



## Agile Software Development



## Goals/Objectives

- Gain an understanding of:
  - Agile Software Development
  - Solve problems experienced by Agile teams today and tomorrow
- Be able to ... speak **confidently** and **eloquently** about Agile and provide solutions for the problems that exist within Agile teams



# Agenda

- Introduction to Agile
- Principles of Agile (Agile Manifesto)
- The People Involved
- Product Manager vs. Product Owner
- Release vs. Sprint vs. Iteration
- Product Backlog vs. Sprint Backlog
- Agile Reports
- Agile Solution Providers
- Problems with Agile and how we can Help
- Resources



## Introduction to Agile (video)

Get **Agile: Agile** vs Waterfall

<http://www.youtube.com/watch?v=XokJLWp7icI>



# Introduction to Agile

Agile software development refers to a group of software development methodologies that are based on similar principles. Agile methodologies generally promote:

- A **project management process** that encourages frequent inspection and adaptation;
- a leadership philosophy that encourages **team work, self-organization** and **accountability**;
- a set of engineering best practices that allow for **rapid delivery of high-quality** software;
- and a business approach that aligns development with customer needs and company goals.

MYTH: No documentation

MYTH: Undisciplined

MYTH: Agile is a process



# Manifesto for Agile Software Development

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

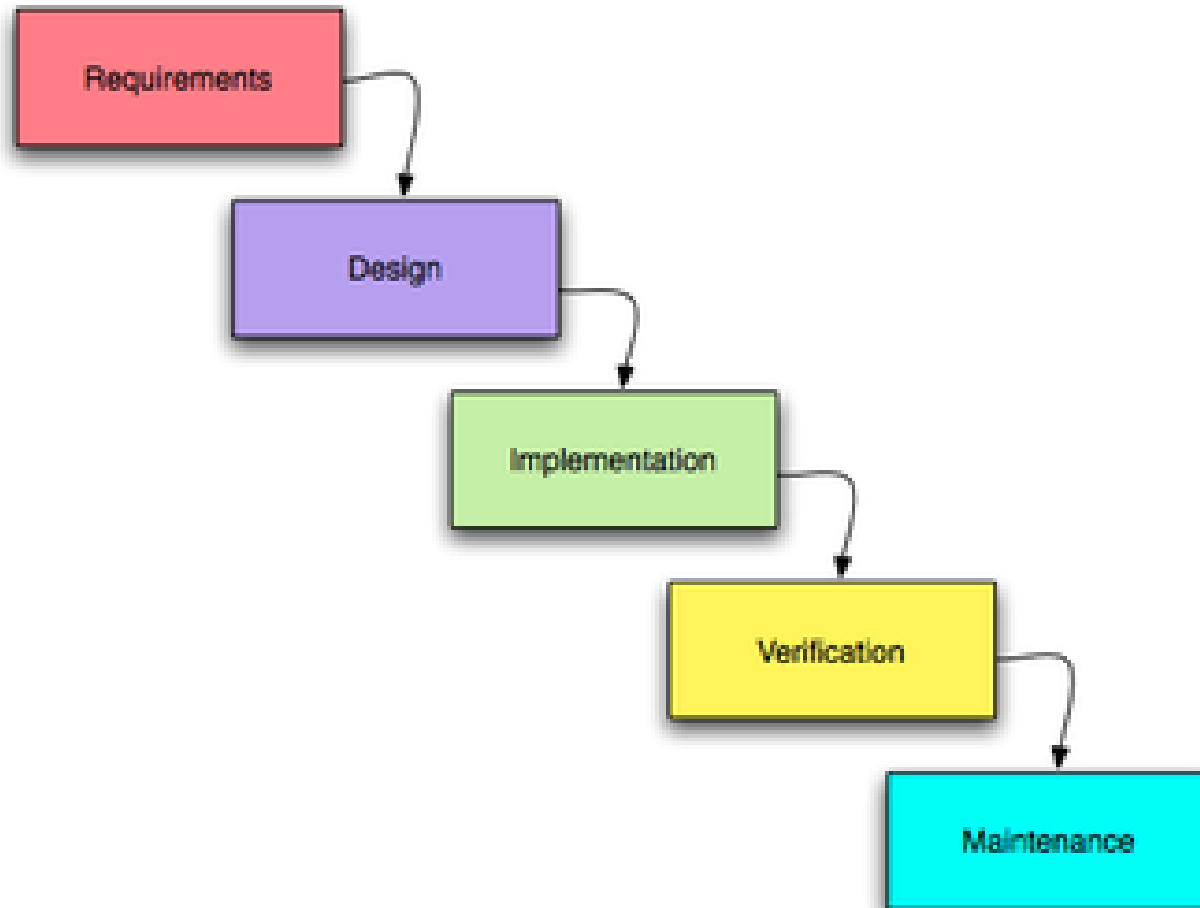


# Principles of Agile

Some of the principles behind the Agile Manifesto are:

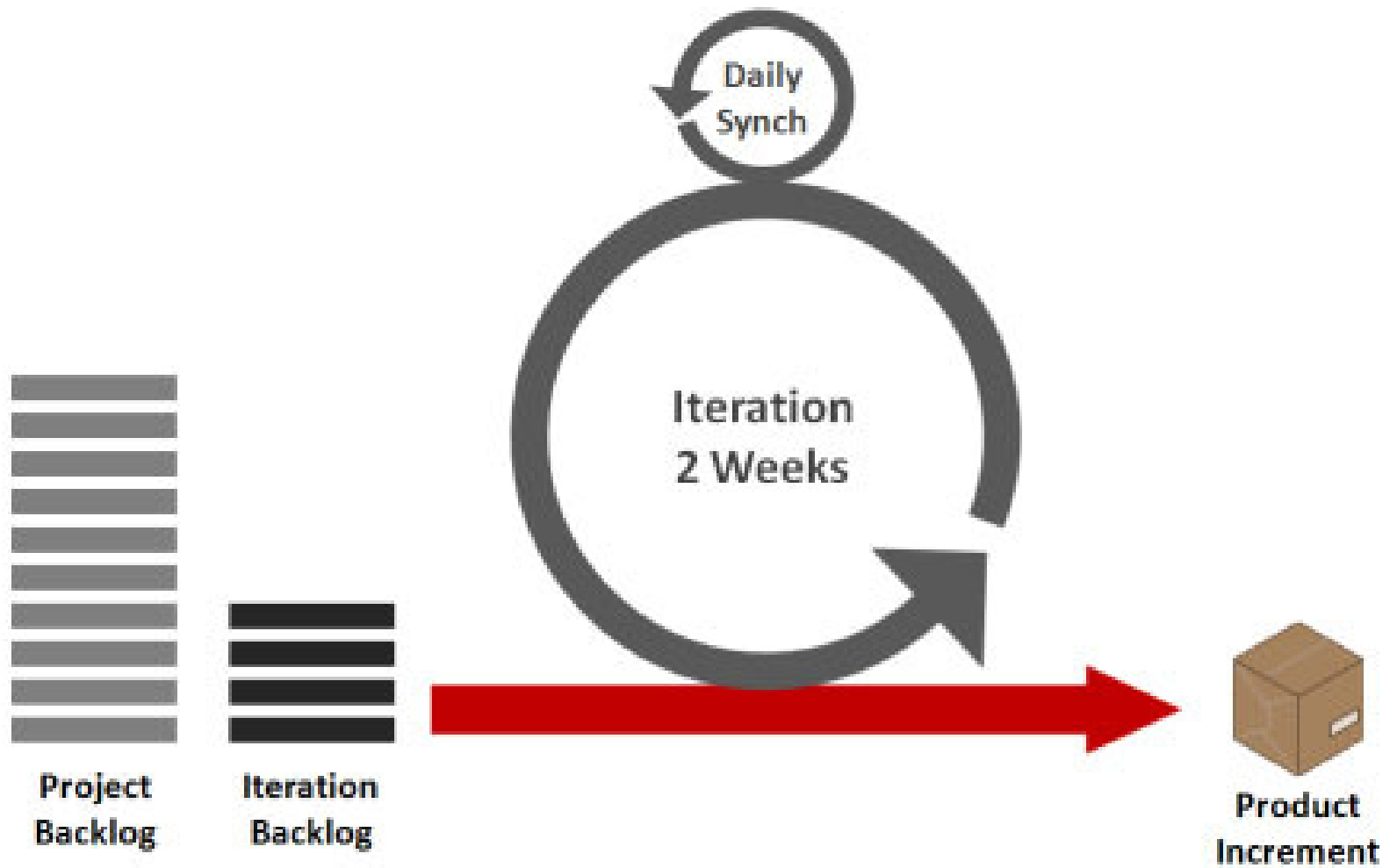
- **Customer satisfaction** by rapid, continuous delivery of useful software
- **Working software** is delivered frequently (weeks rather than months)
- Working software is the principal **measure of progress**
- Even **late changes in requirements** are welcomed
- Close, daily cooperation between business people and developers
- **Face-to-face** conversation is the best form of communication (Co-location)
- Projects are built around motivated **individuals**, who should be trusted
- Continuous attention to **technical excellence** and good design
- **Simplicity**
- **Self-organizing** teams
- Regular **adaptation** to changing circumstances

# Waterfall





# Agile





# Introduction to Agile Methodologies

- SCRUM (most popular) is an iterative process of development used with agile software development. The roles in Scrum are the ScrumMaster, the Product Owner and the Team. During each sprint the team creates an increment of potential shippable software. The set of features that go into each sprint come from the product backlog. Which backlog items go into the sprint is determined during the sprint planning meeting. The team then determines how much of this they can commit to complete during the next sprint. During the sprint, no one is able to change the sprint backlog.
- Extreme Programming (XP) (third) is a traditional software engineering practices (daily interactions, working software, testing, etc.) taken to so-called "extreme" levels, leads to a process that is more responsive to customer needs ("agile"), while creating software of better quality.
- SCRUM/XP (2<sup>nd</sup>)



# Introduction to Agile Methodologies (cont'd)

- Feature-driven Development (rare) an iterative development process. This practice is driven from a client-valued functionality (feature) perspective.
- Test-driven Development (rare) is a software development technique that uses short development iterations based on pre-written test cases that define desired improvements or new functions. Each iteration produces code necessary to pass that iteration's tests.
- Lean (rare) is a process designed to eliminate waste, amplify learning, make decisions late, fast delivery, empowered teams, product integrity and an fully understood system.
- Custom (rare)

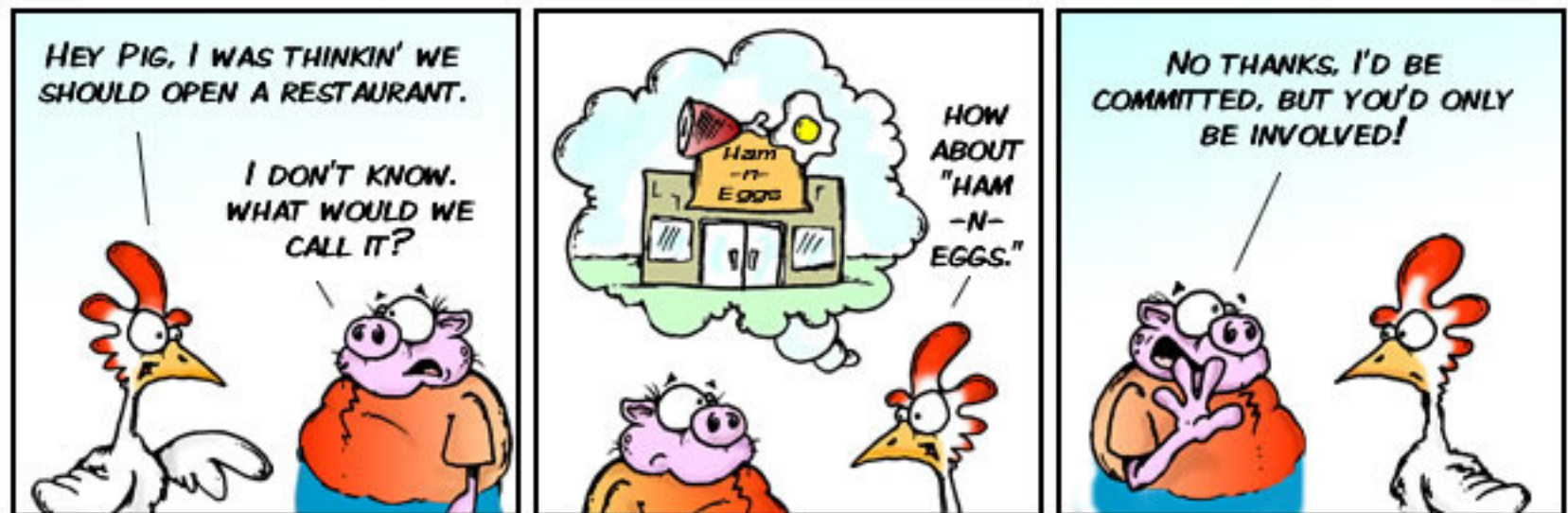


# Why They Move

- Projects too long to estimate accurately
- Increase short-term predictability
- Transparency
- Flexibility
- Agile builds empowered, motivated and self organizing teams
- Clear expectations are set and communicated
- Customers communicate directly with the team and provide timely feedback
- Teams feel a sense of accomplishment and recognition

# The People Involved

- Pigs and Chickens



By Clark & Vizados

© 2006 implementingscrum.com



## The People Involved (cont'd)

- The Agile Team
  - Architects
  - Developers, Analysts, Designers
  - Quality Assurance, Testers
  - Tech Writers
- Product Owners / Customer Representatives
- Project Managers / ScrumMasters
  
- Note: Teams are small (~7), preferably co-located, cross functional. Scale by adding teams, not more people



# The “Other” People Involved

- Product Marketing
  - Marketing Communications
  - Directors
  - Executives
- 
- Note: Can attend dailies but cannot speak. Can participate in planning. Interact with team through Project Manager / ScrumMaster



# Product Manager vs. Product Owner (video)

**David Howard**

**Agile development isn't product management**

- <http://www.youtube.com/watch?v=sRWJF6o-jlQ>





# Product Manager vs. Product Owner

- Product Owner owns the message to the Agile team and the Sprint/Iteration Backlog
- Product Manager owns the Roadmap / Strategy / Vision / Product Backlog
- Often they are the same person and the problems with this are well documented

Note: Only SCRUM calls them Product Owners, other flavours reference customer representative, visionary, business/domain experts

Resource: Scaling Software Agility (blog) by Dean Leffingwell

<http://scalingsoftwareagility.wordpress.com/category/product-ownerproduct-manager/>

# Product Manager vs. Product Owner (cont'd)

Product Manager	Product Owner
Market Sensing / Problem Statements	Tracks internal deliveries
Release Objectives	Iteration Objectives
Strategic Direction	Day-to-day tactical direction
Market Use Cases/Scenarios	System Use Cases/Scenarios
Understands overall solution	Understands architecture and design
Roadmaps	User acceptance tests
Manage Release Portfolios and Backlogs	Manage Iteration and Cross Project priorities
Provides Vision	Provides Implementation
Messaging & Positioning	Unblocks teams throughout iteration
Directs Product Owner	Takes direction from Product Manager
Delivers the Release	Delivers the Iteration

[Agile Product Owner \(.com\) http://agileproductowner.com/?p=15](http://agileproductowner.com/?p=15)



# What is a backlog?

- A backlog is the master list of all functionality
  - Features
  - Epics
  - Stories
  - Requirements
  - Bugs
- Item Attributes:
  - Description
  - Cost estimate (points or size)
  - Business Value
  - Priority



# Product Backlogs vs. Sprint Backlogs

- A **Product Backlog** is the master list of all functionality for the product
  - Features
  - Epics
  - Bugs
- The **Sprint Backlog** is the list of functionality that the team is committing that they will complete in the current iteration
  - Storied
  - Requirements
  - Bugs
- Product Management / Owner owns the backlogs



# Agile Planning

- Beginning of iteration
- Review vision and roadmap
- Review development status, architecture and previous iterations
- Determine velocity
- Review team availability & capacity
- Review definition of done
- **Review product backlog & select items for iteration / sprint**
- **Identify tasks & estimates**
- **Identify challenges & risks**
- Review capacity required
- **COMMIT** (planning not over until entire team commits to sprint plan)



# Velocity

- Velocity is a method for measuring the rate at which teams consistently deliver business value.
- To calculate velocity, simply add up the estimates of the items successfully delivered an iteration.



# Definition of Done

- Is a checklist of valuable activities required to produce software
- The team defines definition of done for each iteration, feature, requirement task, etc.
- Typically
  - Software working?
  - Unit tests written?
  - Code review done?
  - Functional tests written and passed?
  - Non-functional tests written and passed?
  - User documentation done and reviewed?
  - Design documentation updated and reviewed?
  - Release notes updated?



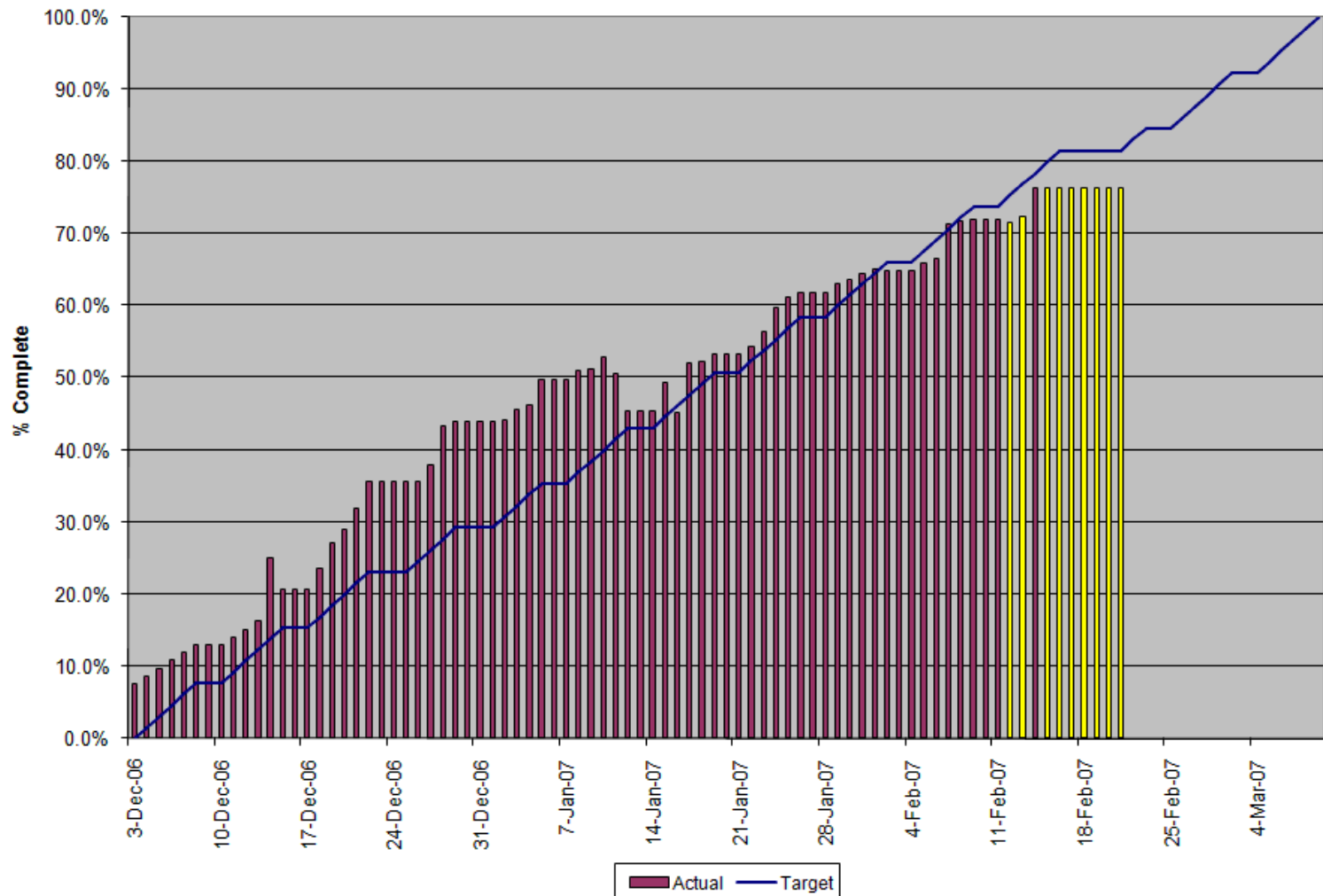
# Agile Reports

- Product / Sprint / Iteration Backlog
- Release Burn-up chart (to show % complete)
- Defects – inflow, outflow, # of open defects per week
- Build quality per day/week
- Number of tests / tests passed per day/week
- Velocity over the last x sprint
- Action items, risks
- Burndown Charts (to show remaining effort)



# Agile Reports (cont'd)

- Release Burn-up chart (to show % complete)



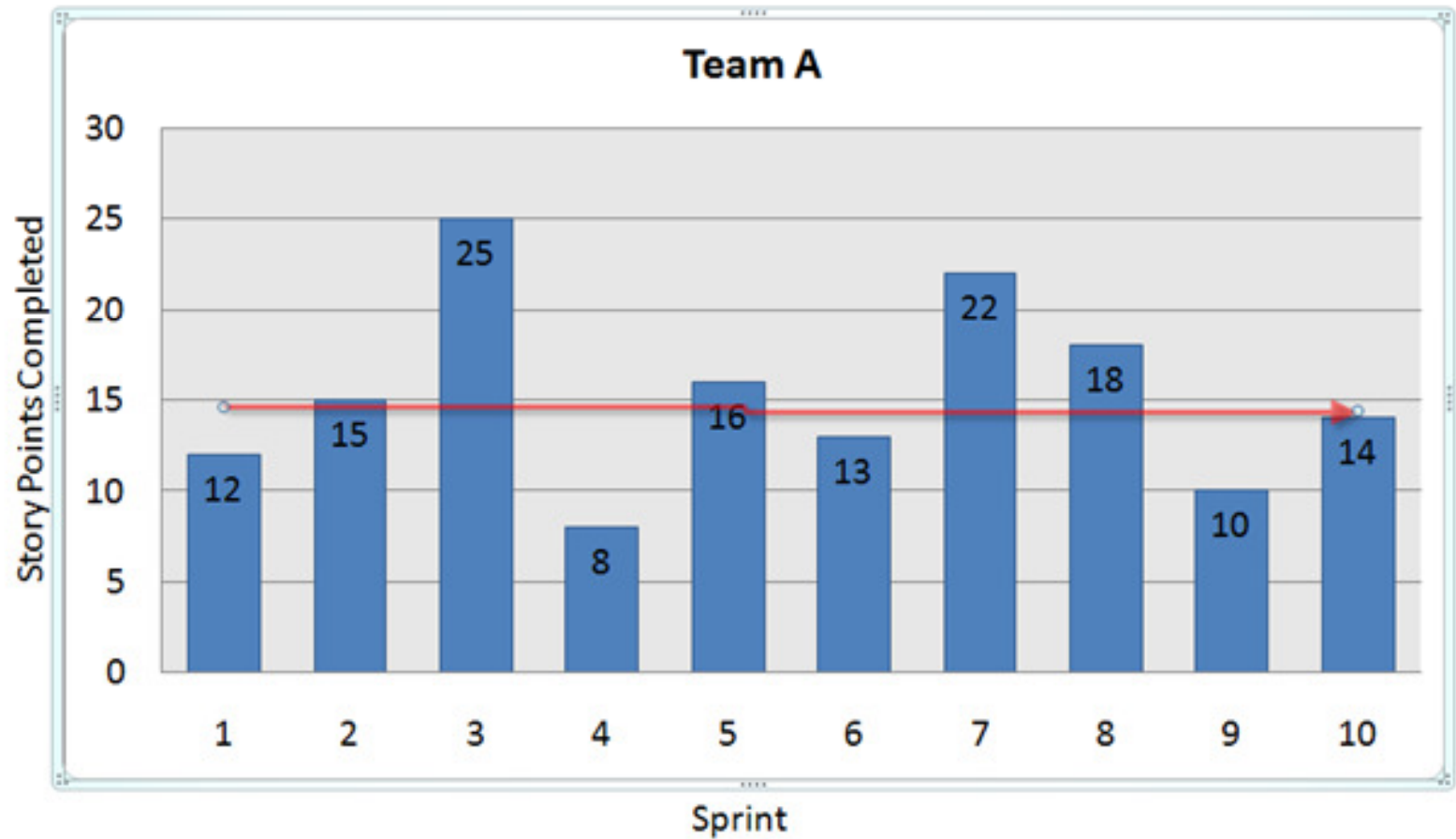
## Agile Reports (cont'd)

- Build quality per day/week

Feature Area	Rating
Setup	Partially Testable
Work Item Tracking	Self Test
Project Management	Self Test
Team Build	Self Toast
Version Control	Self Test
Reporting	Partially Testable
EDT Integration	Self Test

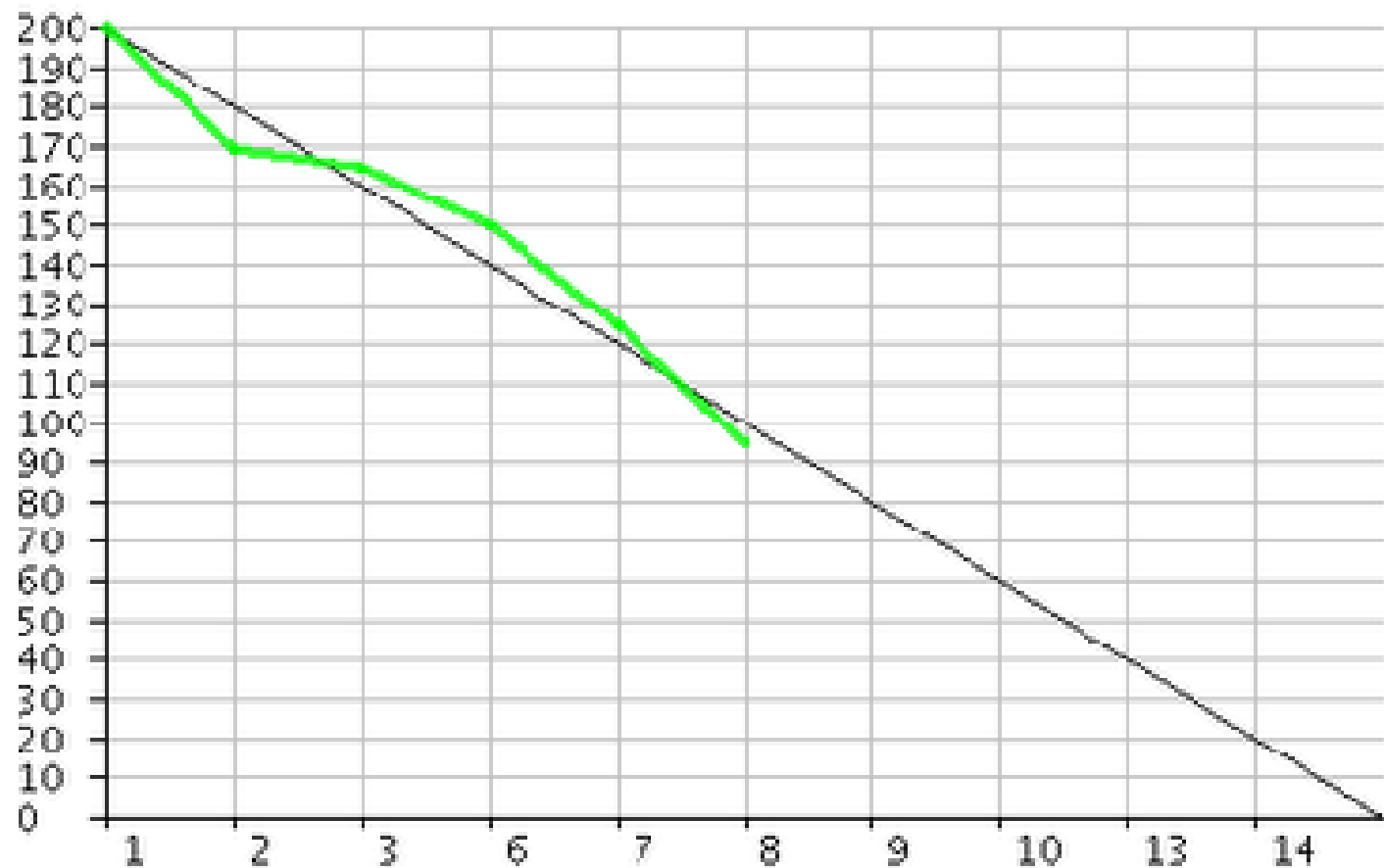
# Agile Reports (cont'd)

- Velocity over the last x sprint



## Agile Reports (cont'd)

- Burndown Charts (to show remaining effort)





# Agile Retrospectives

- Meeting after an increment of work to inspect and adapt their methods and teamwork
  - What went well
  - What could be improved
- Team learning
- Catalyst for change
- Generate action
- Sample Agenda
  - Overview
  - Gather data
  - Prioritize
  - Discuss
  - Action Items



# Agile Estimating

- Typically stories estimated twice
  1. High level estimate provided by team (for the product backlog)
  2. Iteration estimate provided by team (for the sprint)
- Story Points
  - Used to quantify the difficulty, not to identify time
  - Typically done in powers of 2 (1,2,4,8, 16...) or Fibonacci (1,2,3,5,8...)
- Everyone estimates often: at the beginning of the iteration, daily during the iteration to estimate the remaining effort
- The effort remaining (and not the effort already spent) is displayed publicly to enable collaborating teams that work together to meet the target of the iteration



# Agile Solution Providers

- VersionOne, Inc. - V1:Agile Enterprise
- Rally Software Development Corp - Rally Enterprise
- Danube Technologies, Inc. - Scrumworks
- XPlanner (open source)
- ThoughtWorks Inc. – Mingle

These are primarily positions as **Agile Project Management Solutions**



# Resources

## **BLOGS:**

- Agile Advice (<http://www.agileadvice.com/>)
- Implementing Scrum (<http://www.implementingscrum.com/>)
- James Shore (<http://jamesshore.com/>)
- Mike Cohn's Blog Succeeding With Agile <http://blog.mountangoatsoftware.com/>

## **BOOKS:**

- \*Anything\* by Mike Cohn
- The Art of Agile Development by James Shore
- See <http://www.mountangoatsoftware.com/reviews> for more

## **WEBINARS:**

- All presentations on InfoQ about Agile  
<http://www.infoq.com/bycategory/contentbycategory.action?idx=3&ct=5&alias=agile>
- The Product Management View (Past Webinars – search for Agile)