Agility Now:
Preparing for an Agile Future

Camille Bell

camillescareer@yahoo.com

Why industry and government is switching to Agile software development.

Some Comparison Examples:

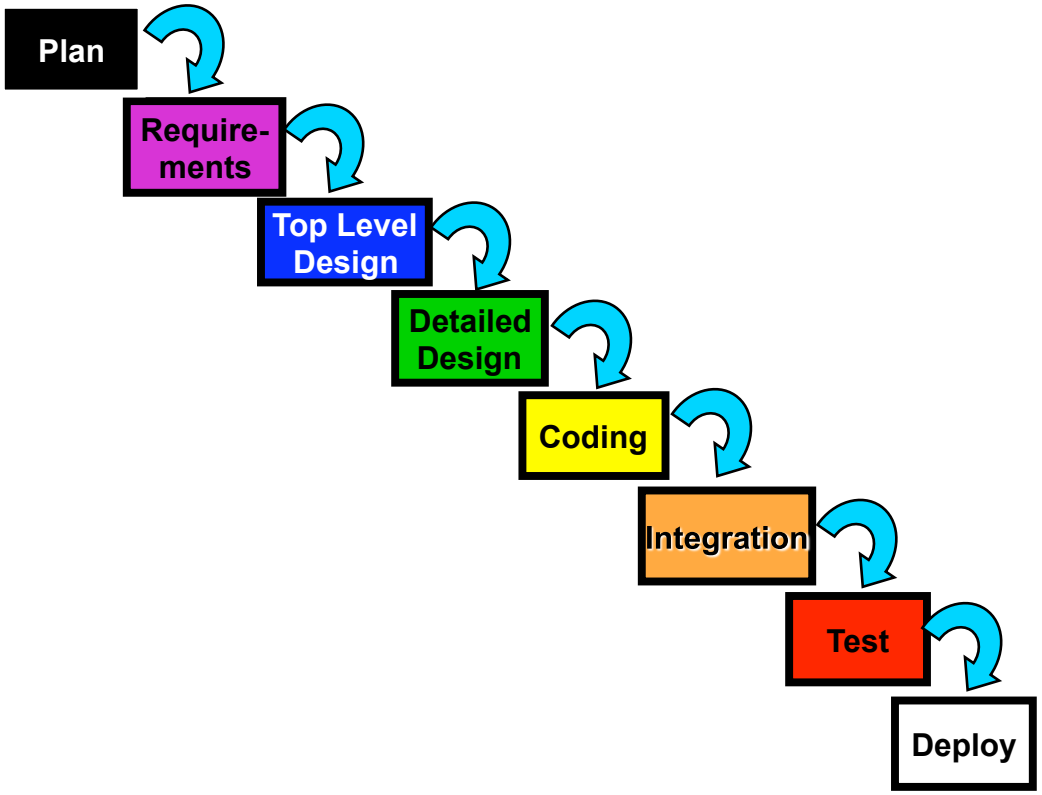
What We Want From Plan Driven Development

- Value
- Timeliness
- Needs Met
- Simple, Solid, Bug Free
- Happy Developers
- Happy Customer

What We Get Instead

- Cost Overruns
- Schedule Overruns
- Wrong & Missed Requirements
- Buggy, Complex SW
- Death Marched Delivery Team Looking for Other Jobs
- Ticked Customer, COTR and Lawyers Looking for New Delivery Team

Traditional Plan Driven Development Doesn't Embrace Change



Because it doesn't adapt to change; it fails.

Large Projects Fail

- Large projects diminish productivity
  - Beyond 25 person months for a given software project, productivity was reduced equally per person (e.g., 250 months reduced productivity 10x)
  - Optimal project size appears to be about 8.3 people for 3 months

Ref: Standish Group Reports

Large Projects Fail

- Large projects diminish chance of success
  - Projects with less than $750K had a 55% success rate
  - $5 – 10 Million projects had an 8% success rate
  - Over $10 million had less than ½% success rate

Ref: Standish Group Reports

Statewide Automated Child Welfare Information System

	State of Florida	State of Minnesota
Initial	**8 years**	**NA**
Initial	**$32 million**	**NA**
Final	**15 years**	**1 year**
Final	**$230 million**	**$1.1 million**

200:1 difference - same Federal mandate

Source: Lean Software Development: An Agile Toolkit

Italian Firm Before CIO Mandate and Agile Practices

1. They would spend two to three months in meetings, aligning all opinions in order to create a huge requirements document;

2. When they finally started developing the systems the business team had disappeared and in most cases forgotten about the project;

3. By the time the project was ready to be tested the key users had changed, the business had changed and the project delivery team immediately entered into a lengthy negotiation phase to reconcile what was delivered versus what the business really needed.

Source: Mike Jones, Driving Agile Success - A CIO's Mandate

Source: Mike Jones, Driving Agile Success - A CIO's Mandate

Italian Firm After CIO Mandate and Agile Practices

1. The business team is more motivated and involved as they are able to see how the projects are progressing on a regular basis;
2. The business gets to be more responsible for the decisions that shape the project direction because they see and constantly test the application;
3. The business and IT avoid the costly, wasteful exercise of building complex requirements documents because they now fully realize that they can never document every detail in a specification;
4. From very early on in the project, IT can see if the project is really what the company needs and identify any mismatches quickly to reduce the amount of time, dollars and resources that might be wasted
5. Even for big projects, Agile methodology is used - and forces the team to split the project into phases. This exercise divides the scope into smaller, manageable projects with incremental releases and decreased risk.

Source: Mike Jones, Driving Agile Success - A CIO's Mandate

Some Personal Experience with CMM/CMMI On Joint Tactical Radio System

- Worst performing JTRS Cluster 1 Team (responsible for stop work order) was CMM Level 5 following strict Waterfall practices

- Best performing JTRS Cluster 1 Team (able to debug HW before it was built) was CMM Level 5 following rigorous Agile practices

CMM/CMMI process itself and certification adds paperwork and cost not value.

General Characteristics of Agile Processes

The Agile Manifesto

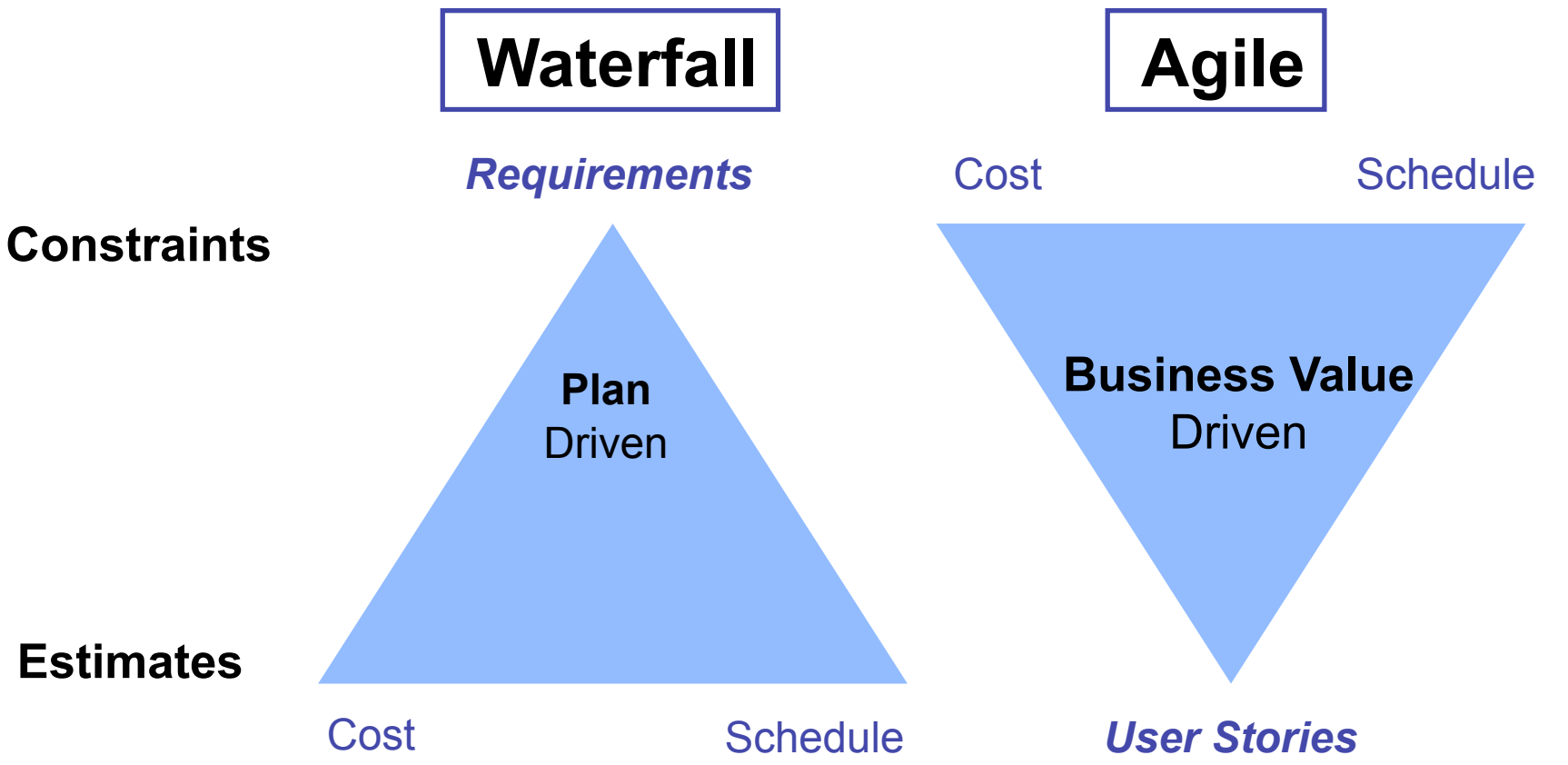
"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **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."

Ref: www.agilemanifesto.org/

Paradigm Shift

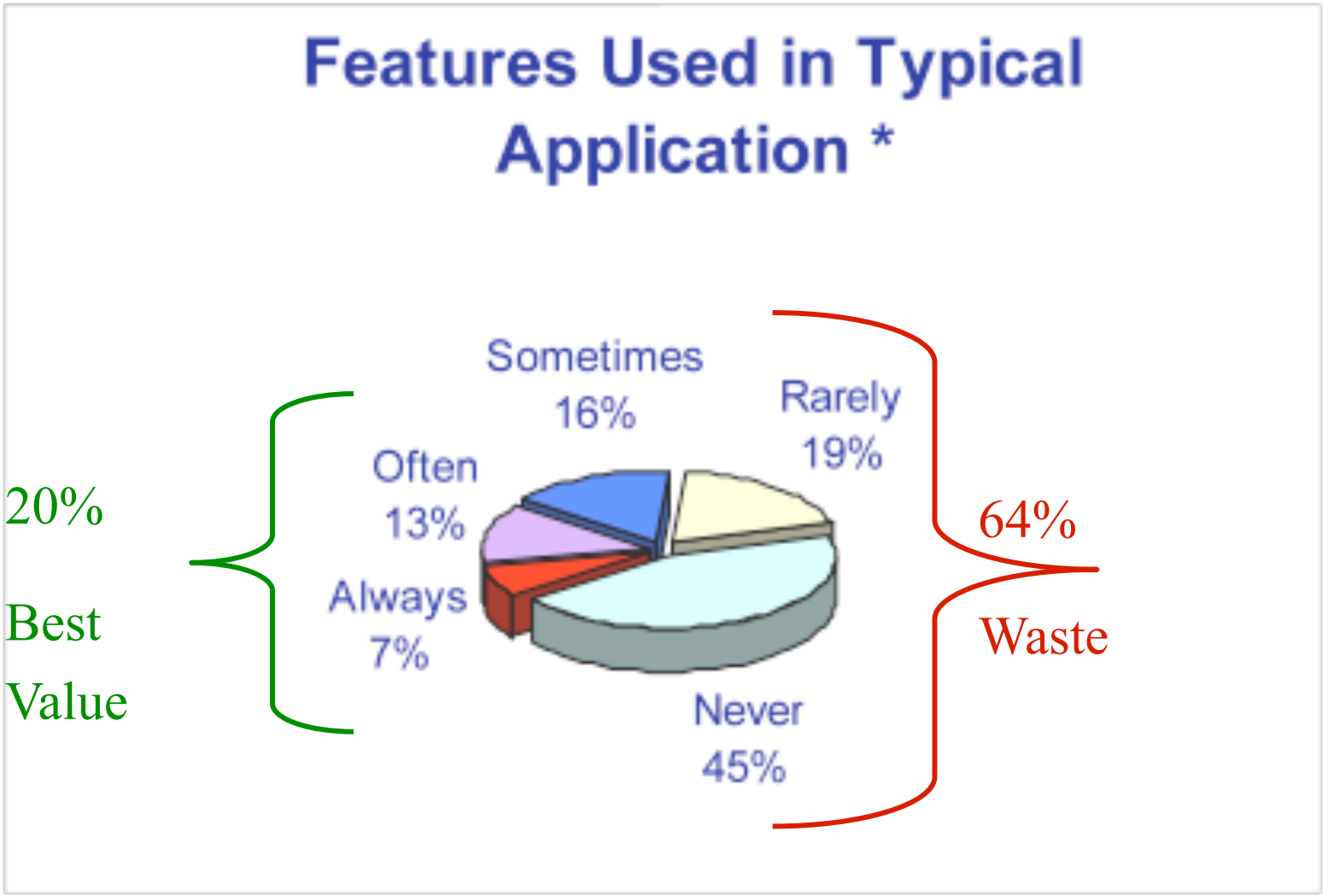


The Plan creates cost/schedule estimates

The Vision creates User Story estimates

Ref: Adapted from Dynamic Systems Development Method

Only Implement the Best Value

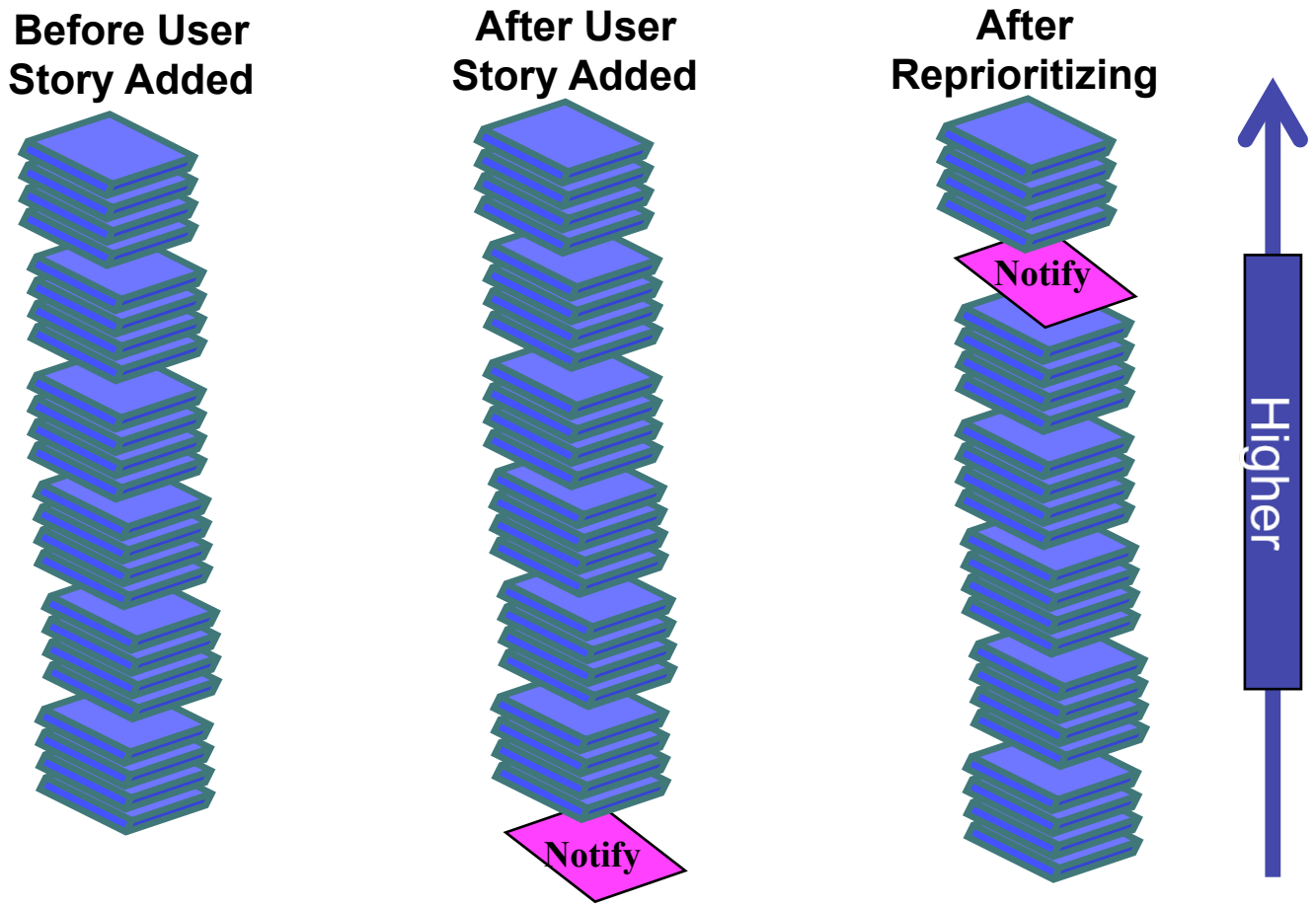


Ref: Standish Group Reports

Common Agile Process Features

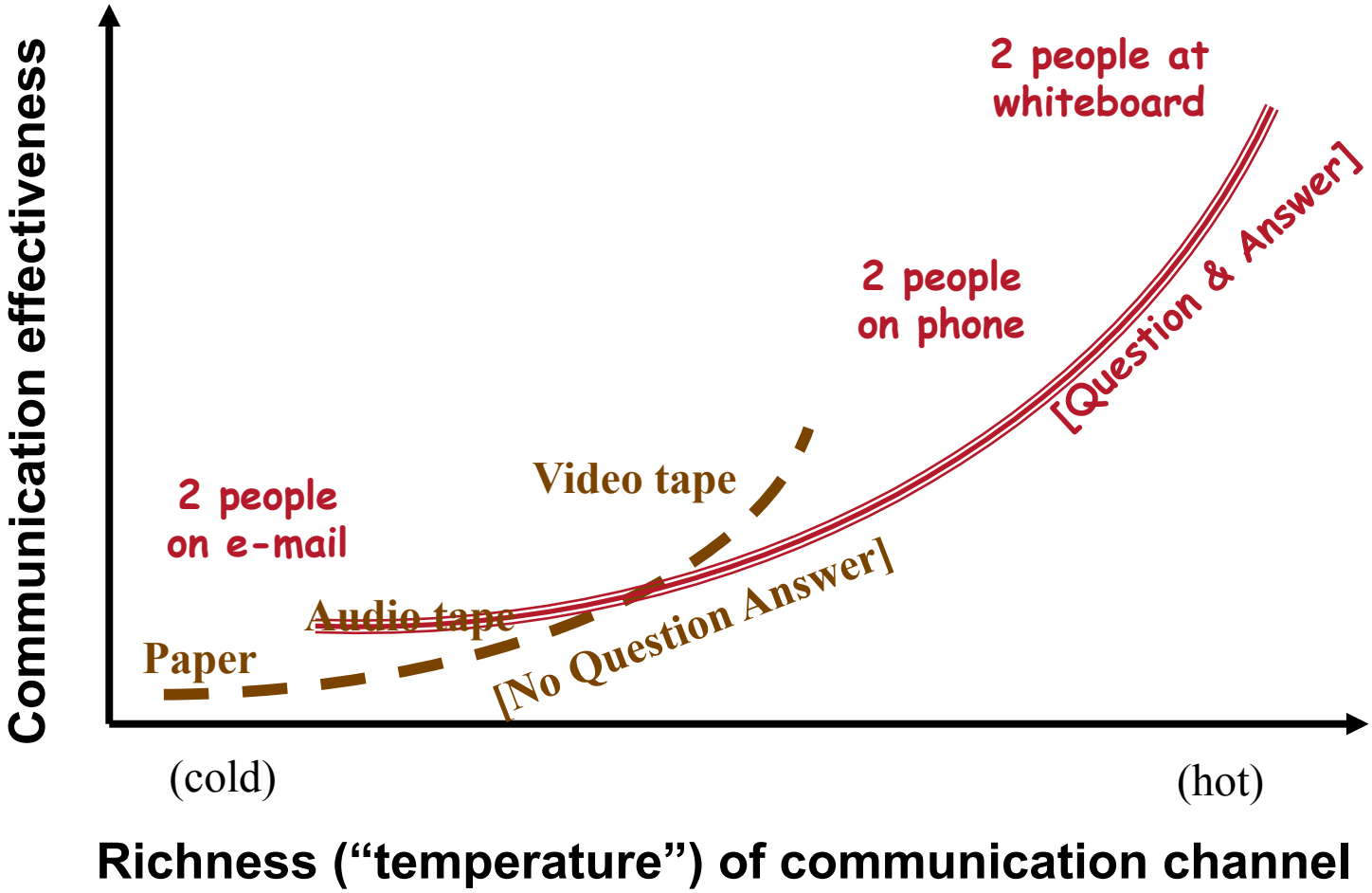
- User Stories
  - byte sized end-to-end customer functionality
- Very Short Iterations
  - 1 or 2 weeks
- No Changes in Priorities During Iteration
- Daily Meetings of a Few Minutes
- Demo Working SW Every Iteration
- Review & Retrospectives Every Iteration
- Development Team Organizes Own Work
- Management Facilitates & Removes Obstacles

User Stories Are Implemented By Business Priority



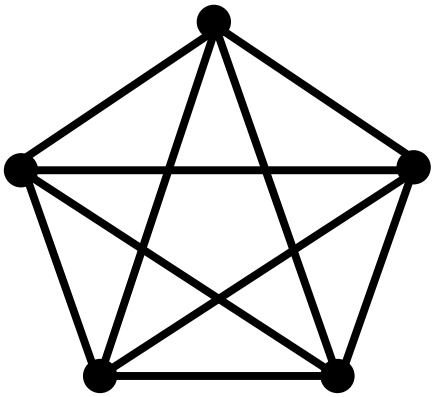
Prioritized Backlog of User Stories for later iterations

Improving Individual Interactions & Customer Collaboration

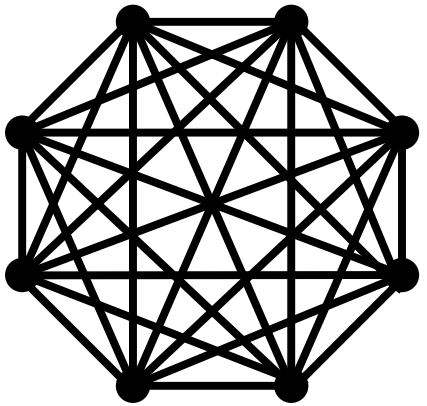


Source: Alistair Cockburn

Smaller Teams Reduce Communications Channels



5 Person Team



8 Person Team

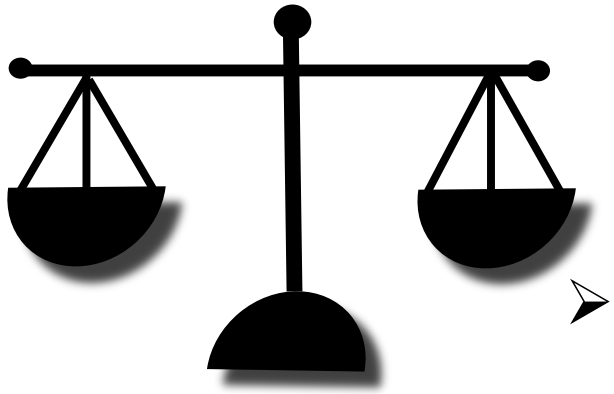
Imagine a 20 person team !

Agile Balance of Power

Customers

Business Decisions:

- Dates
- Scope
- Priority



Developers

Technical Decisions:

- Estimates

Together both ensure the highest priority scope is ready by the release date.

Automate Repeatable Tasks

- Compilation
- Simple Refactoring
- Testing
- Configuration & Archival
- Build
- Continuous Integration
- Deployment

Some Key Features of Leading Agile Processes

- Lean
- Scrum
- eXtreme Programming

Lean Principles

1. Eliminate Waste
2. Build Integrity In
3. Create Knowledge
4. Defer Commitment
5. Deliver As Fast As Possible
6. Respect People
7. Optimize the Whole

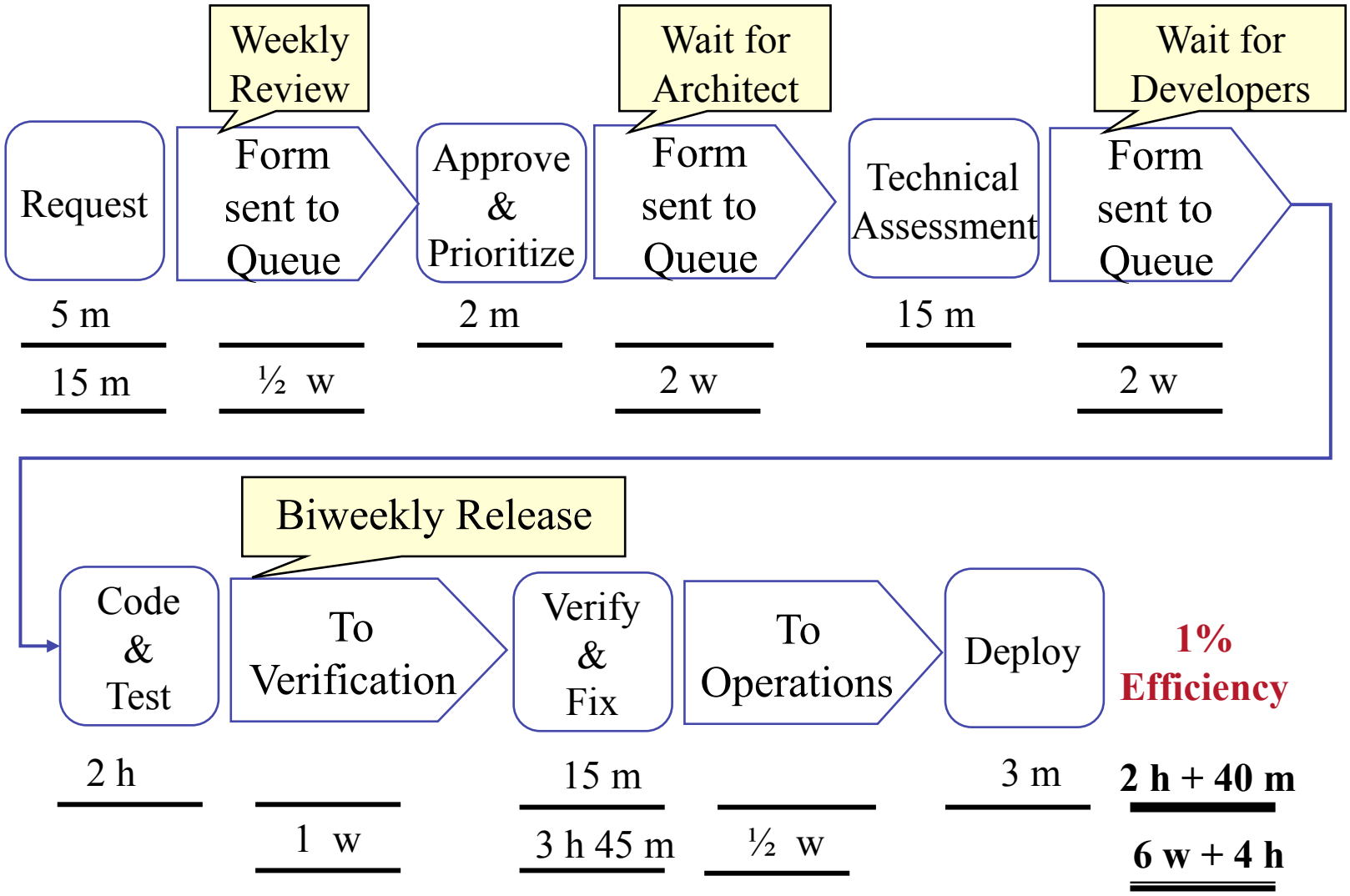
Source: Lean Software Development: An Agile Toolkit

Lean's 7 Wastes in SW Development

- Partially Done Work
- Extra Features
- Relearning
- Task Switching
- Handoffs
- Delays
- Defects

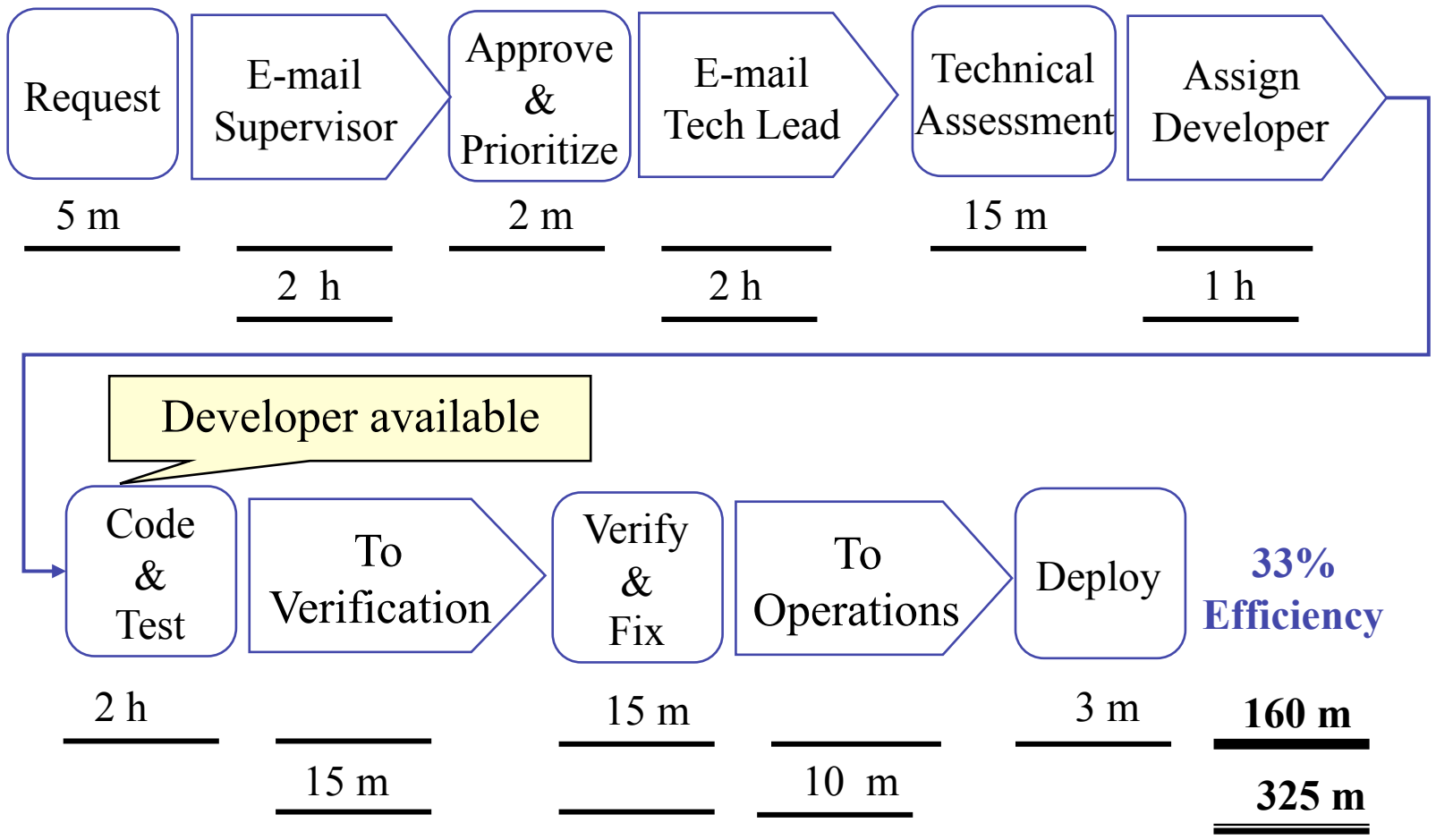
Source: Lean Software Development: An Agile Toolkit

Lean Value Stream Mapping Example:
Most Inefficient



Source: Lean Software Development: An Agile Toolkit

Lean Value Stream Mapping Example:
Better Efficiency



Source: Lean Software Development: An Agile Toolkit

Scrum Roles

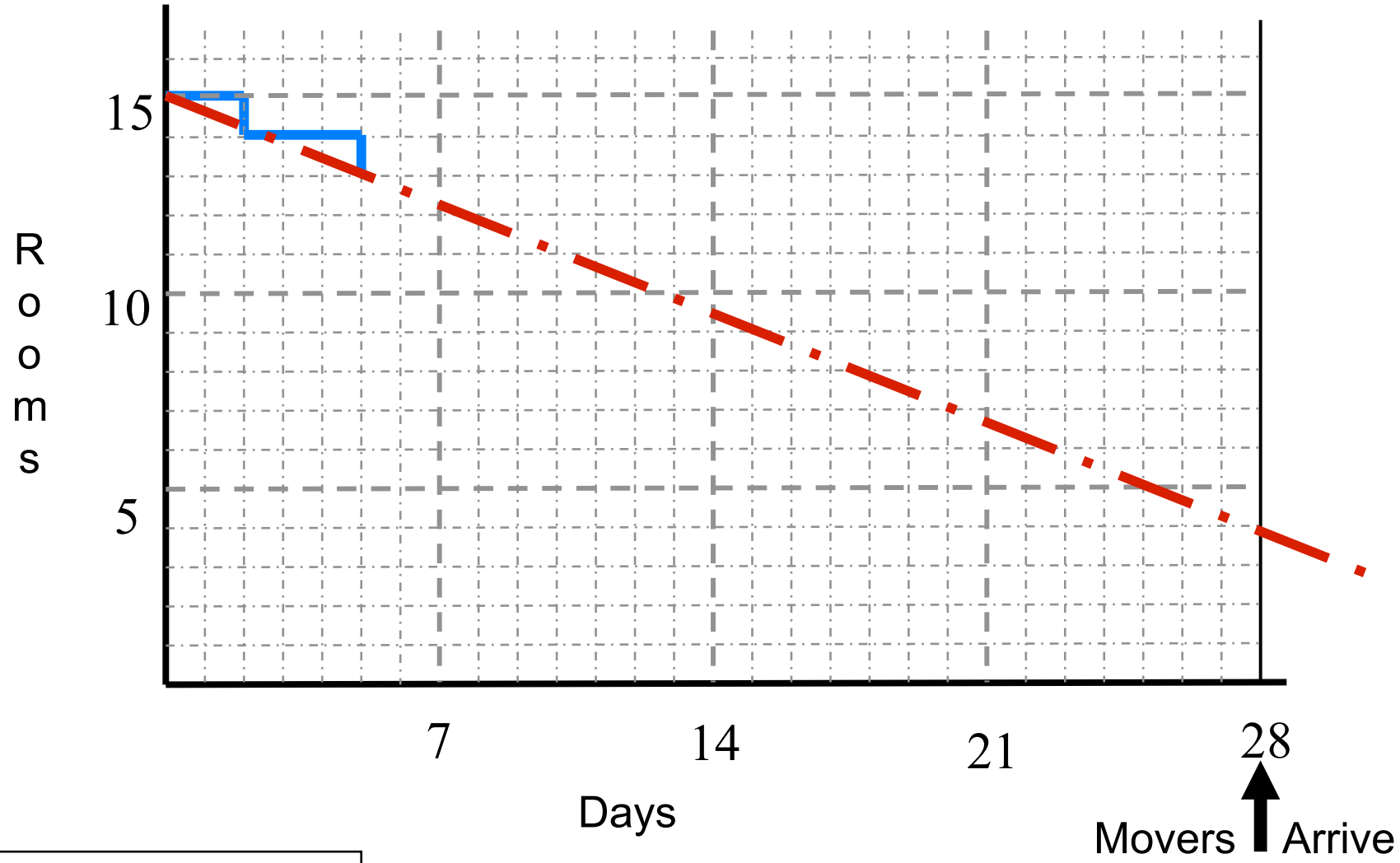
- Product Owner/Customer
- Coach Facilitator – Scrum Master
- Delivery Team
  - Developers
  - Testers
  - Tech Writers
  - Usability Engineers
  - Systems Engineers
  - Architects

Scrum Delivery Team

- Typically 5 to 10 people
- Cross functional
  - Testers, Programmers, Business Analyst
- Members should be full-time
  - May be exceptions (tech writers, DBA)
- Teams are self-organizing
  - It assumes responsibility for planning its own work

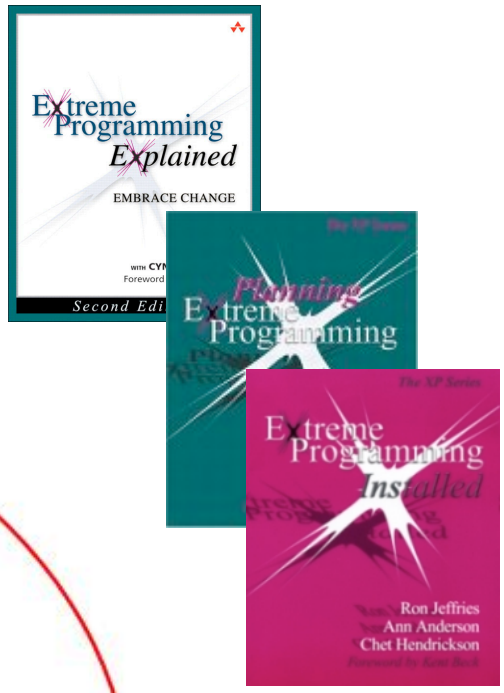
Example Burn down Chart

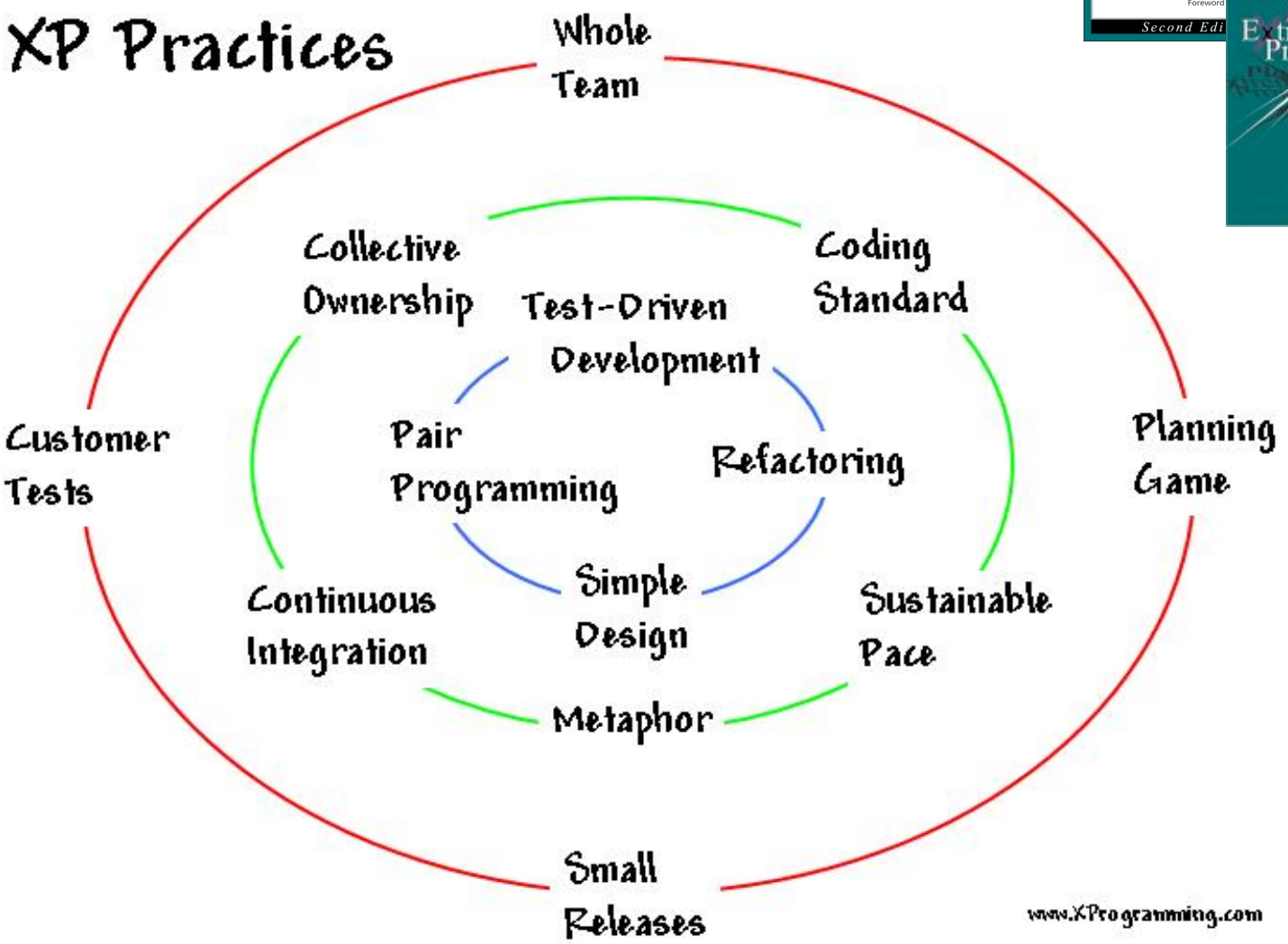
Day 5 – Projected Finish Date



Example: Alistair Cockburn

XP Practices





3 Rules of Test Driven Development

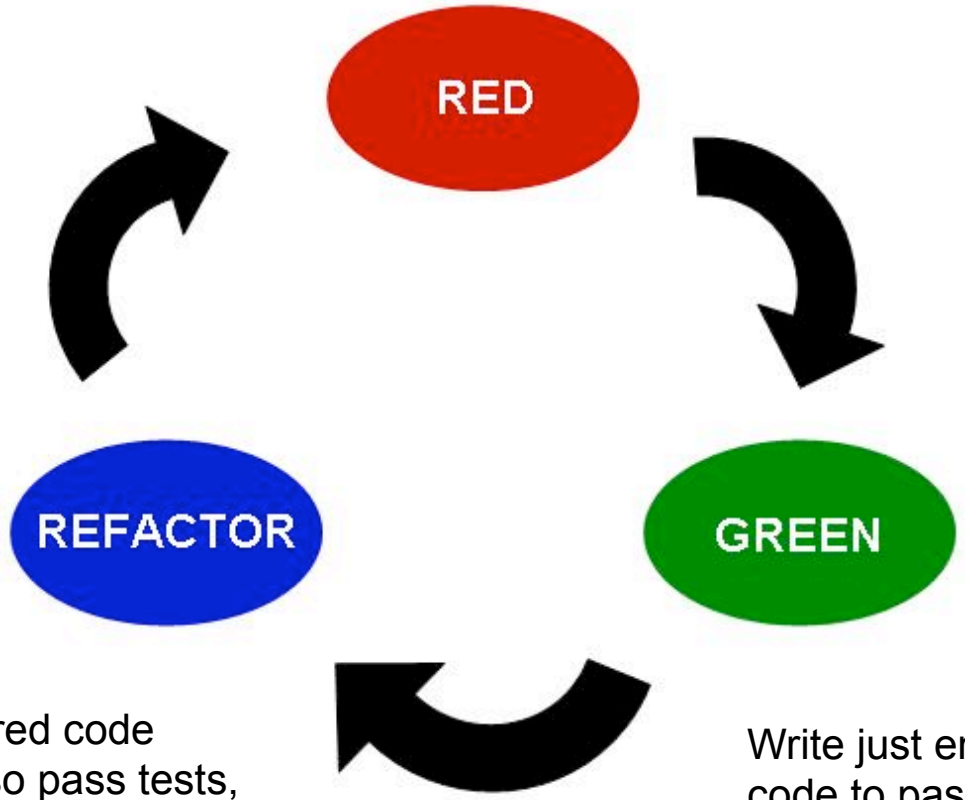
1. You are not allowed to write any production code unless it is to make a failing unit test pass.
2. You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
3. You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

With tests you are in control; without tests you aren't!

Bob Martin - The Three Rules of TDD

The Test Driven Development Cycle

Write a failing test
for new functionality



Refactored code
must also pass tests,
no new functionality

Write just enough
code to pass test

Key Target Audience to Whom Each Process Most Appeals

Lean	• CXO's, large companies, other's familiar with Lean Manufacturing
Scrum	• 1st, 2nd and sometimes 3rd tier management, medium & small companies, those keen on certifications
eXtreme Programming	• Technical staff, small companies, those with constant access to customers

Why Process is Attractive to Audience

Lean	• Toyota's track record with Lean, focus on value stream & waste elimination, solves large enterprise problems, includes SW engineering, Kanban answers when done
Scrum	• Quickest to implement, usually ignores engineering practices, easy certification, burn down answers when done
eXtreme Programming	• Rigorous focus on engineering practices, closest to customers, velocity answers how much can be done

Key Challenges of Each Process

Lean	• Requires CEO level buy-in corporate wide, top down implementation can cause resentment in lower level staff
Scrum	• Clueless teams don't self organize well, without engineering practices productivity and quality improvements stall out, enterprise level impediments difficult to remove in large companies
eXtreme Programming	• Process name, more change than Scrum, engineering focus confuses management, customer participation levels daunting, technical rigor requires lots of practice

General Challenges in Adopting Agility

- Contracts and Profit Models
  - Contracts need to be flexible and short
    - Buy/sell a set number of iterations (sprints)
    - Time box and money box contracts
    - No requirements laundry lists,
      - Hold user story workshop for initial requirements
      - Allow customer to reprioritize each iteration
      - Only focus on highest priority user stories
    - Cancel the contract *early*, if not performing
  - Contracts process needs to be faster
    - Slow and costly government contract process keeps the best companies from bidding
    - Short, low dollar value process grows bidder pool

General Challenges in Adopting Agility

- Contracts and Profit Models (continued)
  - Contract managers need to abandon command and control
    - Instead facilitate to ensure removal of roadblocks to direct customer-developer collaboration
  - Companies need incentives to provide fewer high quality developers instead of many low quality developers
    - Profit model based on bottoms in seats has to change
  - Good agile companies rewarded by more more contracts not contract extension because deadline unmet

General Challenges in Adopting Agility

- Customer Interaction
  - Real end user customers (or as close as possible) are essential
  - Customers need to interact with development teams /wo intermediaries
  - Customers must prioritize their requirements as small end-to-end user stories and reprioritize frequently
  - Customers should expect, attend and provide feedback of demos of working software

General Challenges in Adopting Agility

- Management Resistance
  - Management needs to abandon command and control
    - Difficult to impossible for control freaks
    - Window seat inflexible command and control managers into non-agile, non-critical projects
  - Management facilitates
    - Choose coach style managers
    - Place in Scrum Master or similar roles
  - Management removes road blocks
    - Takes huge courage to remove roadblocks when higher ups are the cause

General Challenges in Adopting Agility

- Technical Resistance
  - Expect resistance if mandated from above
    - Find those who want to be agile first
    - Helps if first Agile projects are cool techie attractors
    - Use social pressure - seed agile teams with > 50% pro-agile
  - Senior staff lacking basic technical professionalism
    - e.g. developers who don't write automated tests are like surgeons who don't wash their hands
    - e.g. developers who don't actively seek learn new technology, skills and techniques
    - e.g. if they aren't humble enough to recognize deficiencies and improve, remove from team

General Challenges in Adopting Agility

- Technical Resistance (continued)
  - Agile technical practices require *discipline*, *lots of practice* and some *social skills*
    - Waterfall developers may find the extreme rigor of agile technical practices like BDD & TDD daunting
    - Practices can slide under pressure, distraction or confusion – stay vigilant
    - Pair programming helps reinforce practices
  - Learning new skills and technologies takes time
    - Training, coaching and patience helps
    - Pair programming helps spread skills faster
  - Command & control tech leads out
  - Collaborative and coaching leaders emerge

General Challenges in Adopting Agility

- Technical Staffing Models
  - Pick the right team
    - Even the best process is worthless with bad people
  - Pyramid style staffing (few senior many drones) not a good fit for Agile
    - Hire fewer, but better techies
    - Use pair programming to raise skill level of entire team
  - Top technical ranks not as well paid
    - Long term adopt dual ladder
    - Short term reward techies with training, perks recognition

Business Consequences of Failing to Adopt Agility

"It is not necessary to change. Survival is not mandatory."

- W. Edwards Deming

What you can do

- Learn more, be change agent, have courage
- Usually best to start small to ensure success
  - Choose one team of the good open developers
  - Choose committed customer
  - Choose highly productive agile technology
    - e.g. Ruby on Rails
  - Get team training
  - Use agile process and technical coaches to ramp up team
  - Use retrospectives to improve
  - Repeat and expand

General Web References

Classics

- http://www.projectsmart.co.uk/docs/chaos-report.pdf
  - A copy of the classic 1995 report on SW project failure by the Standish Group

Agile

- http://www.agilealliance.org
  - The home of the Agile Alliance, with a great library of Agile articles.
- http://www.mountaingoatsoftware.com/system/presentation/file/52/SDWest2007_EUS.pdf
  - Effective User Stories by Mike Cohen
- http://www.bluecollarobjects.com/pub/Main/Agile2009/Federal_Bureaucracy-4.pdf
  - A Retrospective: Managing Agile Transition in Government Bureaucracy by Brandon Raines and Judy Wankerl

Lean Web References

- http://www.poppendieck.com/
  - Mary and Tom Poppendieck's home website, excellent material on Lean
- http://xpday3.xpday.org/slides/LeanTutorial.pdf
  - Overview and Tutorial on Lean by Mary Poppendieck
- http://leansoftwareengineering.com/
  - Good articles on Lean from multiple authors
- http://leansoftwareengineering.com/ksse/scrum-ban/
  - Corey Ladas, Scrumban Intro
- http://leansoftwareengineering.com/2009/06/08/workflow-patterns/
  - Corey Ladas, author of Scrumban - on kanban workflow

Scrum Web References

- www.controlchaos.com/
  - Ken Schwaber's Web site on Scrum.
- jeffsutherland.com/
  - The codeveloper of Scrum's Web site. Jeff Sutherland provides various content related to software programming and technology, particularly objects, components, and Scrum. Very up-to-date and educational.
- www.mountaingoatsoftware.com/scrum/
  - Mike Cohn's great Web site on User Stories, Estimation, Scrum
- www.scrumalliance.org/
  - The home of the Certified Scrum Masters (*caution founders have left*)

eXtreme Programming Web References

- http://www.eXtremeProgramming.org
  - Extreme Programming (XP) home site. The place to start investigating technical agility.
- http://www.xprogramming.com
  - Ron Jeffries's Web site about Extreme Programming (XP), also has good articles on planning and metrics
- http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd/
  - Bob Martin on TDD
- http://www.extremeprogramming.org/rules/testfirst.html
  - XP and TDD
- http://www.agiledata.org/essays/tdd.html
  - Scott Ambler's introduction to TDD
- http://www.testdriven.com/
  - Test Driven Development Community
- http://groups.yahoo.com/group/testdrivendevelopment/
  - Test Driven Yahoo Group
- http://www.objectmentor.com/resources/articles/xpepisode.htm
  - Classic detailed Test First example (uses older Junit)
- http://blog.daveastels.com/files/BDD_Intro.pdf
  - Behavior Driven Development, an improved TDD

Books

- "User Stories Applied" by Mike Cohn: ISBN 0-321-20568-5
- “Lean Software Development: An Agile Toolkit” by Mary and Tom Poppendieck: ISBN-10: 0321150783
- “ Implementing Lean Software Development: From Concept to Cash” by Mary and Tom Poppendieck: ISBN-10 0321437381
- "Scrumban " by Corey Ladas: ISBN-10: 0578002140
- "Agile Project Management with Scrum" by Ken Schwaber: ISBN-10: 073561993X
- “Extreme Programming Explained (second edition)” by Kent Beck and Cynthia Andres ISBN 0321278658
- “Planning Extreme Programming” by Kent Beck and Martin Fowler ISBN 0201710919
- "Extreme Programming Installed” by Ron Jeffries, Ann Anderson and Chet Hendrickson : ISBN 978-0201708424
- “Test Driven Development by Example” by Kent Beck ISBN 0321146530
- “Test-Driven Development: A Practical Guide” by David Astels ISBN 9780131016491
- "Refactoring" by Martin Fowler: ISBN 0201485672
- “Pair Programming Illuminated” by Laurie Williams and Robert Kessler ISBN 0-201-74576-3

