Agile Testing

Embrace Uncertainty

Naresh Jain
naresh@agilefaqs.com

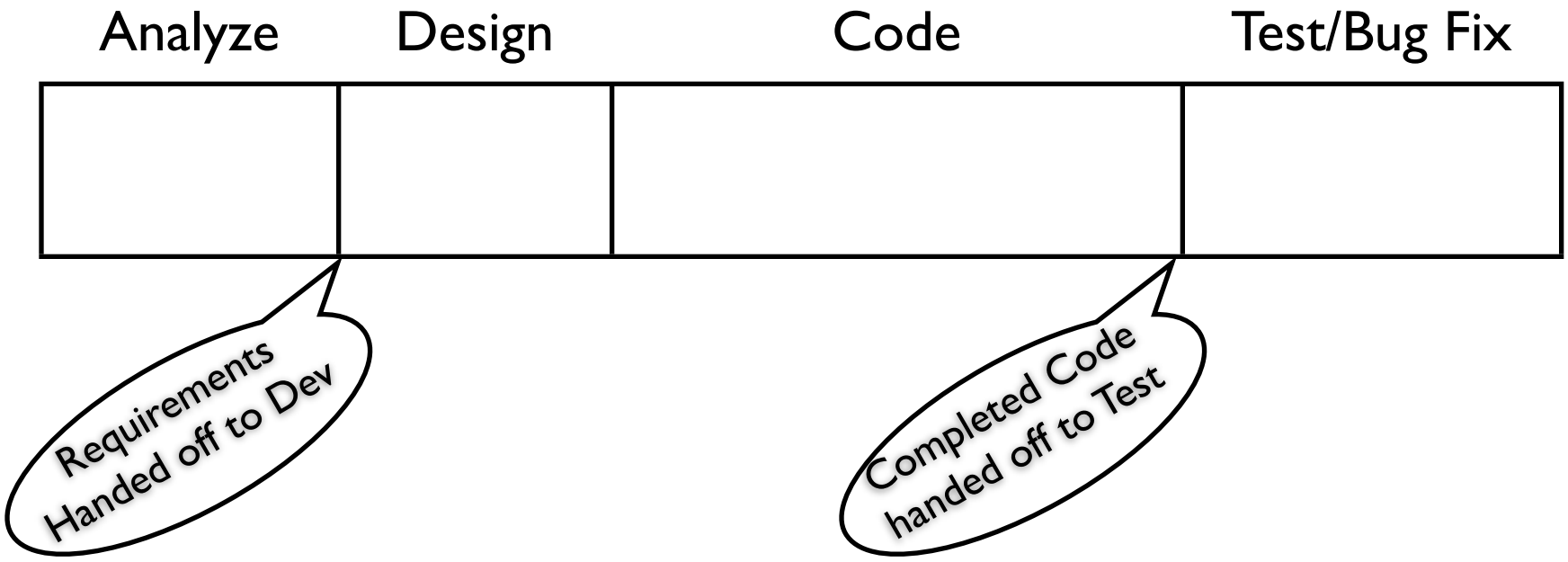
Objective

- ☑ Introduce Agile Testing Concepts
  - ☑ Some background about Agile
  - ☑ Compare with Traditional Testing practices
  - ☑ Highlight the fundamental shift in Thought Process
- ☑ Discuss some tools and techniques used
- ☑ Some pointers to help you try some of these techniques

agile faqs

Traditional Software Development

With great optimism and the best of intentions, The Project Plan is announced

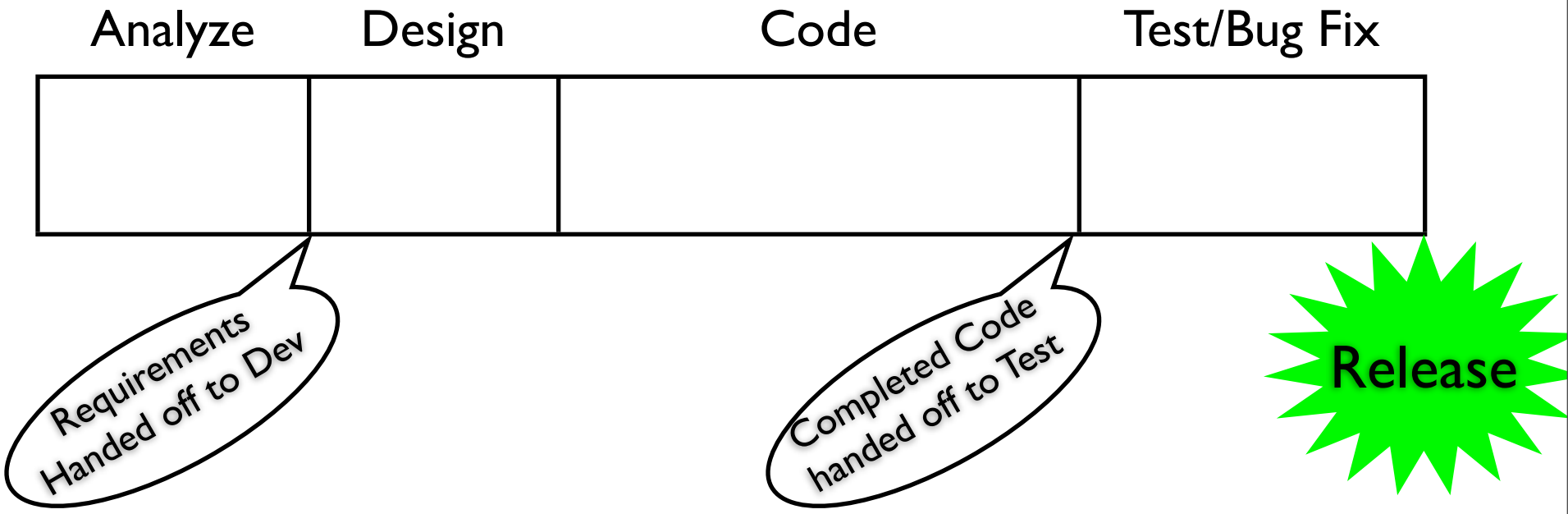


Source Agile/QA Testing - Elisabeth Hendrickson

agile faqs

Traditional Software Development

With great optimism and the best of intentions, The Project Plan is announced

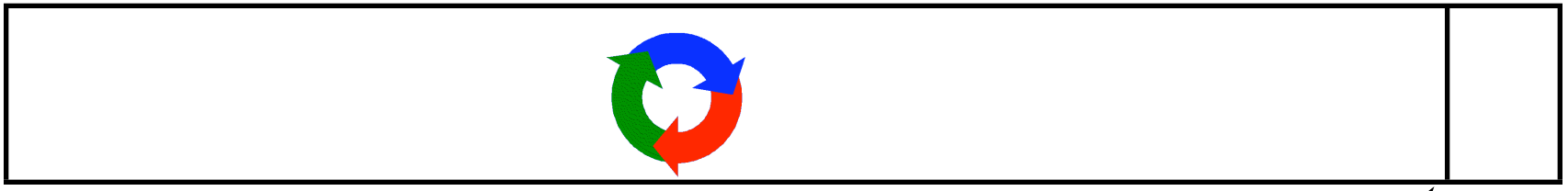


Source Agile/QA Testing - Elisabeth Hendrickson

Reality

Inevitably, the project plan is revised

Analyze, Design and Code | Test/Bug Fix



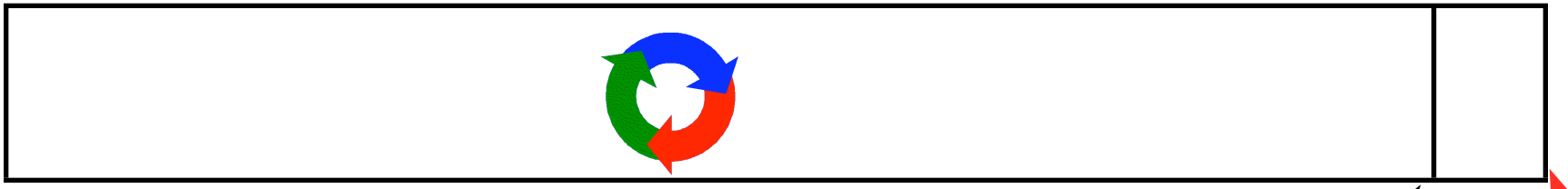
Completed Code handed off to Test

Source Agile/QA Testing - Elisabeth Hendrickson

Reality

Inevitably, the project plan is revised

Analyze, Design and Code | Test/Bug Fix



Completed Code handed off to Test

Release

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance
- ☑ Strict change management

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance
- ☑ Strict change management
- ☑ Detailed preparation and up front planning

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance
- ☑ Strict change management
- ☑ Detailed preparation and up front planning
- ☑ Heavyweight documentation

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance
- ☑ Strict change management
- ☑ Detailed preparation and up front planning
- ☑ Heavyweight documentation
- ☑ Strict entrance and exit criteria with sign-offs

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

- ☑ "Last Defender of Quality" stance
- ☑ Strict change management
- ☑ Detailed preparation and up front planning
- ☑ Heavyweight documentation
- ☑ Strict entrance and exit criteria with sign-offs
- ☑ Heavyweight test automation focused on regression

Source Agile/QA Testing - Elisabeth Hendrickson

Controlling the Chaos

Traditional testing practices attempt to manage the chaos (or at least avoid the blame):

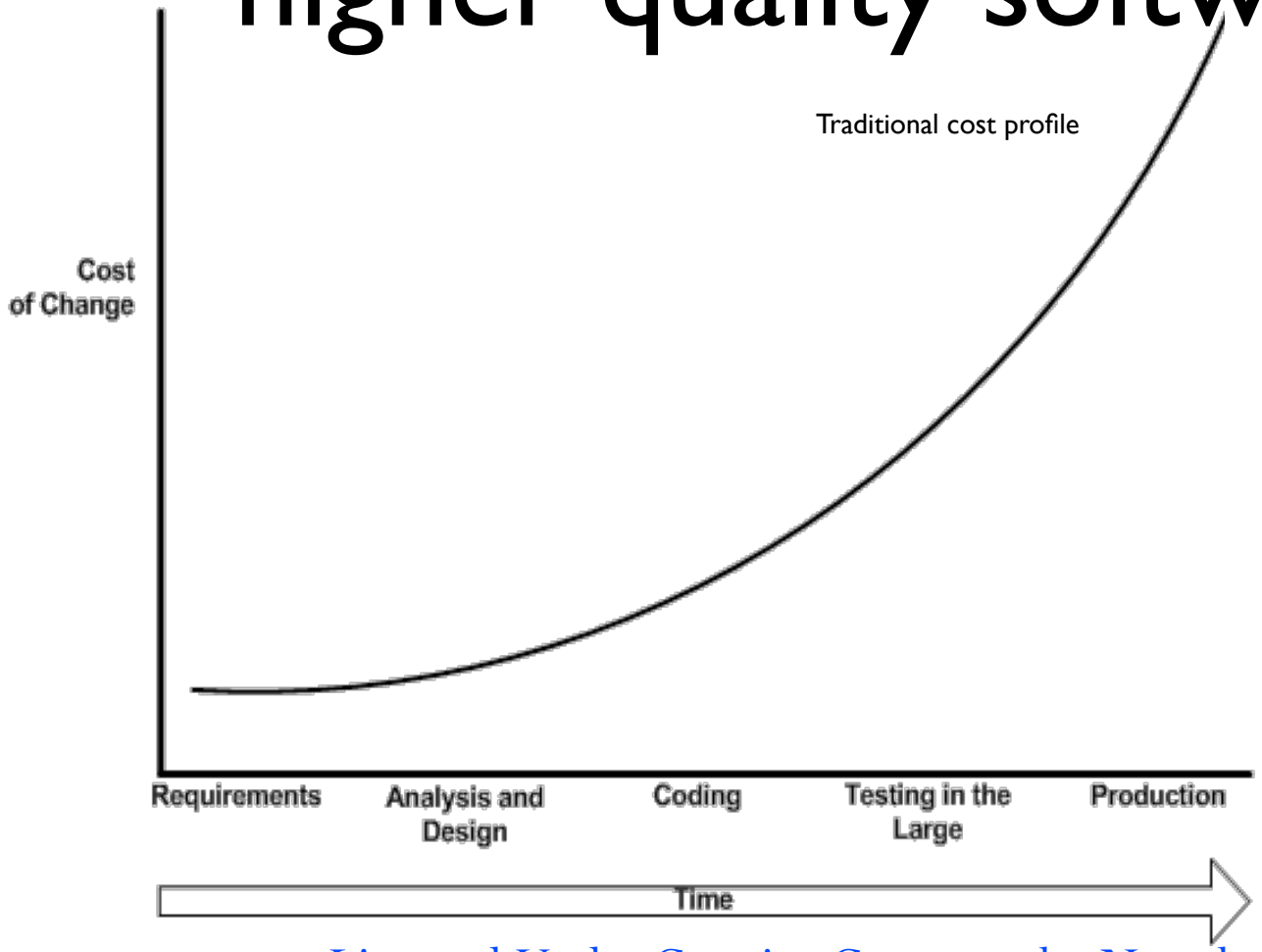
- ☑ "Last Defender of Quality" stance
- ☑ Strict change management
- ☑ Detailed preparation and up front planning
- ☑ Heavyweight documentation
- ☑ Strict entrance and exit criteria with sign-offs
- ☑ Heavyweight test automation focused on regression
- ☑ Attempts at process enforcement

Source Agile/QA Testing - Elisabeth Hendrickson

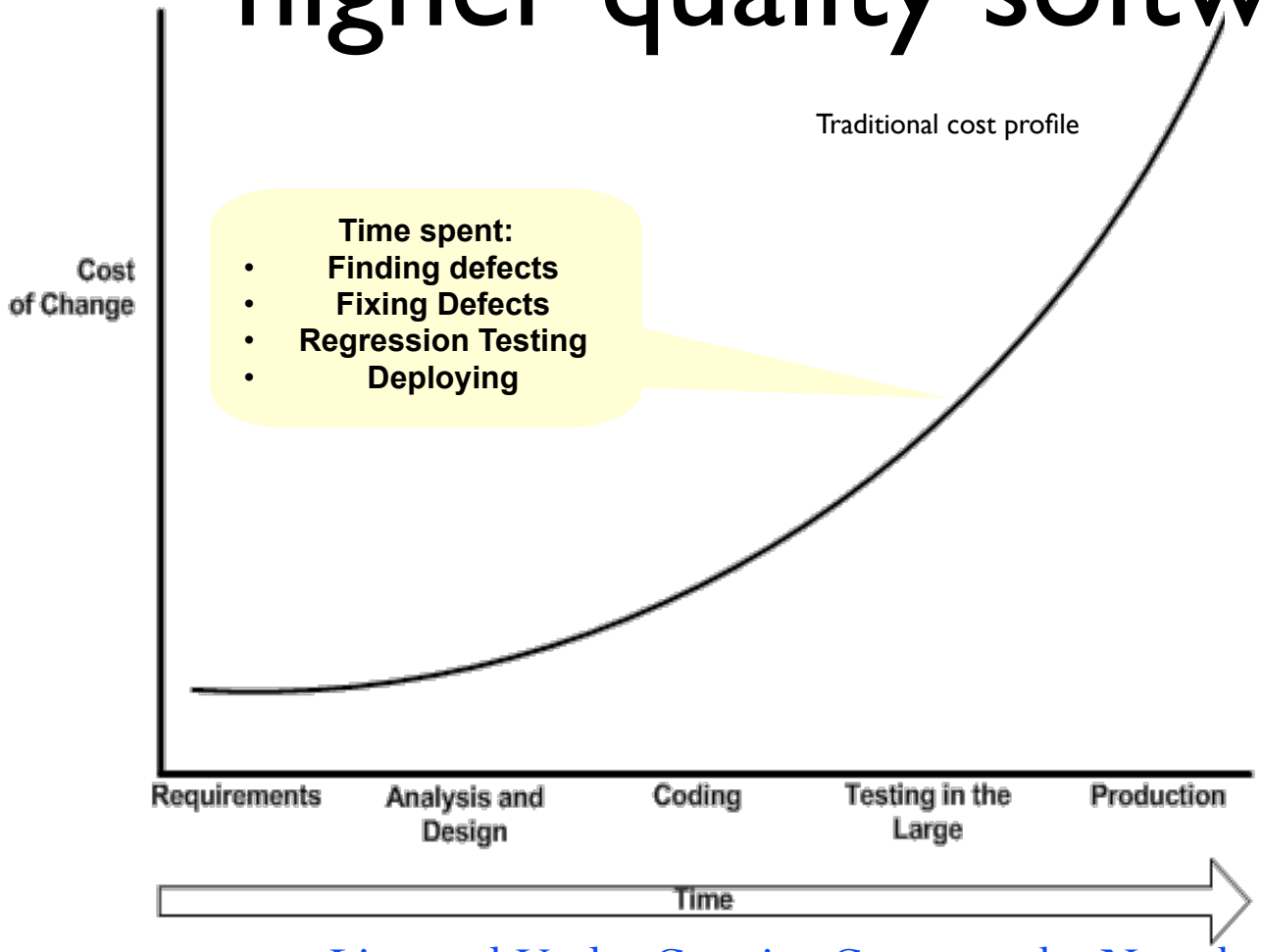
Project Management Triangle

Project Management Triangle

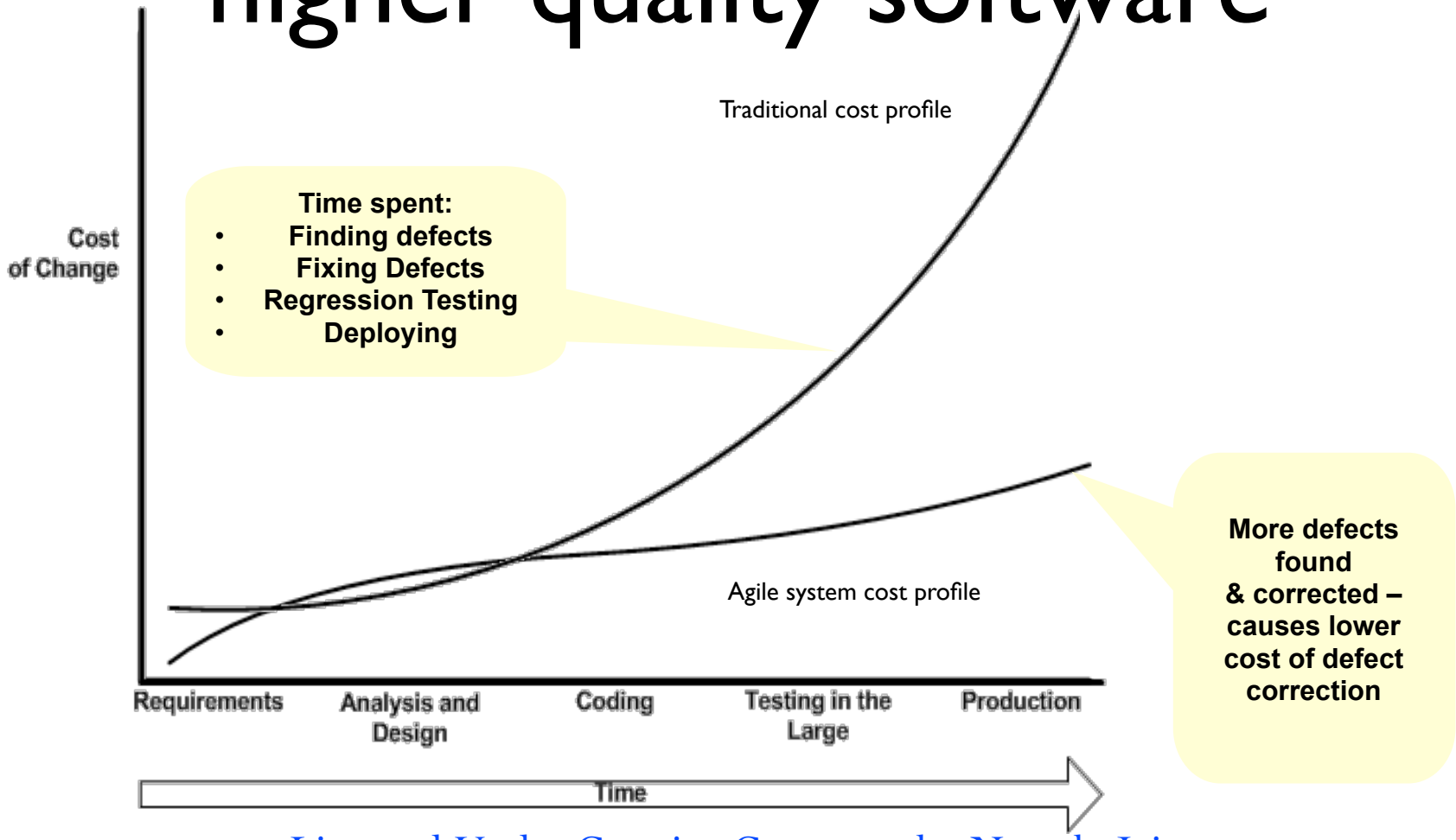
Lower cost of change through higher quality software



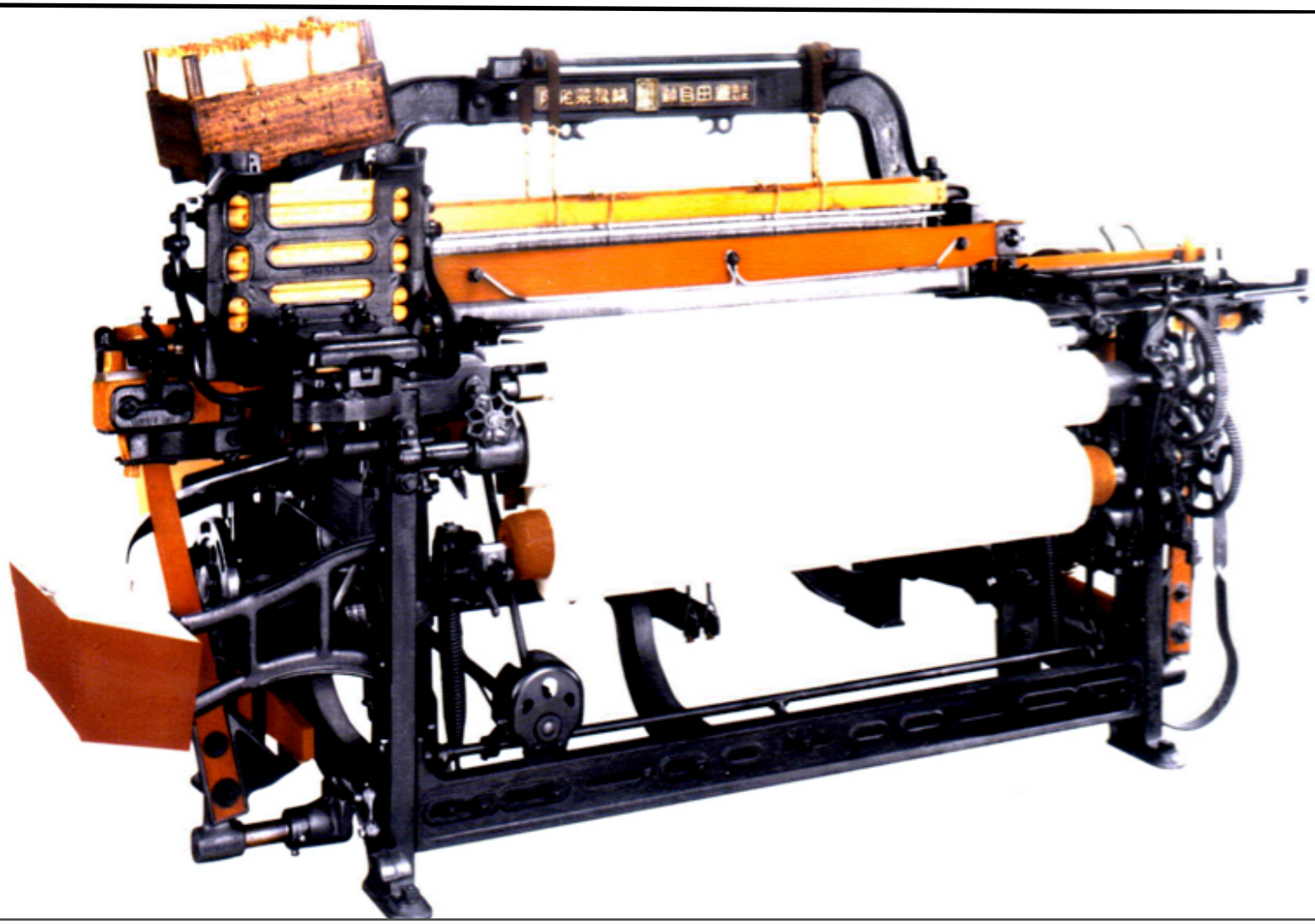
Lower cost of change through higher quality software



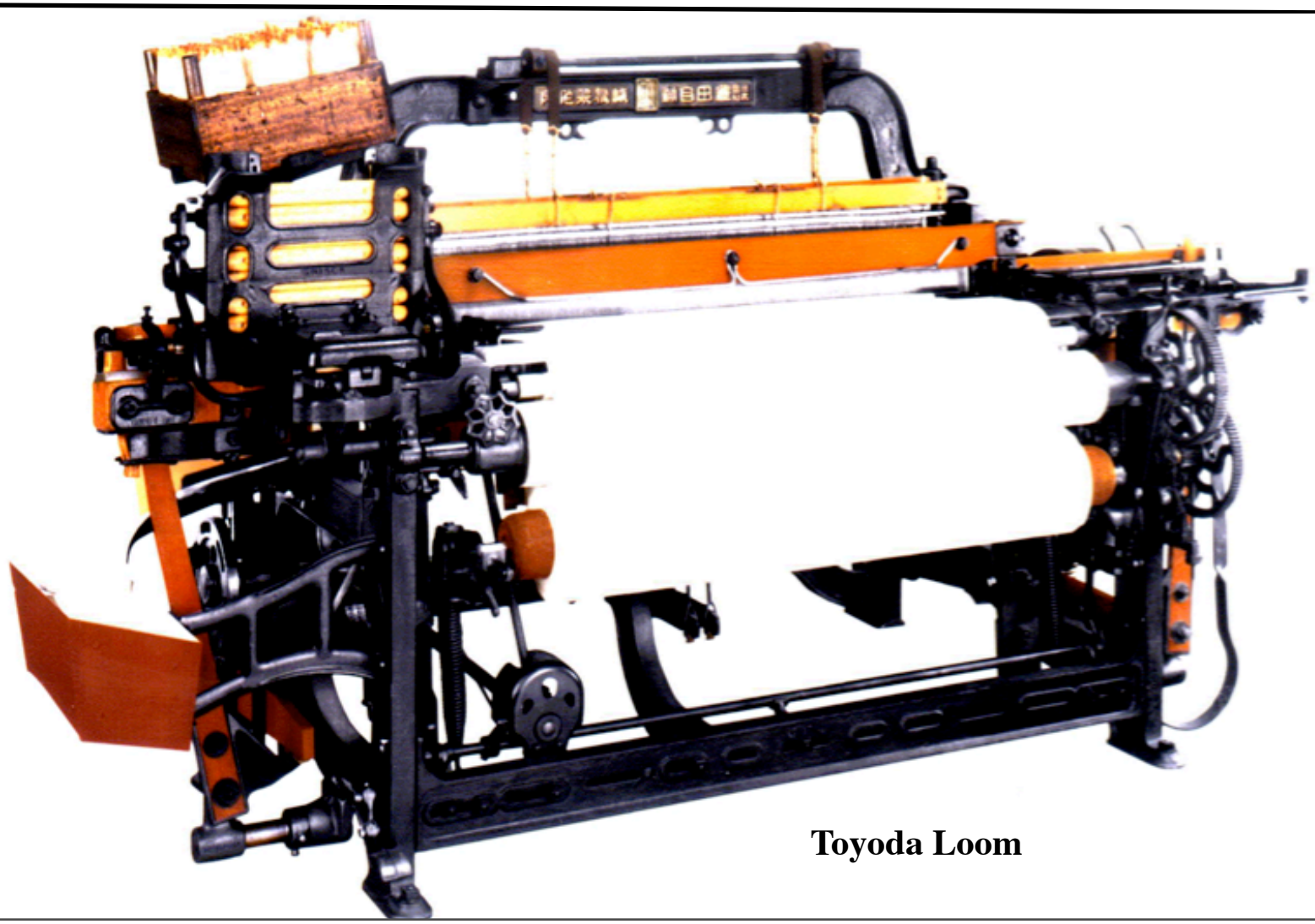
Lower cost of change through higher quality software



Building Quality into the Process

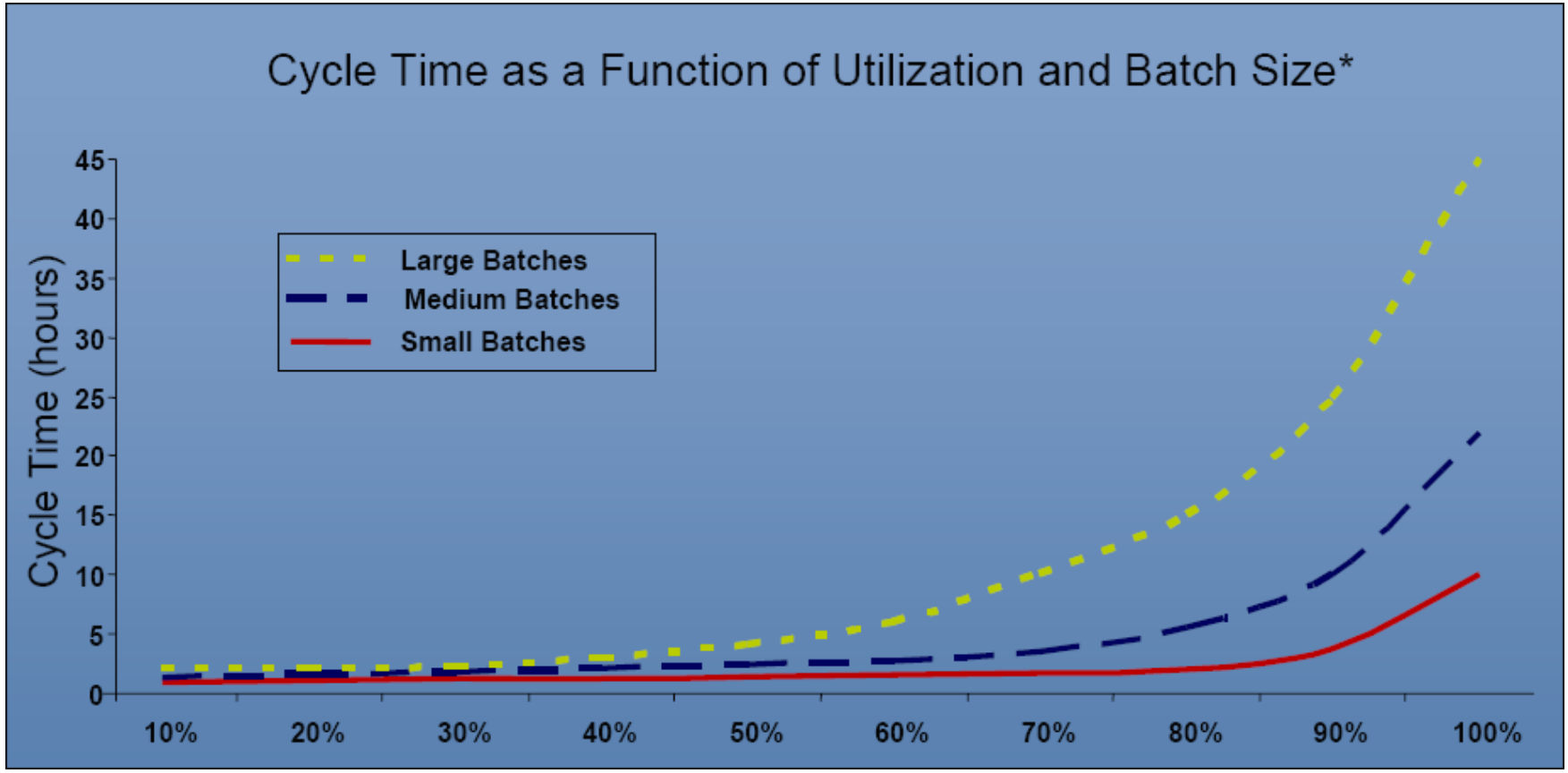


Building Quality into the Process



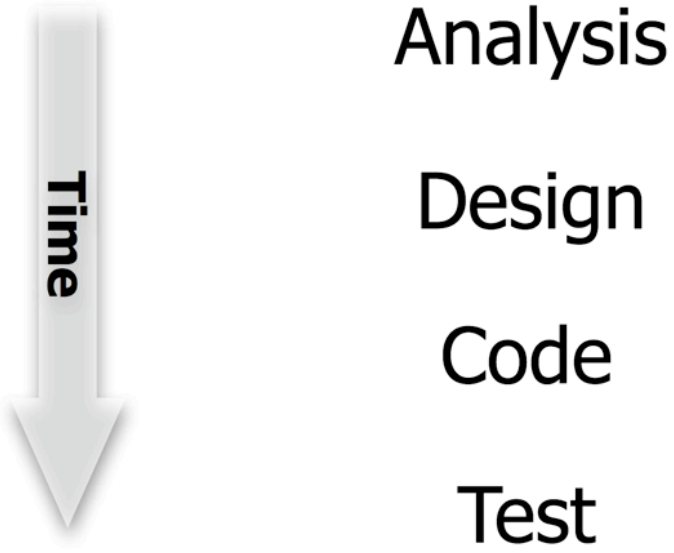
Toyoda Loom

Lessons from Queuing theory

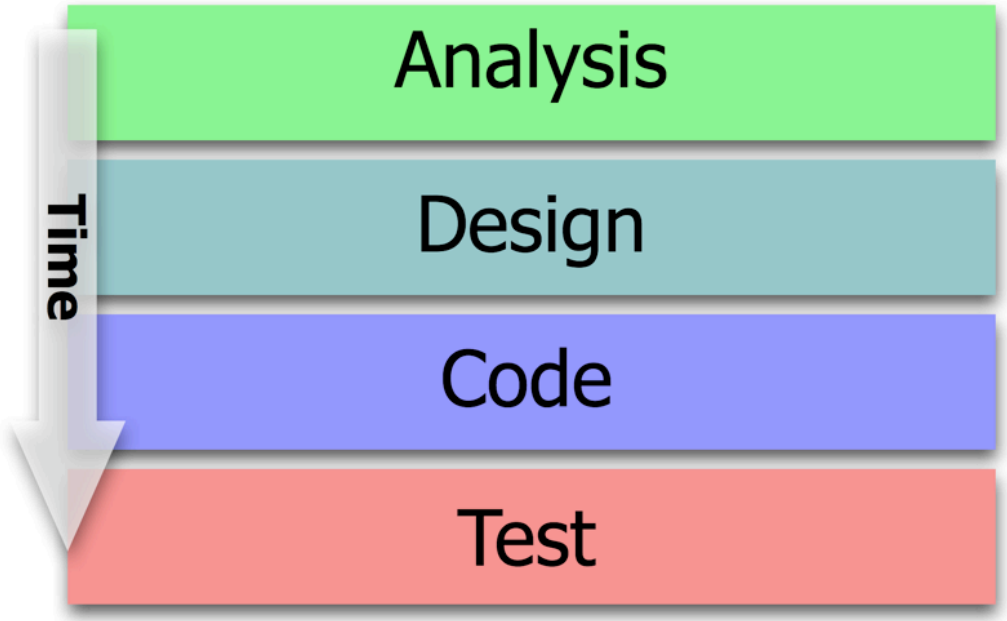


Source: Beyond Agile Software Development Becoming Lean, Mary Poppendieck, Poppendieck.llc

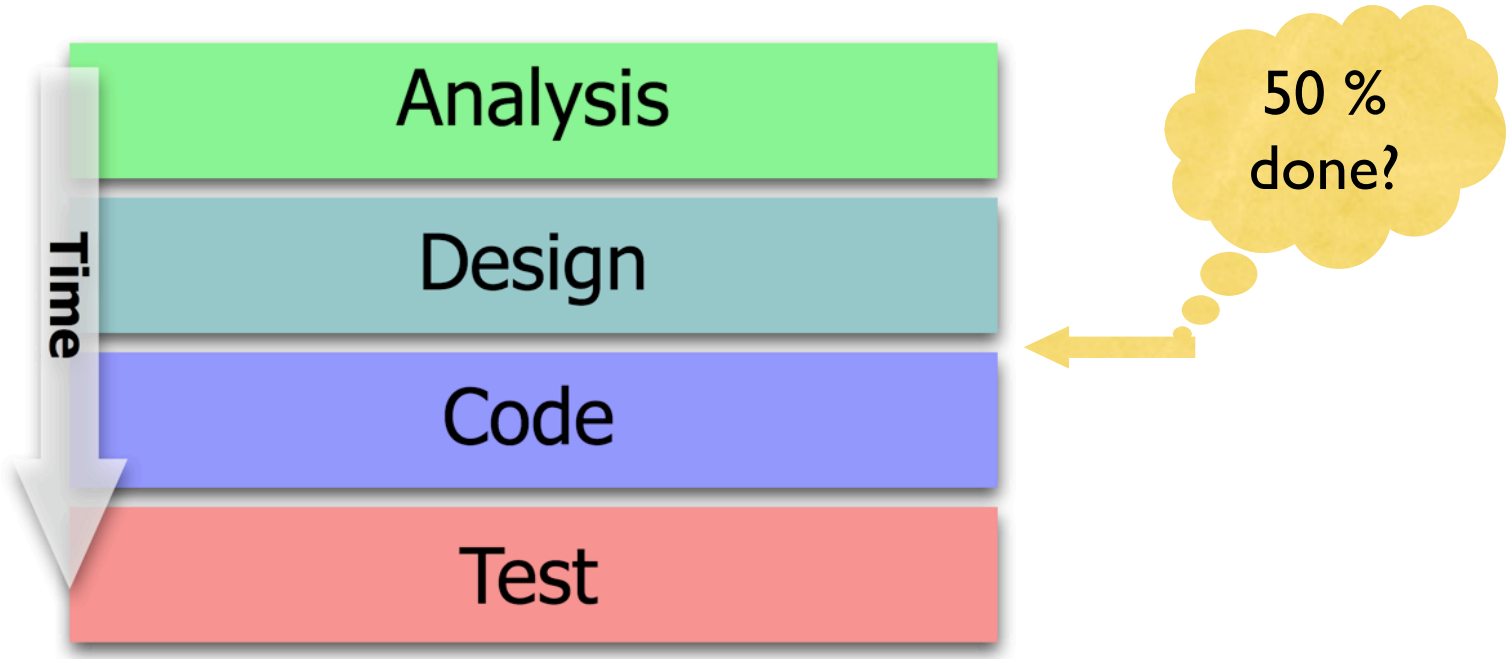
Traditional Software Development



Traditional Software Development



Traditional Software Development



Applying Lean Principles to Software Development ...

A better way of doing the same

Analysis

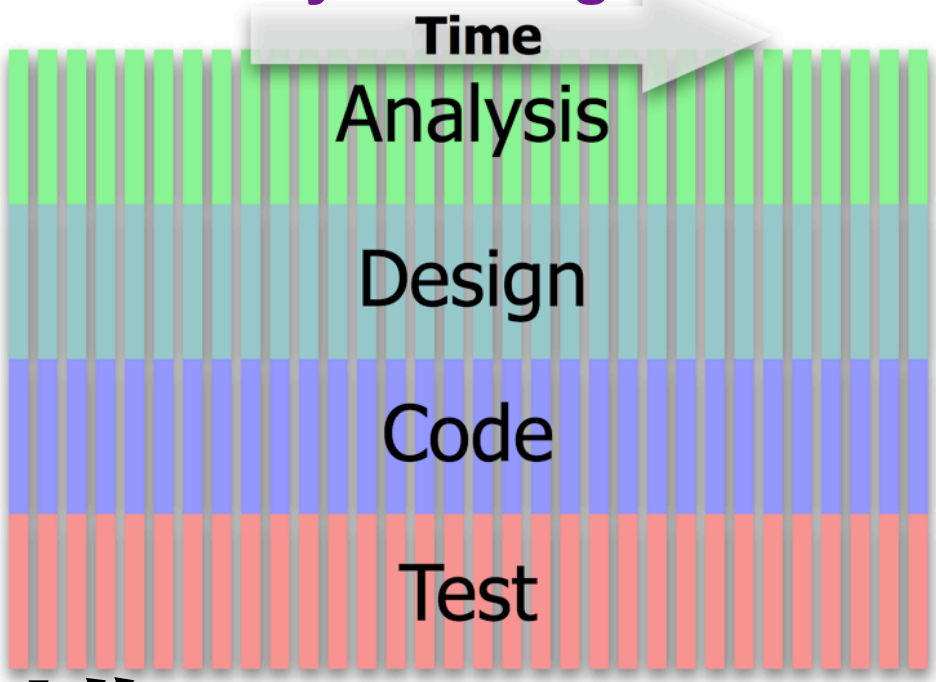
Design

Code

Test

Applying Lean Principles to Software Development ...

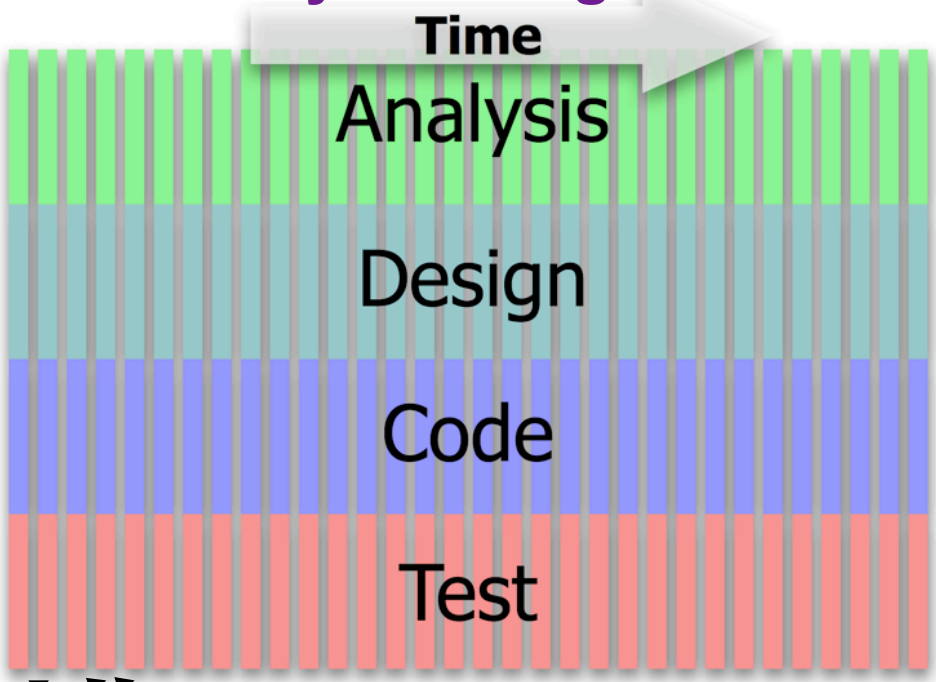
A better way of doing the same



End-to-End small slices of work

Applying Lean Principles to Software Development ...

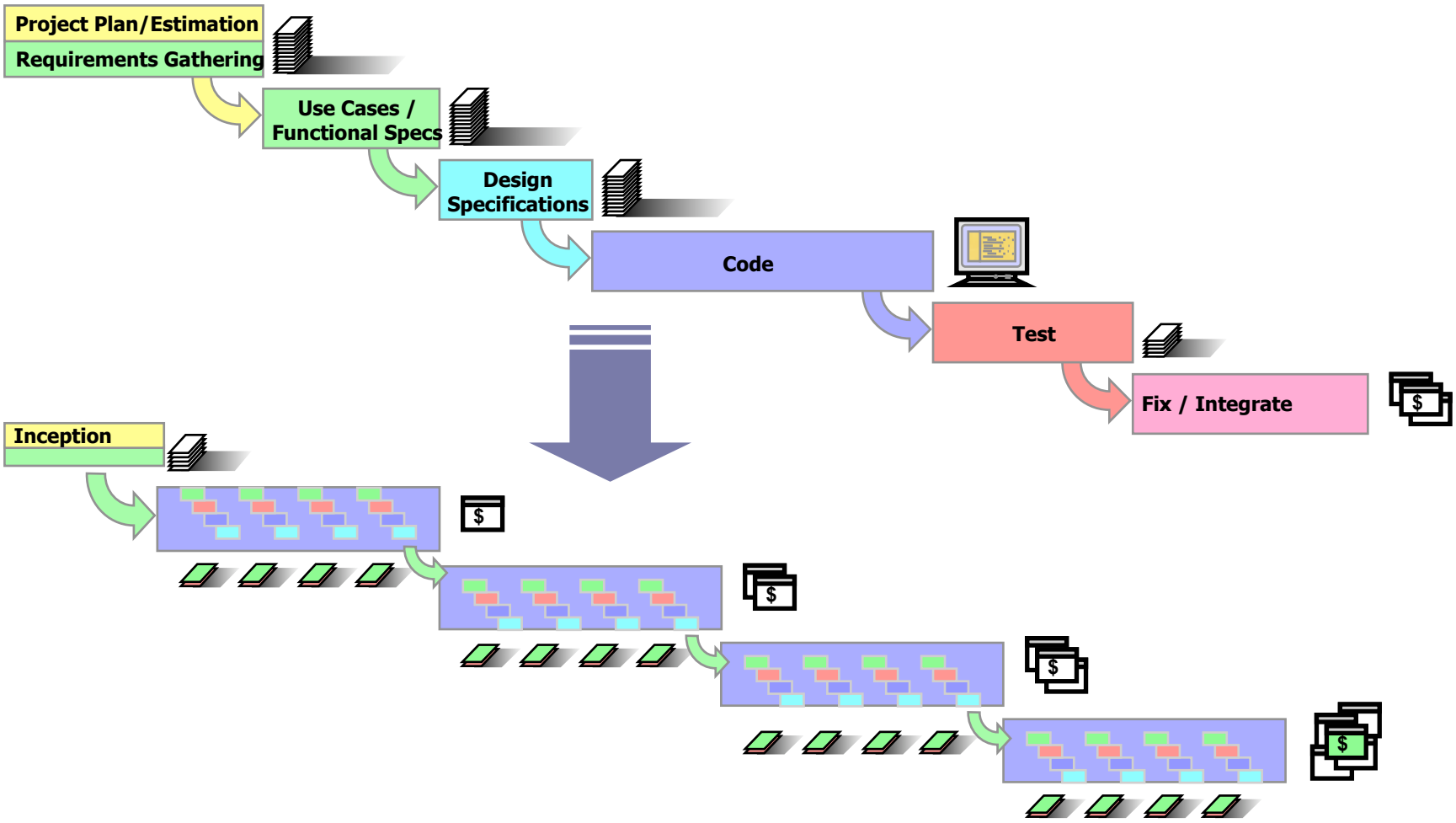
A better way of doing the same



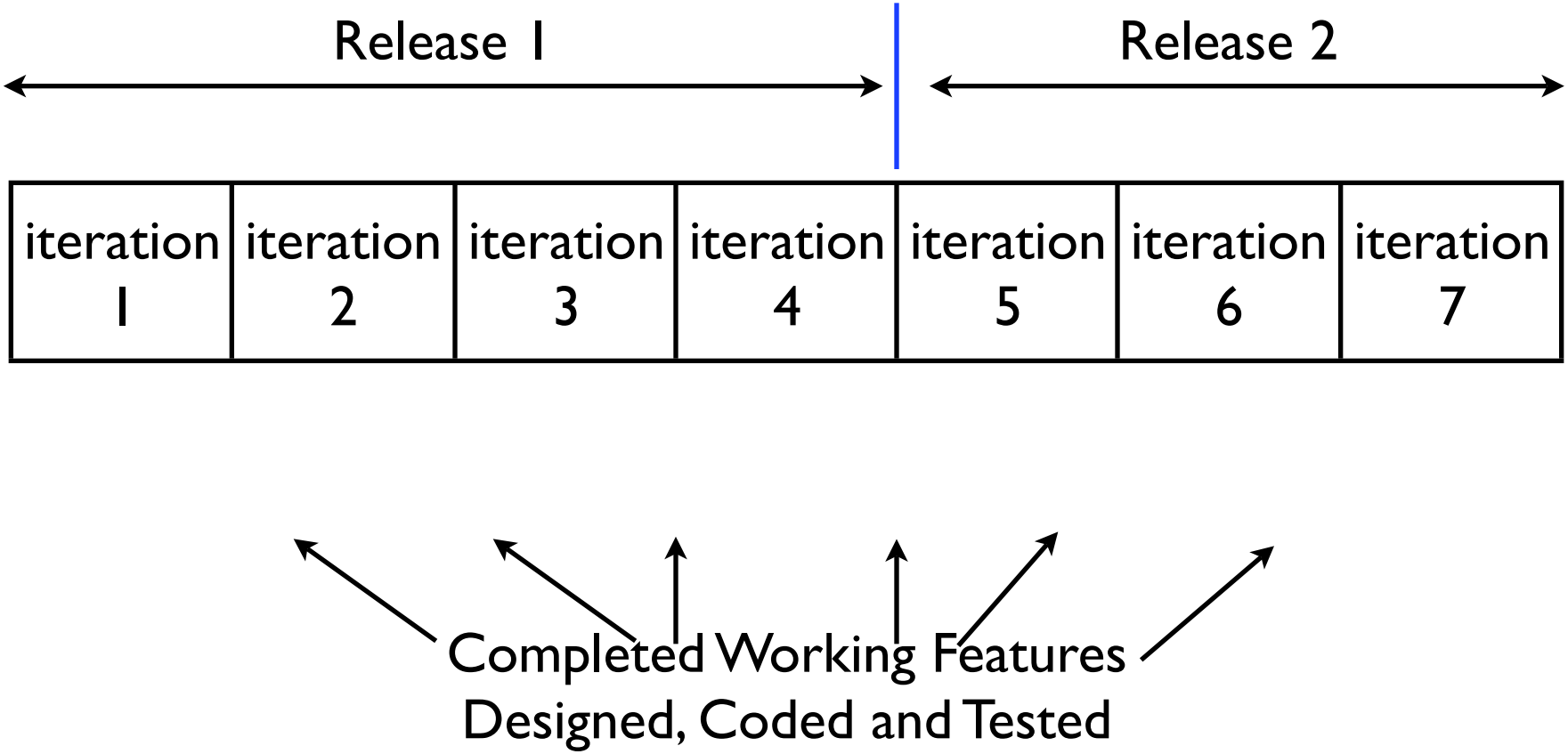
End-to-End small slices of work

20 % done = 100 % usable

Traditional vs Agile

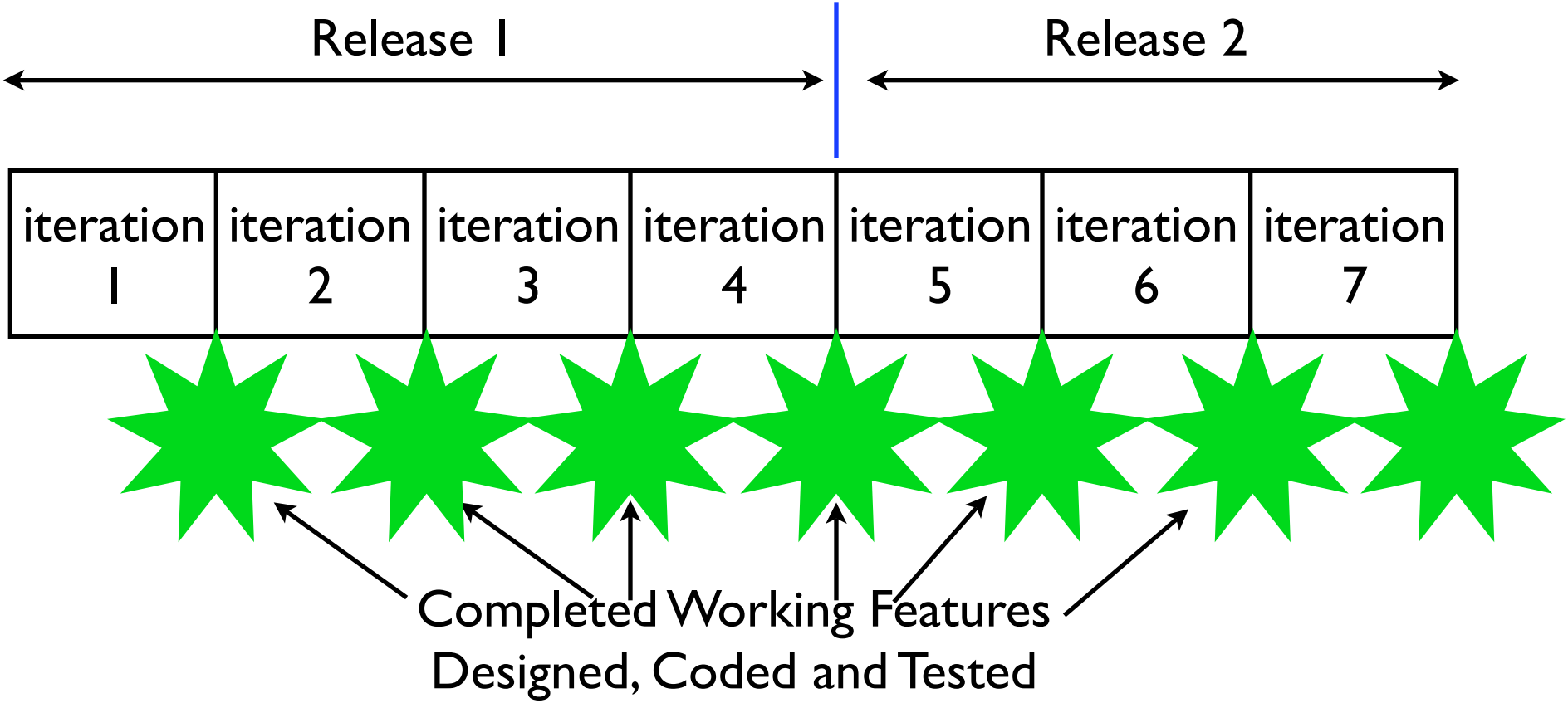


Agile = Continuous Stream of Value



Source Agile/QA Testing - Elisabeth Hendrickson

Agile = Continuous Stream of Value



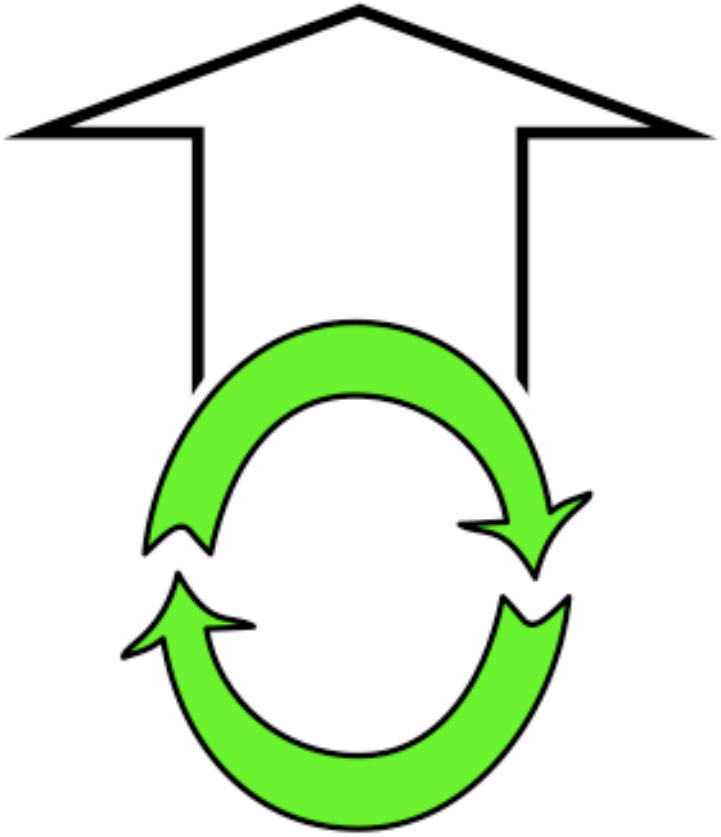
Source Agile/QA Testing - Elisabeth Hendrickson

Becoming Agile : Delivering Value

Increase the rate of
delivery
(usually with smaller increments)

Source Agile/QA Testing - Elisabeth Hendrickson

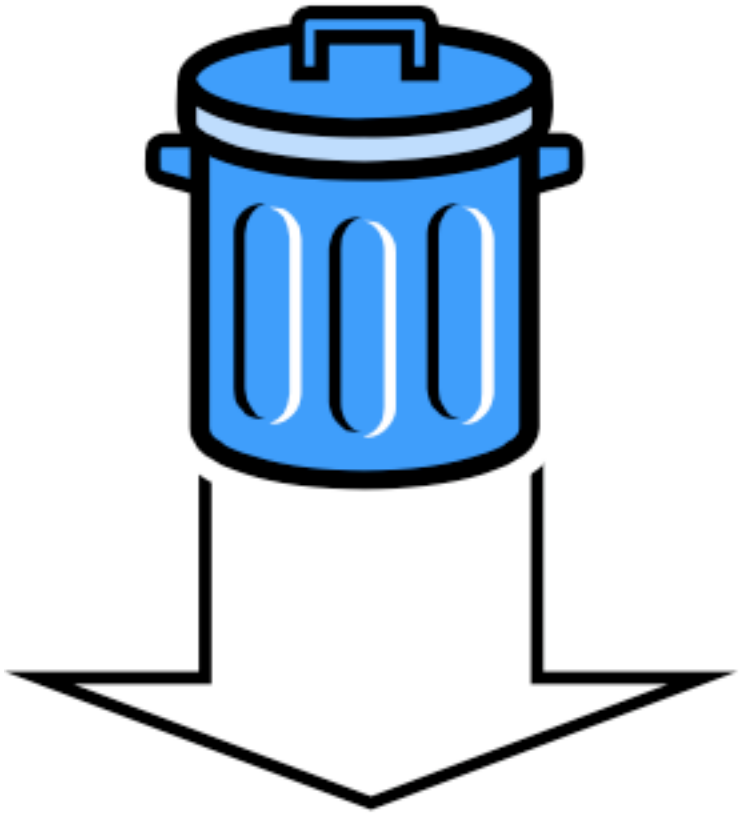
Becoming Agile : Increasing Feedback



Increase the rate
and quality
of ***feedback***

Source Agile/QA Testing - Elisabeth Hendrickson

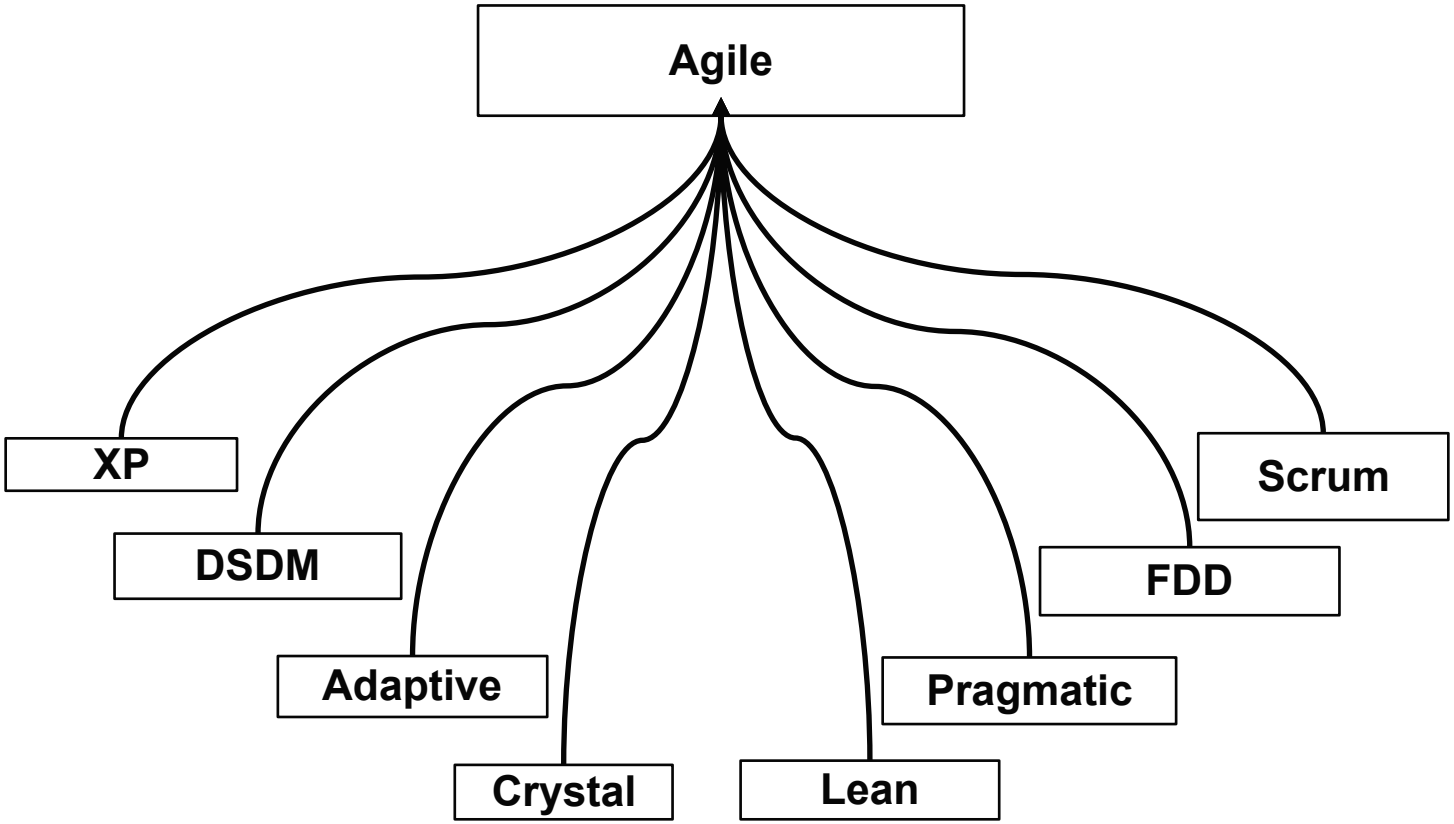
Becoming Agile : Reducing Waste



Reduce ***waste***

Source Agile/QA Testing - Elisabeth Hendrickson

Agile Umbrella



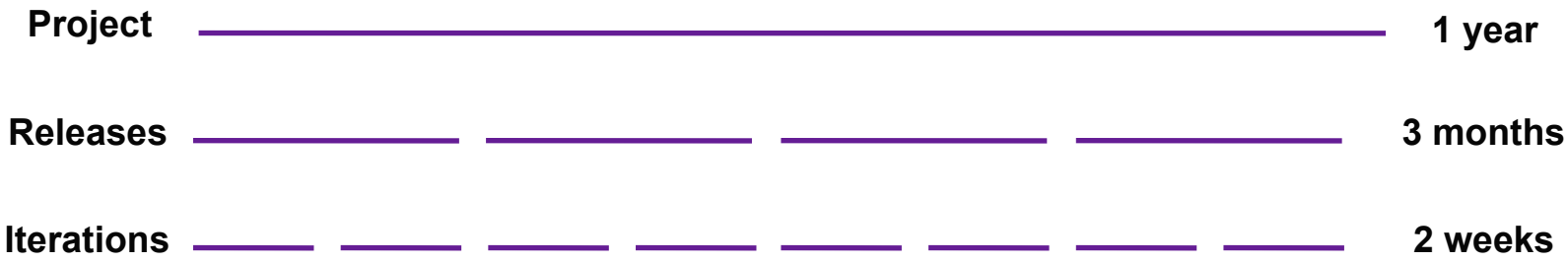
A typical XP Project

A typical XP Project

A typical XP Project

agile faqs

A typical XP Project

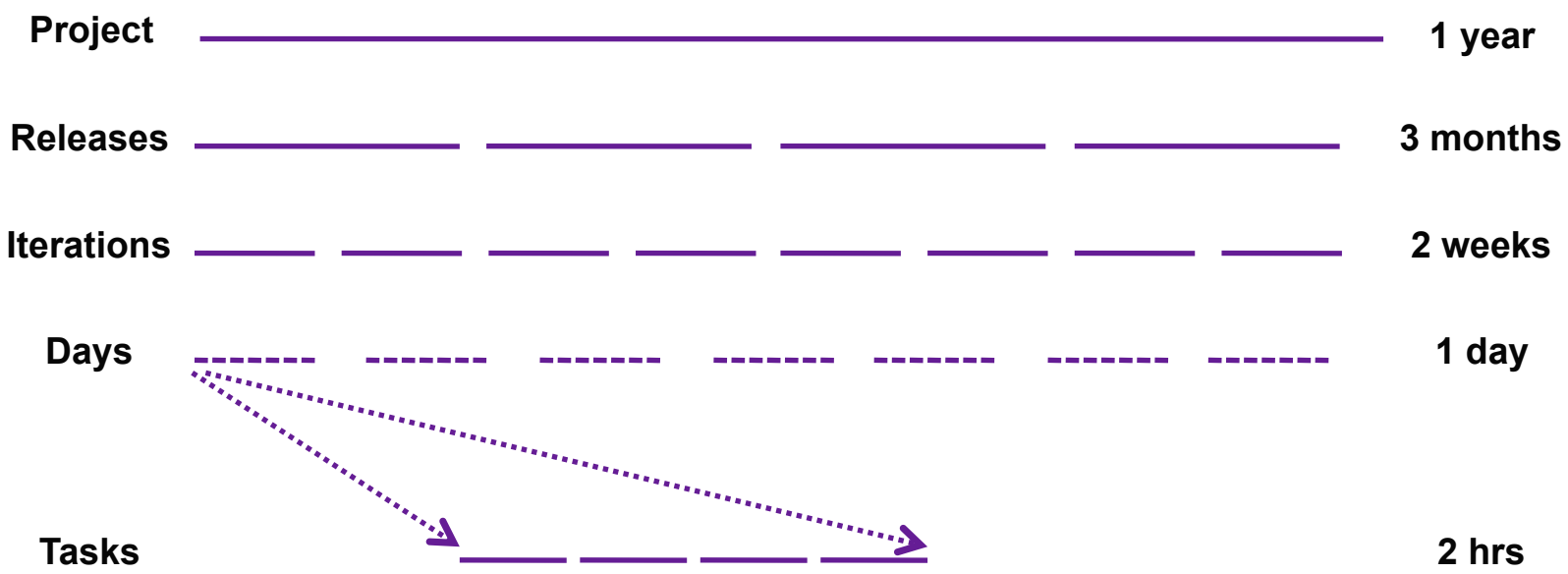


agile faqs

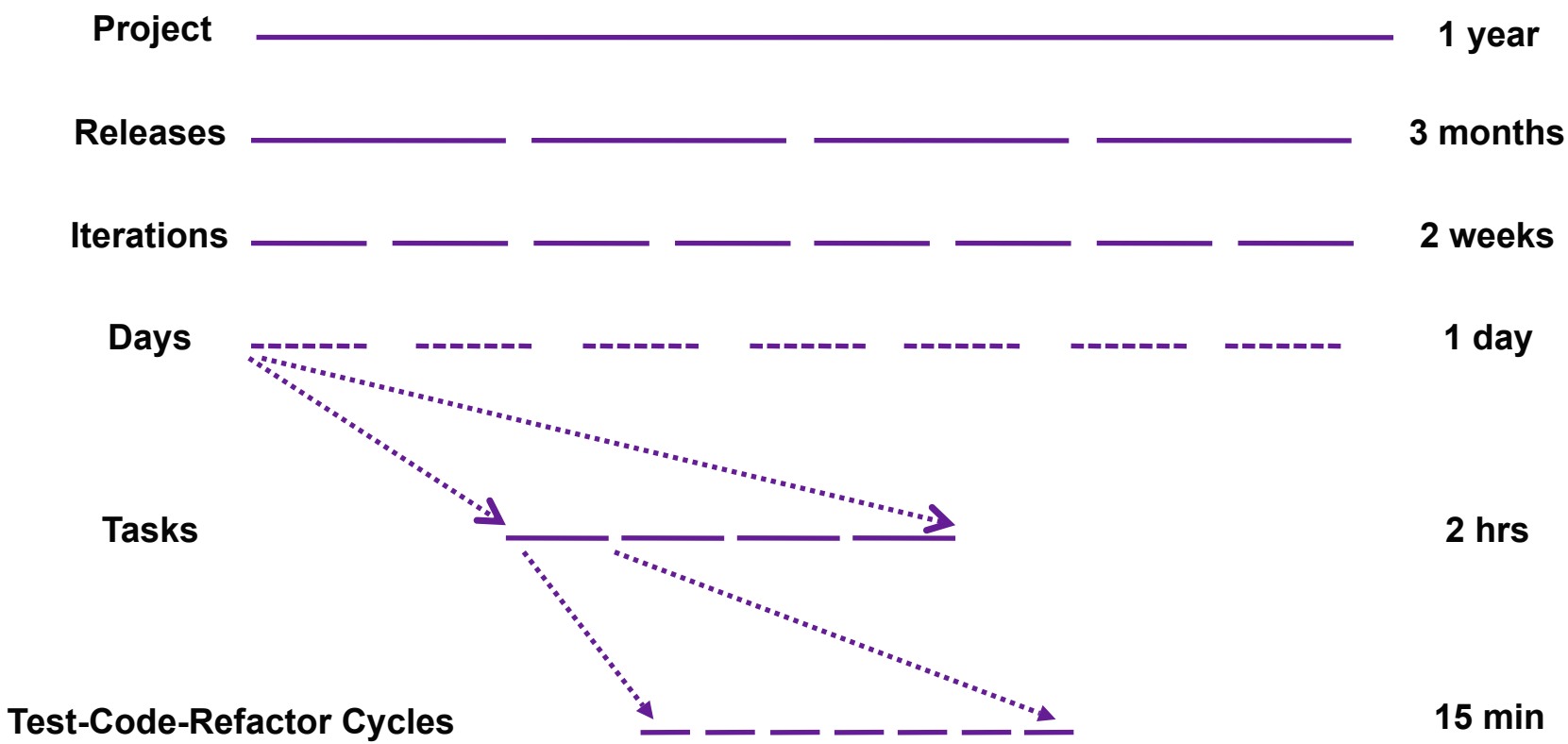
A typical XP Project

agile faqs

A typical XP Project

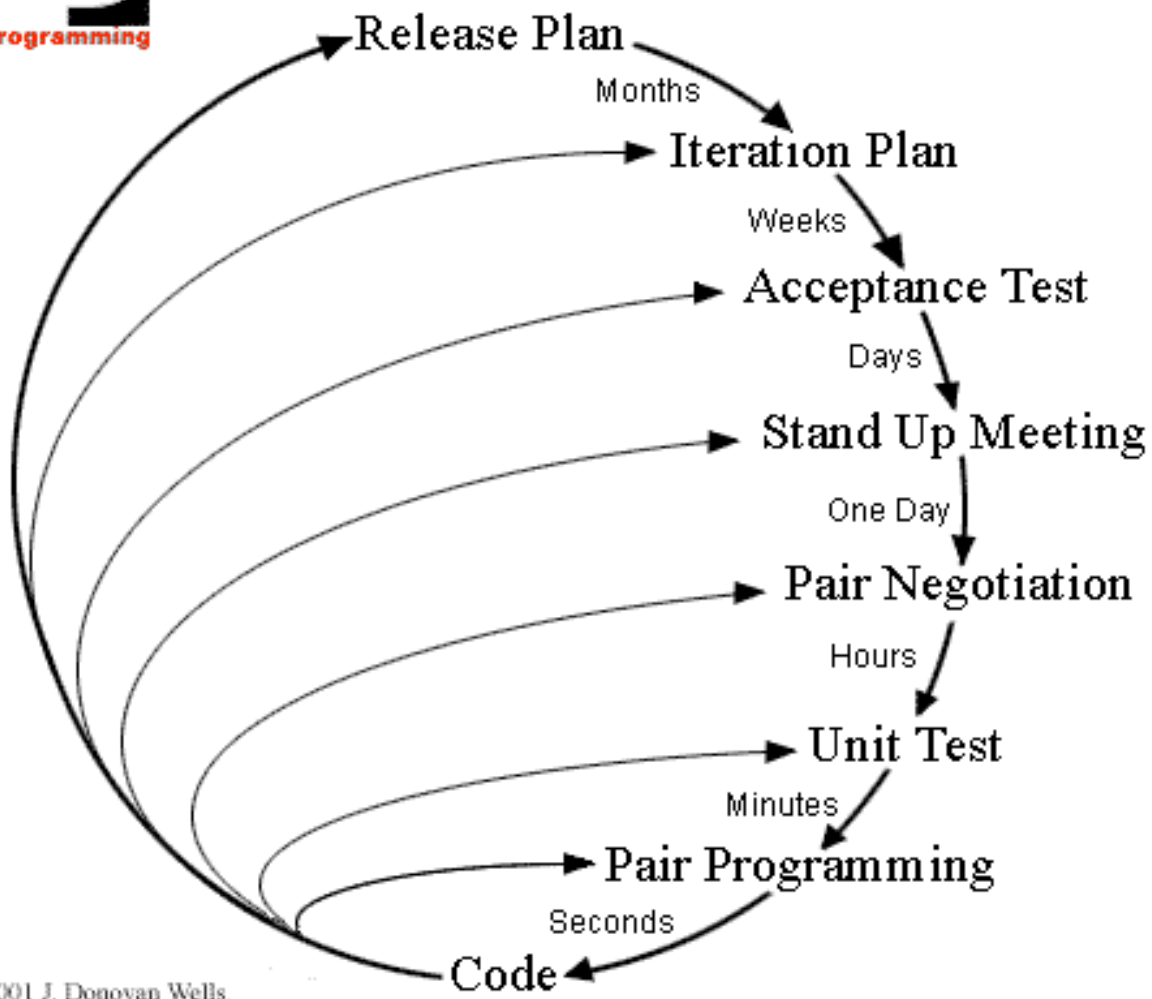


A typical XP Project



Planning/Feedback Loops

Zoom Out



Source : http://www.extremeprogramming.org

Key Questions

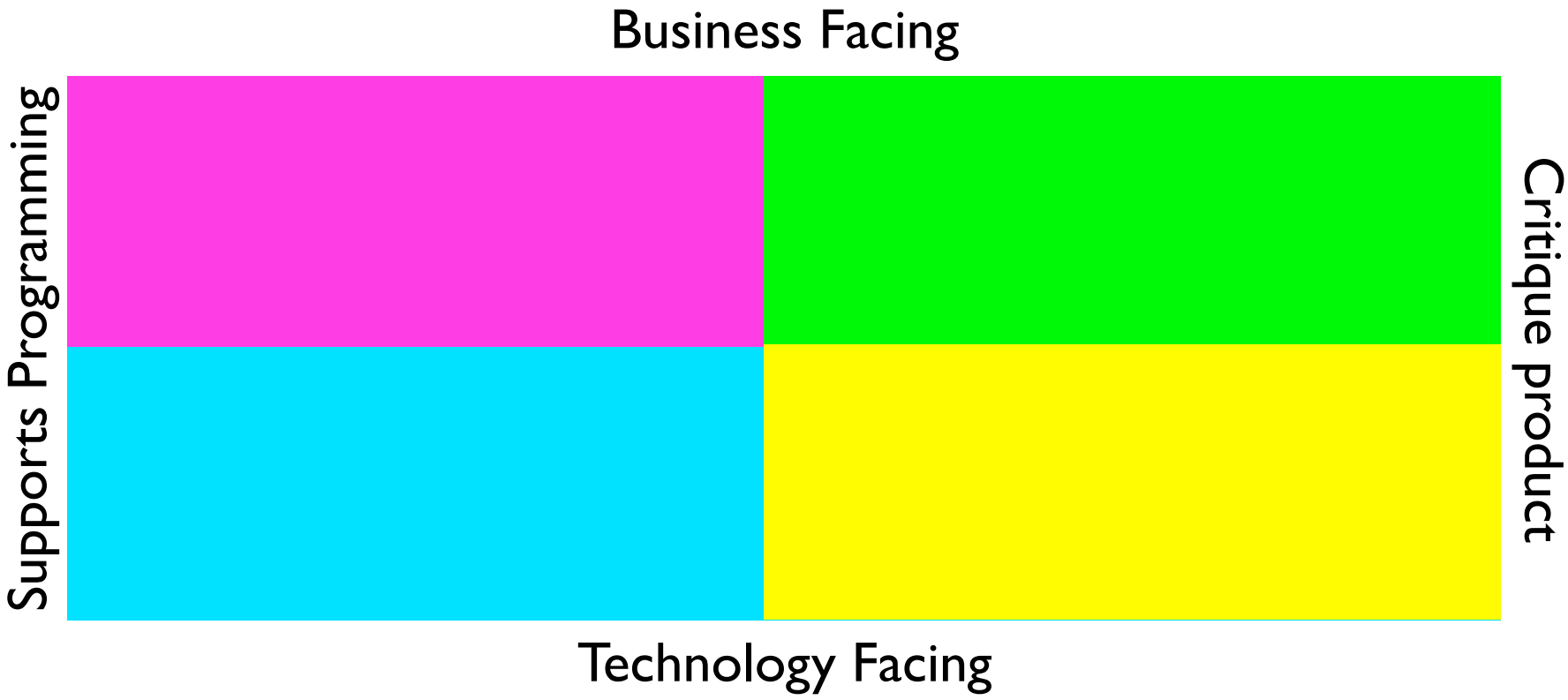
Business Facing

Are we building the right product?

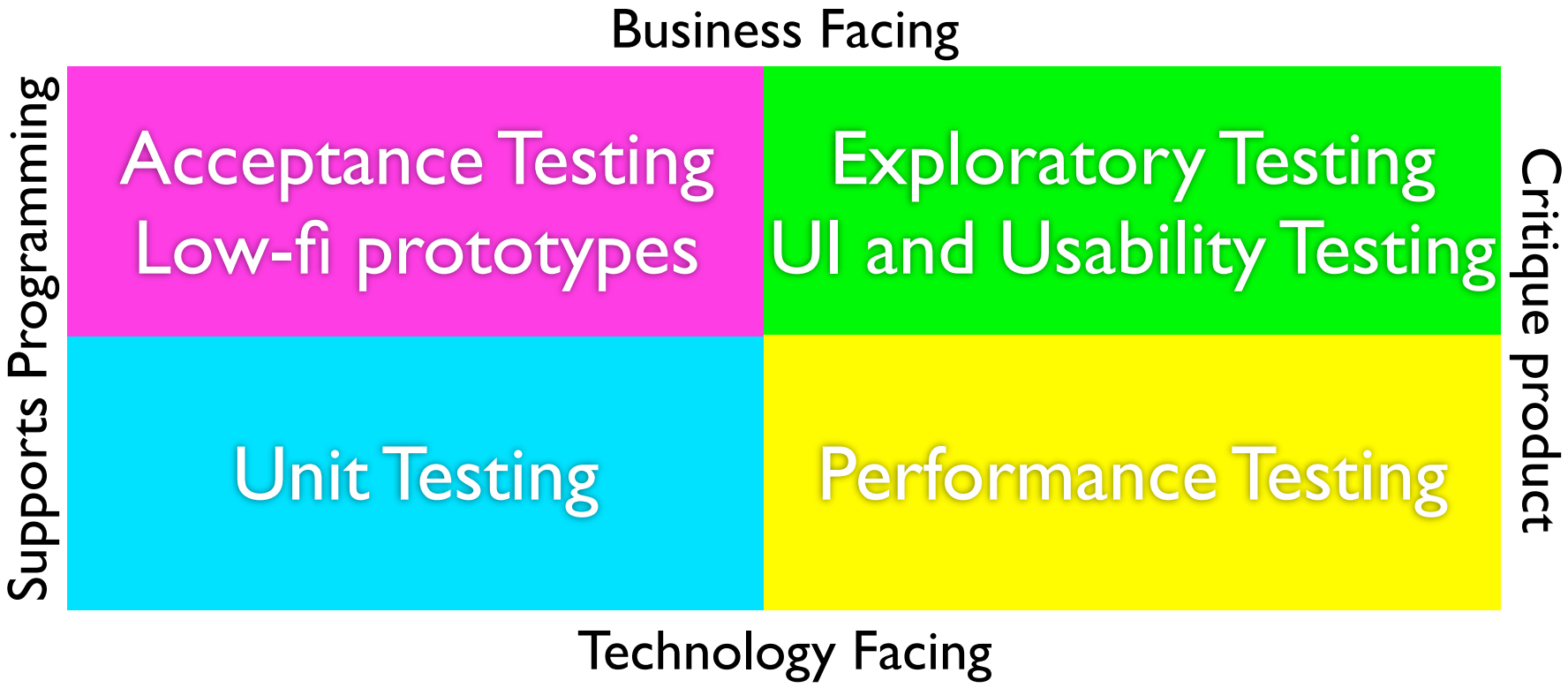
Are we building the product right?

Technology Facing

Brian Marick's Test Categorization



Brian Marick's Test Categorization

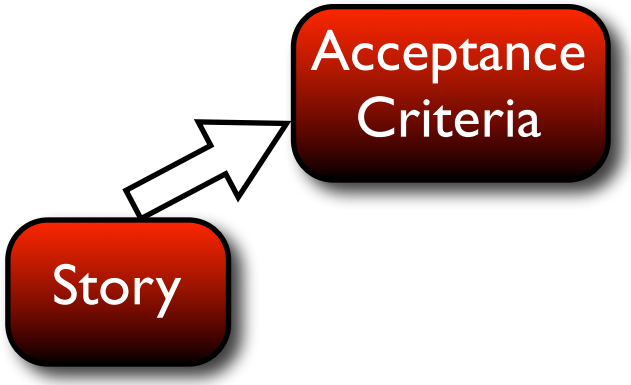


agile faqs

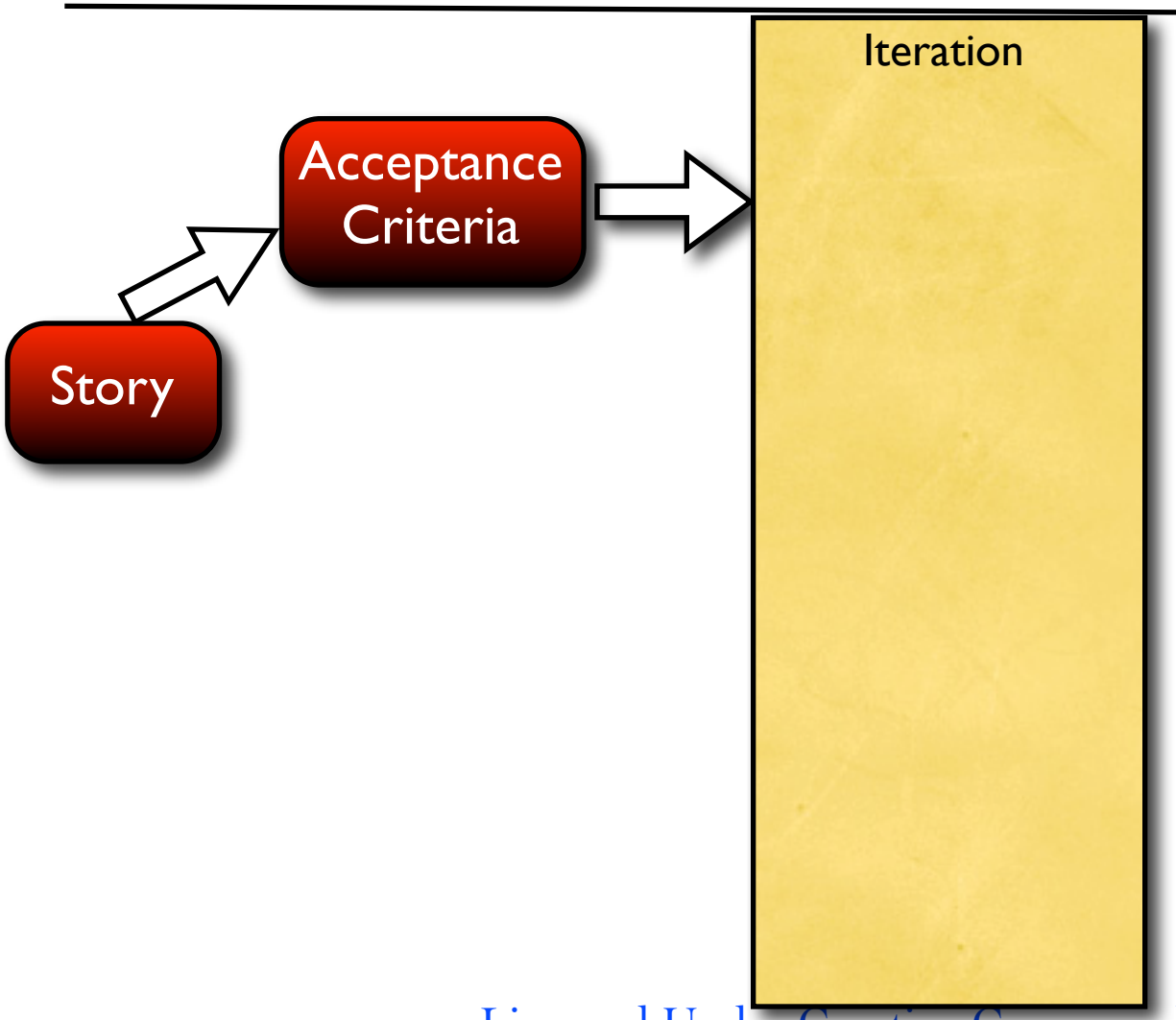
Acceptance Test Driven Development

Acceptance Test Driven Development

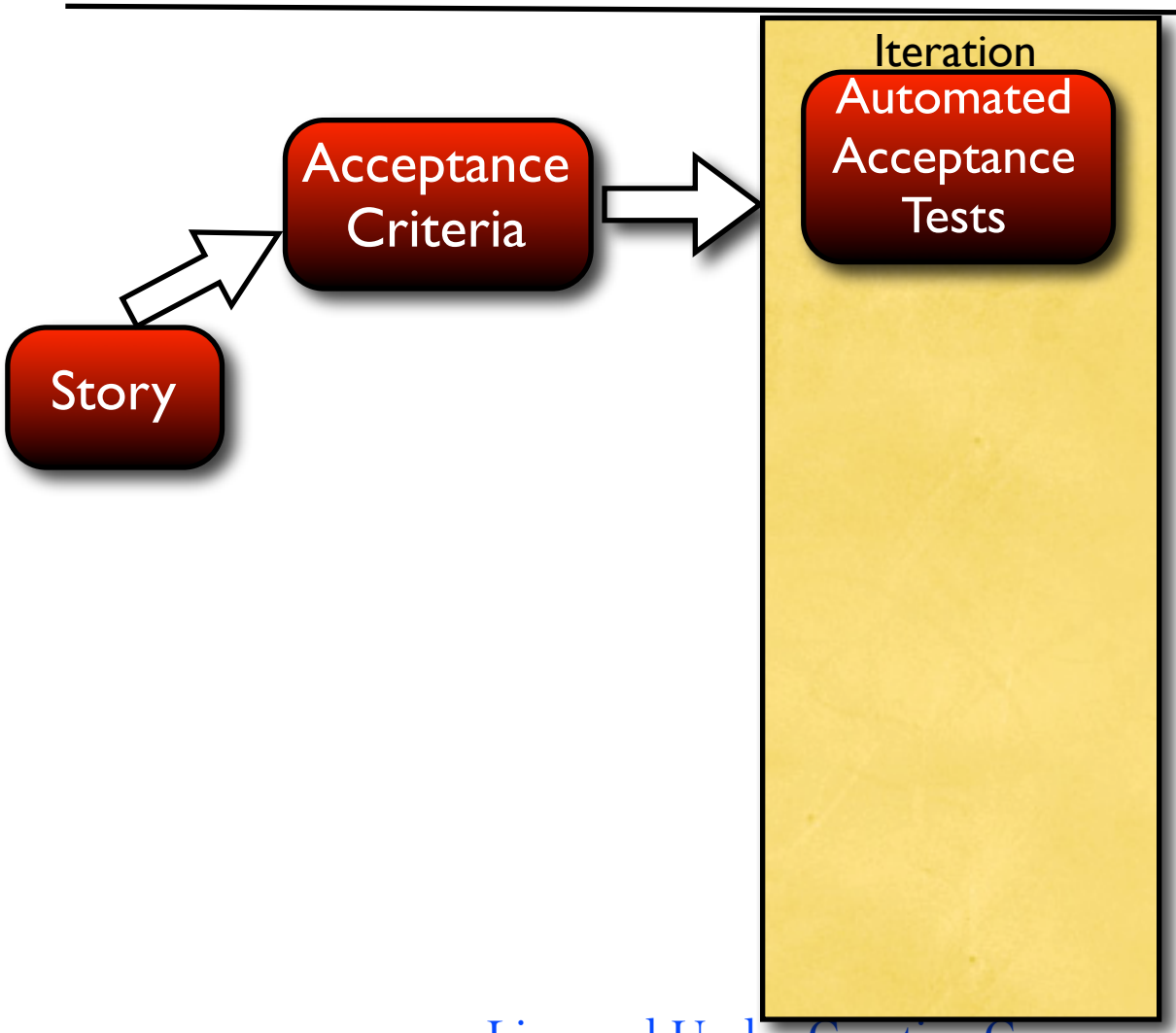
Acceptance Test Driven Development



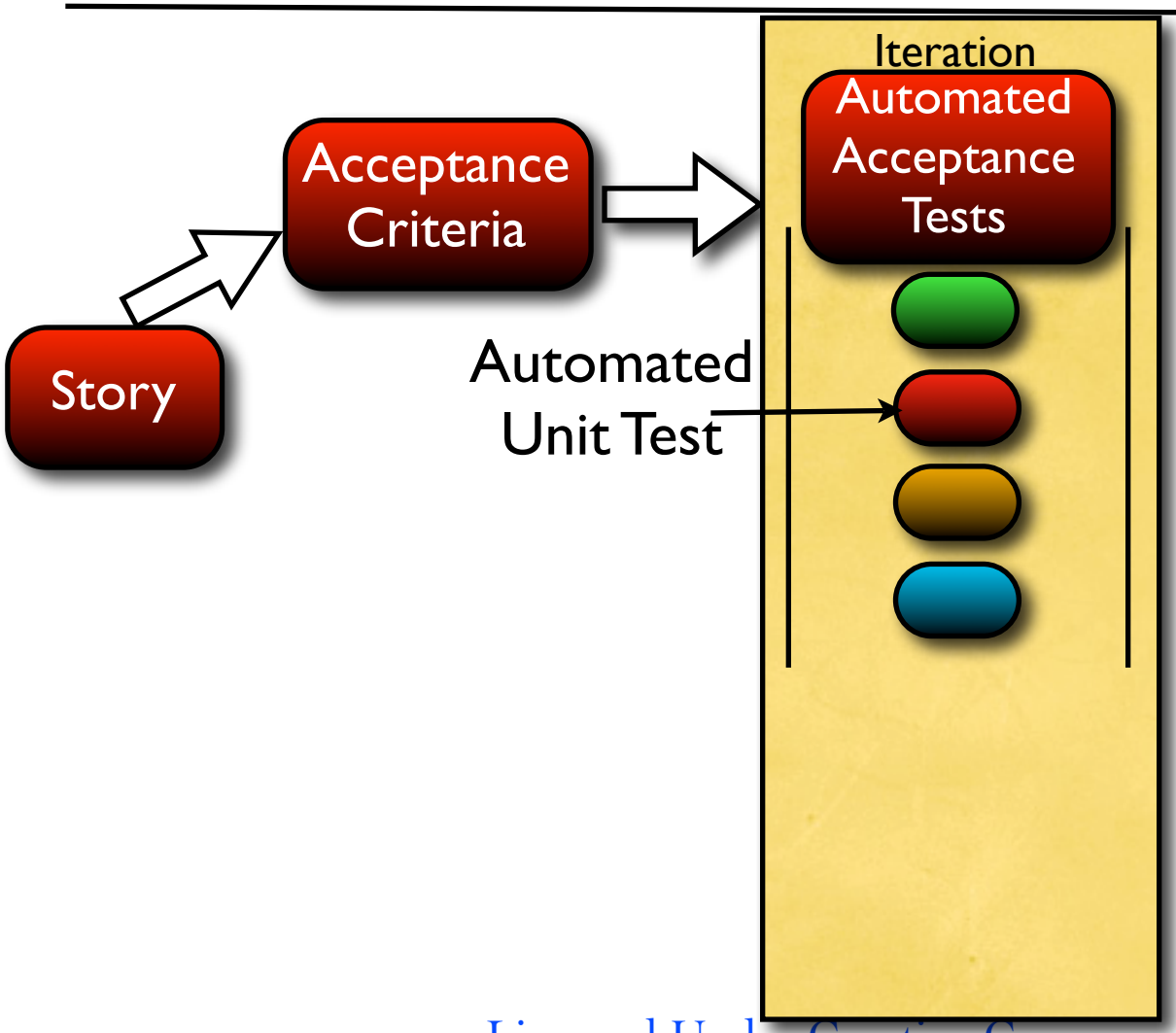
Acceptance Test Driven Development



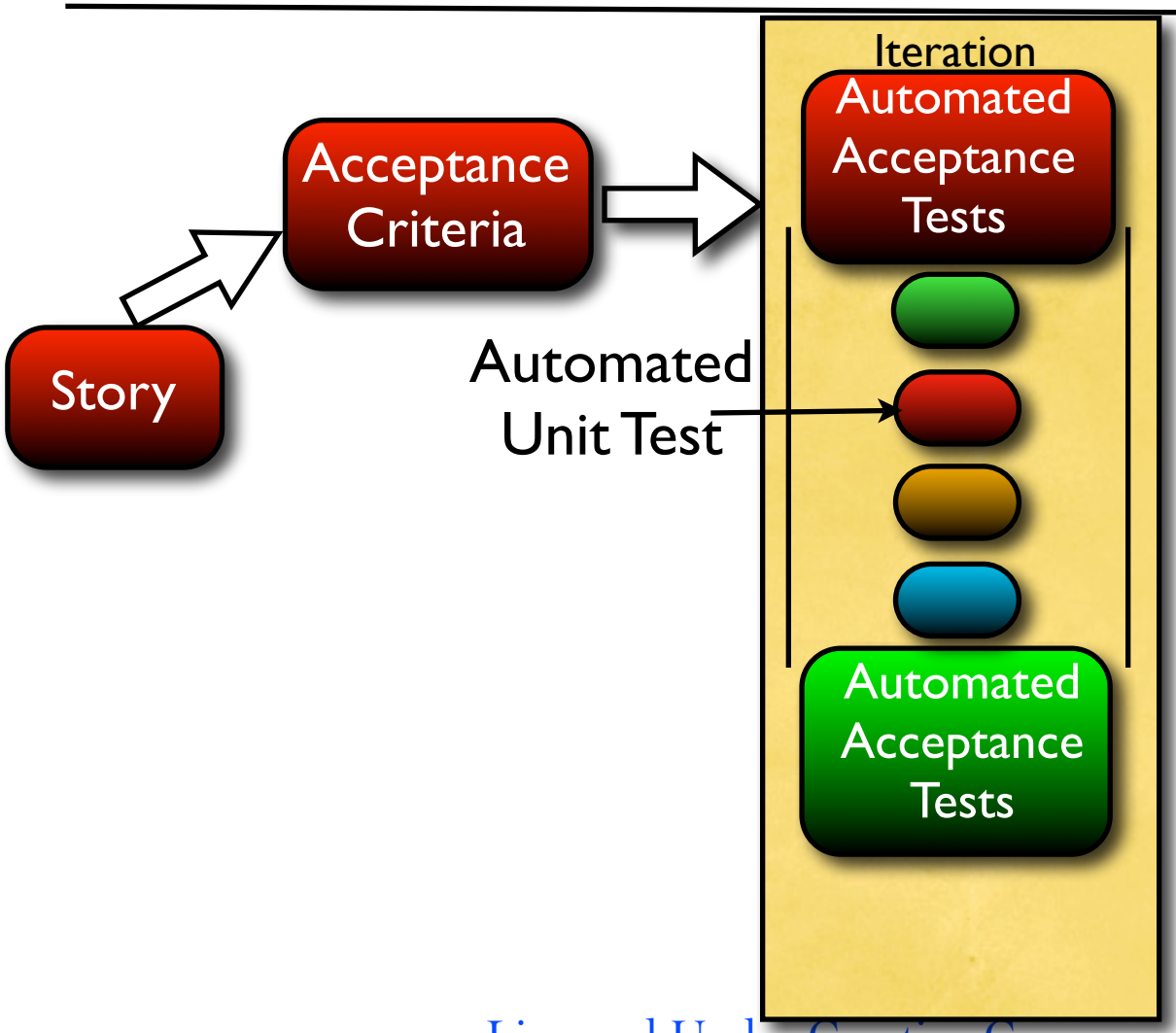
Acceptance Test Driven Development



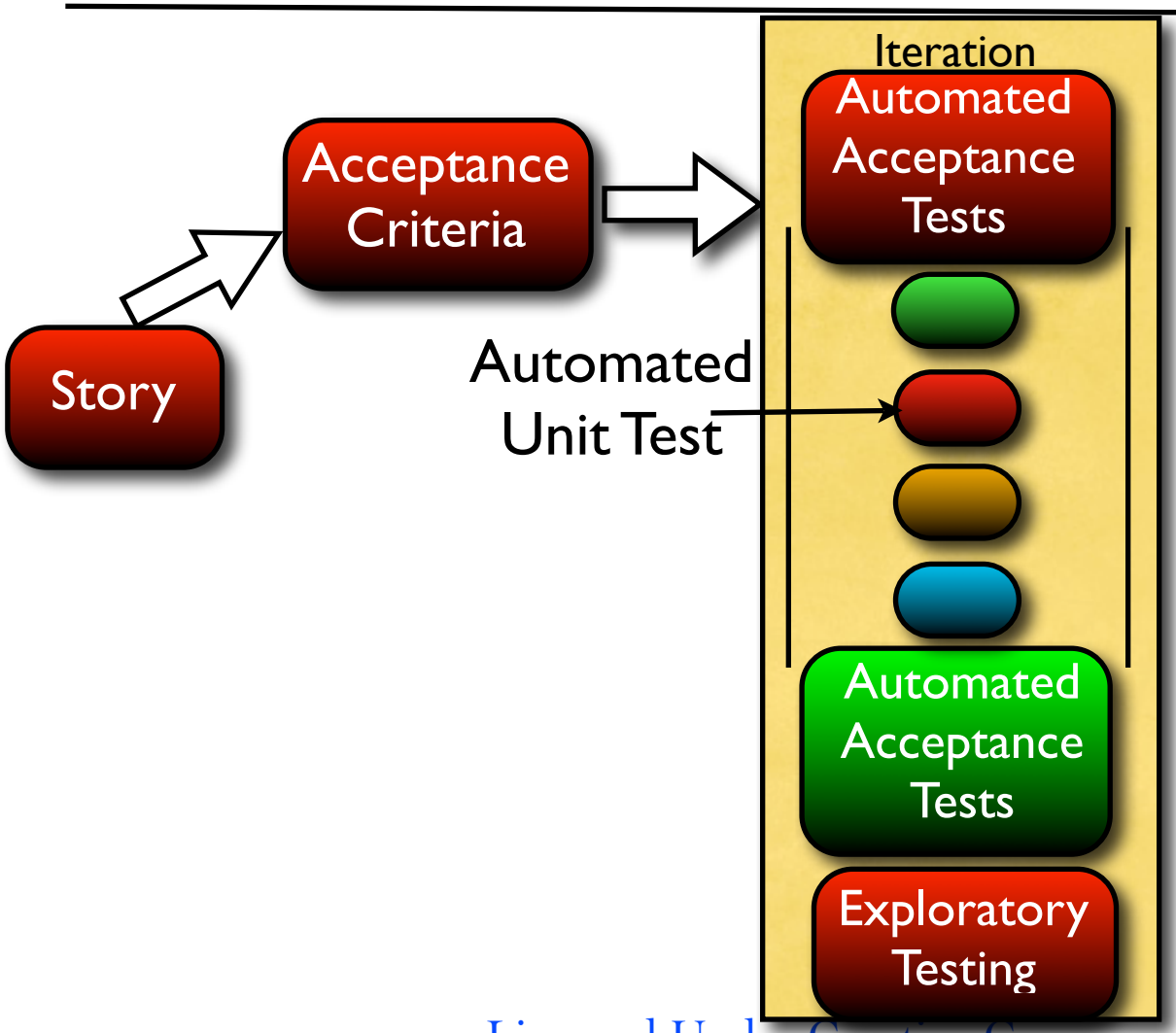
Acceptance Test Driven Development



Acceptance Test Driven Development

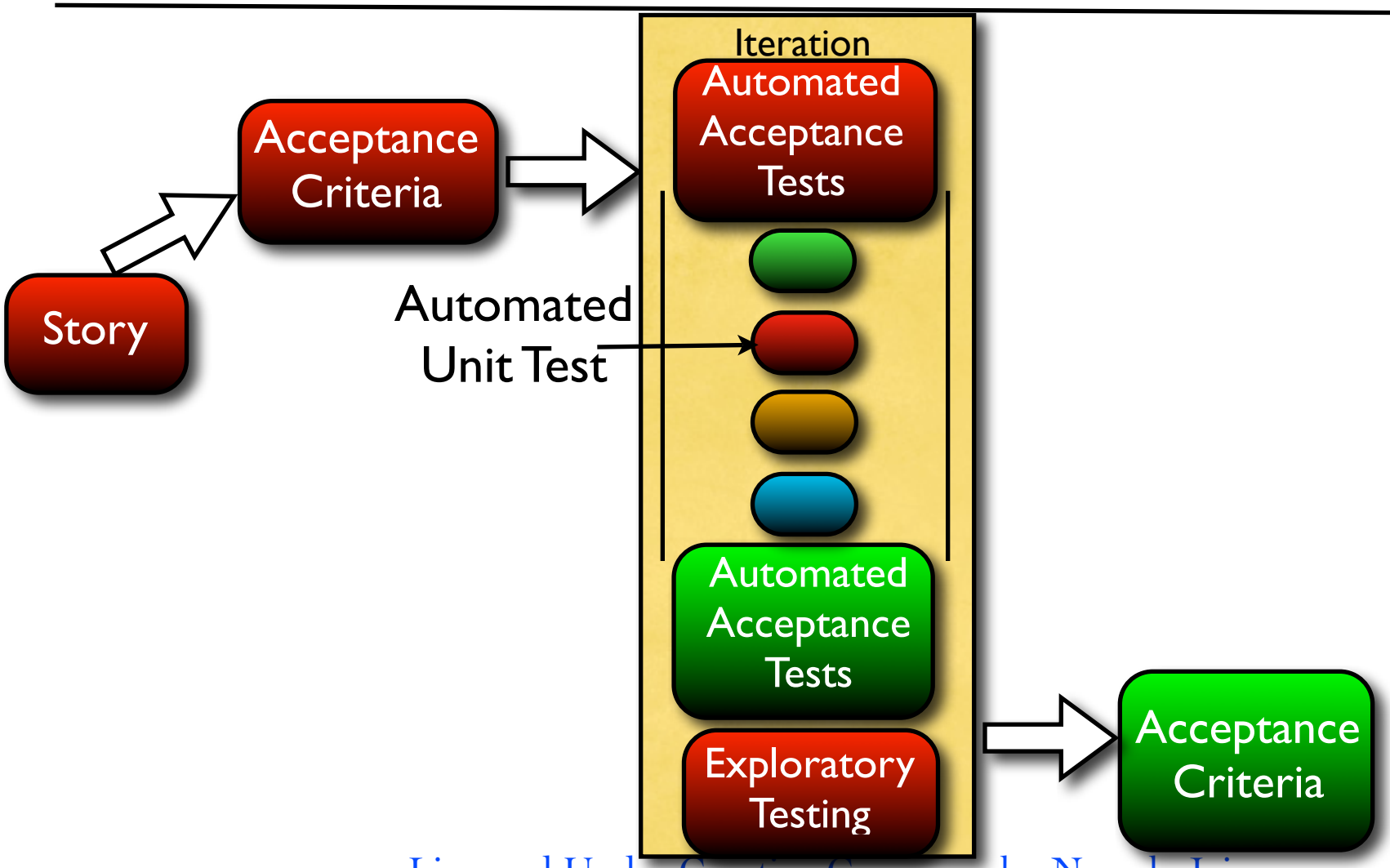


Acceptance Test Driven Development

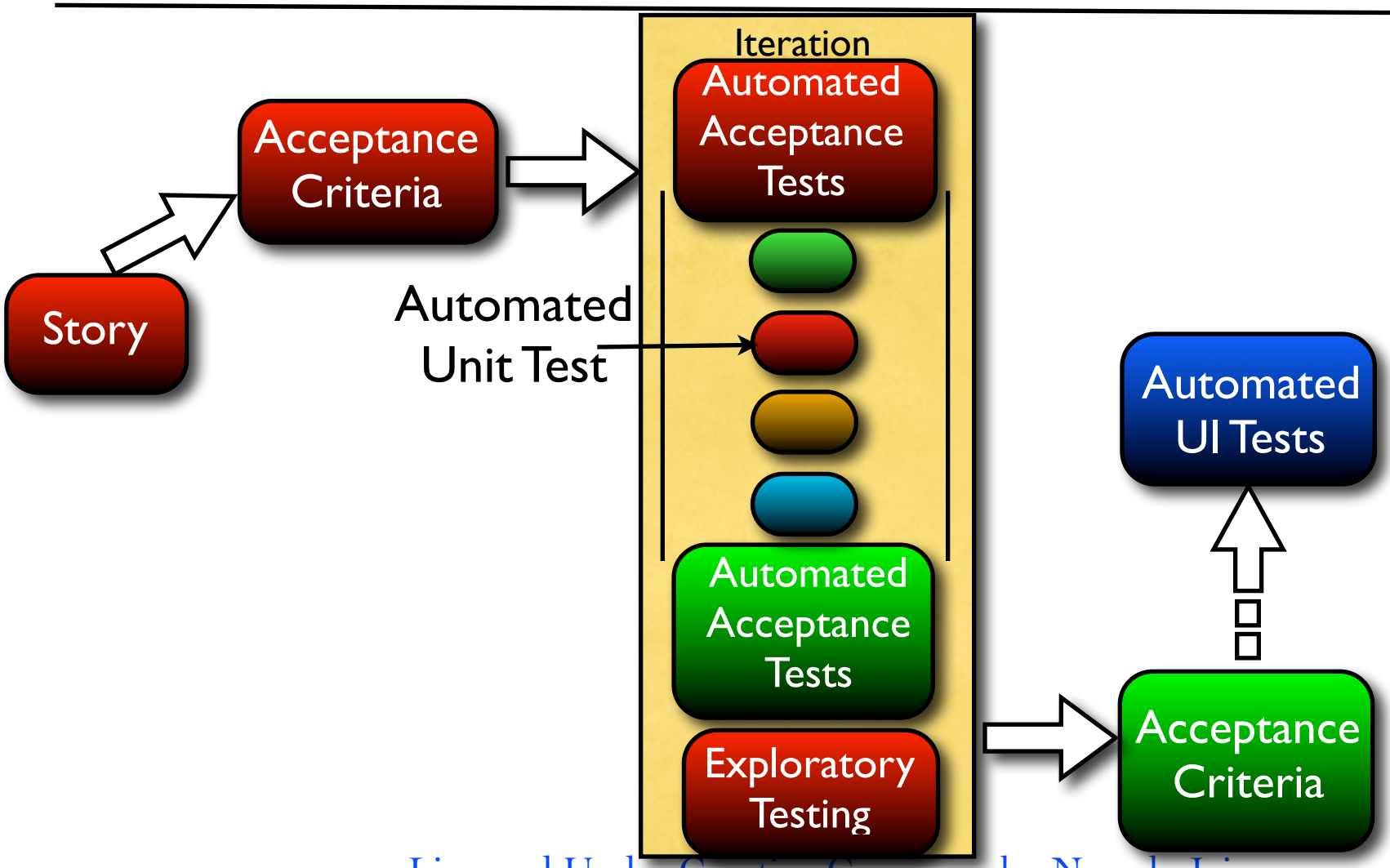


agile faqs

Acceptance Test Driven Development

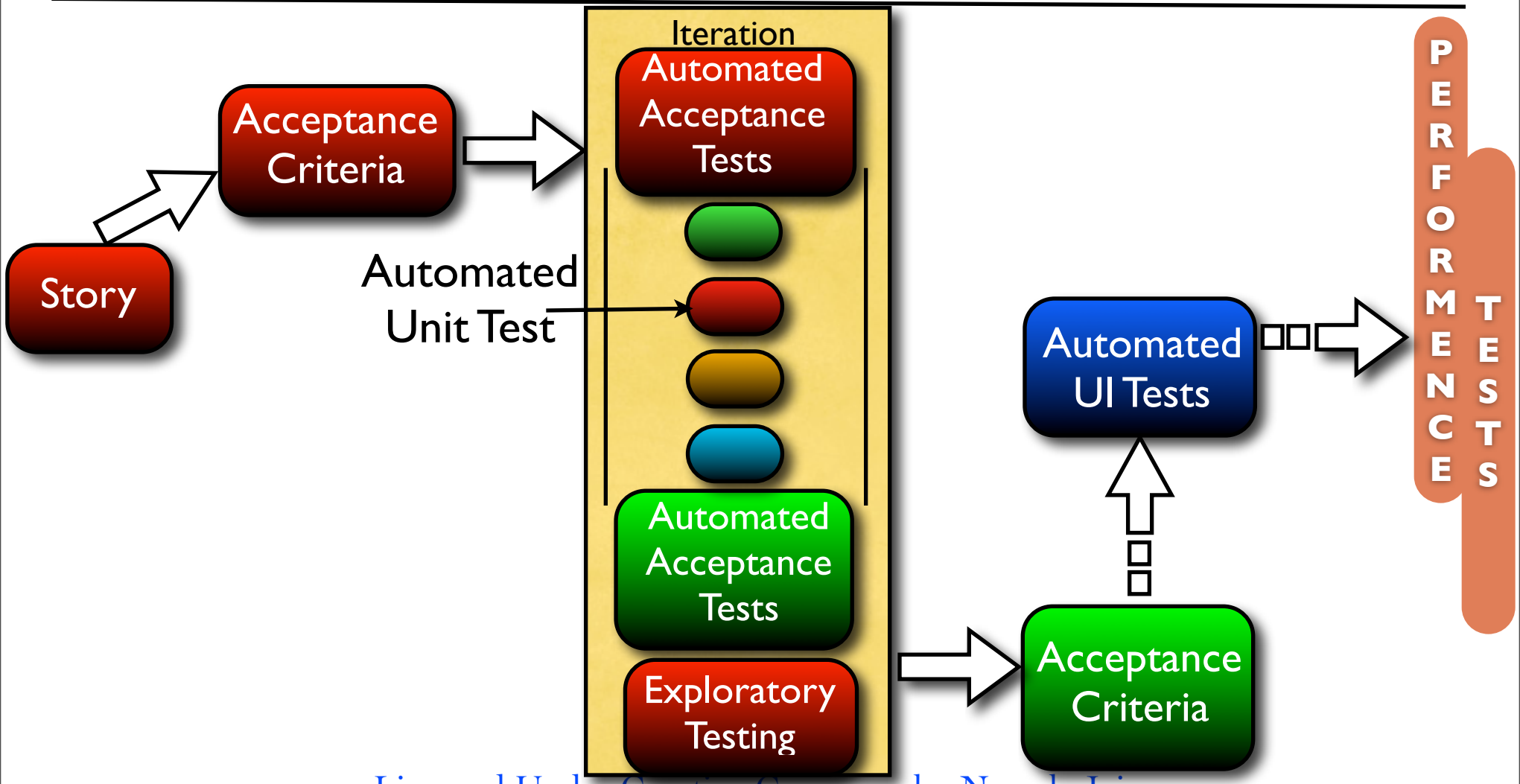


Acceptance Test Driven Development



agile faqs

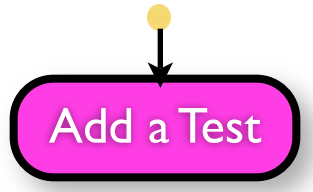
Acceptance Test Driven Development



Test Driven Development

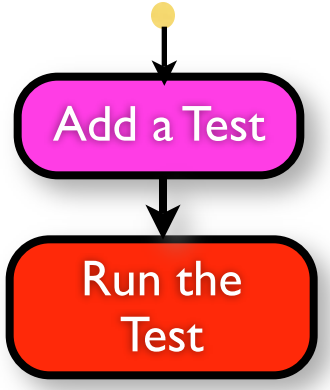
TDD Rhythm - Test, Code, Refactor

Test Driven Development



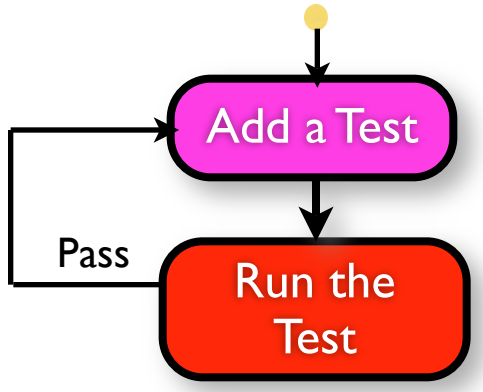
TDD Rhythm - Test, Code, Refactor

Test Driven Development



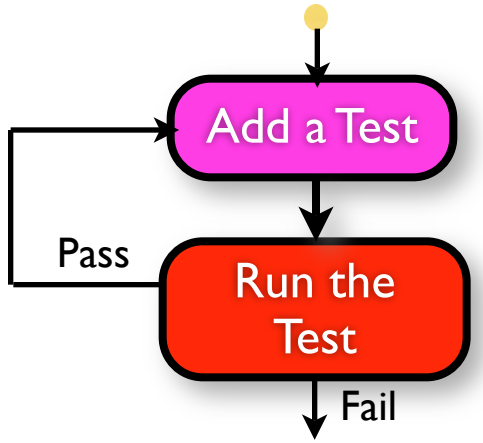
TDD Rhythm - Test, Code, Refactor

Test Driven Development



TDD Rhythm - Test, Code, Refactor

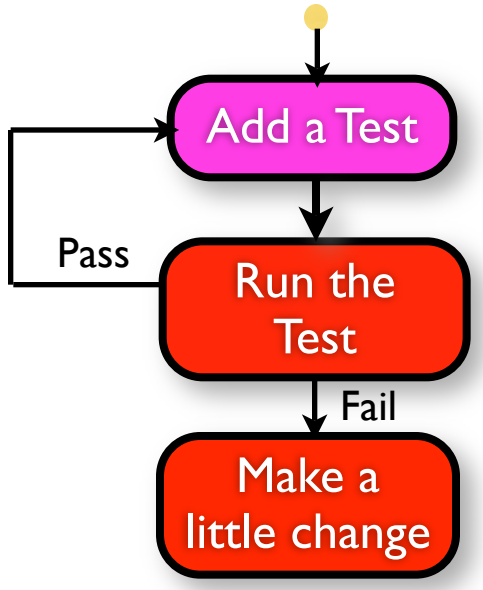
Test Driven Development



TDD Rhythm - Test, Code, Refactor

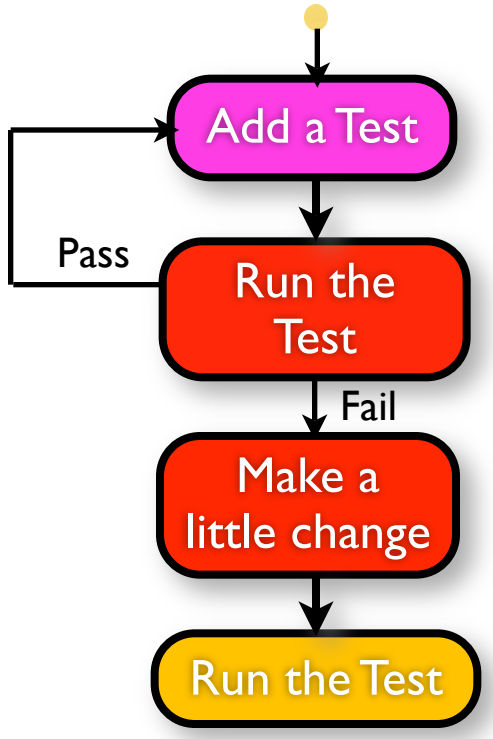
Test Driven Development

TDD Rhythm - Test, Code, Refactor



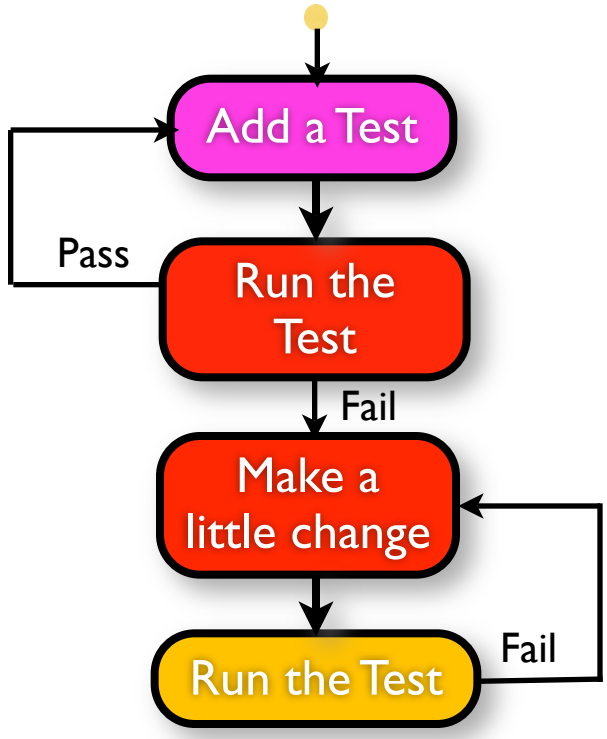
Test Driven Development

TDD Rhythm - Test, Code, Refactor



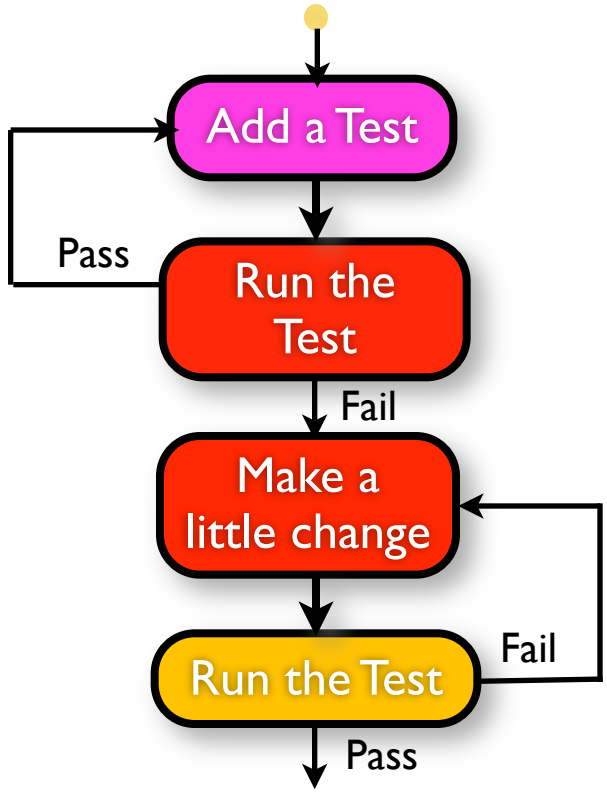
Test Driven Development

TDD Rhythm - Test, Code, Refactor



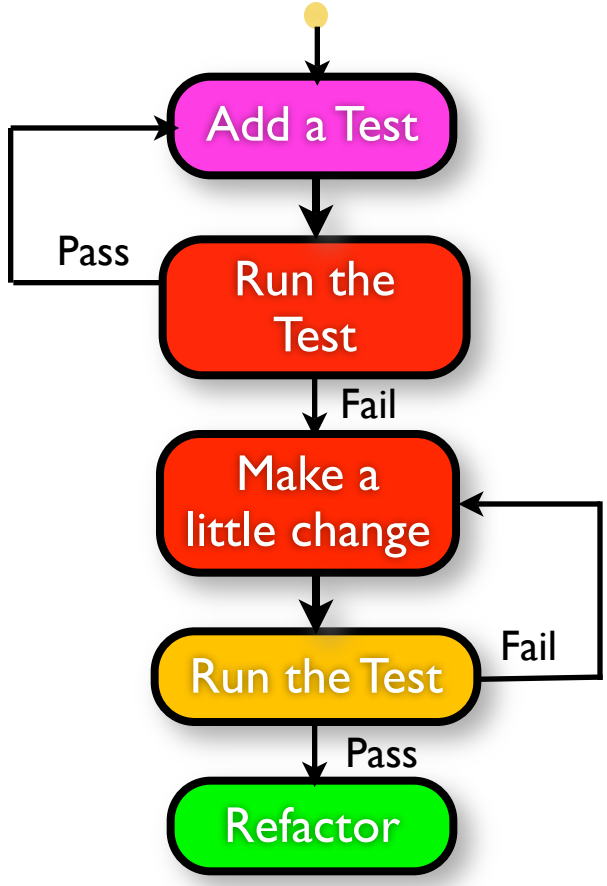
Test Driven Development

TDD Rhythm - Test, Code, Refactor



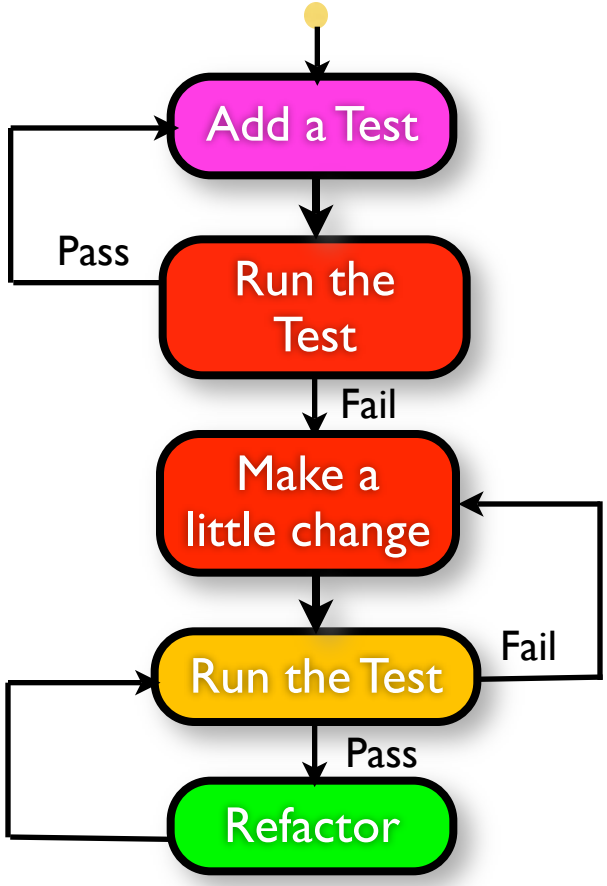
Test Driven Development

TDD Rhythm - Test, Code, Refactor



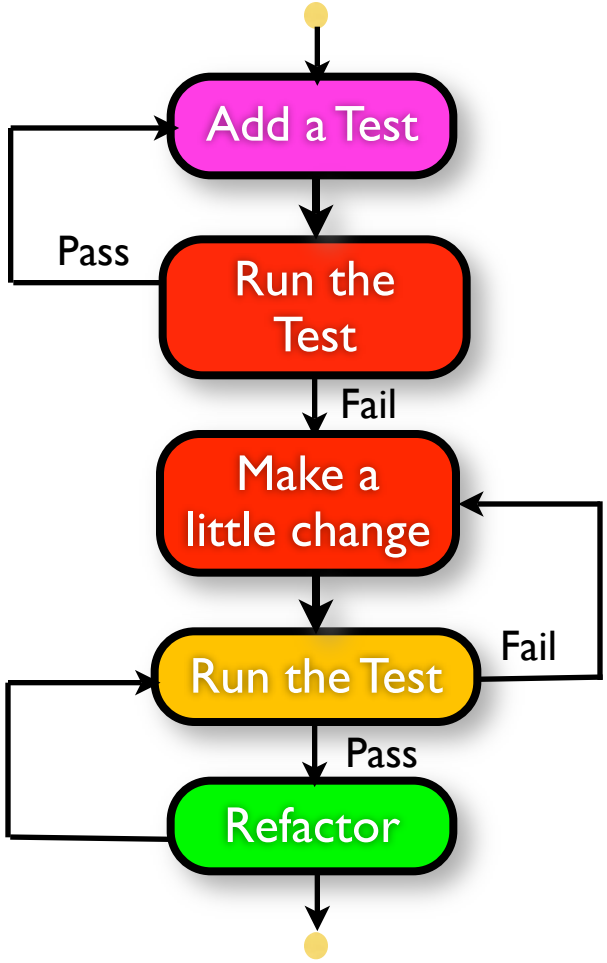
Test Driven Development

TDD Rhythm - Test, Code, Refactor



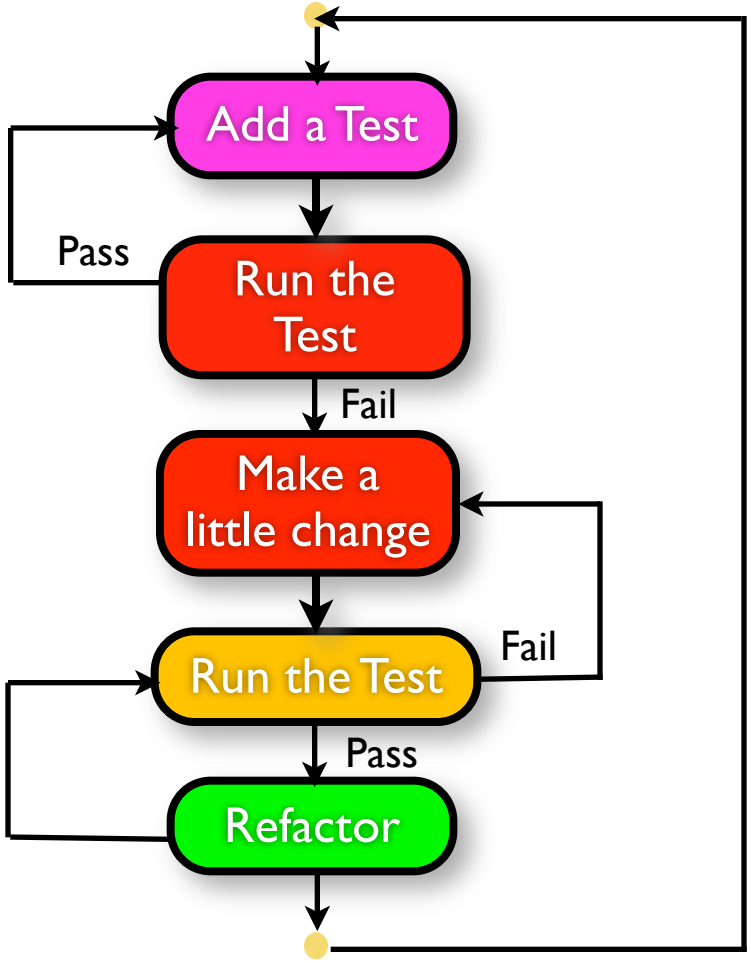
Test Driven Development

TDD Rhythm - Test, Code, Refactor



Test Driven Development

TDD Rhythm - Test, Code, Refactor



Mike Cohn's Testing Pyramid

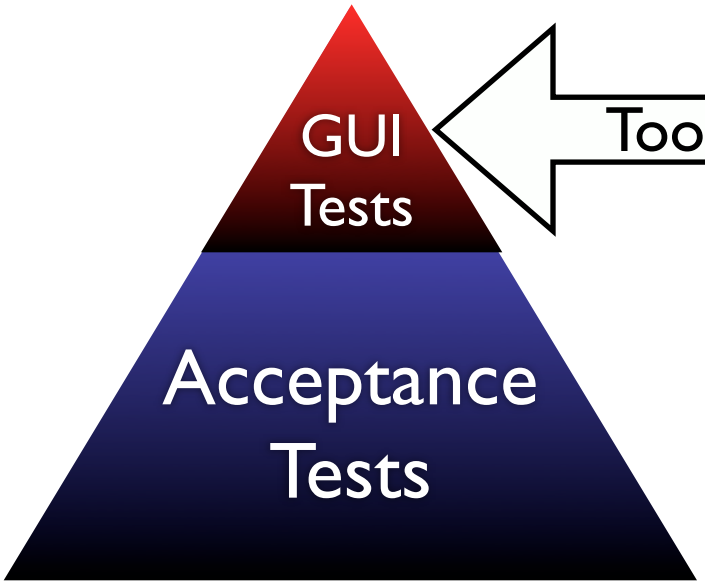
Mike Cohn's Testing Pyramid

Mike Cohn's Testing Pyramid

Small in Number

Tools: Selenium, Sahi, Watir, Abbot, Frankenstein

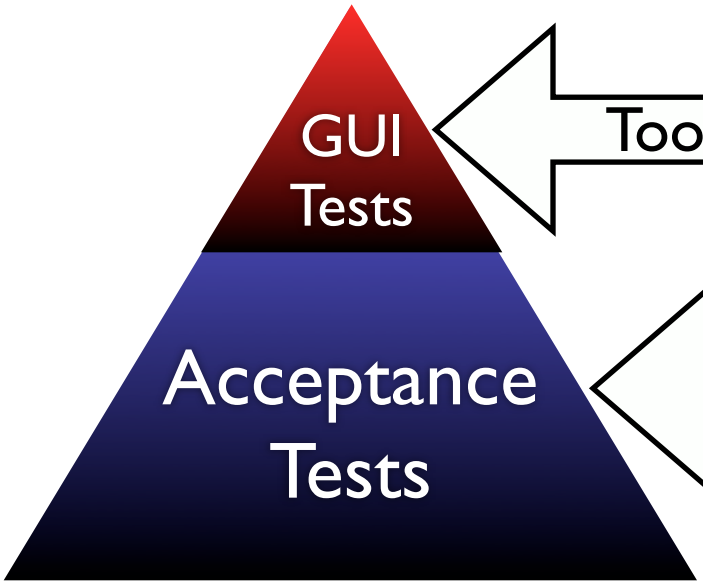
Mike Cohn's Testing Pyramid



Small in Number

Tools: Selenium, Sahi, Watir, Abbot, Frankenstein

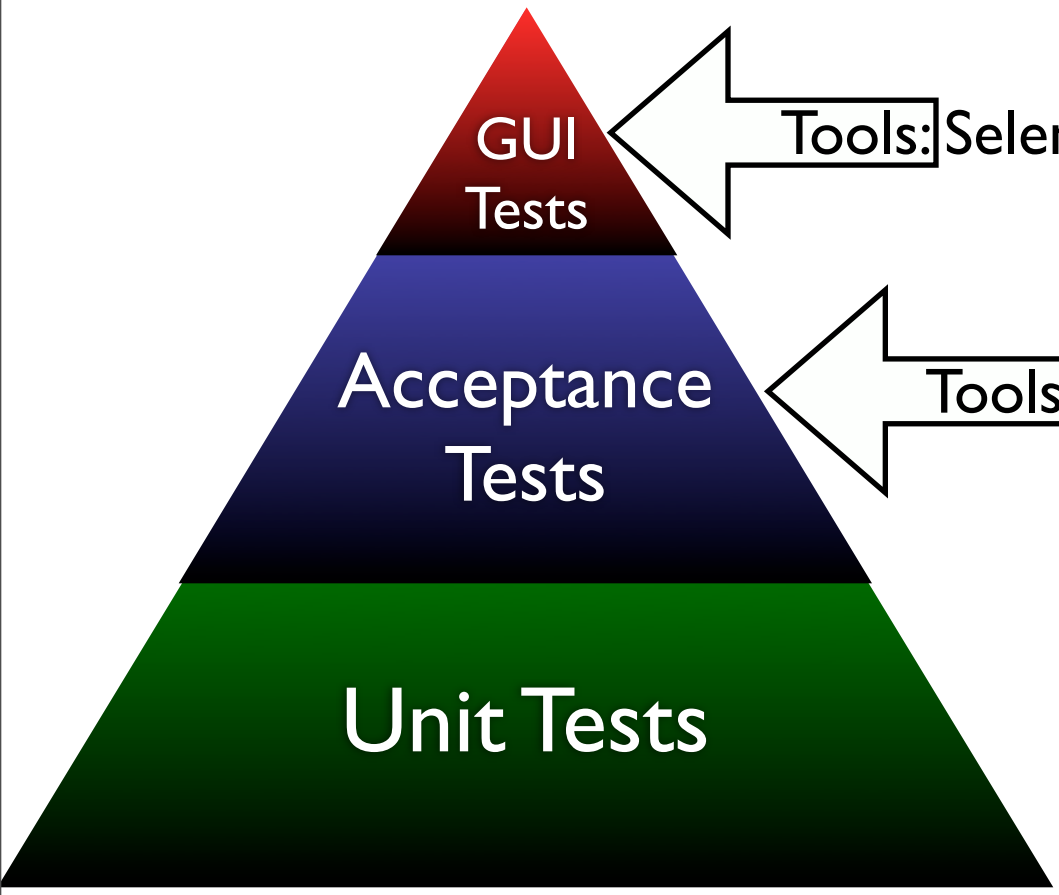
Mike Cohn's Testing Pyramid



Small in Number
Tools: Selenium, Sahi, Watir, Abbot, Frankenstein

At least one per story
Tools: Fit, FitNesse, RSpec, JBehave

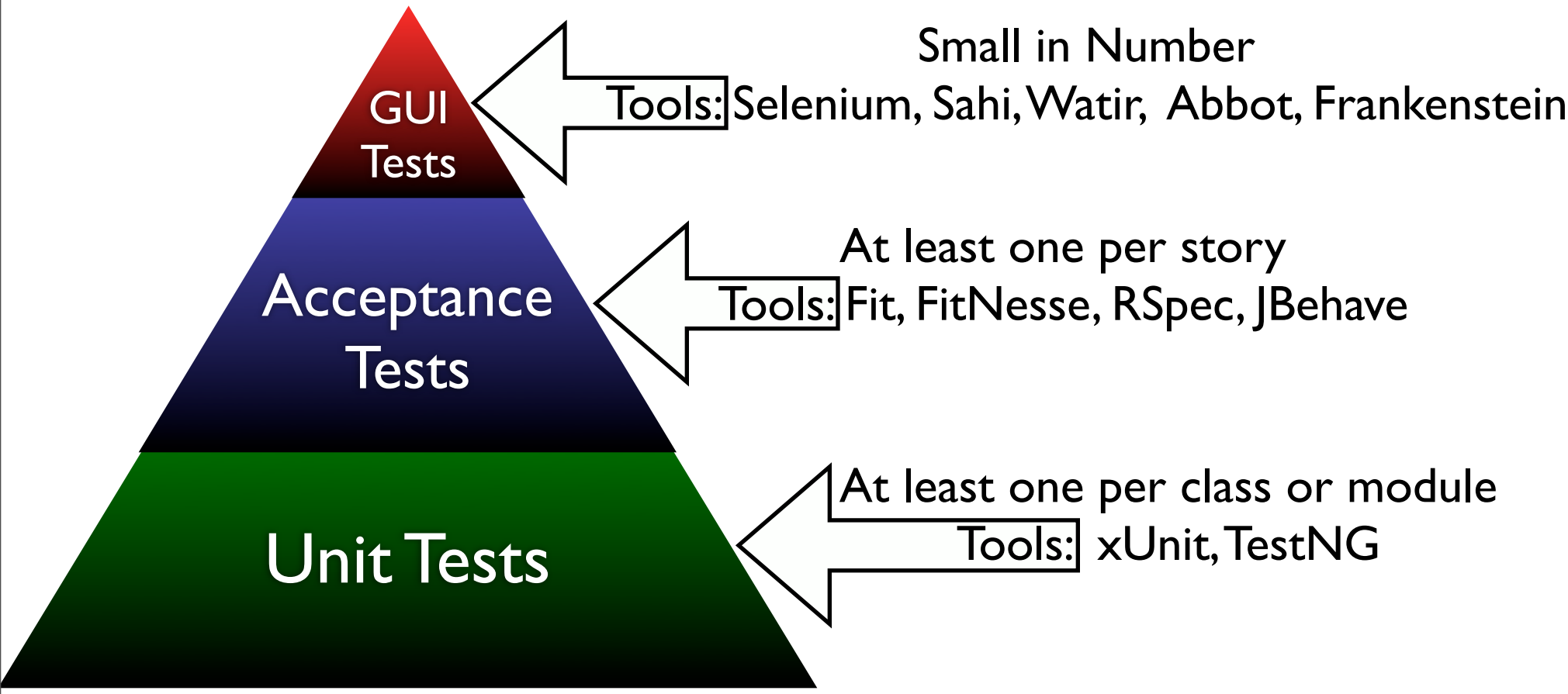
Mike Cohn's Testing Pyramid



Small in Number
Tools: Selenium, Sahi, Watir, Abbot, Frankenstein

At least one per story
Tools: Fit, FitNesse, RSpec, JBehave

Mike Cohn's Testing Pyramid



FitNesse and Selenium Demo

What is the Role of a Tester on Agile Projects?

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team
- ☑ Works closely with customers to define acceptance tests for each story

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team
- ☑ Works closely with customers to define acceptance tests for each story
- ☑ Tests each story as it is complete

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team
- ☑ Works closely with customers to define acceptance tests for each story
- ☑ Tests each story as it is complete
- ☑ Practices pair testing

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team
- ☑ Works closely with customers to define acceptance tests for each story
- ☑ Tests each story as it is complete
- ☑ Practices pair testing
- ☑ Provides continuous feedback to the team

What is the Role of a Tester on Agile Projects?

- ☑ Testers are part of the team
- ☑ Works closely with customers to define acceptance tests for each story
- ☑ Tests each story as it is complete
- ☑ Practices pair testing
- ☑ Provides continuous feedback to the team
- ☑ Works closely with developers to do performance and other types of testing

Challenges

Challenges

- Technical
  - Requirements are changing
  - Bringing Testing Forward
  - Moving from Manual Testing to Automation

Challenges

- Technical
  - Requirements are changing
  - Bringing Testing Forward
  - Moving from Manual Testing to Automation
- Organizational
  - Tester/Developer roles are blurred
  - Dev/Test Teams might be separate
  - Everyone has not bought into agile practices

Challenges

- Technical
  - Requirements are changing
  - Bringing Testing Forward
  - Moving from Manual Testing to Automation
- Organizational
  - Tester/Developer roles are blurred
  - Dev/Test Teams might be separate
  - Everyone has not bought into agile practices
- People

References

- *Agile/QA Testing - Elisabeth Hendrickson*
- *"Agile Testing Directions" - Brian Marick*

- *Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley.*
- *Cockburn, A. (2004). Crystal Clear: A Human- Powered Methodology for Small Teams.*
- *Crispin, L., & House, T. (2002). Testing Extreme Programming. Addison-Wesley.*
- *Poppendieck, M. & Poppendieck, T. (2003). Lean Software Development. Addison-Wesley.*
- *Schwaber, K. & Beedle, M. (2001). Agile Software Development with SCRUM. Prentice Hall.*
- *http://www.extremeprogramming.org*