Agile and Why It Works

An Overview of Agility

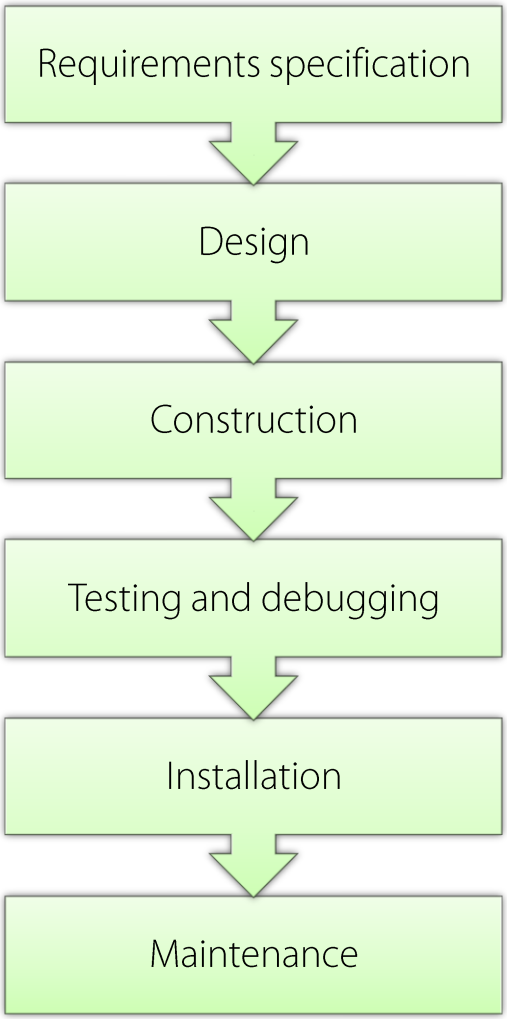
Overview

- Comparing Development Processes
- What Agile Is and Is Not
- Contemporary Agile Methodologies

Comparing Development Processes

Agile and Plan Driven

Winston Royce's Waterfall Model



From the 1970 IEEE paper:
"Managing the Development of Large Software Systems"

A careful reading of Royce's paper reveals:

1. **Each phase should pass iteratively to the next**
2. **The entire process should be exercised twice before release**
3. **Royce knew that a single pass will fail**

"Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps."

The Agile Manifesto

Original Signatories	
Kent Beck	Ron Jeffries
Mike Beedle	Jon Kern
Arie van Bennekum	Brian Marick
Alistair Cockburn	Robert C. Martin
Ward Cunningham	Steve Mellor
Martin Fowler	Ken Schwaber
James Grenning	Jeff Sutherland
Jim Highsmith	Dave Thomas
Andrew Hunt	

Manifesto for Agile Software Development

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.

http://agilemanifesto.org

The Big Difference

Plan Driven Software Development

Planning	Analysis	Develop	Test	Integrate	Validate	Deploy

Time

Agile Software Development

Planning									
Analysis	D	Analysis	D	Analysis	D	Analysis	D	Analysis	Deploy
Develop		Develop		Develop		Develop		Develop	
Test		Test		Test		Test		Test	
Integrate		Integrate		Integrate		Integrate		Integrate	
Validate		Validate		Validate		Validate		Validate	

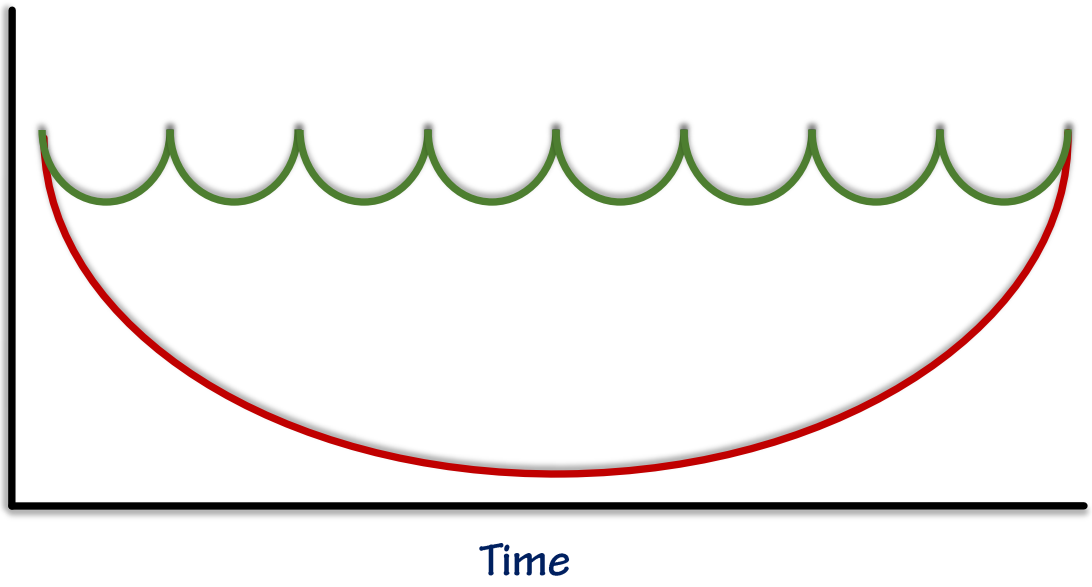
Comparing Methodologies

Plan Driven Methodologies	Agile Methodologies
Change is bad, therefore discouraged and actively controlled	Change is inevitable and valuable, therefore encouraged and embraced
Adherence to the plan determines success or failure	Incentives are often based on customer satisfaction and ROI
I am done when my part of the plan is signed off	I am done when the customer is happy

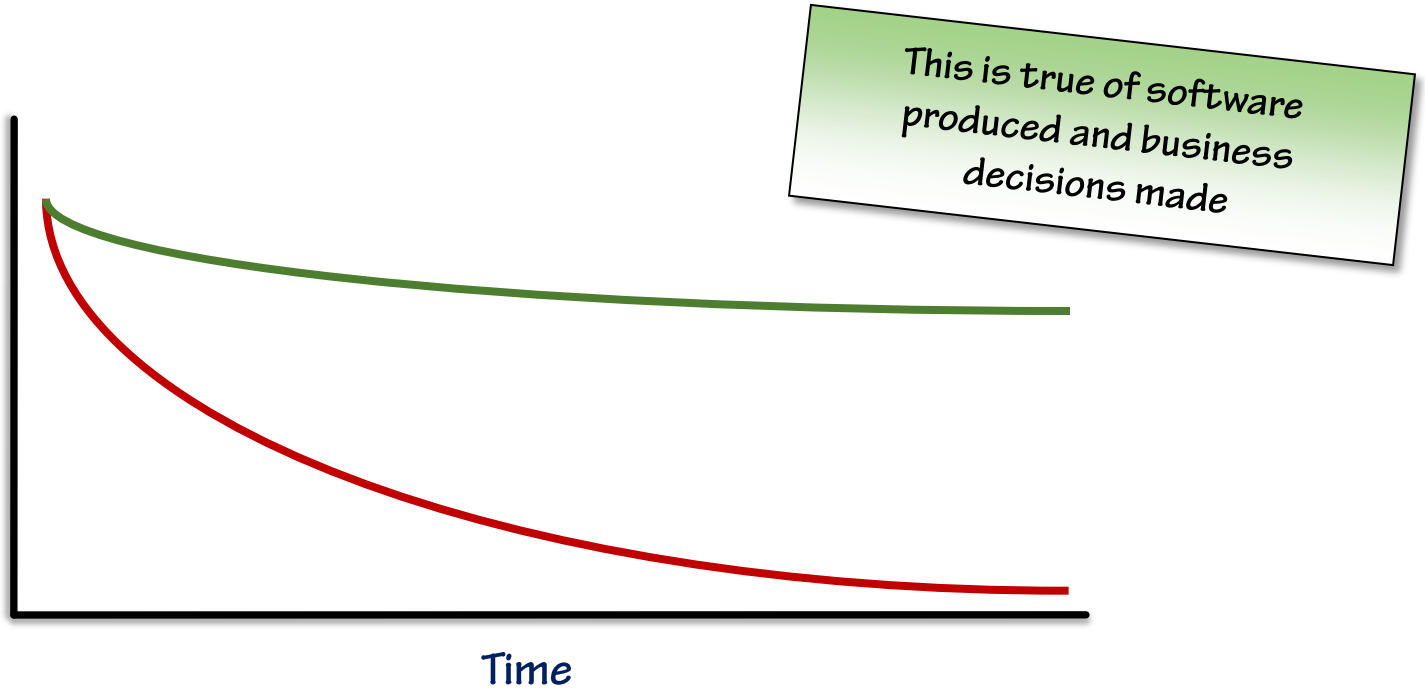
Comparing Methodologies

Plan Driven Methodologies	Agile Methodologies
Lots of gates to control quality	Highly iterative to achieve quality
Inspect product when it is complete	Inspect work as it is being done
Start by predicting what will be delivered	Start with a goal of filling a need

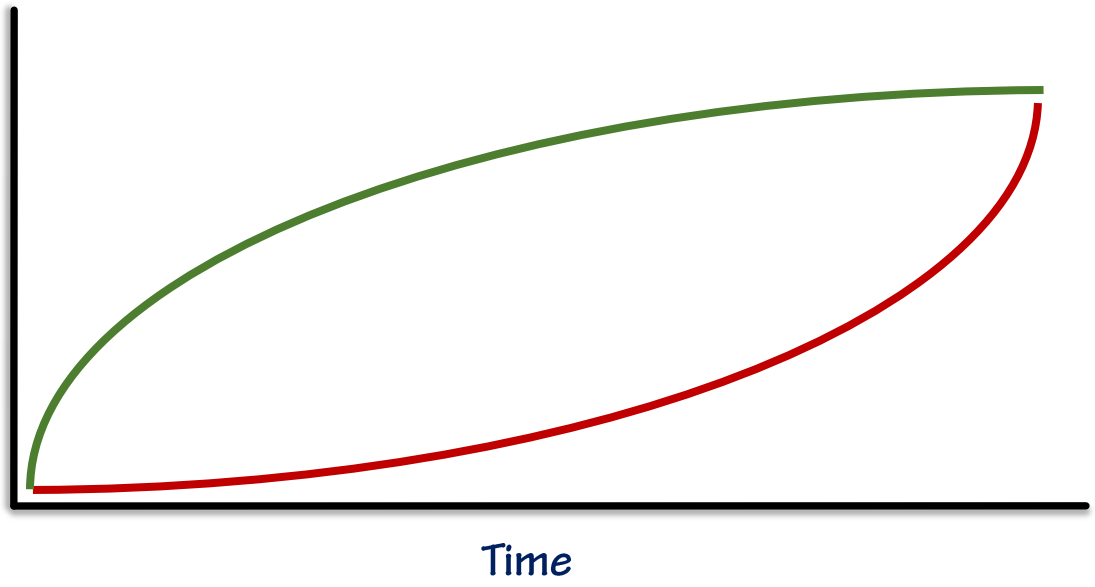
Visibility



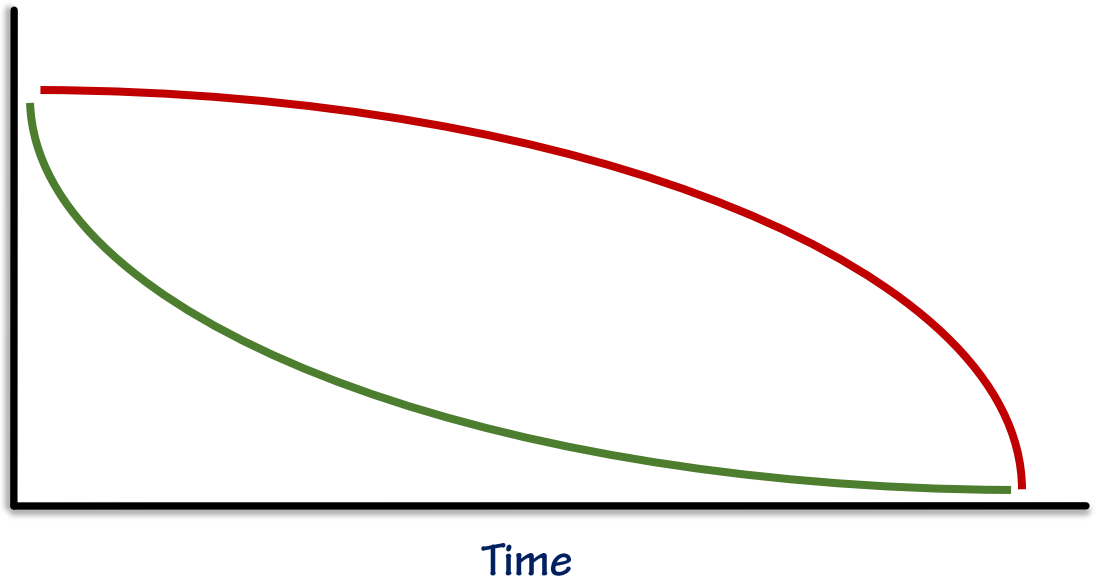
Ability to Change



Business Value



Risk



What Agile Is and Is Not

Dispelling Common Myths

What is Agile?

Plenty of opinions
and
a lot of white noise

RALLY SOFTWARE

industrial logic

Agile Alliance

Agile2009 Conference

GreenPepper

ScrumAlliance transforming the world of work

AgileLog

APLN

Cyber Korp The Agile Company

VERSIONONE Simplifying Software Delivery

Buy the Scrum Game!

Agile LEADERSHIP SUMMIT

Orlando Scrum Gathering

pluralsight
see what you can learn

Touchstones of Agility

Documents

- The Manifesto
- The Declaration for Interdependence

Principles

- Iterative Delivery
- Frequent Feedback
- Transparency and Openness
- Lean Principles

Come back to these things to recall what we are trying to accomplish

The 2 Parts of the Agile Discussion

Processes and Methodologies

Working with people

- Planning
- Teamwork
- Engaging customers
- Providing leadership
- Collaboration
- Learning

Techniques and Practices

Working with software

- Design
- Coding
- Testing
- Deploying
- User experience

The 2 Parts of the Agile Discussion

Processes and Methodologies

- Adaptive Software Development
- Agile Modeling
- Agile Unified Process
- Crystal Clear
- Dynamic Systems Development Method
- eXtreme Programming
- Feature Driven Development
- Lean Software Development
- Scrum
- Feature Driven Development
- User Stories

Techniques and Practices

- Automated Regression
- Behavior Driven Development
- Continuous Integration
- Design Patterns
- Domain Driven Design
- DRY, SoC, Reuse, Testability
- Incremental Design
- JIT Architecture
- Separation of Concerns
- Pair Programming
- Refactoring
- Test Driven Development

Agile Is

- Iterative
- Adaptive
- Value based
- Easy to understand
- Hard to implement

True for practices and methodologies

Agile Is Not

- Just about writing code
- Undisciplined
- Unstructured
- Whatever you want it to be
- A placebo for those pesky developers

Contemporary Agile Methodologies

Proven Practices

Extreme Programming (XP)

- The ancestor of most Agile methodologies
- Originated with Kent Beck in 1999
- Blends processes and practices
- Found success is smaller teams
- XP became controversial early on due to advocates teaching it as dogma

Extreme Programming (XP)

Kent Beck's basic idea

1. Take observed effective team practices
2. Push them to extreme levels

Good Practice	Pushed to the Extreme
Code Reviews	Pair Programming
Testing	TDD and constant regression
Software Design	Relentless Refactoring
Simplicity	The simplest thing that could possibly work
Integration Testing	Continuous Integration
Short Iterations	The Planning Game

XP's 12 Practices

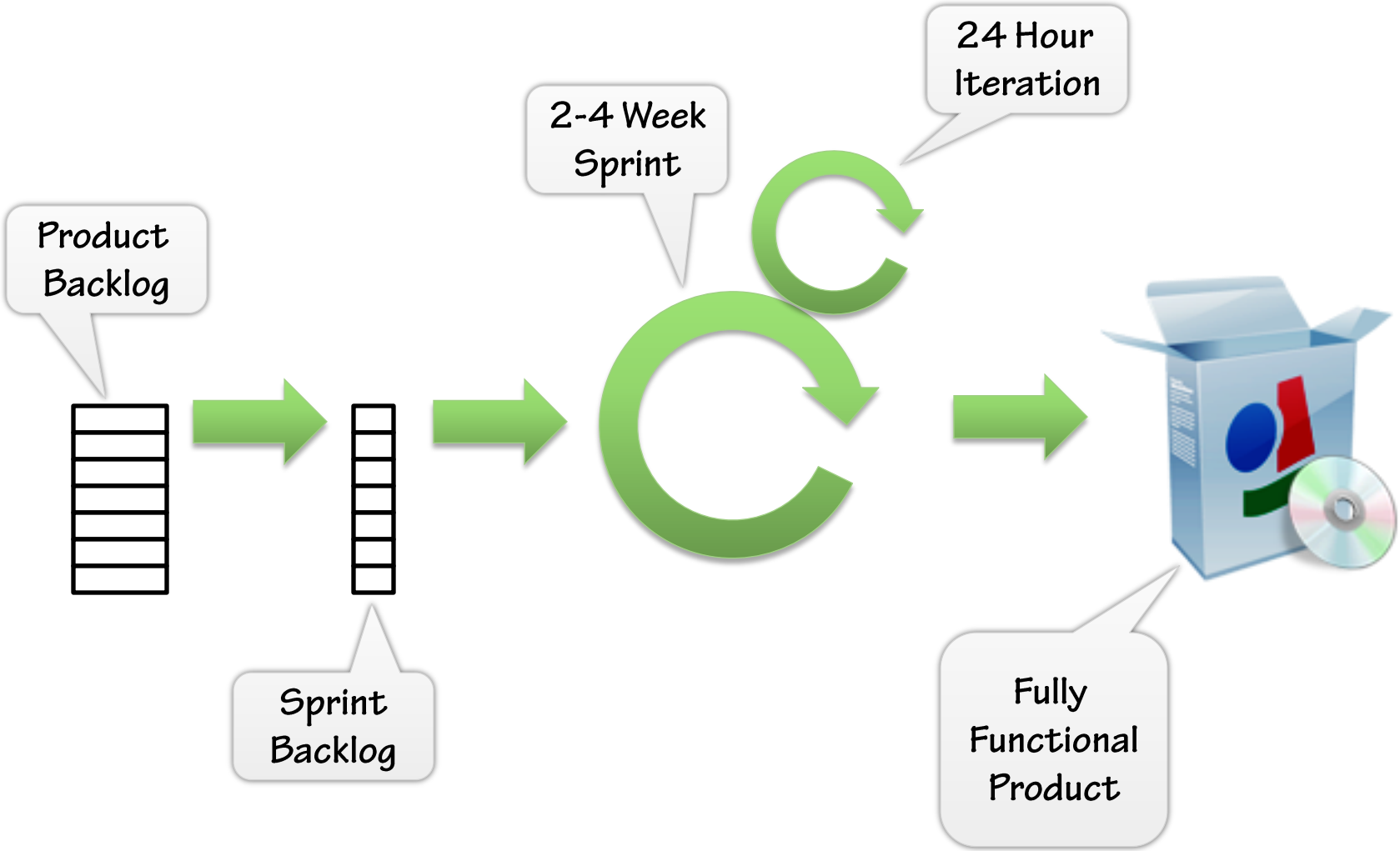
1. The Planning Game
2. Small Releases
3. Metaphor
4. Simple Design
5. Testing
6. Refactoring
7. Pair Programming
8. Continuous Integration
9. Collective Ownership
10. On-site Customer
11. The 40-hour Week
12. Coding Standards

This one is the subject of much debate

Scrum

- An iterative Project Management process
- Originated with Ken Schwaber and Jeff Sutherland in 1990s
- Does not advocate any specific engineering techniques
- Used beyond software development
- Blends well with XP practices
- Simple

Scrum


24 Hour Iteration
2-4 Week Sprint
Product Backlog
Sprint Backlog
Fully Functional Product

Lean Software Development

- More a set of guidelines than a formal methodology
- Originally applied in manufacturing, now used in software development
- Can be applied to improve any processes
- Focuses on Continuous Improvement (Kaizen) and value flow

Principles of Lean Software Development

- Eliminate waste
- Amplify learning
- Respect people
- Build quality in
- Defer commitment
- Deliver fast
- Recognize and optimize the whole

Feature Driven Development

- Originated with Jeff De Luca while working on a 50 person project at Singapore bank in 1997
- Adheres to strict process
- Based on time-honored engineering practices
- Advocates modeling as the base currency of process
- Some reports claim FDD scales more effectively than Scrum

Feature Driven Development

Activities	Milestones
1. Develop Overall Model	1 Domain Walkthrough
2. Buil	
3. Plan	
4. Desi	
5. Buil	

<action> <result> <object>

Sum the total for monthly sales
Validate the password of the user

FDD Practices

- Domain Object Modeling
- Developing by Feature
- Individual Code Ownership
- Feature Teams
- Inspections
- Configuration Management
- Regular Builds
- Highly Visibility Progress and Results

Summary

- **Agile is a set of principles**
  - Individuals and interactions **over** processes and tools
  - Working software **over** comprehensive documentation
  - Customer collaboration **over** contract negotiation
  - Responding to change **over** following a plan
- **Agile is not a distinct process or methodology**
- **Agile does have some distinct attributes**
  - Frequent and iterative product delivery
  - Adaptive and responsive to feedback
  - Close customer relationships
  - Highly transparent

References

- **The Manifesto for Agile Software Development**
  AgileManifesto.org
- **Project Manager's Declaration of Interdependence**
  www.pmdoi.org
- **Agile Alliance**
  agilealliance.org
- **Iterative and Incremental Development: A Brief History**
  http://www2.computer.org/portal/web/csdl/abs/mags/co/2003/06/r6047abs.htm
- **Benefits of Agile Development**
  http://www.versionone.com/Resources/AgileBenefits.asp
- **Managing the Development of Large Software Systems**
  http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf