

# Align Business Goals and Operational Excellence through Agile Practices

Declan Whelan

[dwhelan@dpwhelan.com](mailto:dwhelan@dpwhelan.com)

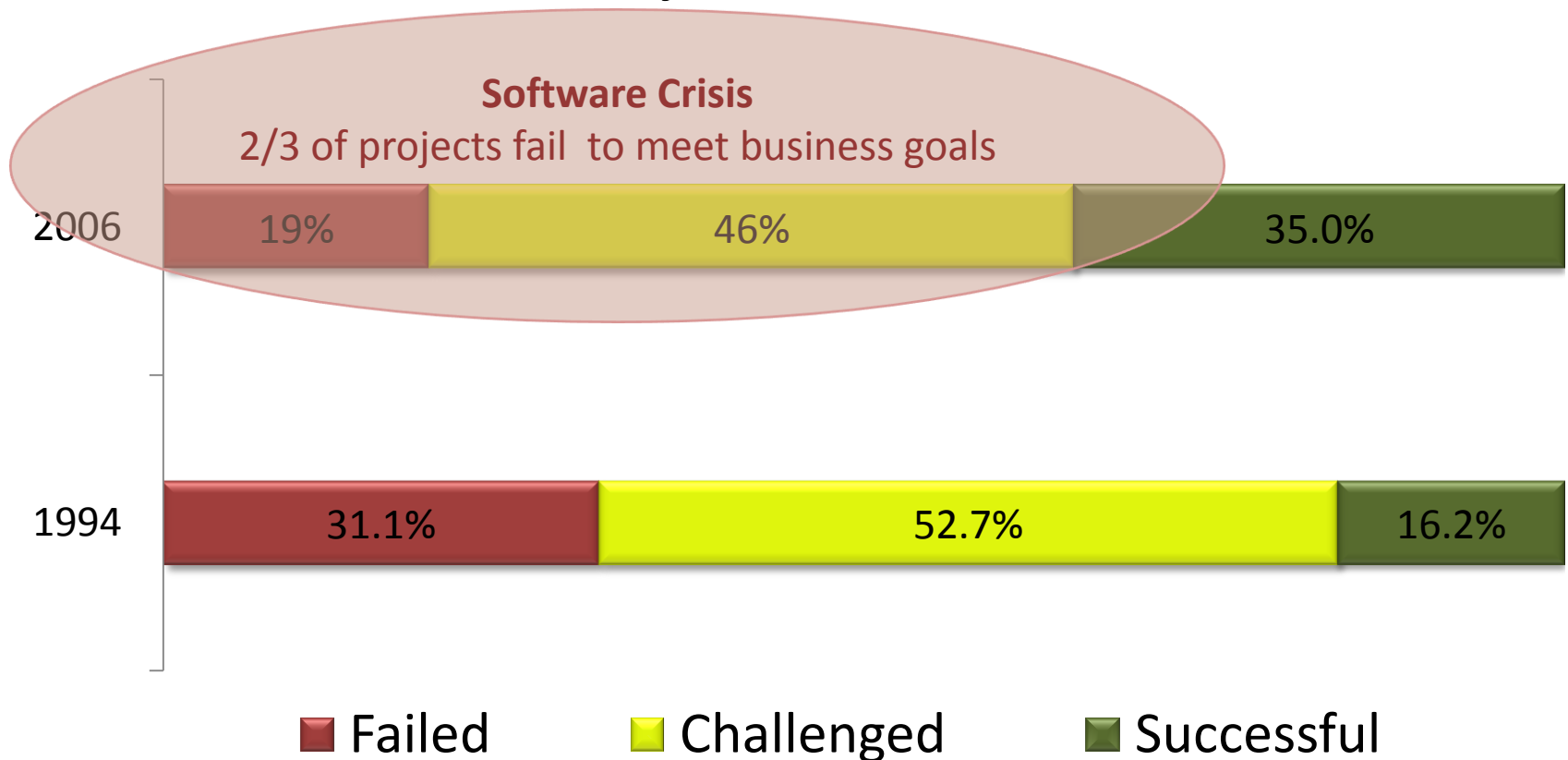
This presentation is licensed under a [Creative Commons Attribution 2.5 License](https://creativecommons.org/licenses/by/2.5/), which means you can share and adapt it, including commercial and derivative works, as long as you include attribution to Declan Whelan.

# Agenda

- The Software Crisis
- What is Agility?
- ROI & Time to Market
- Quality
- Risk
- When Does Agility Fit and When Does it Not?
- Agile Myths
- Panel Discussion

# The CHAOS Chronicles

## Software Project Success – 1994, 2006



"The CHAOS Chronicles" 1994, 2006 The Standish Group

© 2007 Whelan & Associates

# CHAOS Report - Top 10 Success Factors

	<i>Reason</i>	<i>Agile Impact*</i>
1	User Involvement	★★★★★
2	Executive Management	★★★
3	Clear Business Objectives	★★★
4	Optimizing Scope	★★★★★
5	Agile Process	★★★★★
6	Project Management Expertise	★★★
7	Financial Management	★★★
8	Skilled Resources	★★★
9	Formal Methodology	★★★★
10	Standard Tools and Infrastructure	★★★★

Success Factors from "The CHAOS Chronicles" 2006 The Standish Group \* Agile Impact from Declan Whelan

# Jim Johnson on Agility

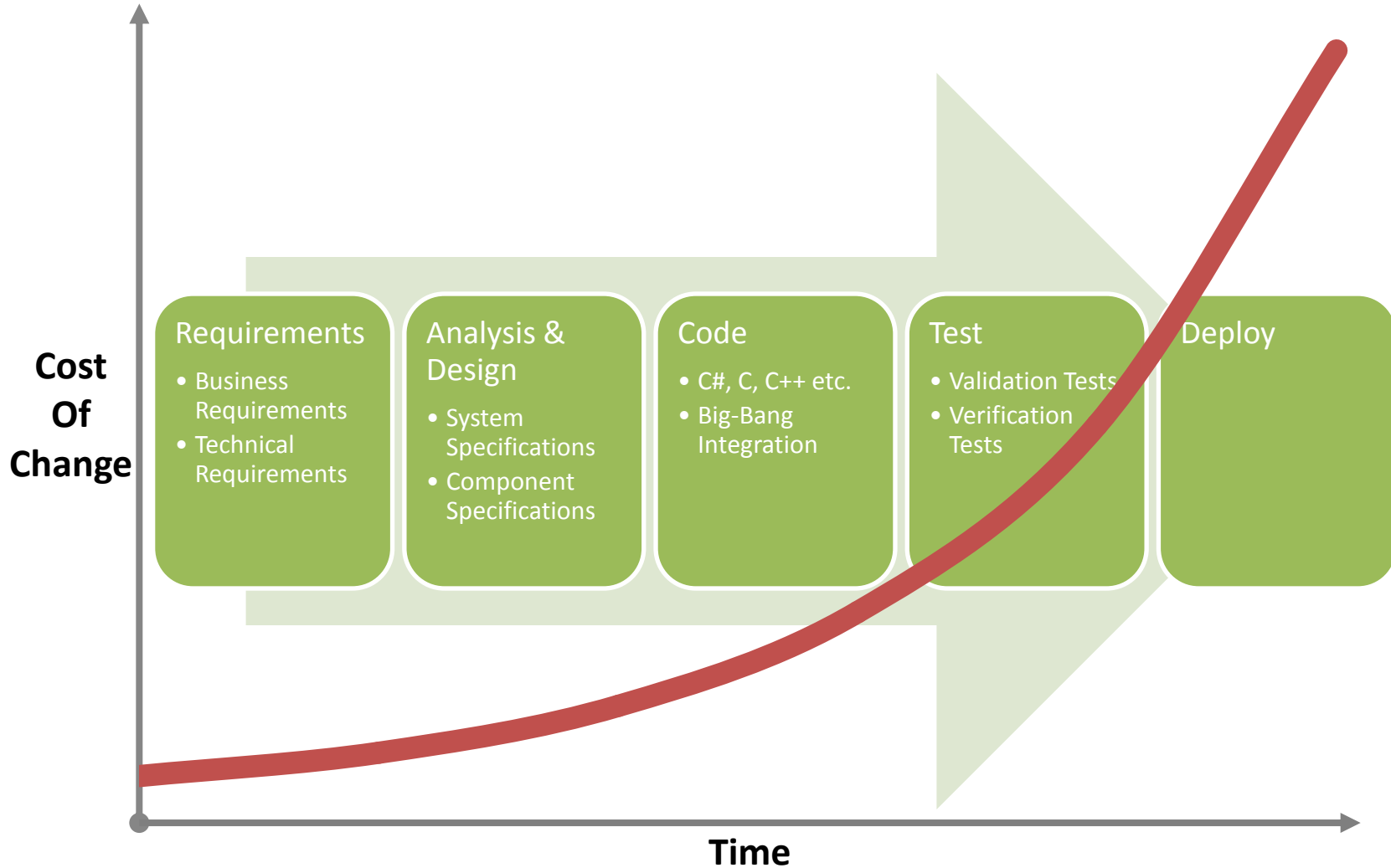
*I'm a big believer in Agile, having introduced iterative process in the early 90s ...we're a real flag waver for small projects, small teams, Agile process.*

*A big problem was project bloat, causing projects to go over time, over budget, and creating features and functions not required ... Agile really helps this - small increments.*

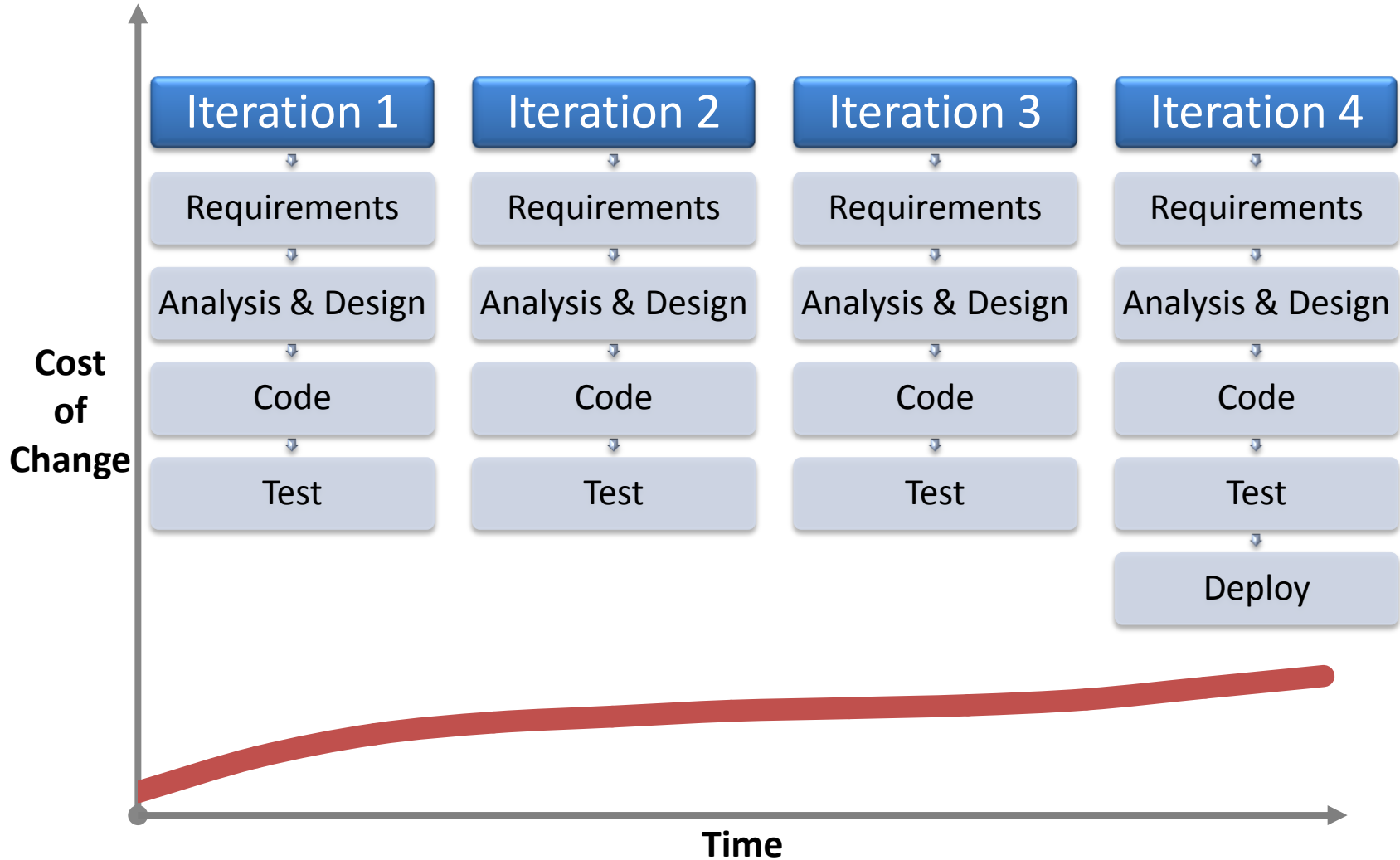
*Companies like Webex, Google, Amazon, eBay - they're doing something called "pipelining" instead of releases. "Whatever is done in 2 weeks, we'll put that up." They're successful because users only get small, incremental changes.*

Jim Johnson  
Chairman, Standish Group

# Traditional Approach - Waterfall



# Agile Approach



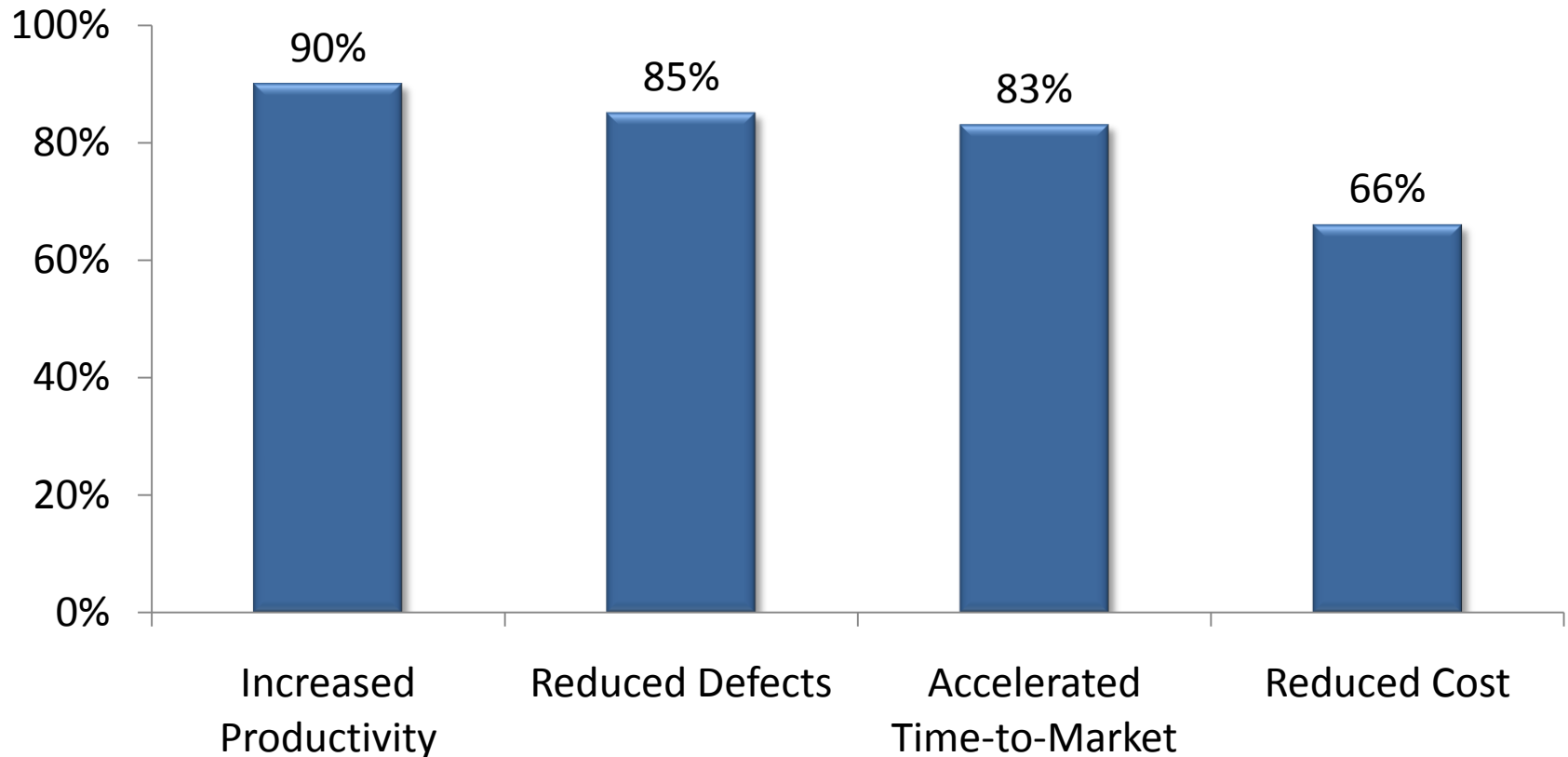
# Two Responses to Software Crisis

Traditional Plan Driven	Agile
Requirements, Design, Develop, Test, Deploy <u>once</u>	<u>Iterative</u> delivery with all S/W phases
Make decisions <u>early</u>	Defer decisions to <u>last responsible moment</u> - JIT
Project gates and <u>hand-offs</u> of <u>documents</u>	Iterations with <u>working software</u>
<u>Adhere to plan</u> and carefully control change	<u>Embrace change</u> and use it to competitive advantage
<u>Documentation</u> for control & review	<u>Face-to-face</u> communications
<u>Specialists</u>	Cross-functional <u>teams</u>



# Main Agile Benefits

**Specific improvements you have actually realized from implementing Agile Practices.**



*"The State of Agile Development"* Annual Survey – July 2007 – VersionOne: [http://www.versionone.com/pdf/StateOfAgileDevelopmet2\\_FullDataReport.pdf](http://www.versionone.com/pdf/StateOfAgileDevelopmet2_FullDataReport.pdf)

# What is Agility?

***“Practices that focus on team communication and feedback to regularly deliver customer value through working software.”***

# Agile Values

Individuals & Interactions

Processes & Tools

Working Software

Comprehensive  
Documentation

Customer Collaboration

Contract Negotiation

Responding to Change

Following a Plan

<http://agilemanifesto.org/>

# Agile Principles

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity - the art of maximizing the amount of work not done - is essential

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

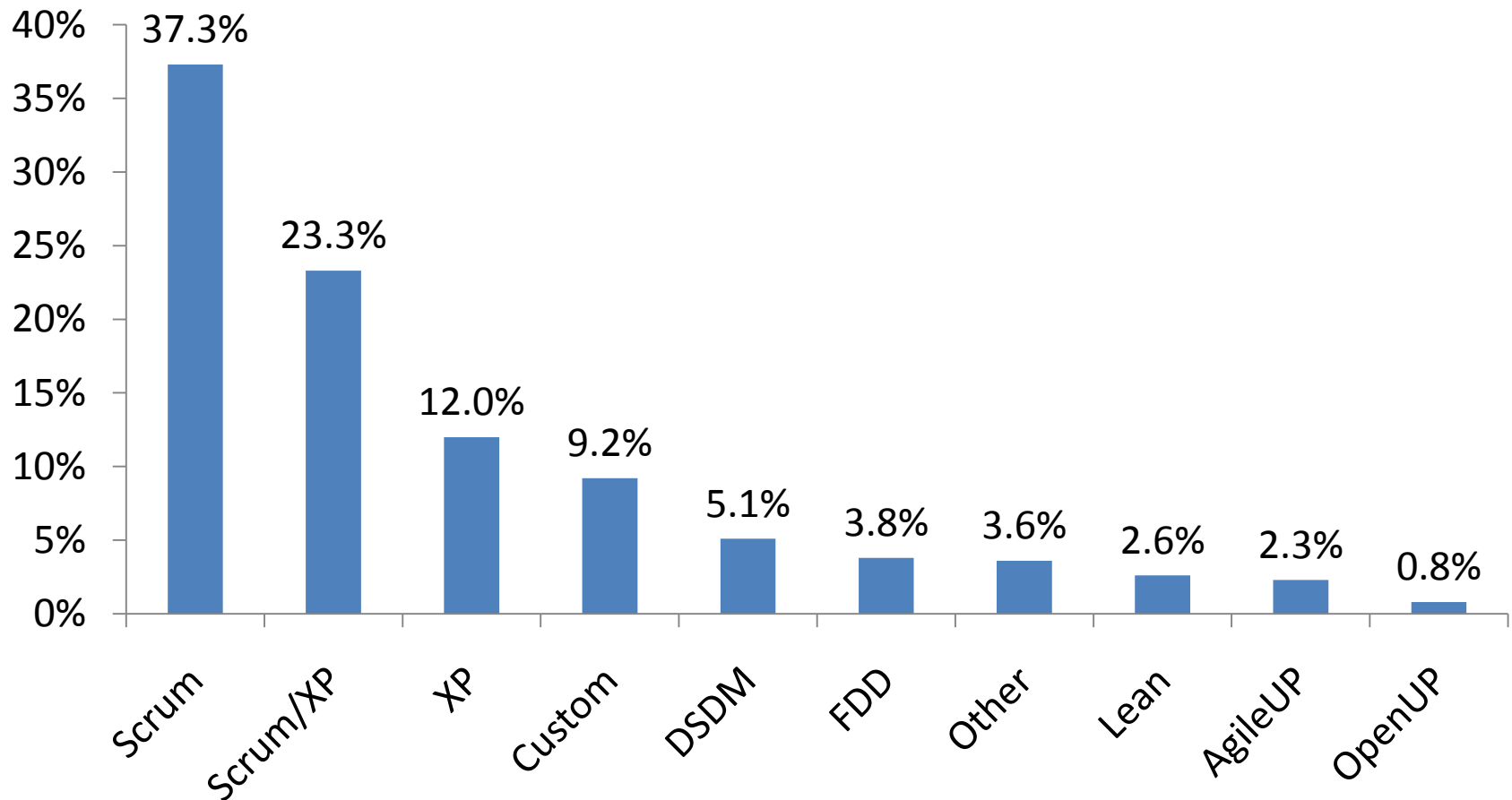
<http://agilemanifesto.org/principles.html>

# Agile Practices – They Aren't New!

Practice	Introduced
Requirements	00's Beginning of time
Pairs Programming	50's von Neumann/IBM
Test-Driven Design	60's NASA Project Mercury
Project Planning	60's NASA Project Mercury
Risk Management	60's NASA Project Mercury
Software Reuse	60's AT&T McIlroy
Software Architecture	60's Brooks/Dijkstra/Parnas
Data Hiding & Abstraction	70's Parnas
Simple Design	70's Basili/Turner
Continuous Integration	70's FSC Integration Engineering
Documentation	70's Parnas
Collective Ownership	70's Unix/Open Source
Incremental Releases	70's Basili/Turner
Coding Standards	70's Kernighan/Plauger
On-site Customer	70's Mills/IBM FSD
Software Metrics	70's Gilb/Halstead
Evolutionary Design	80's Gilb
Patterns	80's DeMarco/Lister/GoF
Peopleware/Sustainable Pace	80's DeMarco/Lister
Refactoring	90's Opdyke and Fowler
Metaphor	90's Beck/Fowler/Cunningham
Retrospectives	90's Kerth/Rising

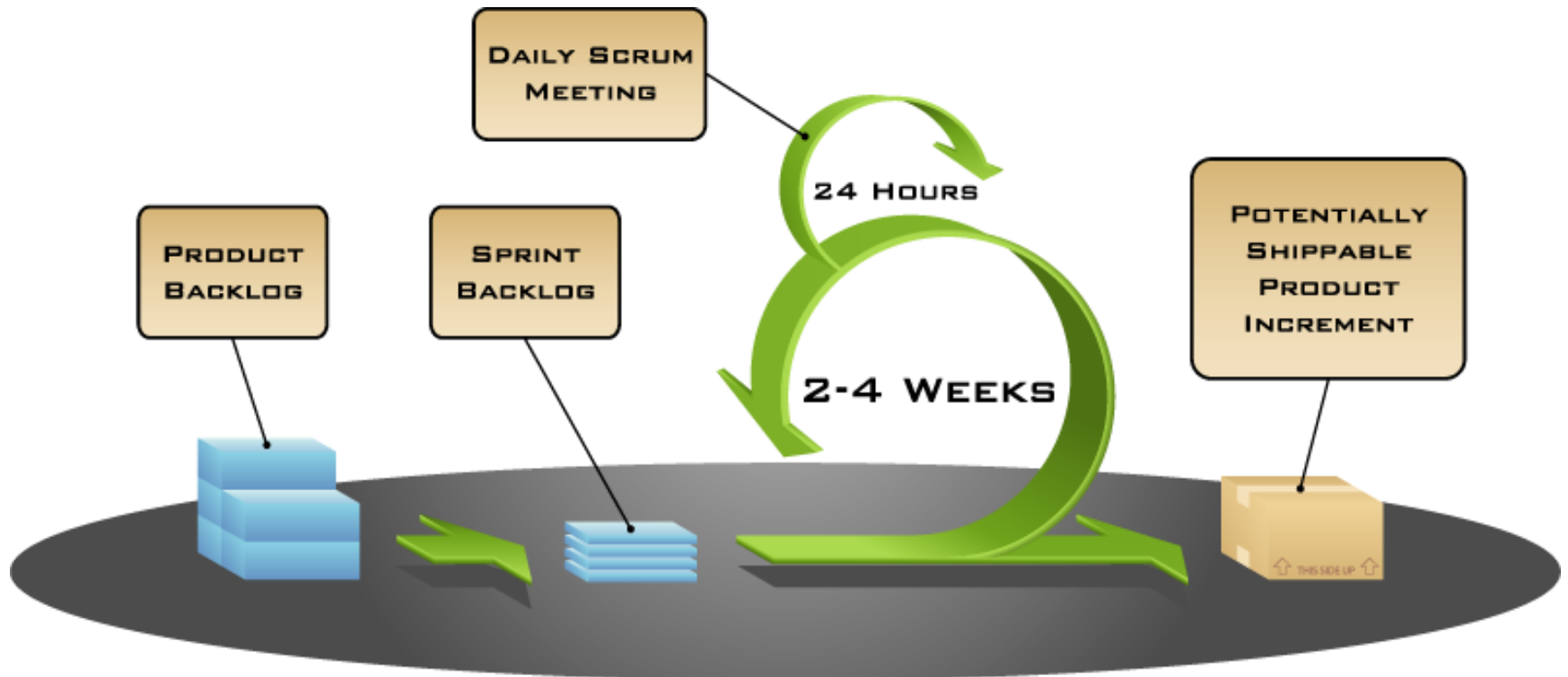
Source: Software Best-Practices: Agile Deconstructed - Steven Fraser OOPSLA 2007

# Agile Methodology Adoption



"The State of Agile Development" Annual Survey – July 2007 – VersionOne: [http://www.versionone.com/pdf/StateOfAgileDevelopmet2\\_FullDataReport.pdf](http://www.versionone.com/pdf/StateOfAgileDevelopmet2_FullDataReport.pdf)

# Agile Practices - Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

[http://www.mountaingoatsoftware.com/scrum\\_figures](http://www.mountaingoatsoftware.com/scrum_figures)

© 2007 Whelan & Associates

# Scrum Team

## Product Owner

- Feature definition
- Release dates
- Single decision point
- Accepts or rejects work
- ROI

## ScrumMaster

- Represents management
- Removes obstacles
- Ensures Scrum process
- Servant leader

## Team

- Self organizing
- Cross-functional
- Estimates
- Tracks
- Gets 'er done



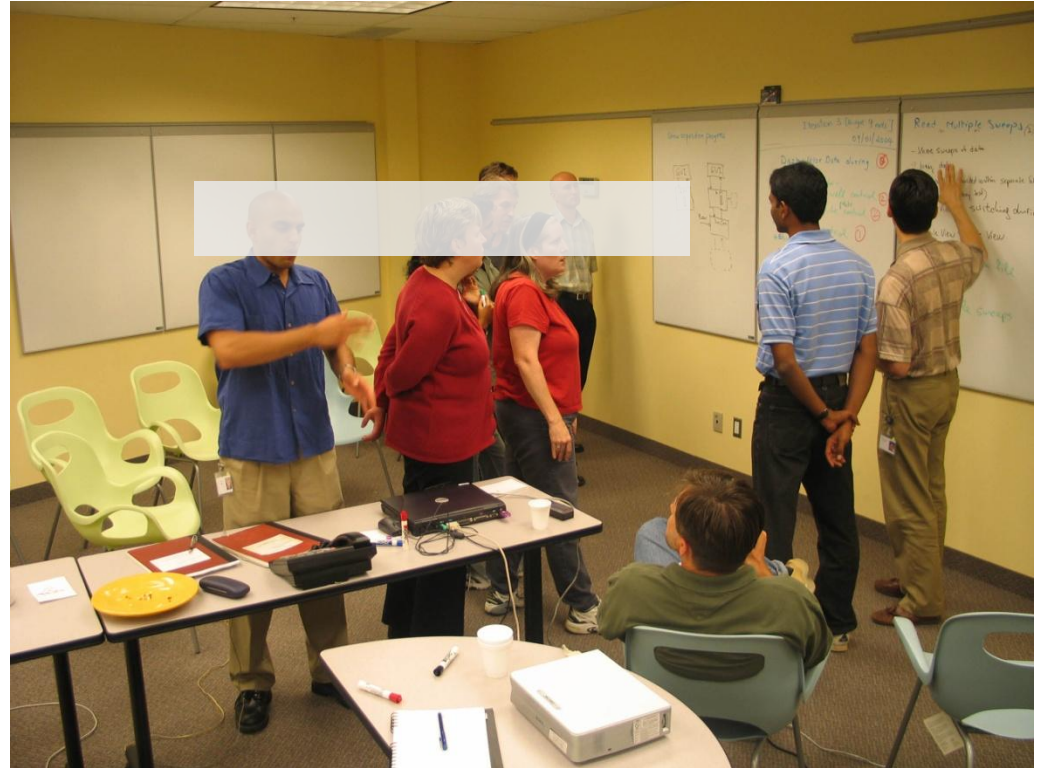
# Product Backlog

- Master list of all “features”
- High priority features are split into “stories” achievable within an iteration.
- Each “story” is prioritized and scoped.



# Sprint Planning Meeting

- Highest priority stories are reviewed.
- Team selects stories
- Team breaks stories down into tasks & re-estimates.
- Team commits to next iteration's deliverables.



# Sprint Backlog



# Daily Scrum

- Each team member describes:
  - What they did
  - What they plan to do
  - Obstacles
- ScrumMaster tracks and resolves obstacles
- 10 – 15 minutes



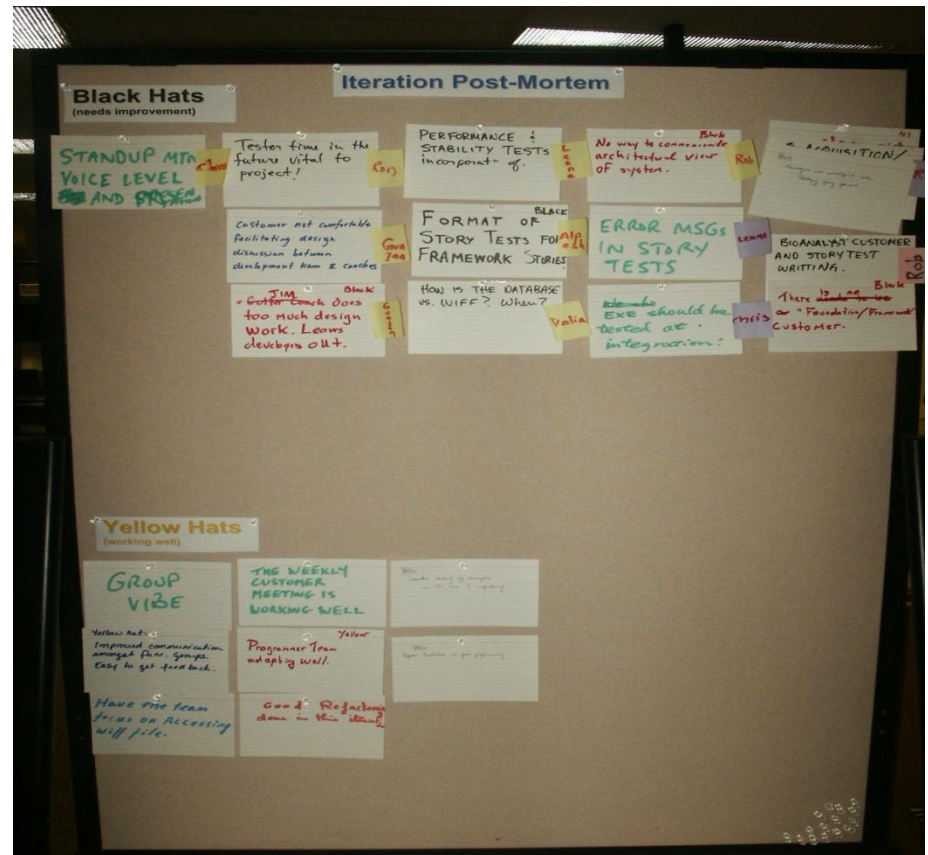
# Sprint Demo

- Team demonstrates working software to product owner
- Product owner accepts or rejects completed work
- Result should be potentially shippable

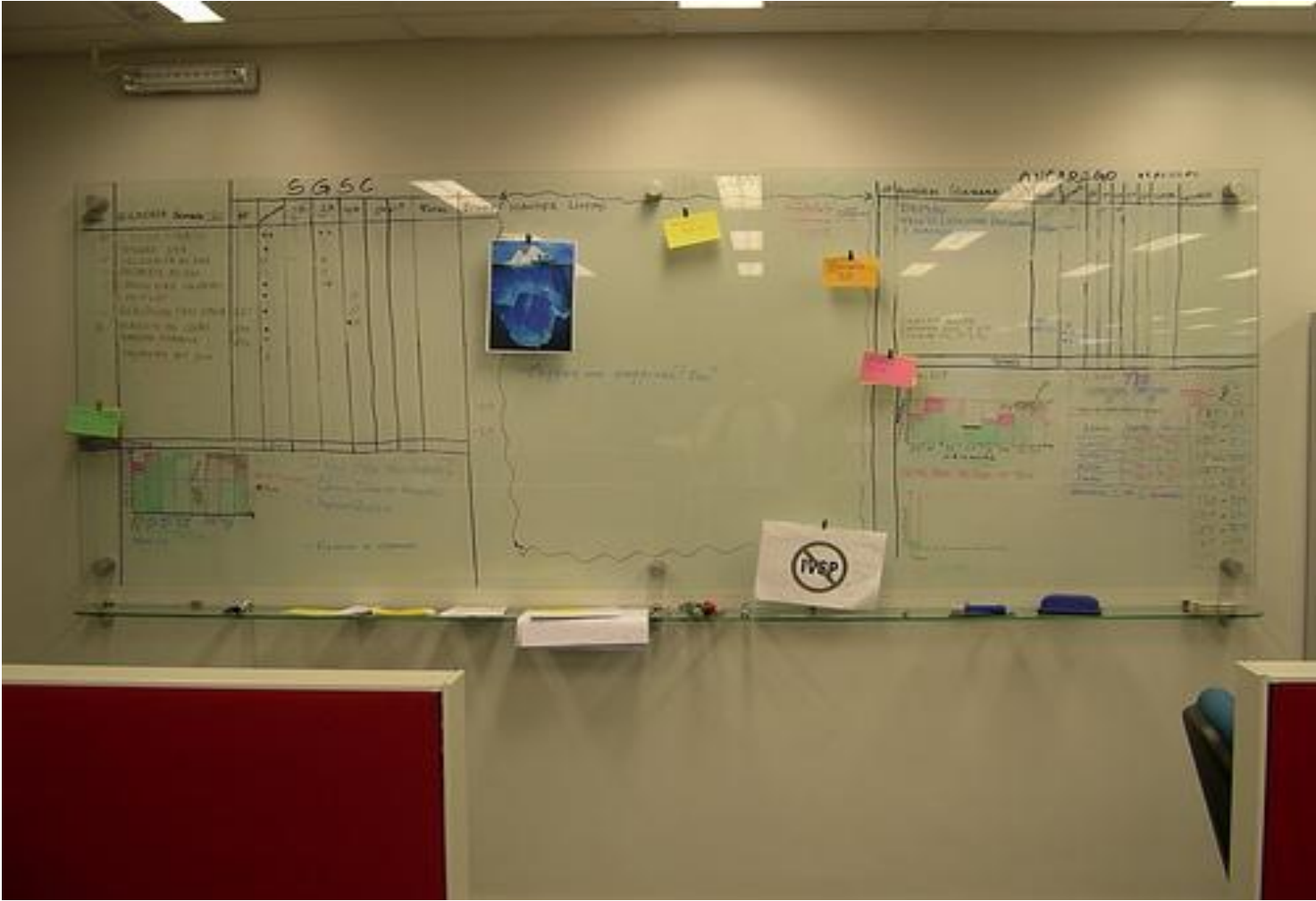


# Sprint Retrospective

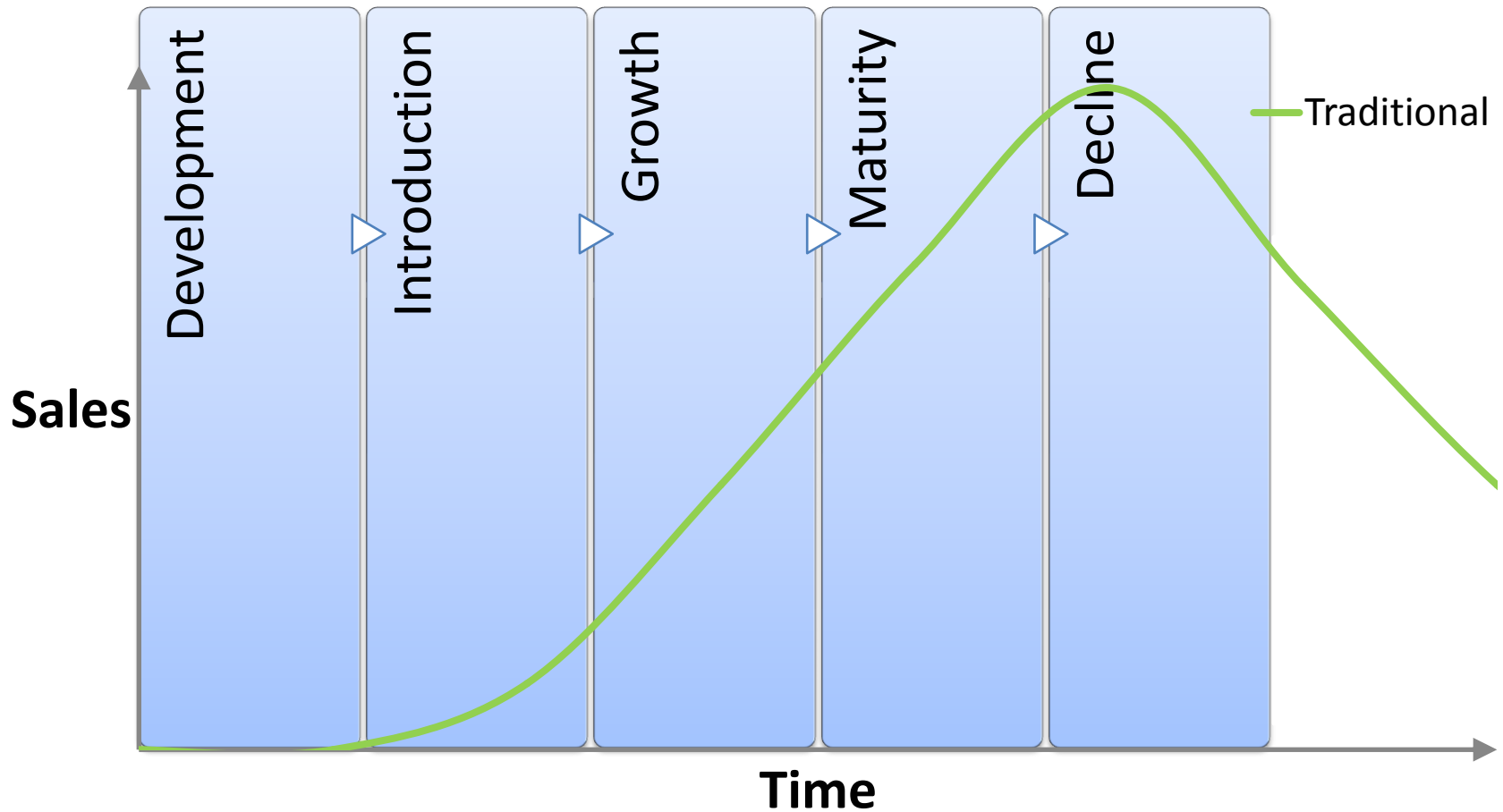
- Team meets to review:
  - What is working?
  - What is not working?
- Team adds tasks for immediate actions for working better



# Informative Collaborative Workspace



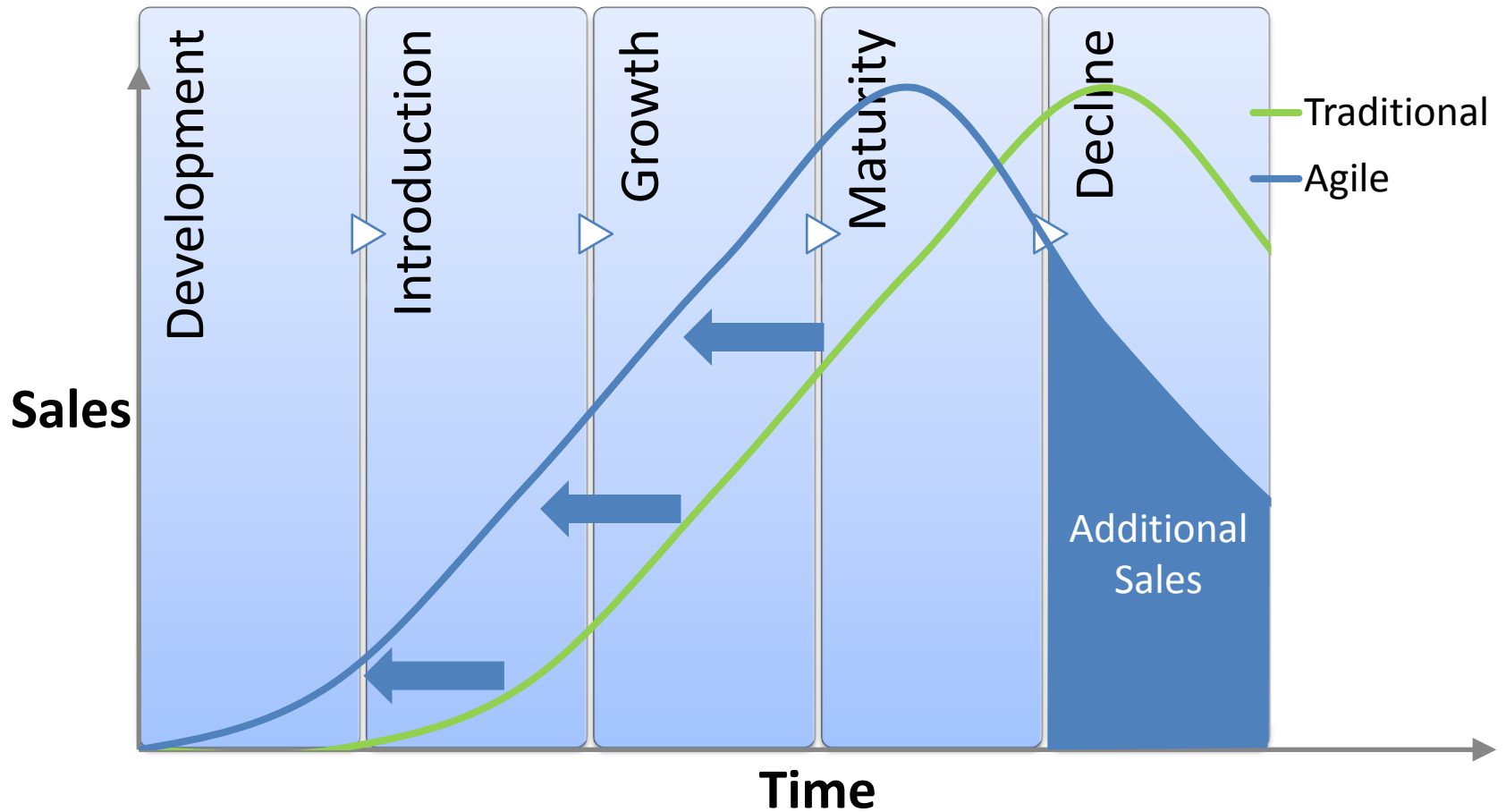
# Product Life Cycle



<http://tynerblain.com/blog/2007/02/27/agile-development-roi-1/>



# Product Life Cycle



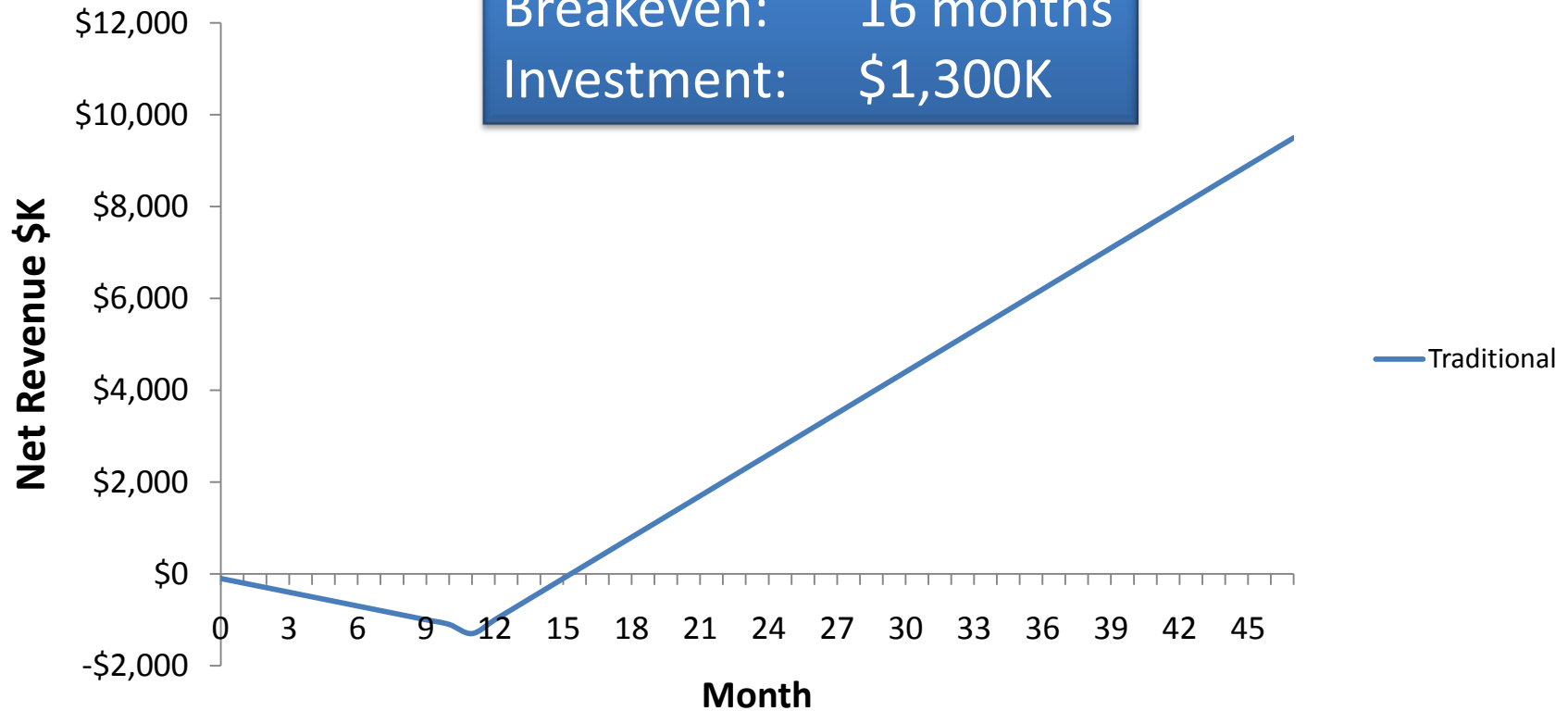
<http://tynerblain.com/blog/2007/02/27/agile-development-roi-1/>

# An Example Project

- Expected revenue of \$300K/month
- Development costs \$100K/month
- Release costs \$100K/release
- Estimated at 12 months effort
- Looking at 4 year project window

# Traditional Approach

IRR: 11.32%  
Breakeven: 16 months  
Investment: \$1,300K



Adapted from "[Incremental Releases Users & Stakeholders will Love](#)" OOPSLA 2007

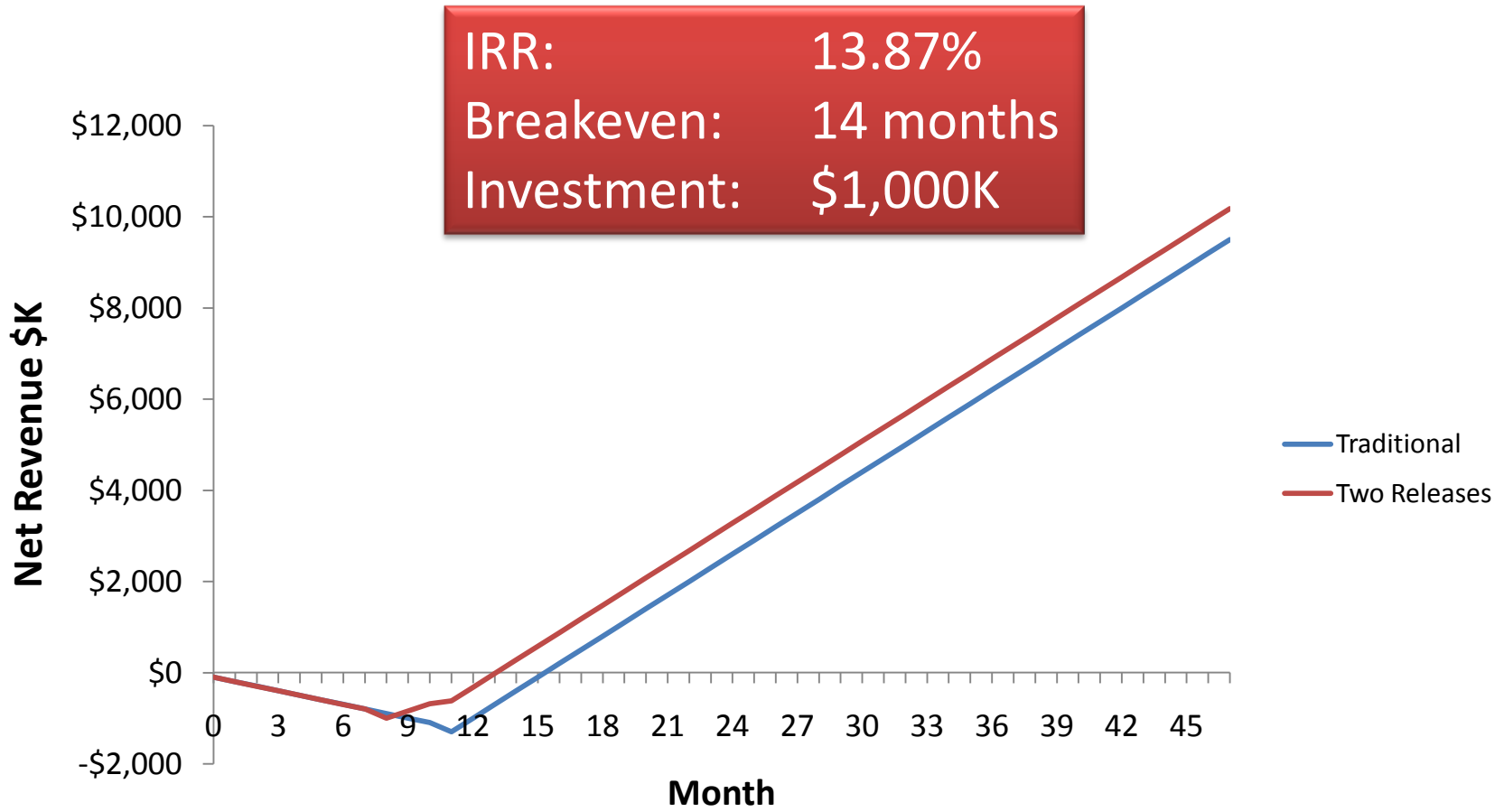
# What if We Release More Often?

- Not all features generate same value:

Feature Set	Value
Highest Value 25%	\$120K
Next Highest Value 25%	\$80K
Next Highest Value 25%	\$60K
Lowest Value 25%	\$40K

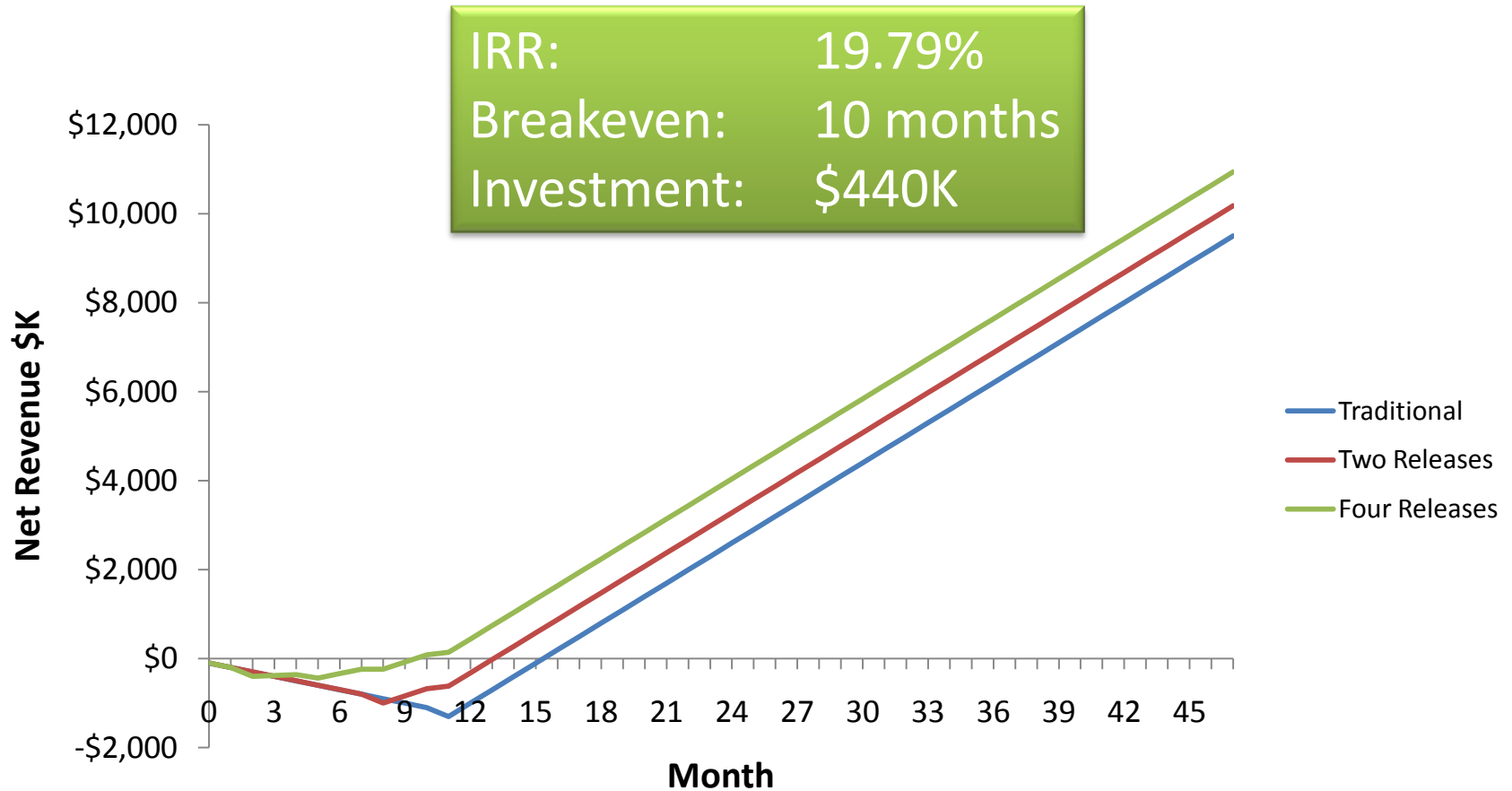
Adapted from "[Incremental Releases Users & Stakeholders will Love](#)" OOPSLA 2007

# Agile – Two Releases



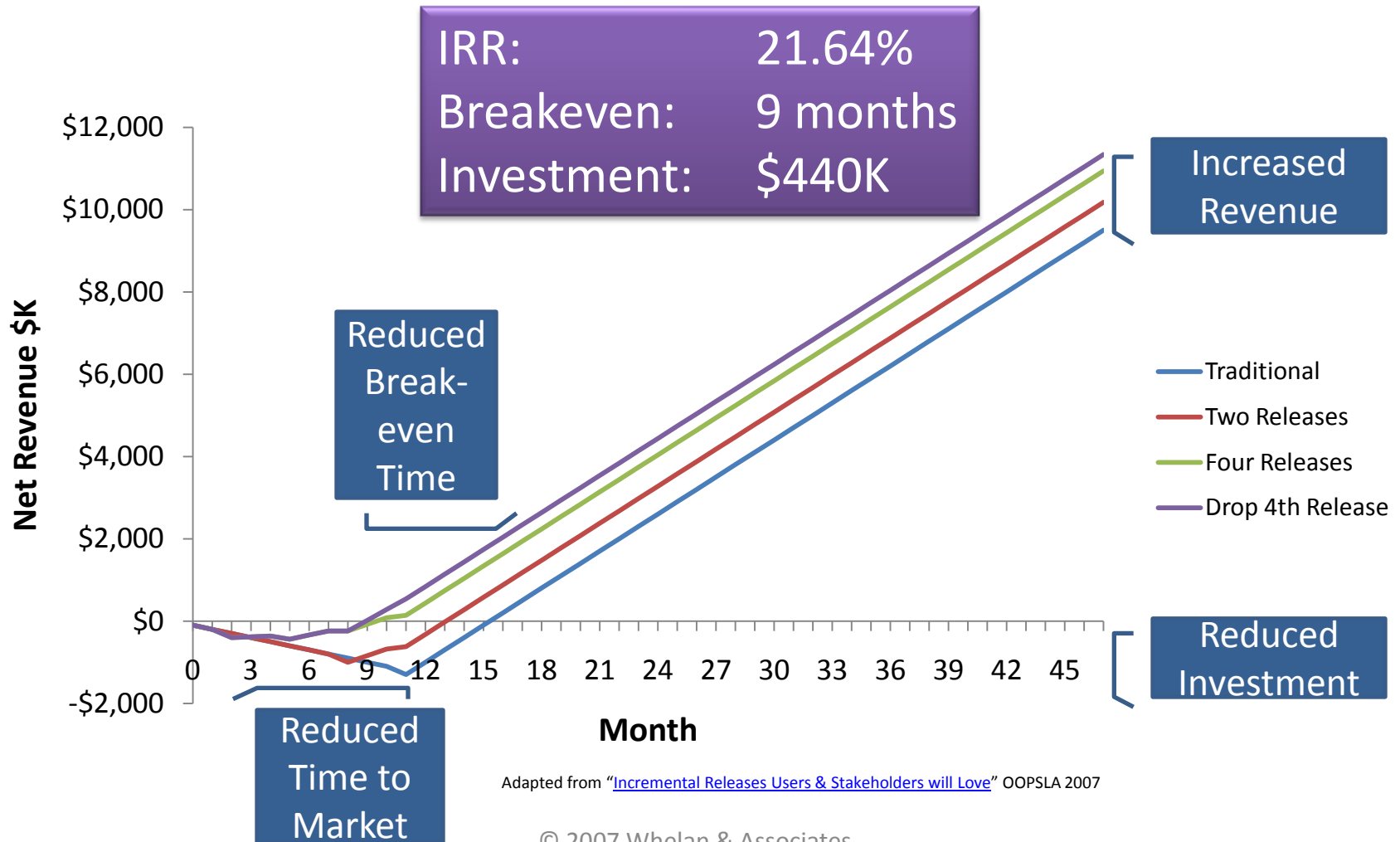
Adapted from "[Incremental Releases Users & Stakeholders will Love](#)" OOPSLA 2007

# Agile - 4 Releases



Adapted from "[Incremental Releases Users & Stakeholders will Love](#)" OOPSLA 2007

# Agile – Drop Last Release



# ROI & Time-To-Market Comparison

	Traditional	Agile	Improvement
Time to Market*	12 months	3 months	75%
Break Even	16 months	9 months	44%
Investment	\$1,300K	\$440K	66%
Net Revenue	\$9,500K	\$11,430	19%
IRR	11.32%	21.64%	91%

Adapted from "[Incremental Releases Users & Stakeholders will Love](#)" OOPSLA 2007



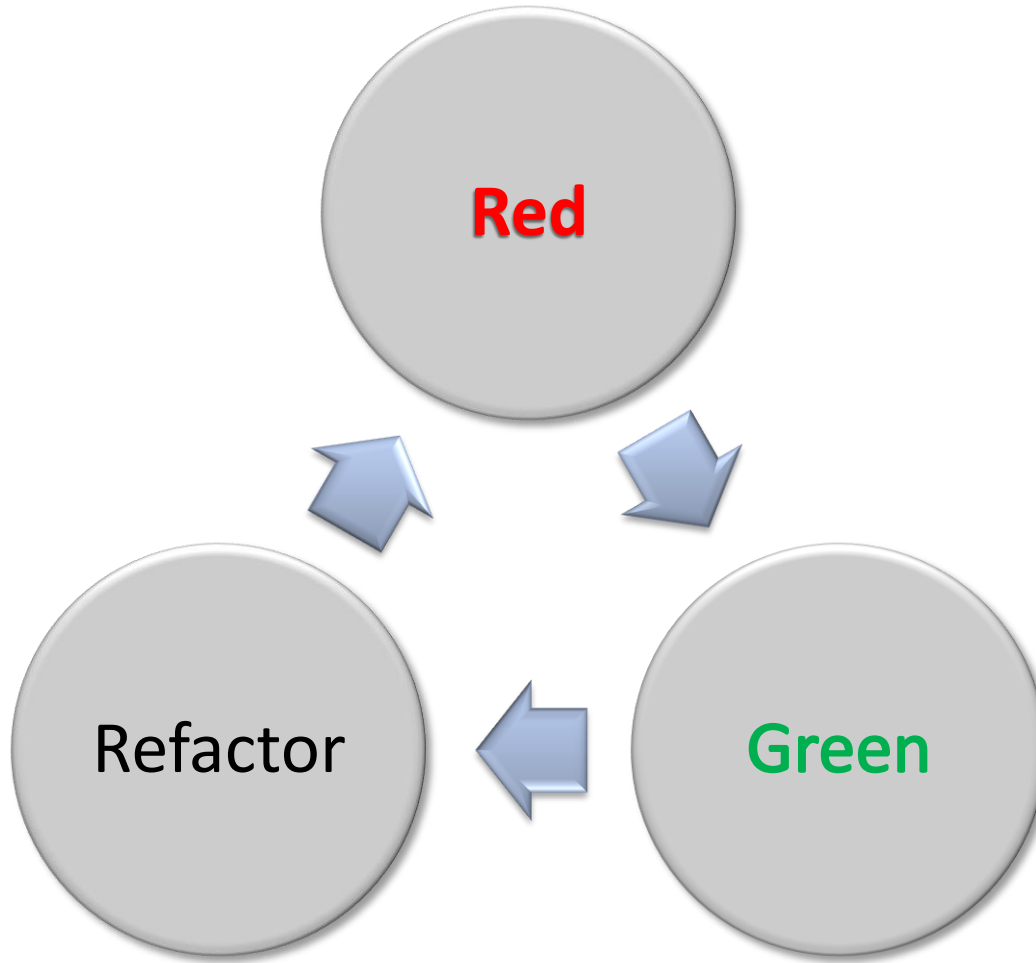
# Agile ROI – Other Factors

- Cost reduction delivering only what is really needed
- Faster to market could drive additional sales
- Customer feedback should improve product fit driving additional sales
- If there is a fixed market window agile development can help hit the window

# Agile Quality – Acceptance Tests

Agile Practice	Benefits
Requirements specified using acceptance tests	<ul style="list-style-type: none"><li>• Eliminates mismatch between requirements and test cases</li></ul>
Acceptance tests written by product owner with team support	<ul style="list-style-type: none"><li>• Common understanding of functionality.</li><li>• Helps flush out requirement inconsistencies</li><li>• Encourages alternate approaches early</li></ul>
“Customer driven” acceptance tests	<ul style="list-style-type: none"><li>• Connects features to customer value</li><li>• Using customer/business terms</li></ul>
Automated acceptance tests	<ul style="list-style-type: none"><li>• Previous functionality is retained – keeps product notching forward</li><li>• Focuses development on business value</li><li>• Reduces scope creep - the best way to write defect-free code is to not write it at all</li></ul>

# Test Driven Development



# Agile Quality – A Team Deliverable

Agile Practice	Benefits
Whole Team	<ul style="list-style-type: none"><li>• Quality is not just a QA responsibility</li><li>• QA role shifts to quality infusion throughout project life cycle</li><li>• QA is more than just testing</li></ul>
Continuous Integration	<ul style="list-style-type: none"><li>• Developers cannot check in code with failing tests</li></ul>
Continuous Testing	<ul style="list-style-type: none"><li>• Avoids long delays with “big-bang” testing after the “final build”</li><li>• Bugs found closer to when they are introduced making them easier to fix</li></ul>

# Software Project Risks

Inherent schedule flaws

Scope creep

Employee turnover

Specification breakdown

Poor productivity

Source: *Agile Project Management* – Jim Highsmith, 2004

© 2007 Whelan & Associates

# Agility & Schedule Flaws

## Schedule Flaws

- Scope is grossly misestimated or
- Impossible date is mandated

## Agility reduces risk by:

- Whole team planning & estimating
- Early feedback on delivery velocity
- Balancing features against schedule
- Keeping the product defect-free

Source: *Agile Project Management* – Jim Highsmith, 2004

# Scope Creep

## Scope Creep

- Customers change requirements indiscriminately
- Developers change requirements indiscriminately

## Agility reduces risk by:

- Communications with customers
- Whole team planning & tracking
- Simple design
- Daily scrum meetings

Source: *Agile Project Management* – Jim Highsmith, 2004

# Employee Turnover

Agility  
reduces  
risk by:

- Whole team ownership
- Specializing generalists
- Pair Programming

Source: *Agile Project Management* – Jim Highsmith, 2004

© 2007 Whelan & Associates



# Specification Breakdown

Customers fail  
to agree on  
specifications

- Team may get stalled
- Development may need to be backed out
- Code duplication

Agility  
reduces risk  
by:

- Product owner as single decision point
- Product owner can halt project

Source: *Agile Project Management* – Jim Highsmith, 2004

# Poor Productivity

Results  
from 3 main  
sources

- Wrong people on the bus
- A team that does not work well together
- Poor morale

Agility  
reduces risk  
by:

- Hard to hide on an agile bus
- Self-organizing teams
- Feedback loops
- Agile teams are more motivated

Source: *Agile Project Management* – Jim Highsmith, 2004

# Agility & Risk

*“The best bang-for-the-buck risk mitigation strategy that we know is incremental delivery.”*

*Waltzing With Bears*

Tom DeMarco & Tim Lister

2003

# Agility Introduces New Risks

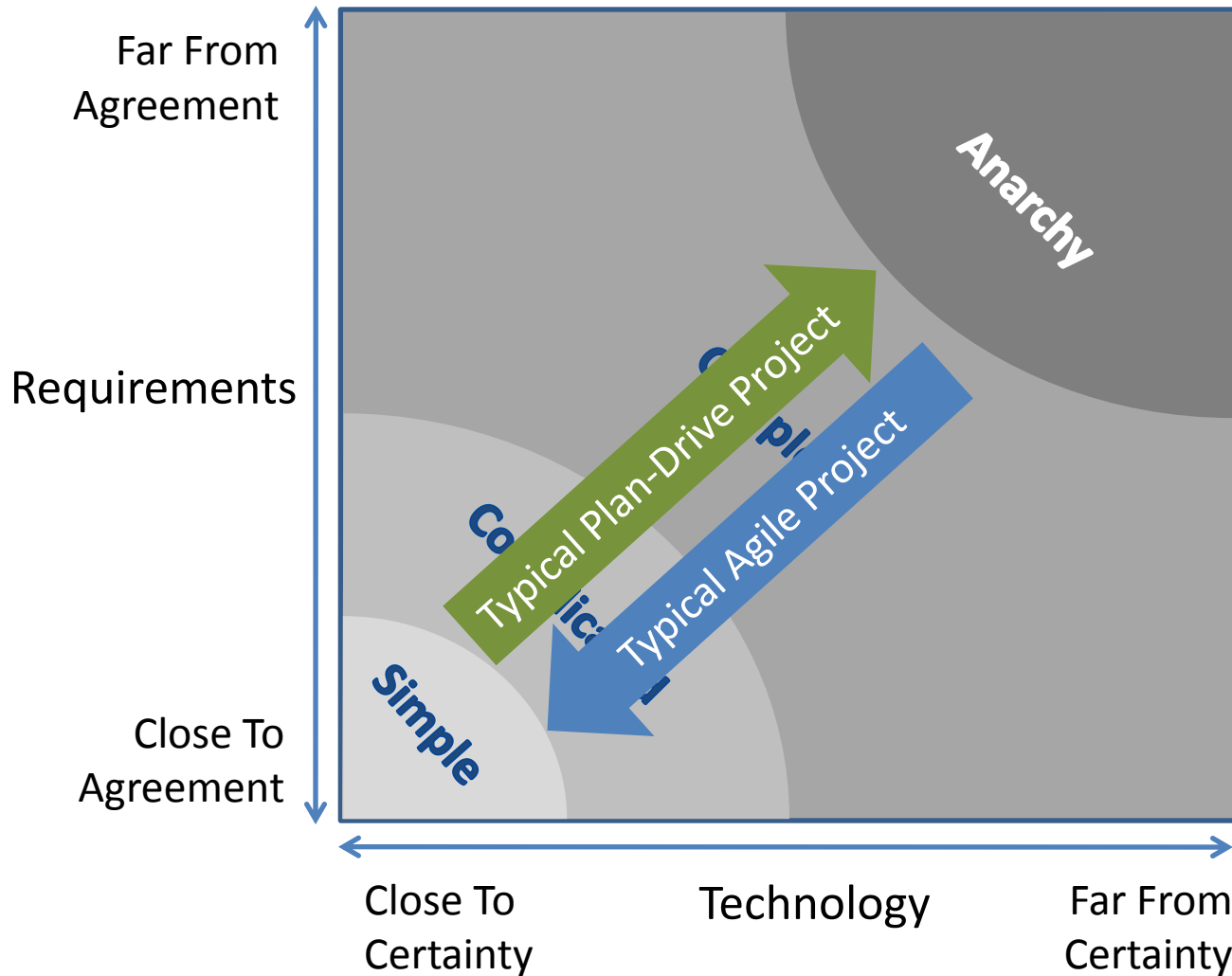
Too little planning or specification could lead to major rework

- Use “simple design” principles to lower rework costs
- Use test-driven design to reduce risk & rework costs

Overhead of collaboration and feedback could impact project

- Yes, but you save by not doing large up-front work
- Focus on continual process improvement

# Agile Applicability



# Agile Impediments

Command and control  
company culture

- Customer Ownership
- Management Commitment
- Developer Cooperation

Lack of trust and  
honesty within company

- You can't hide on an agile project
- Agility will reveal company dysfunction

Simple low-risk projects  
or complete anarchy

- Overkill for simple projects
- Nothing can save you from complete anarchy

Regulatory Governance:

- ISO9000
- FDA
- SOX

# Agile Success Factors

Servant leadership & team empowerment

- Trust , honesty & transparency
- Ability to face and “handle the truth”
- Bottom-up and top-down commitment to agility

Motivated and skilled teams

- Training, coaching. mentoring
- Commitment to teamwork
- Egoless engagement

Complex projects with business value

- Agility manages risk and complexity
- Consider ROI and P&L project tracking

Willingness to “embrace change”

- Courage
- Creativity
- Openness to new ways of working

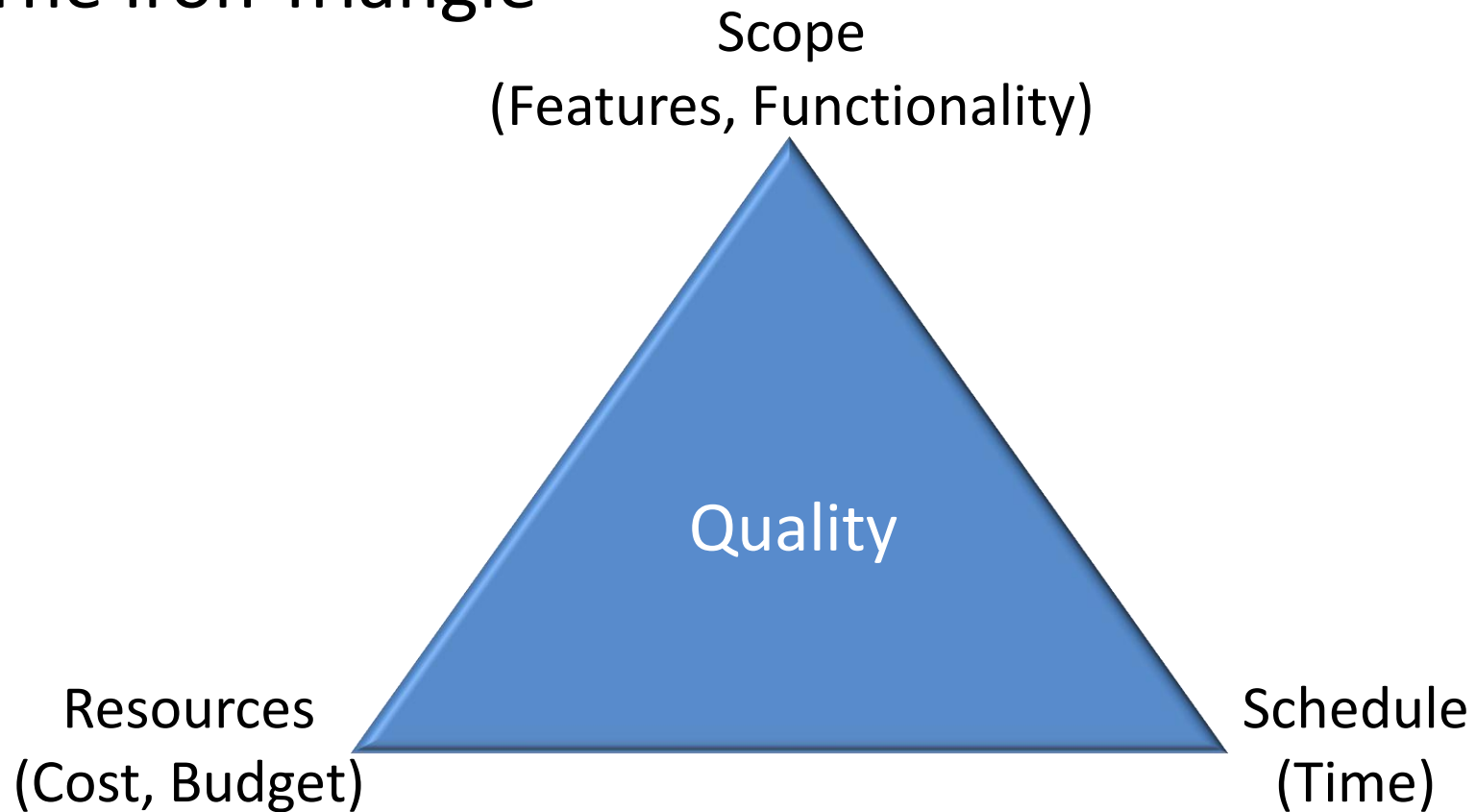
# Agile Myths

Myth	Reality
No discipline (just hacking)	<b>False</b> Agile practices require more discipline. Delivering working S/W every week takes considerable discipline.
No documentation	<b>False</b> Stakeholders often find more effective investments. An agile team will make documentation costs explicit. Focus on stable concepts; avoid volatile concepts like requirements. More documentation likely <u>increases</u> the chance of project failure.
No planning	<b>False</b> Continual just-in-time planning. Agility focuses on self-organizing teams that plan just enough to focus on business value and to get the job done.
No scalability	<b>False</b> Most literature focuses on small teams. Agile practices can and do scale both in locality and scope: e.g. IBM Eclipse
Another Fad	<b>False</b> Has crossed the chasm and is becoming mainstream. 45% of companies have adopted one or more agile methods. 65% have adopted one or more agile practices.
No predictability	<b>False</b> Nature of predictability shifts Agile teams will predictably deliver high quality S/W and the highest possible business value.



# Predictability

## The Iron Triangle



Source: <http://www.ambysoft.com/essays/brokenTriangle.html>

# Predictability

- There is a false sense of predictability when companies do not honor the iron triangle
- Agility provides increased predictability through the project life cycle where traditional projects have decreasing predictability
- Agility reliably predicts that the maximum business value will be produced within a given time-frame

Source: <http://www.ambysoft.com/essays/brokenTriangle.html>

# Fixed Price Work

- Suffers from the delusion that stakeholders know what want and that requirements can be effectively captured, recorded, understood and fixed at the start of a project
- Requires risk mitigation through padding
  - Increases overall costs to stakeholders
  - Projects always expand to fill the padding
- Encourages costly change-control procedures
- Encourages team to build to spec rather than what stakeholders actually need

Source: <http://www.ambysoft.com/essays/brokenTriangle.html>

# Quoting Fixed Price Contracts

- Estimate as traditional projects
- Use a mix of T&M with fixed price
- Vary scope and/or schedule
- Add terms for additions and then allow customer to drop “fixed” requirements and grow the additional budget accordingly

Source: <http://www.ambysoft.com/essays/brokenTriangle.html>

# Managing Fixed Price Contracts

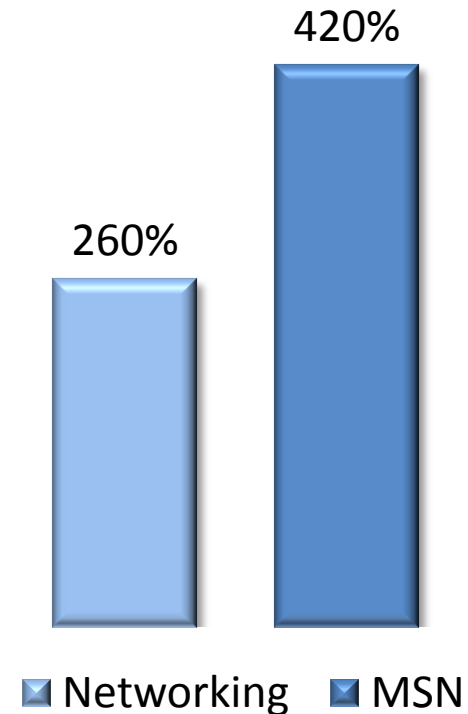
- Use estimate ranges
  - accuracy shifts with planning horizon
- Split large requirements into smaller pieces
- Update estimates continuously through project
- People who do the work should own estimates
- Estimate with whole team

Source: <http://www.ambysoft.com/essays/brokenTriangle.html>

# Agility at Microsoft

- Visual Studio Team Foundation Server supports agile
- Process Templates available for agile
- April 2007 released eScrum a web-based Scrum tool
- Internal teams have adopted and continue to adopt agile practices

## Quality Improvement Using Test Driven Development



# Wrap Up

- Agility can deliver better ROI, time-to-market, quality and reduce risk
- Agility is hard work - requires discipline
- Agility can help almost all software companies – probably yours!
- Agility is not binary
  - a little is better than none
- Greatest impediment is company culture
- Greatest success factor is company culture

# Next Steps

- Attend agile events planned by Communitech
  - December 17<sup>th</sup>, Scott Ambler
  - Other events in 2008 being planned
- Encourage staff to educate themselves on agility
  - see reading list and web sites
- Identify pilot project in your organization
  - Should be risky and of high value
  - Have team self-select for project
- Train team on agile practices
- Just do it!



# Agile Reading List

- *Agile Project Management*  
Jim Highsmith; 2004
- *Lean Software Development*  
Mary & Tom Poppendieck; 2003
- *Extreme Programming Explained 2<sup>nd</sup> Edition*  
Kent Beck, Cynthia Andres; 2004
- *Agile Software Development with Scrum*  
Ken Schwaber, Mike Beedle; 2002

# Agile Web Sites

- [www.agilemanifesto.com](http://www.agilemanifesto.com)
- [www.agilealliance.org](http://www.agilealliance.org)
- [www.ambyssoft.com](http://www.ambyssoft.com)
- [www.scrumalliance.org](http://www.scrumalliance.org)
- [www.xprogramming.com](http://www.xprogramming.com)
- [www.agileadvice.com](http://www.agileadvice.com)

# Panel Discussion

Ray Simonson  
*Software Innovation*

Jeff Fedor  
*aideRSS*

Lance Mohring  
*emedia interactive*

Jeff Berardine  
*Innosphere*

Ed Bianchin  
*Virtek*

Declan Whelan  
*Whelan & Associates*