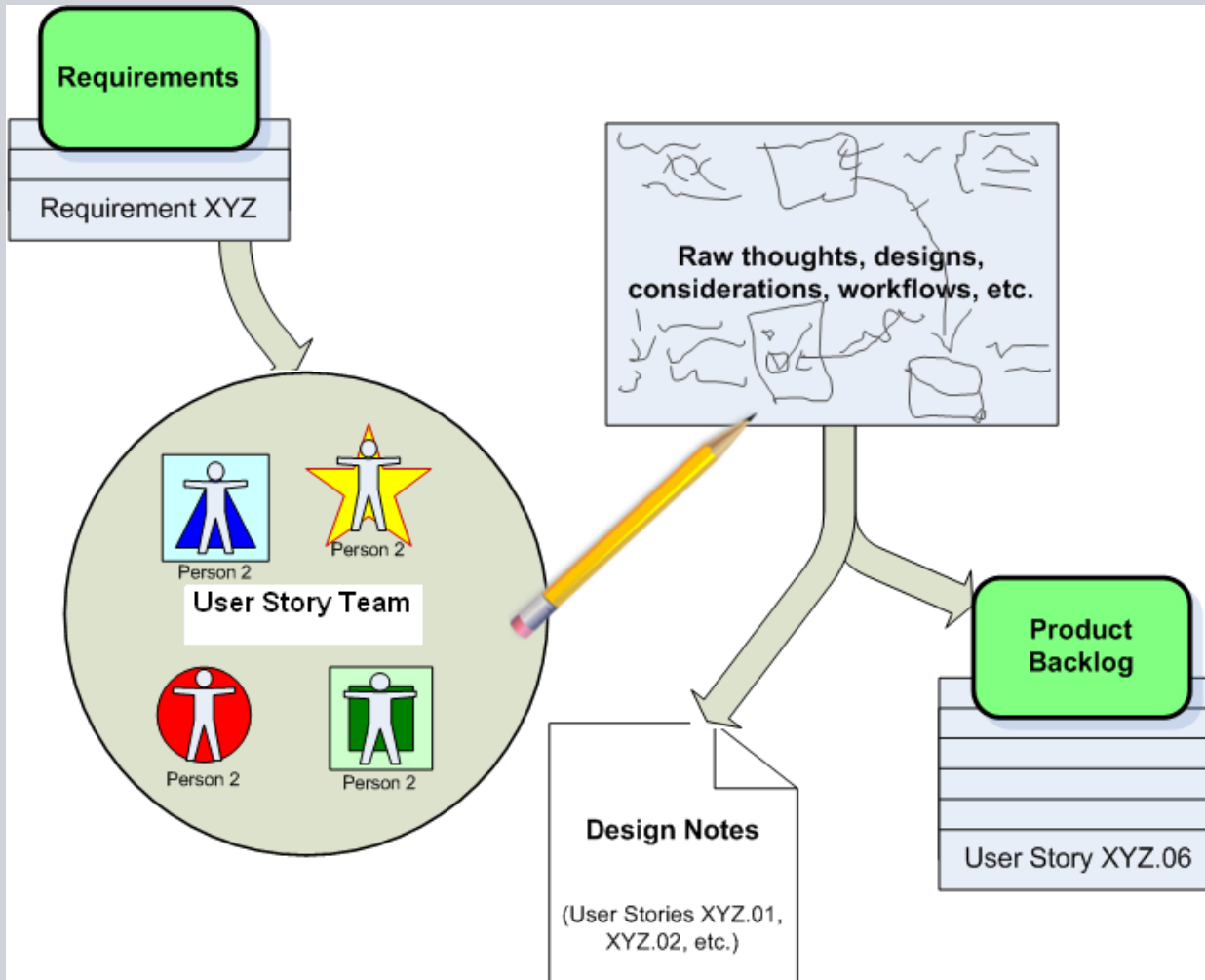
Helpful User Story format

As a _____<type of user>_____ user

I would like to _____<action>_____

because _____<value>_____.

User Stories: Process Matters



When?

During Sprint

How Long?

1 hour = 1
requirement -> 2-3
user stories. About
8 hours per sprint

User Stories: QA Role

- External perspective creates the best User Stories
- QA involvement is vital to defining good user stories
- More efficient and complete understanding of the functionality results from QA involvement
- QA is well prepared to work in parallel with developers

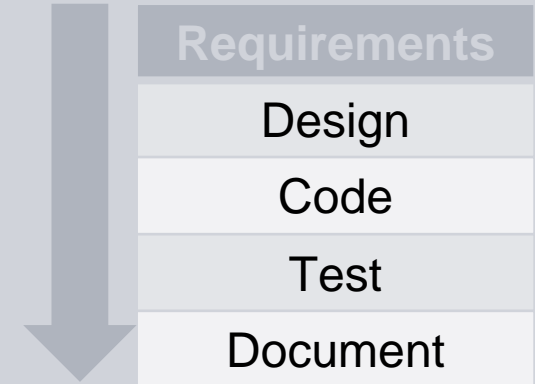
Discussion Questions

What are some best practices you have seen with respect to creating quality User Stories?

What are your organizations greatest challenges with respect to creating quality User Stories?

What is Done

- Do you have a “What is Done” list that is visible?
- Do you make sure it is followed?
- Does it drive tasking?
- Are you able to achieve parallel development and testing? If not... get creative!
- Are you producing a working product each iteration?



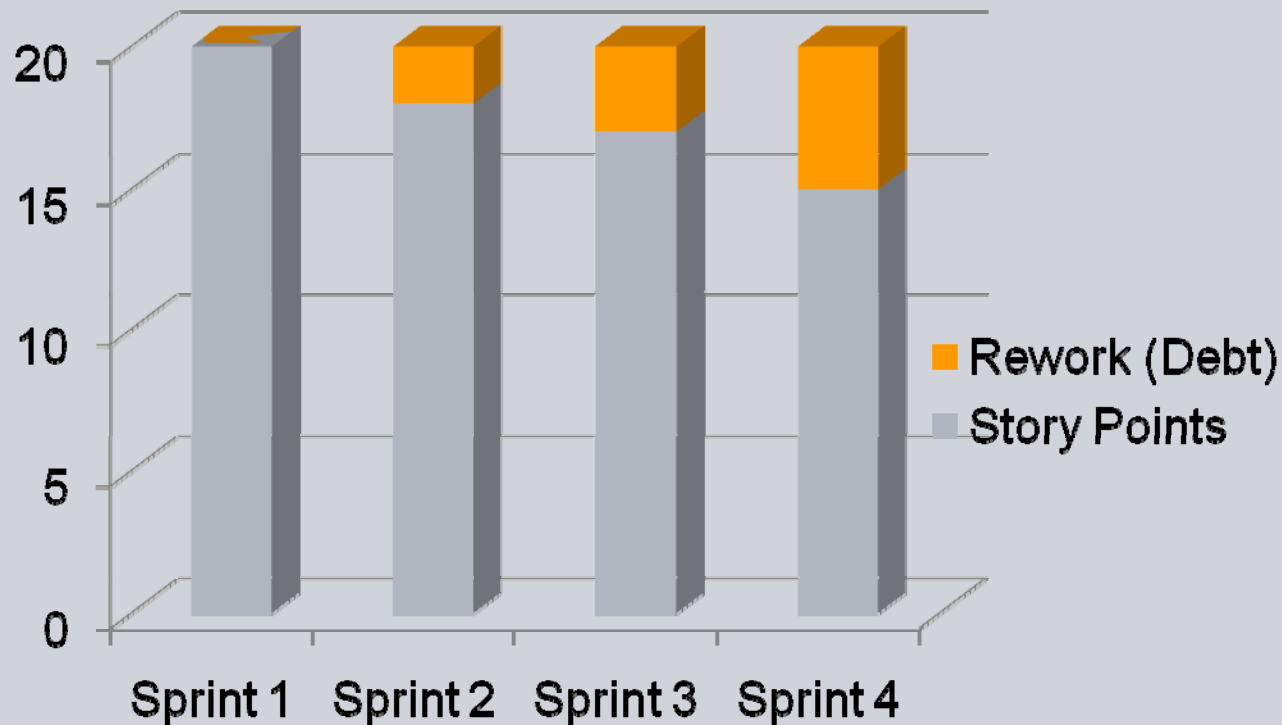
What is Done: Example

Full project lifecycle every iteration

- Design
- User Story update
- UTS Documentation
- Coding
- Design Doc
- Unit Testing
- Bug Fixing
- Design Review
- Code Review
- UTS Review
- Test Case Review
- Tech. Comm. Review
- Peer Review of Test Cases
- Test Data generation
- Test cases preparation
- Incorporating review comments
- Build verification
- Test case execution
- Retesting
- Regression / Ad-hoc testing
- Test execution report generation
- Update help topic
- Incorporating review comments
- Traceability Matrix updated

Managing Technical Debt

What's wrong with this picture?



Managing Technical Debt: Key Questions

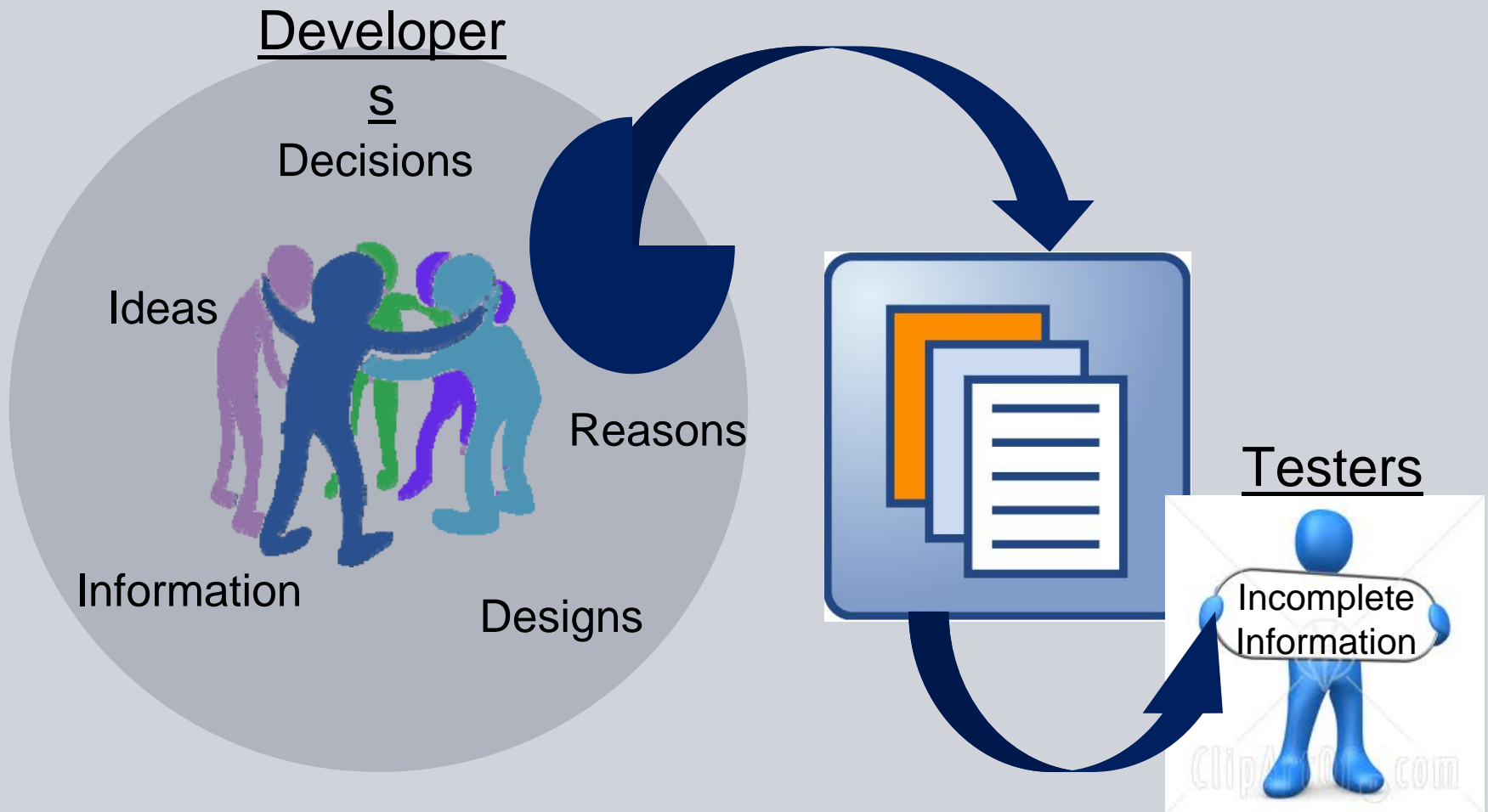
- Is the team taking too much work?
- Are testers able to test throughout the sprint, or just at the end?
- Is your “What is Done” list complete and strict enough?
- Are there external factors driving the team toward this behavior?
- What are teams being rewarded for? (Speed vs. Quality)
- Is the team implementing TDD, automated unit testing, automated functional testing, etc.?

Collaboration

- Importance of shared vision
- Efficiency, really?
- Meetings vs. Email
- Relationship challenges
- Silos of knowledge
- Daily Stand-up, Planning Meetings, Demos



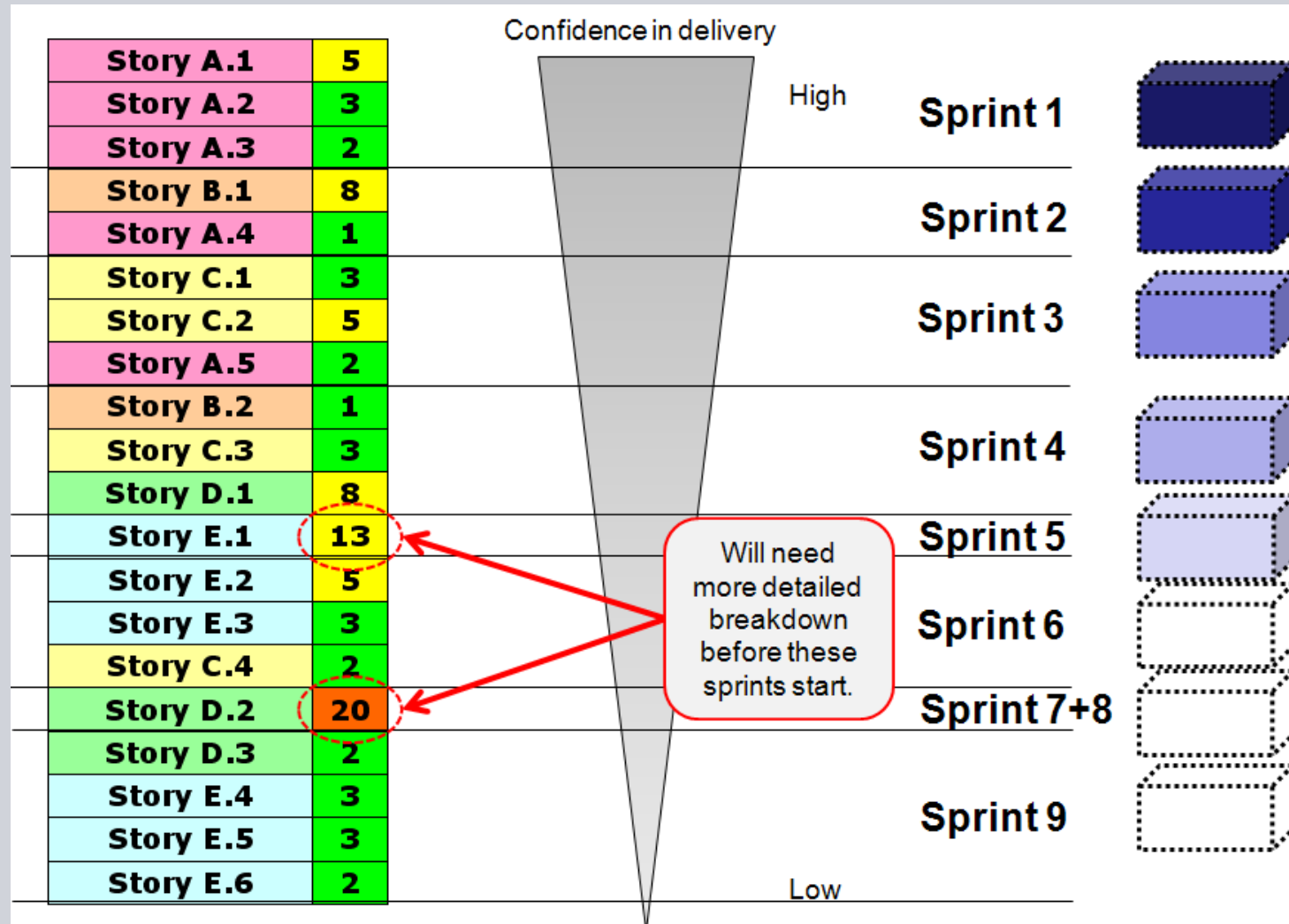
Collaboration: Wasted time?



Discussion Question

In what ways can your Scrum practice be more collaborative? What would be the benefit?

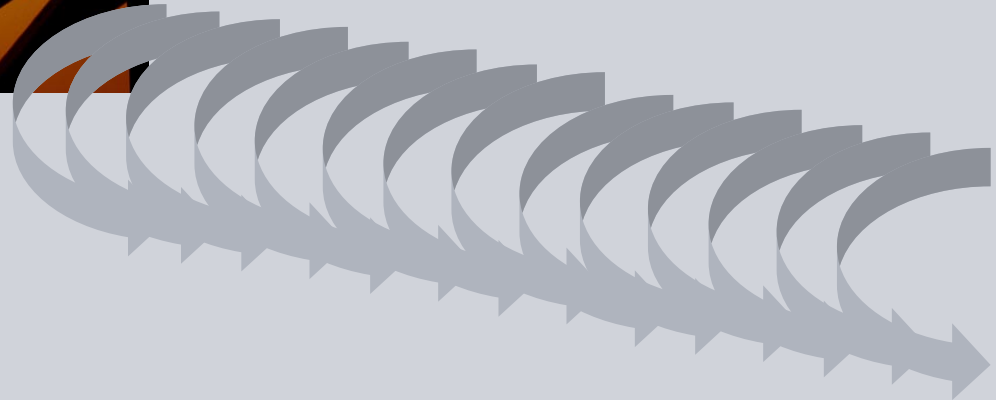
Backlog Maintenance



Continuous Integration

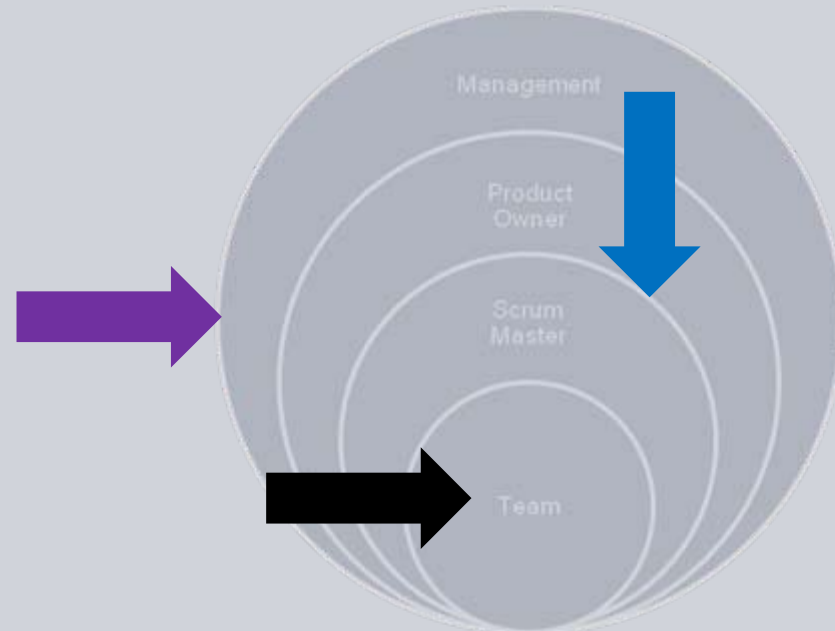


- Are you doing daily builds?
- Is integration done at least weekly?
- What is the value of a product that always “works”?



Scope

- Is the Scrum team given every opportunity to succeed?
- Do team members expand the scope during the sprint?
- Does QA raise the “red” flag when this happens?



Inspect and Adapt



- Do you look at the data that you have?
 - Estimates vs. Actuals
 - Rework vs. new development (technical debt)
- Are your retrospectives meaningful?
- Do teams implement at least one change per sprint?
- Do you try to analyze issues objectively?
(Example: fish-bone)

You get what you measure

- If teams are measured purely on speed, they will make a mess fast.
- Consider measuring and rewarding “doneness”.
- Consider measuring automated unit test coverage.
- Consider measuring structural quality. (things like complexity, function and class size, dependencies, etc.)
- Consider reading “**The Land that Scrum Forgot**” by Bob Martin:

<http://www.scrumalliance.org/articles/300-the-land-that-scrum-forgot>

Case Study – Discussion

You are one of two QA people on your Scrum team. Your team is incurring more and more technical debt each sprint. The developers are developing up to the last few days of the sprint, and QA is always pressed for time to finish. Bugs are showing up in later sprints when more pieces of the application are integrated. It doesn't look like Scrum is helping your team deliver a higher quality product.

What do you do?

Questions



Thank you for your attention!

patricia.rotman@siemens.com



Resources

Agile Alliance

www.agilealliance.org

Control Chaos

www.controlchaos.com

Scrum Alliance

www.scrumalliance.org