



# **Quality Management in the World of Agile Product Development**



***Russell Pannone  
February 10, 2009  
webeagile@aol.com***



**Creative Agile Thinking™**

# About Me

- 27 years of System/Software Product Development Experience
  - Developer
  - Data Modeler
  - Team Lead
  - Project Manager
  - Certified Scrum Master/Certified Scrum Product Owner
  - Bachelor of Science/Computer Science
  - Master of Business Administration/MIS



## My Motto:

"Being agile combines leading change, practicing the 4 agile values & 12 principles, using scrum, applying iterative/incremental systems/software product development and takes wisdom, common sense, passion, courage, a desire to be better and openness, especially to change"

# What We Will Cover.....

## ❖ Overview of What it Means to “Be Agile”

1. Leading Change
2. The 4 Agile Values & 12 Principles
3. Iterative and Incremental System/Software Product Development
4. SCRUM
5. People
6. Practices

## ❖ Where Quality Management Fits

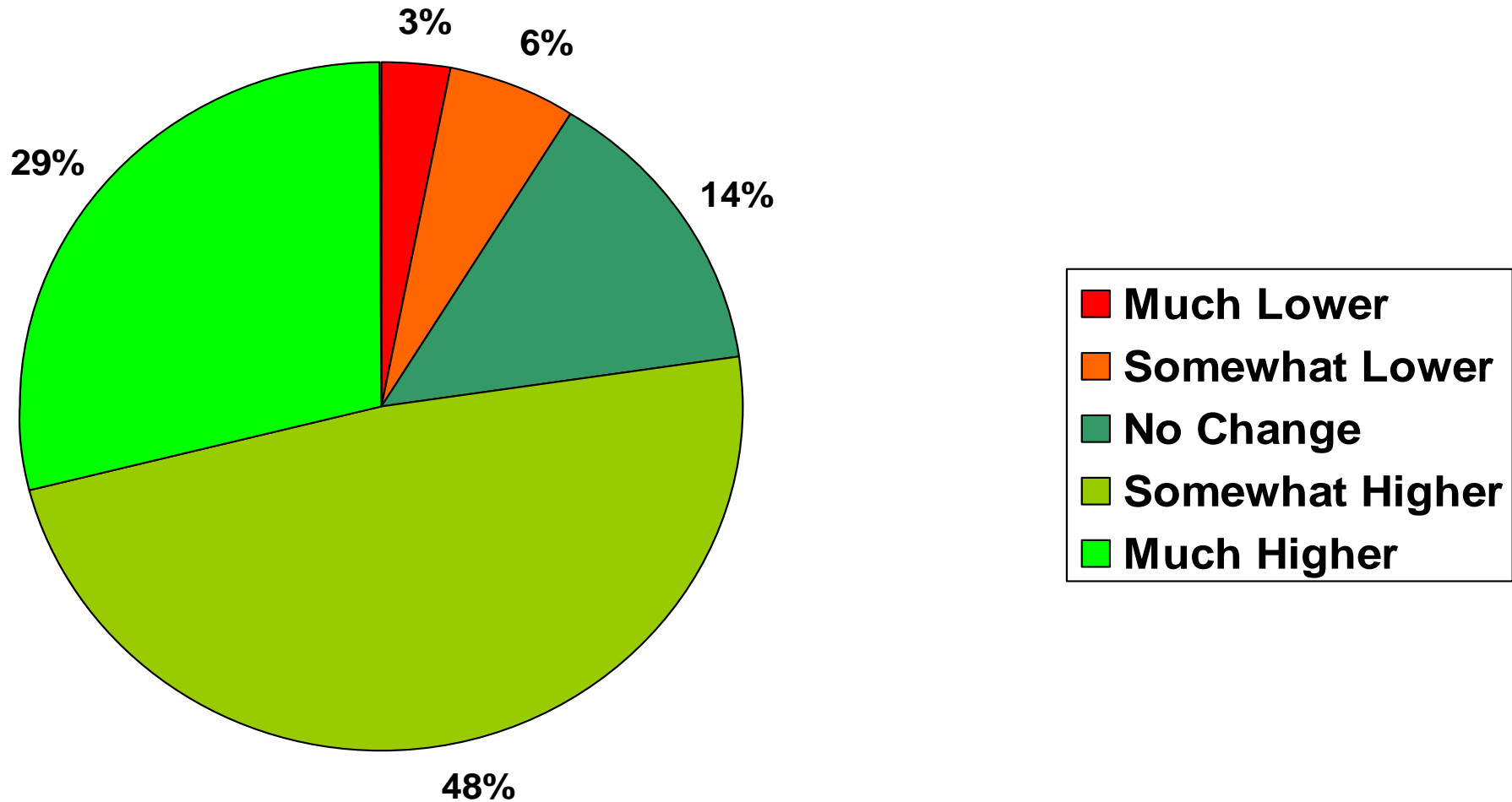
- Preventing Defects of Intent and Defects of Implementation by:
  - ✓ Ensuring We are Doing the Right Things – “*Fit for Purpose*”
  - ✓ Ensuring We are Doing Things Right – “*Fit for Use*”





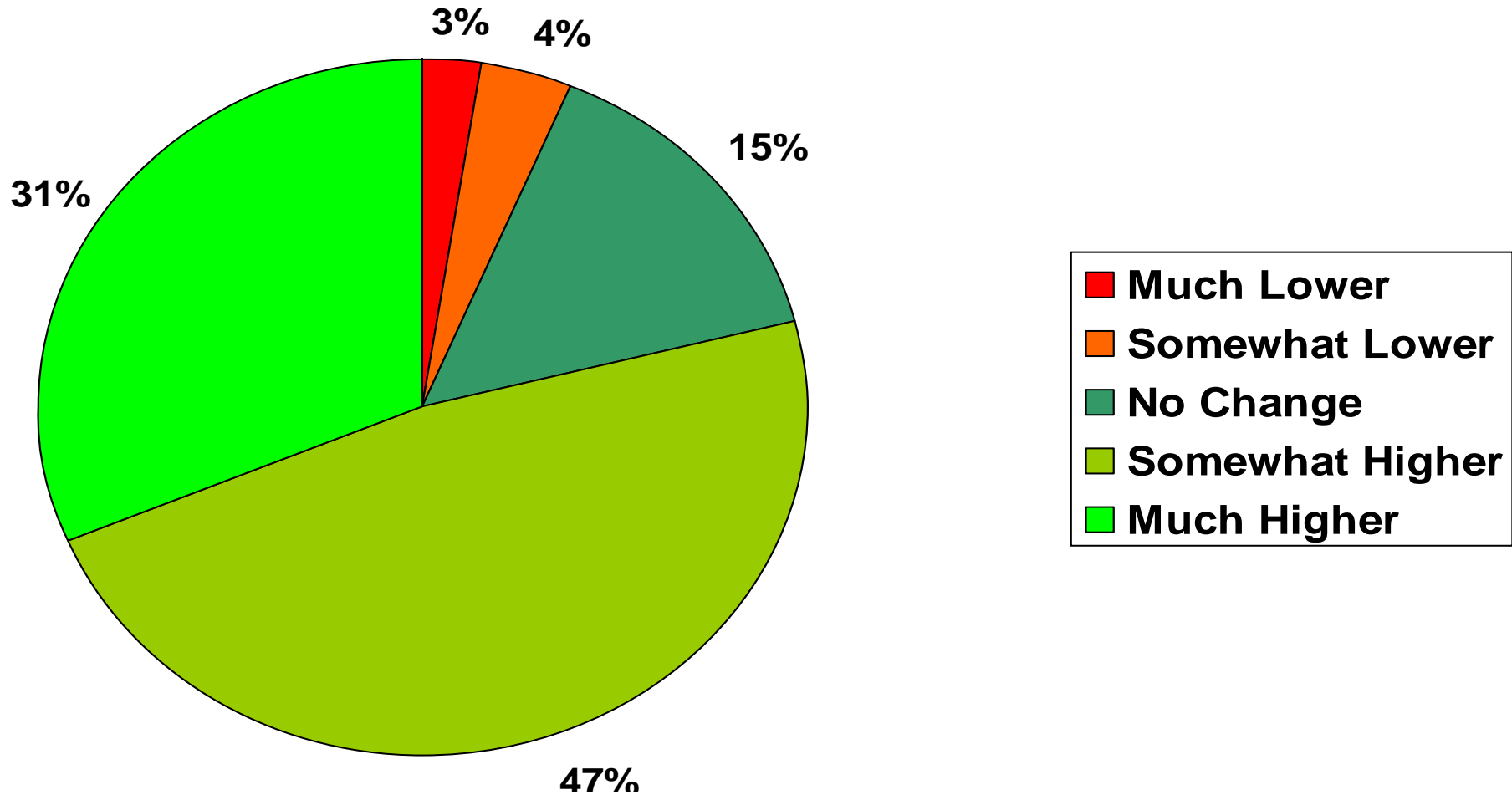


## How Has Being Agile Affected the Quality of Deployed Systems?



Results from Scott Ambler's February 2008 Agile Adoption Survey posted at <http://www.ambysoft.com/surveys/agileFebruary2008.html>

## How Has Being Agile Affected the Business Stakeholder Satisfaction?



Results from Scott Ambler's February 2008 Agile Adoption Survey posted at <http://www.ambysoft.com/surveys/agileFebruary2008.html>



# Barriers to Becoming Agile

Ability to change organizational culture - 45%

General resistance to change - 44%

Personnel with the necessary Agile experience - 42%

Management Support - 32%

Project Complexity or Size - 23%

Customer Collaboration - 22%

Confidence in ability to scale Agile methods - 17%

Perceived time to transition - 14%

Budget Constraints - 10%



Source: VesionOne 2008 State of Agile Development Survey

**2,300 Respondents from 80 Countries**

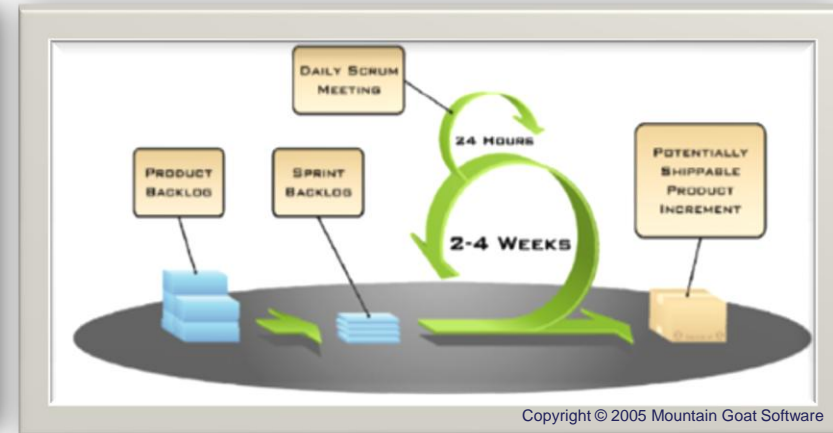
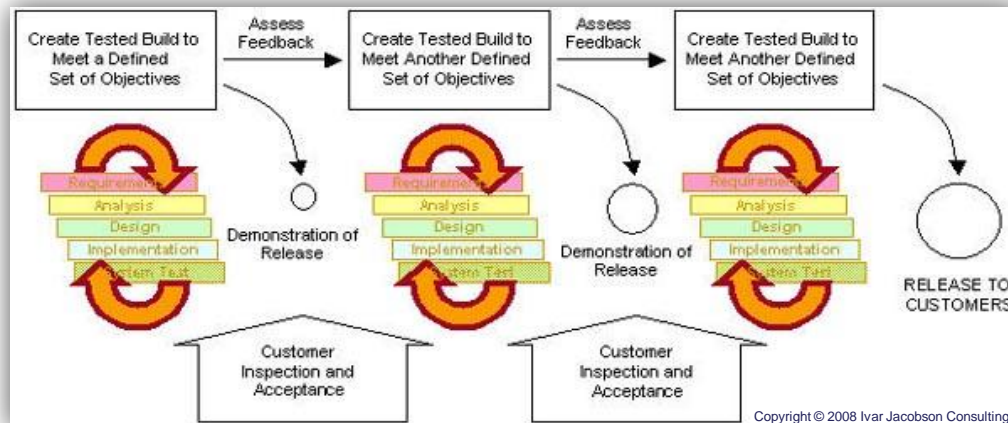
# What Does an Agile Team Look Like

- ✓ **Work as one**
- ✓ **Highly collaborative & self-directed**
- ✓ **Work in short iterations**
- ✓ **Deliver something each iteration**
- ✓ **Focus on business priorities**
- ✓ **Inspect and adapt**



## Roles

- Product Owner
- Scrum Master
- Team







# Navigating the Sea of Agility

**Agile Manifesto**

**Sprint Backlog**

**Collaboration**

**User Stories**

**Test Driven Development**

**Velocity**

**Continuous Integration**

**Co-Located**

**Burndown Chart**

**Iterative & Incremental Development**


**Acceptance Test Driven Development**

**Product Backlog**

**Sprint Review**

**Scrum**

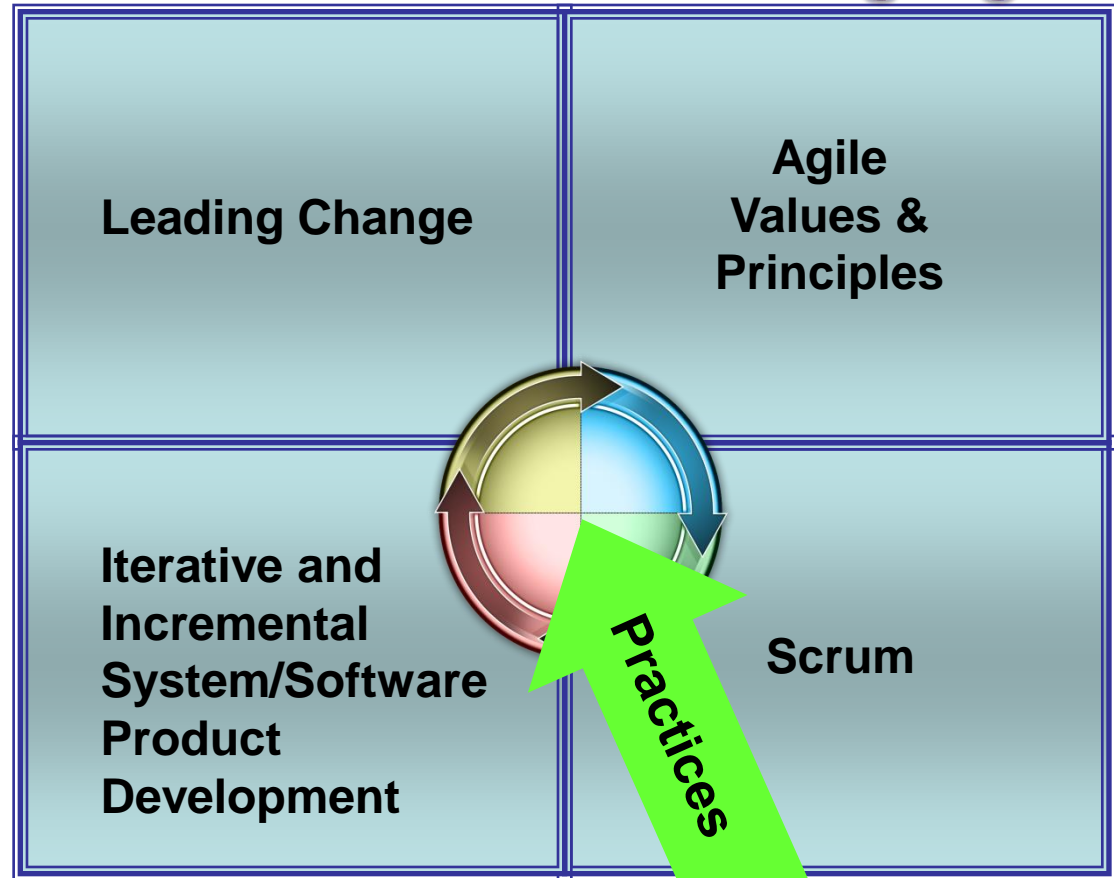
**Retrospective**

The **Key** is to  
  
gain a common  
understanding  
of what it means  
to **YOU**  
the **TEAM** & the  
**Enterprise**  
to



**“Be Agile”**

## The 6 Elements of Being Agile



**People**

## Transformational Challenge

- ✓ Re-Engineering
- ✓ Re-Strategizing
- ✓ Cultural Renewal

**Leading Change**

Agile  
Values & Principles

Iterative and Incremental  
System/Software Product  
Development

Scrum

Practices



The **1st** Element of Being Agile



## Barriers to Becoming Agile

Ability to change organizational culture - 45%

General resistance to change - 44%

Personnel with the necessary Agile experience - 42%

Management Support - 32%

Project Complexity or Size - 23%

Customer Collaboration - 22%

Confidence in ability to scale Agile methods - 17%

Perceived time to transition - 14%

Budget Constraints - 10%

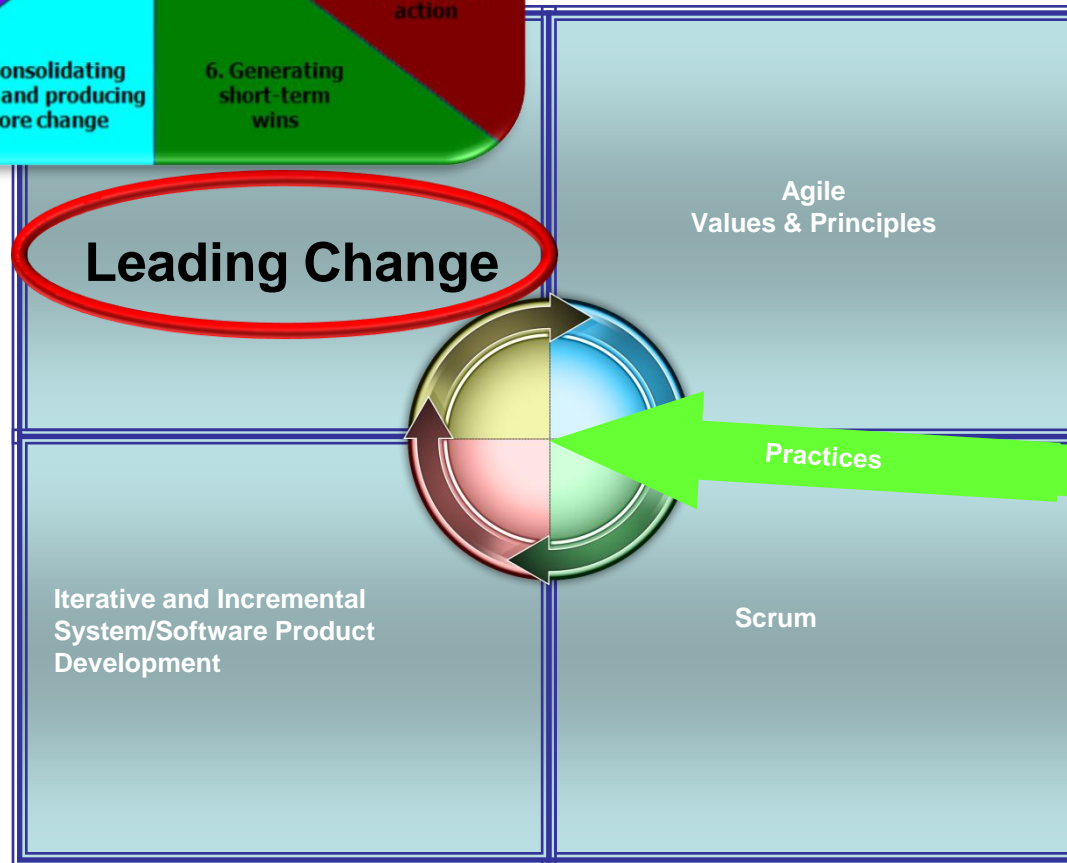


**From 2,300 respondents in 80 different countries**

Source: ~~VersionOne~~ 2008 State of Agile Development Survey



# Your Change/Action Plan



## The **1st** Element of Being Agile

\* Taken from [Leading Change](#) by John Kotter - 1996



# Your Change-Action Plan

(continued on next slide)

1. Establishing a sense of urgency
  - Identifying and discussing Strengths, Weaknesses, Opportunities, Threats
2. Creating the guiding coalition
  - Putting together a group with enough power to lead the change
  - Getting the group to work together as a team
3. Developing a vision and strategy
  - Creating a vision to help direct the change effort
  - Developing strategies for achieving that vision
4. Communicating the change vision
  - Using every vehicle possible to constantly communicate the new vision and strategies
  - Having the guiding coalition role model the behavior expected of employees



# Your Change-Action Plan

(continued from previous slide)

## 5. Empowering broad-based action

- Getting rid of obstacles
- Changing policies, procedures and structures that undermine the change vision
- Encourage risk taking and nontraditional ideas, activities, and actions

## 6. Generating short-term wins

- Planning for visible improvements in performance, or “wins”
- Creating those wins
- Visibly recognizing and rewarding people who make wins possible



## 7. Consolidating gains and producing more change

- Using increased credibility to change all policies, procedures and structures that don't fit the transformation vision
- Hiring, promoting, and developing people who can implement the change vision
- Reinvigorating the cultural renewal with new projects, themes and change agents

## 8. Anchoring new approaches in the culture

- Creating better performance through customer and productivity oriented behavior, more and better leadership, and more effective management
- Articulating the connections between new behaviors and original success
- Developing means to ensure leadership development and succession

## 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:

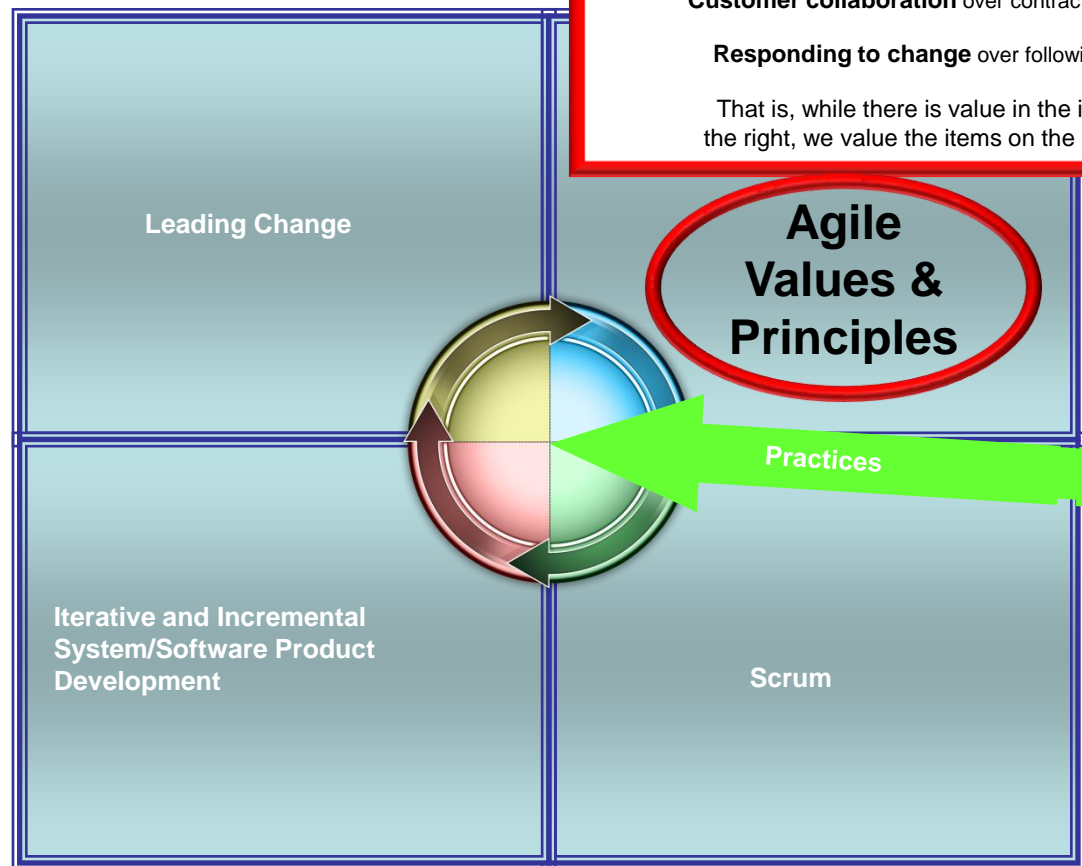
**Individuals and interactions** over processes and tools

**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.



## The **2nd** Element of Being Agile

# 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

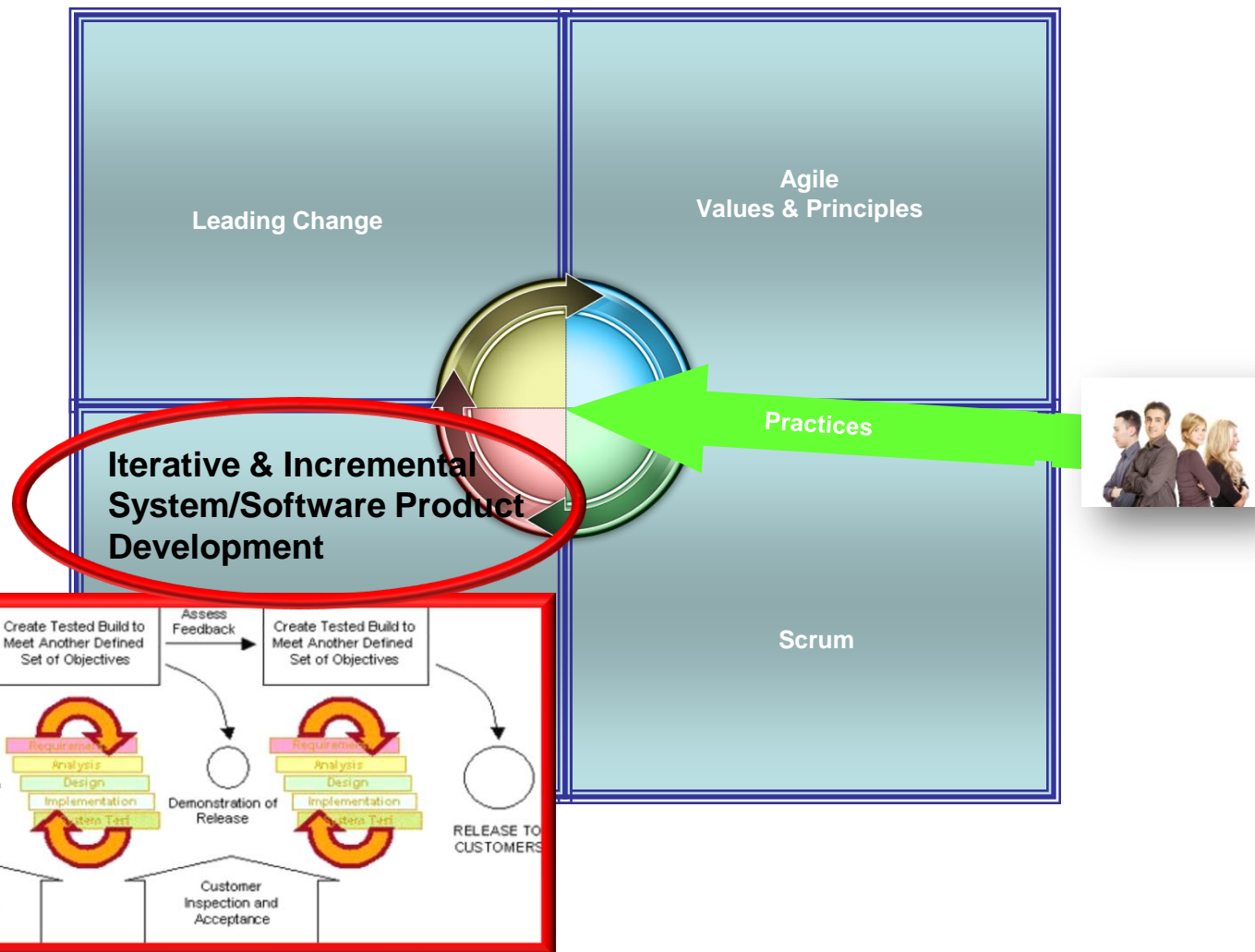
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

- 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.
- Working software is the primary measure of progress.

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

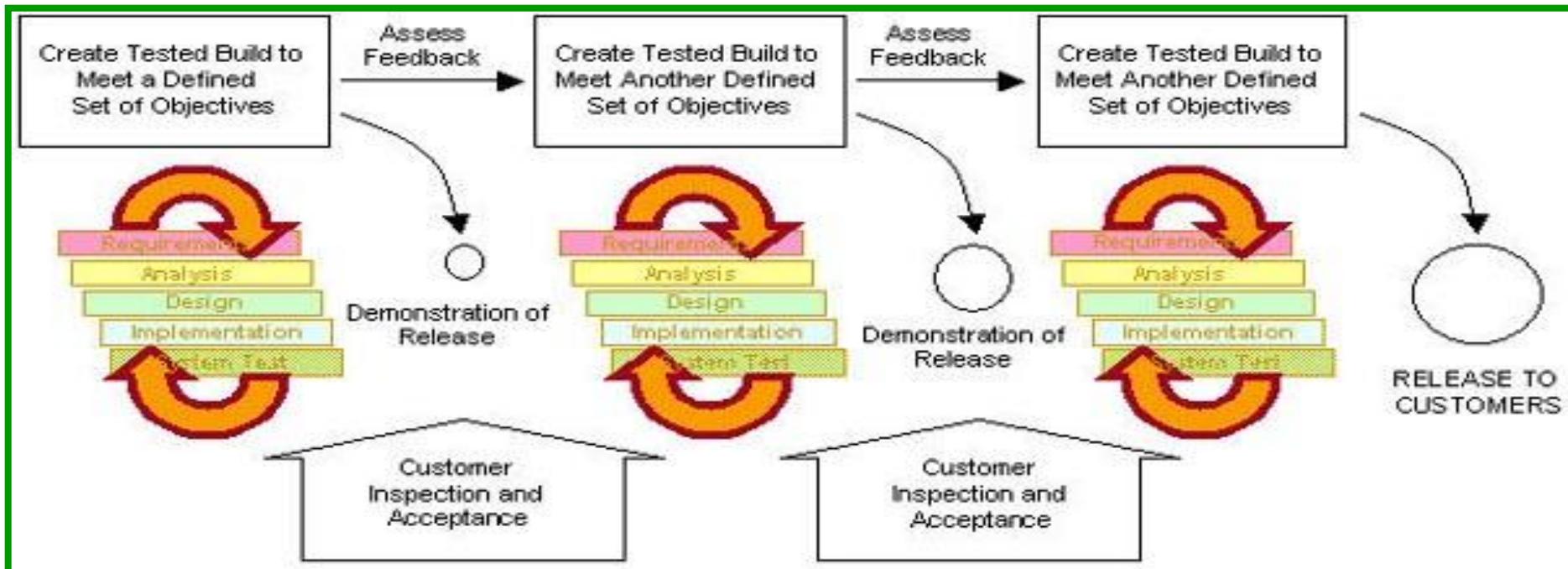
# The **3rd** Element of Being Agile





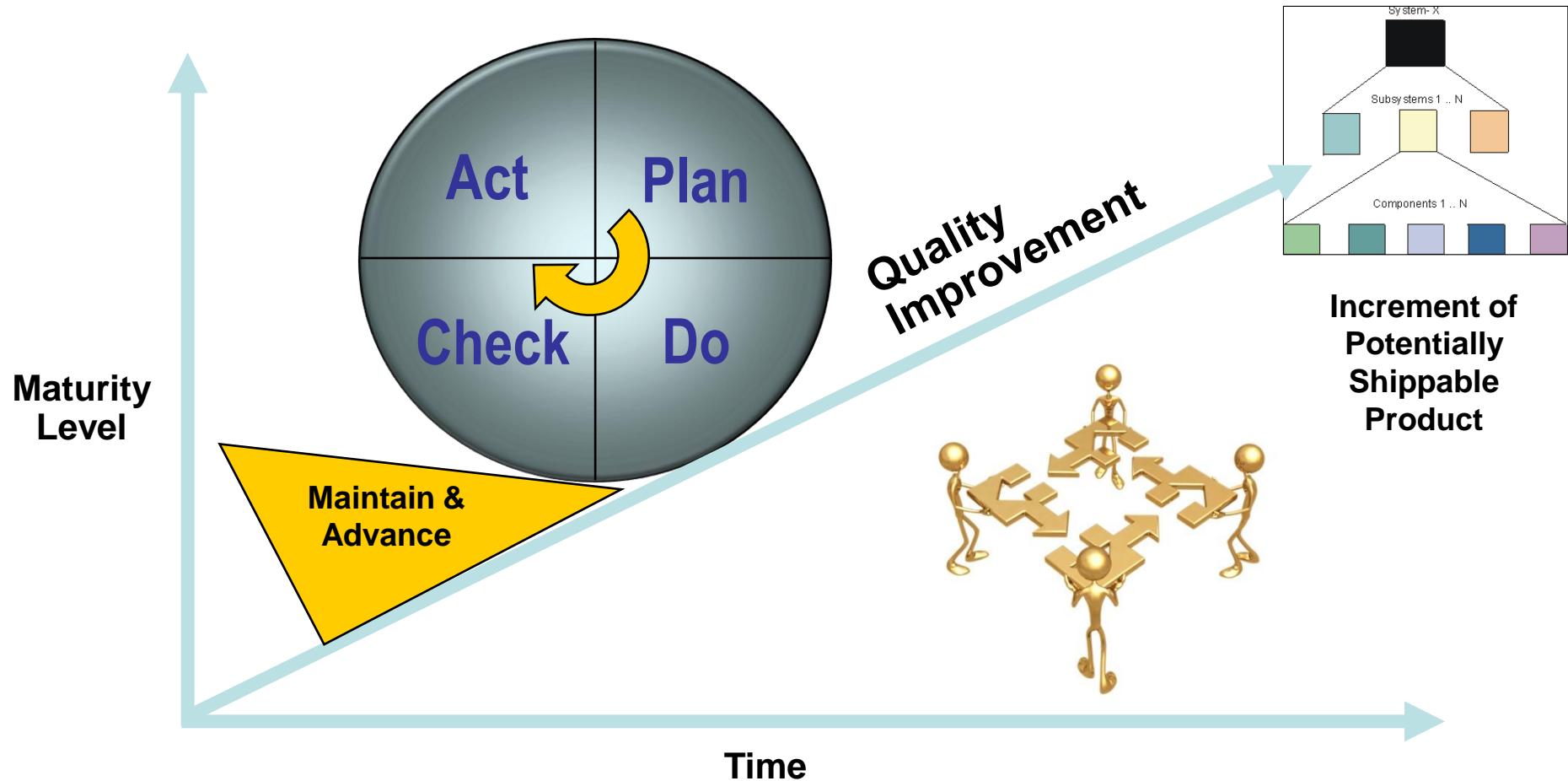
# What is Iterative and Incremental Development?

- The definition of "iterative" is to involve repetition
- Iterative Development is a development approach that "cycles" through a set of activities, from understanding requirements to *incrementally* produce and refine an effective solution
- Iterative Development involves the successive refinement of the solution definition and implementation by the repetitive application of the core development activities to *incrementally* produce and refine an effective solution

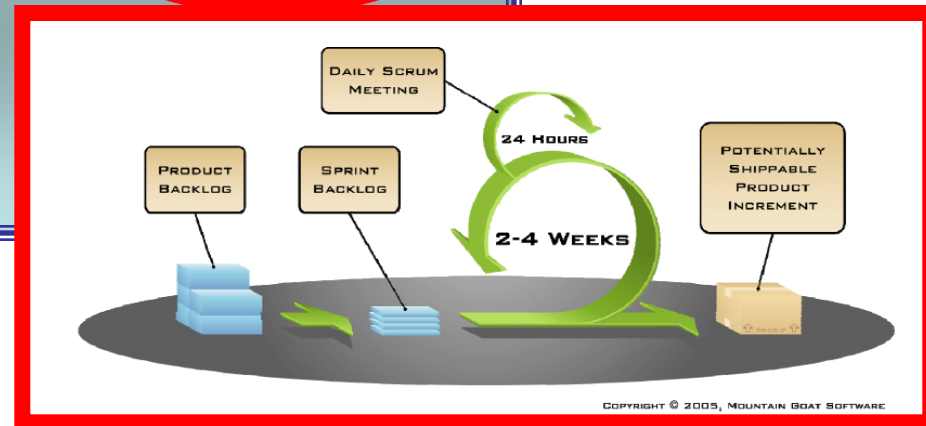
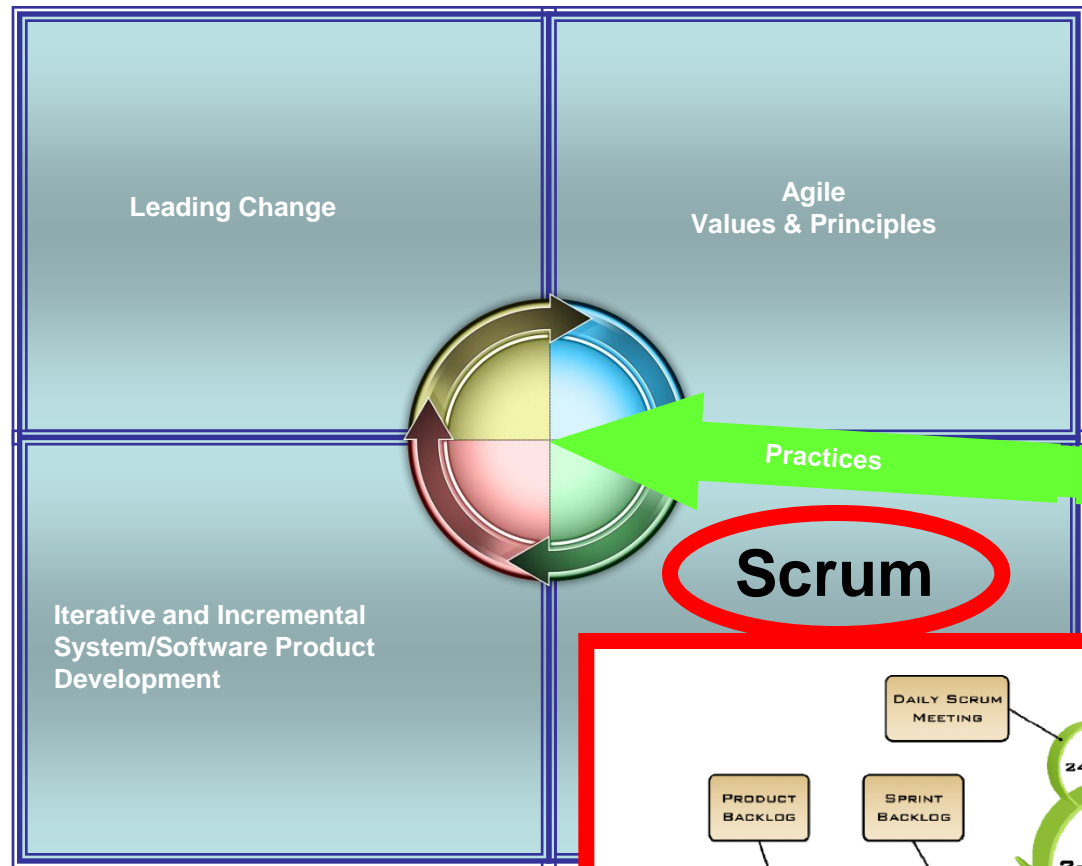


# The General Pattern of Agile Development

## Deming's Incremental Quality Improvement Cycle PDCA



# The **4th** Element of Being Agile



# Scrum Explained

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or *‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—* may better serve today’s competitive requirements.”

Copyright © 2008 Oobeyagroup.

Hiroataka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”, *Harvard Business Review*, January 1986.

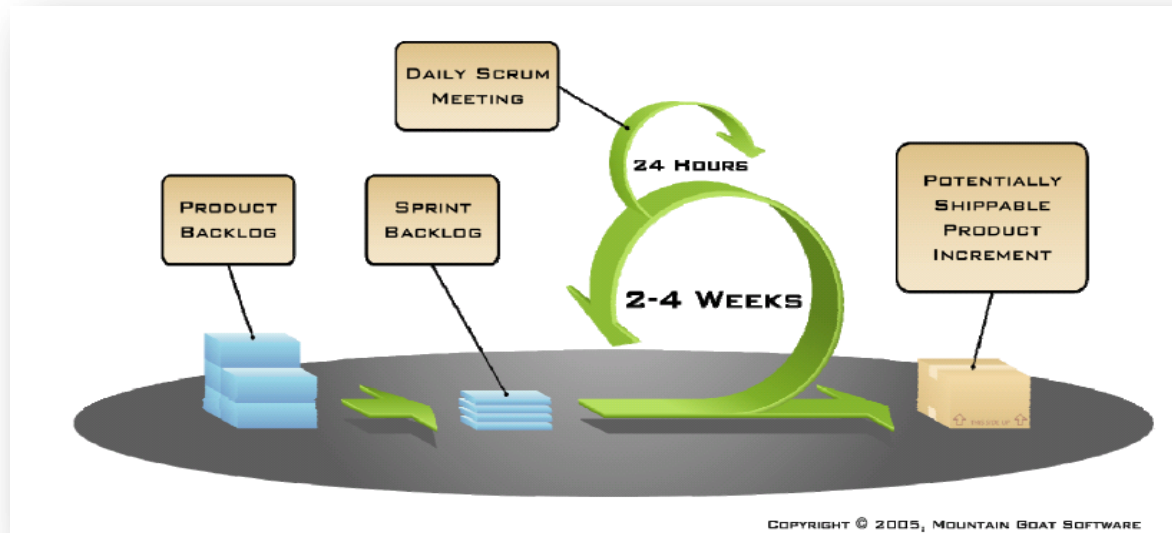


# Think of Scrum as an Agile Framework Embodying Iterative and Incremental Product Development

## Roles



- Product Owner
- Scrum Master
- Team



- Planning
- Daily Standup
- Sprint Review
- Retrospective



# Scrum Roles & Definitions

(continued on next slide)



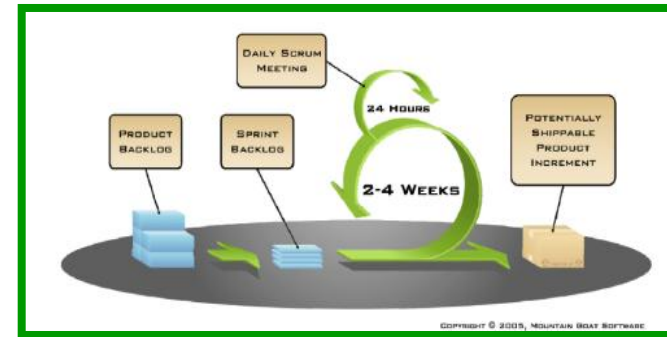
## ❖ Product Owner

- In Scrum, a single person must have final authority representing the customer's interest in backlog prioritization and requirements questions.
- This person must be available to the team at any time, but especially during the sprint planning meeting and the sprint review meeting.
- Challenges of being a product owner:
  - Resisting the temptation to "manage" the team. The team may not self-organize in the way you would expect it to. This is especially challenging if some team members request your intervention with issues the team should sort out for itself.
  - Resisting the temptation to add more important work after a Sprint is already in progress.
  - Being willing to make hard choices during the sprint planning meeting.
  - Balancing the interests of competing stakeholders.

# Scrum Roles & Definitions

(continued on next slide)

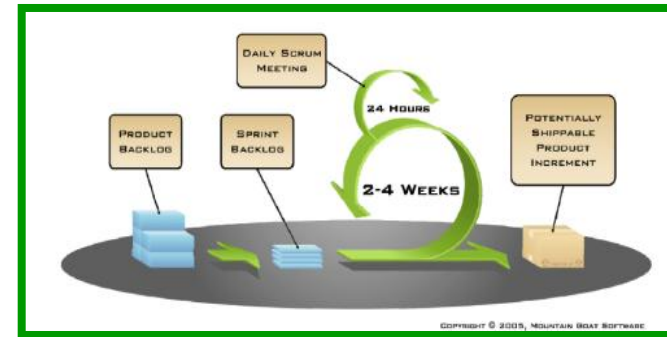
## ❖ Scrum Master



- The Scrum Master is a facilitator for the team and product owner. Rather than manage the team, the Scrum Master works to assist both the team and product owner in the following ways:
  - Remove the barriers between the development and the product owner so that the product owner directly drives development.
  - Teach the product owner how to maximize return on investment (ROI), and meet his/her objectives through Scrum.
  - Improve the lives of the development team by facilitating creativity and empowerment.
  - Improve the productivity of the development team in any way possible.
  - Improve the engineering practices and tools so that each increment of functionality is potentially shippable.
  - Keep information about the team's progress up to date and visible to all parties.

# Scrum Roles & Definitions

(continued from previous slide)



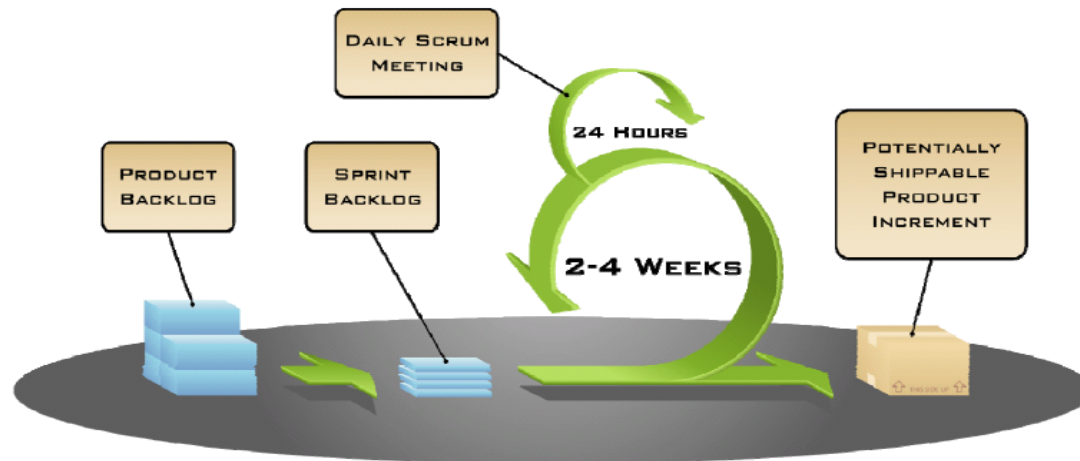
- Team

- A team (or "Scrum team") is optimally comprised of seven plus or minus two people.
- For software development projects, the team members are usually a mix of software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc.
- This is often called "cross-functional project teams".
- Agile practices also encourage cross-functional team members.
- During a sprint, the team self-organizes to meet the sprint goals. The team has autonomy to choose how to best meet the goals, and is held responsible for them.
- Scrum also advocates putting the entire team in one team room.

# Looking at SCRUM from a Different Perspective

## Roles

- Product Owner
- Scrum Master
- Team



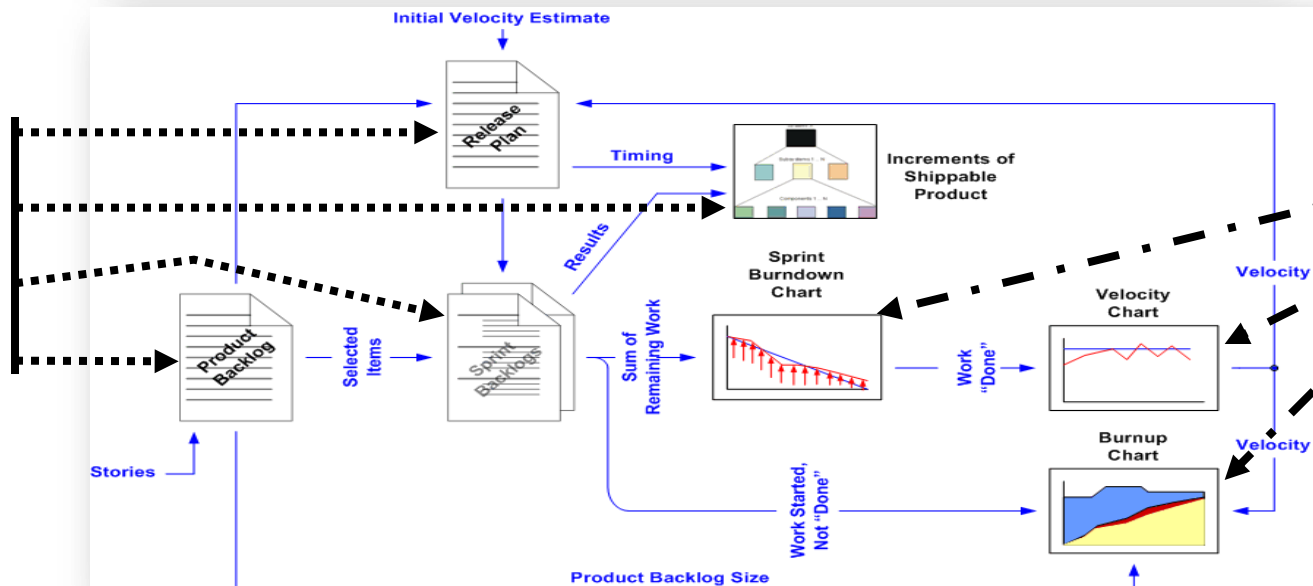
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



- Planning
- Daily Standup
- Sprint Review
- Retrospective



## Pivotal Points



## Progress Items

# Product Backlog

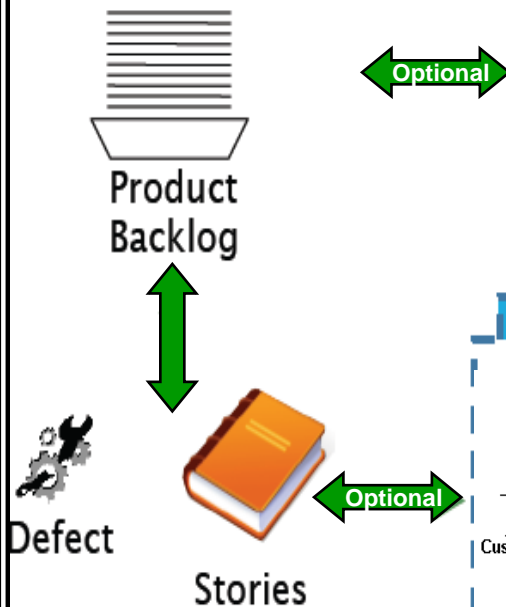
User Stories	Business Priority	Story Points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2
Story H	8	8
Story I	9	5
Story J	10	1



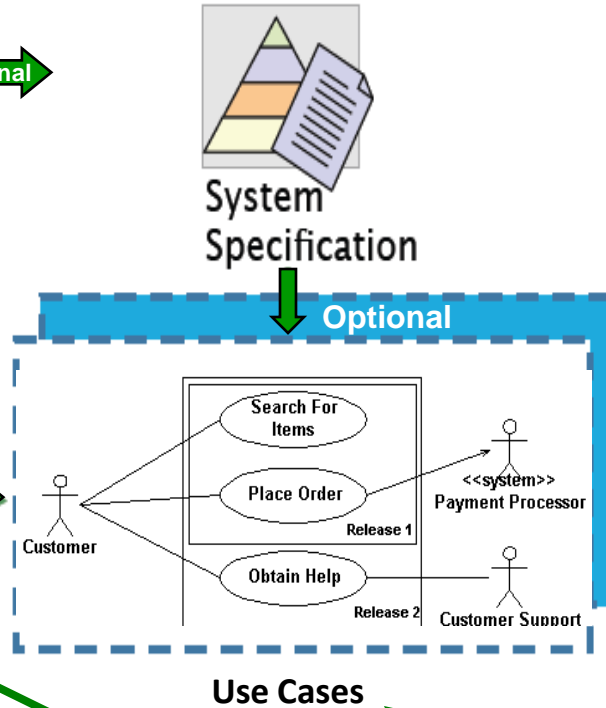
Sometimes You Have to See the Big Picture  
to Know How the Pieces Fit Best Together



## Project Life Span

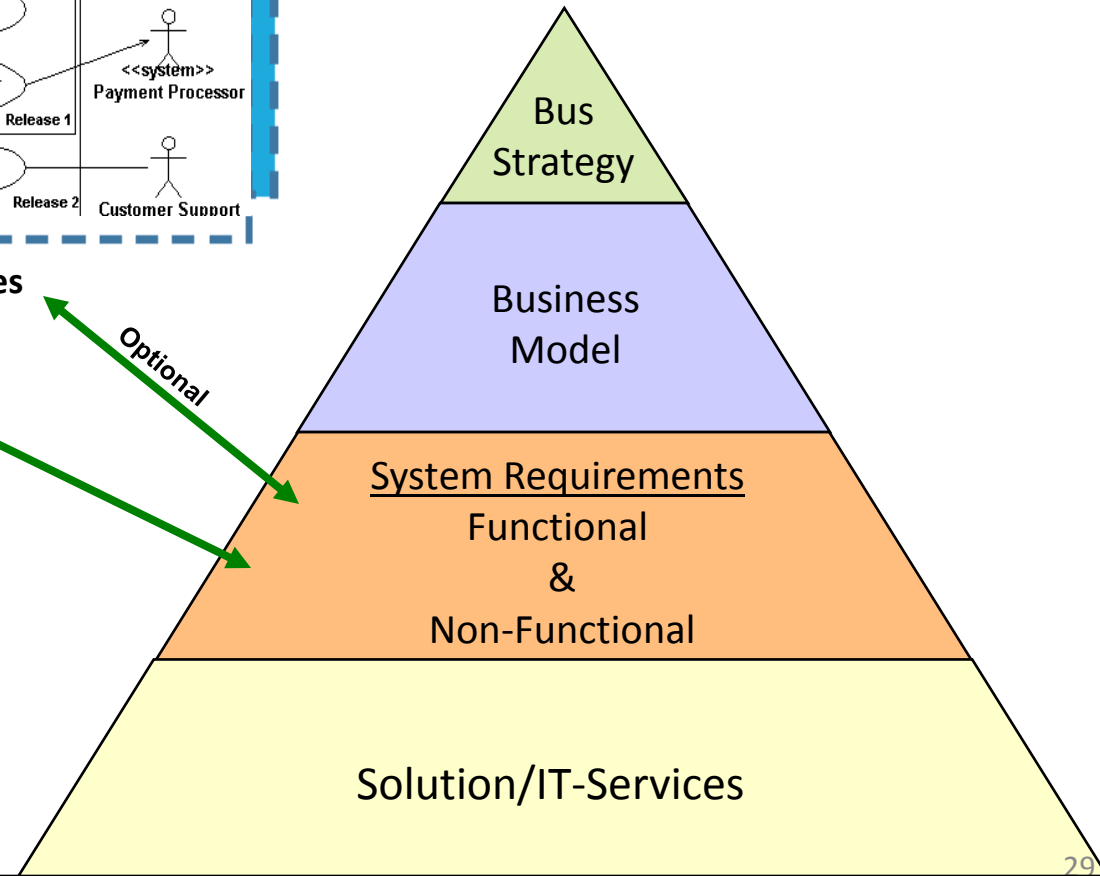


## System Life Span



Use Cases

Optional



Bus Strategy

Business Model

System Requirements  
Functional  
&  
Non-Functional

Solution/IT-Services

# Prioritizing



**There are four factors to consider when prioritizing**

1. The amount and significance of learning and new knowledge gained by developing the product increment
2. The amount of risk removed by developing the product increment
3. The financial value of having the product increment
4. The cost of developing the product increment

# Story Points: Relative Measure of the Size of a User Story

## Product Backlog

- ✓ What matters are the relative values
- ✓ The raw values we assign are unimportant
- ✓ A story assigned a two should be twice as much as a story that is assigned a one; it should be two-thirds of a story that is estimated as three story points
- ✓ Estimating in story points completely separates the estimation of effort from the estimation of duration

User Stories	Business Priority	Story Points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2
Story H	8	8
Story I	9	5
Story J	10	1

# Team Velocity

**Velocity** is a measure of a team's rate of progress per Sprint





# Let's Reflect

**Painting** the interior of your house

Upstairs		Downstairs	
Bathroom 4	Laundry 4	Family Room 8	
Kid's Room 4	Kid's Room 4	Dining Room 6	Bathroom 2
Master Bedroom 8			Bathroom 2
		Office 4	Kitchen 5

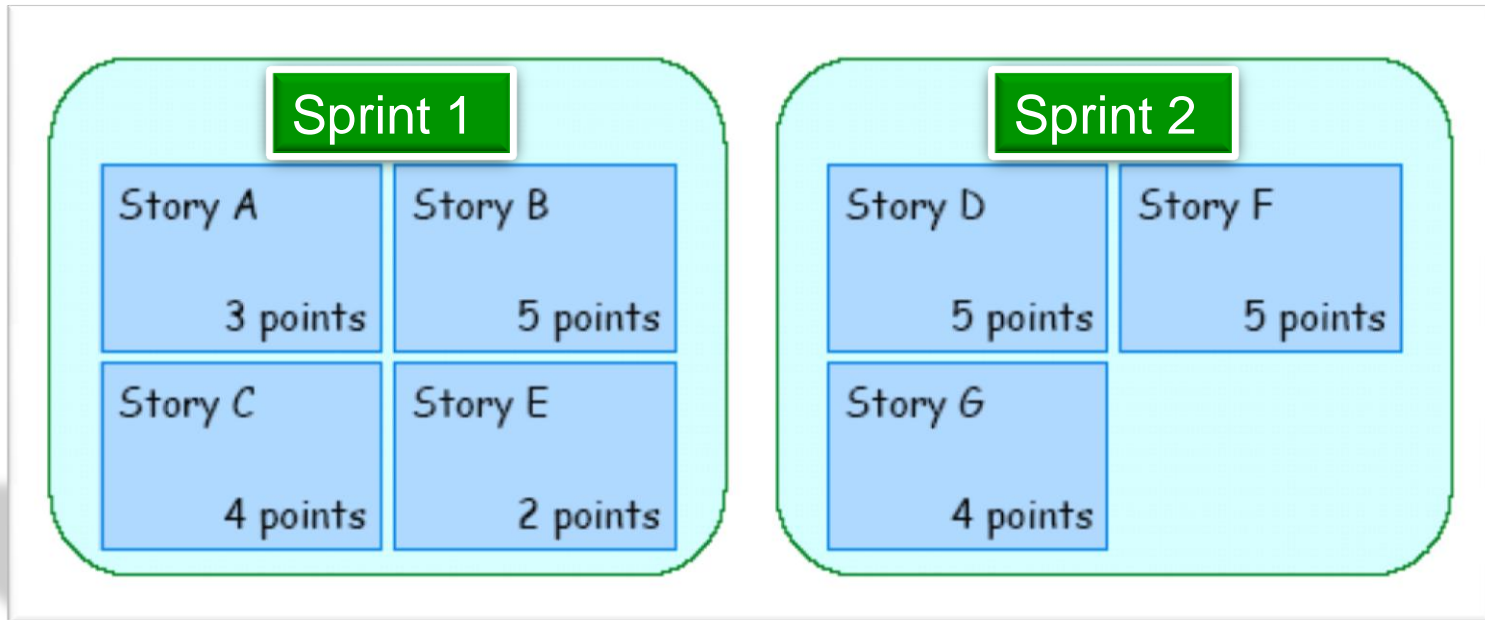




# Let's Reflect

It will take ? weeks to complete our project

- ❖ We have a total of 28 story points
- ❖ Our team velocity is 14 story points per sprint

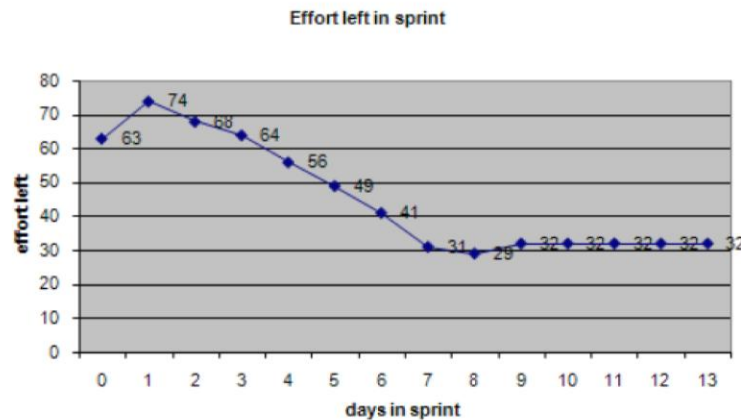


***All Stories → Cumulative Relative Size → Divided-By Velocity  
→ Multiplied-By Length-of-Sprint = Duration***



# Sprint Backlog

Story ID	Story/task	days in sprint / effort left													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
		63	74	68	64	56	49	41	31	29	32	32	32	32	32
10	Fetch one day temperature data from the weather provider system														
	Make our server connect and authenticate to the provider system	4	16	12	8	3	3	3	3	3	3	3	3	3	3
	Read provider's data directory	8	7	7	7	4	0	0	0	0	0	0	0	0	0
	Parse the current temperature out of the data	6	6	4	4	4	1	1	1	1	1	1	1	1	1
	Push the temperature data to the client	16	16	16	16	16	16	8	2	0	0	0	0	0	0
11	Fetch rain, snow, etc details from the provider														
	Parse snow/rain data from the provider's data	4	4	4	4	4	4	4	0	0	0	0	0	0	0
	Push the snow/rain data to the client	4	4	4	4	4	4	4	4	0	0	0	0	0	0
	Redesign client screen a bit										3	3	3	3	3
	Refactor the server code										4	4	4	4	4
12	Fetch several days data from the provider														
	Parse the weather data in day packs	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	Push several days data to the client	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	Auto-refresh feature														
	Make the client ping server once per 4 hours	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	Make the server update the client	2	2	2	2	2	2	2	2	2	2	2	2	2	2



Backlog state taken after day 9

based on: <http://anilesoftwaredevelopment.com/scrum/simple-sprint-backlog>

## Product Backlog

User Stories	Business Priority	Story Points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2
Story H	8	8
Story I	9	5
Story J	10	1

# Progress Tracking Concepts

## ➤ Velocity Chart

- Depicts how much product backlog effort a team completes in one sprint



## ➤ Burn-down Charts

- Depicts how much work remains

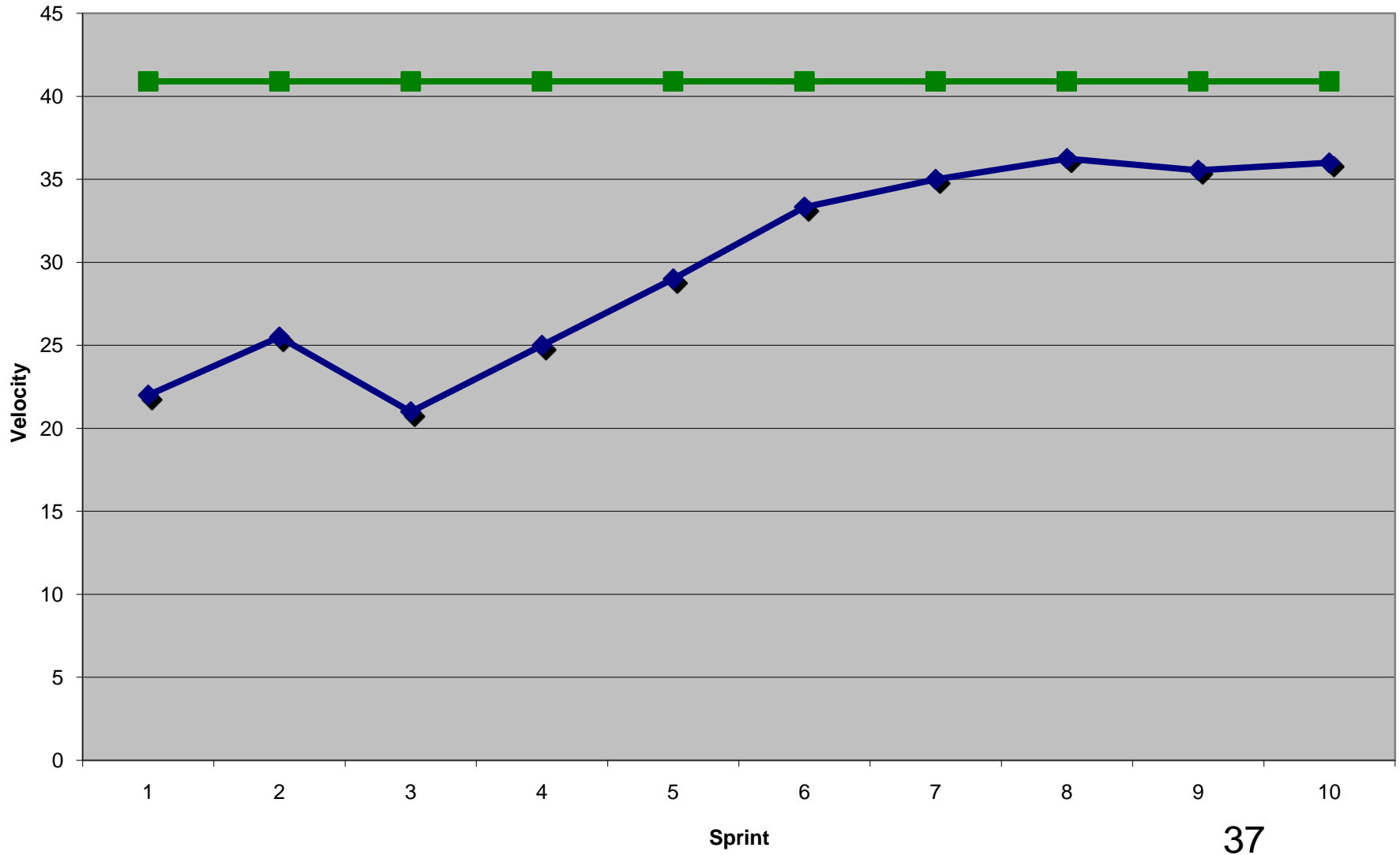


## ➤ Burn-up Charts

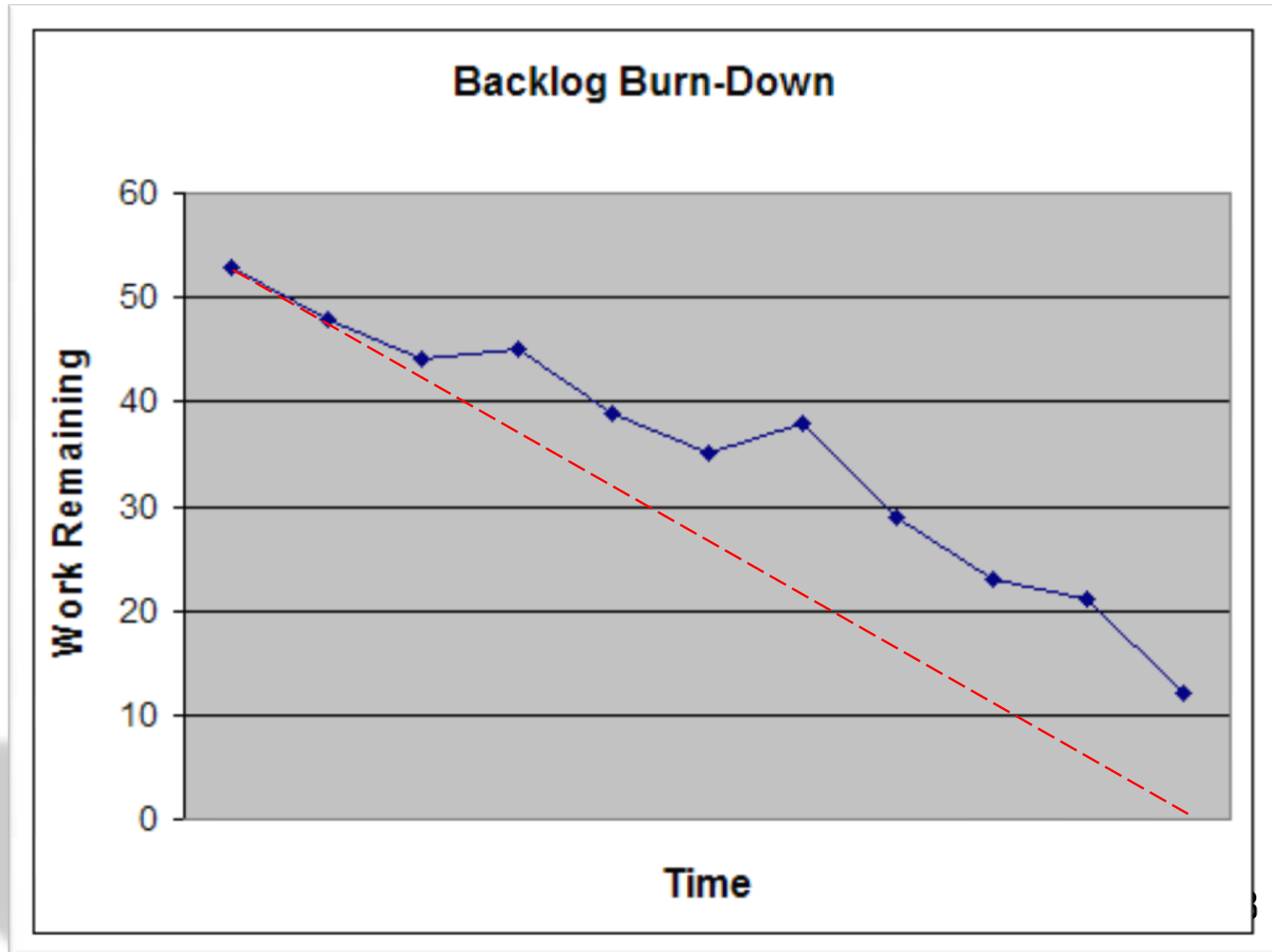
- Depicts total work, work in-progress and work completed



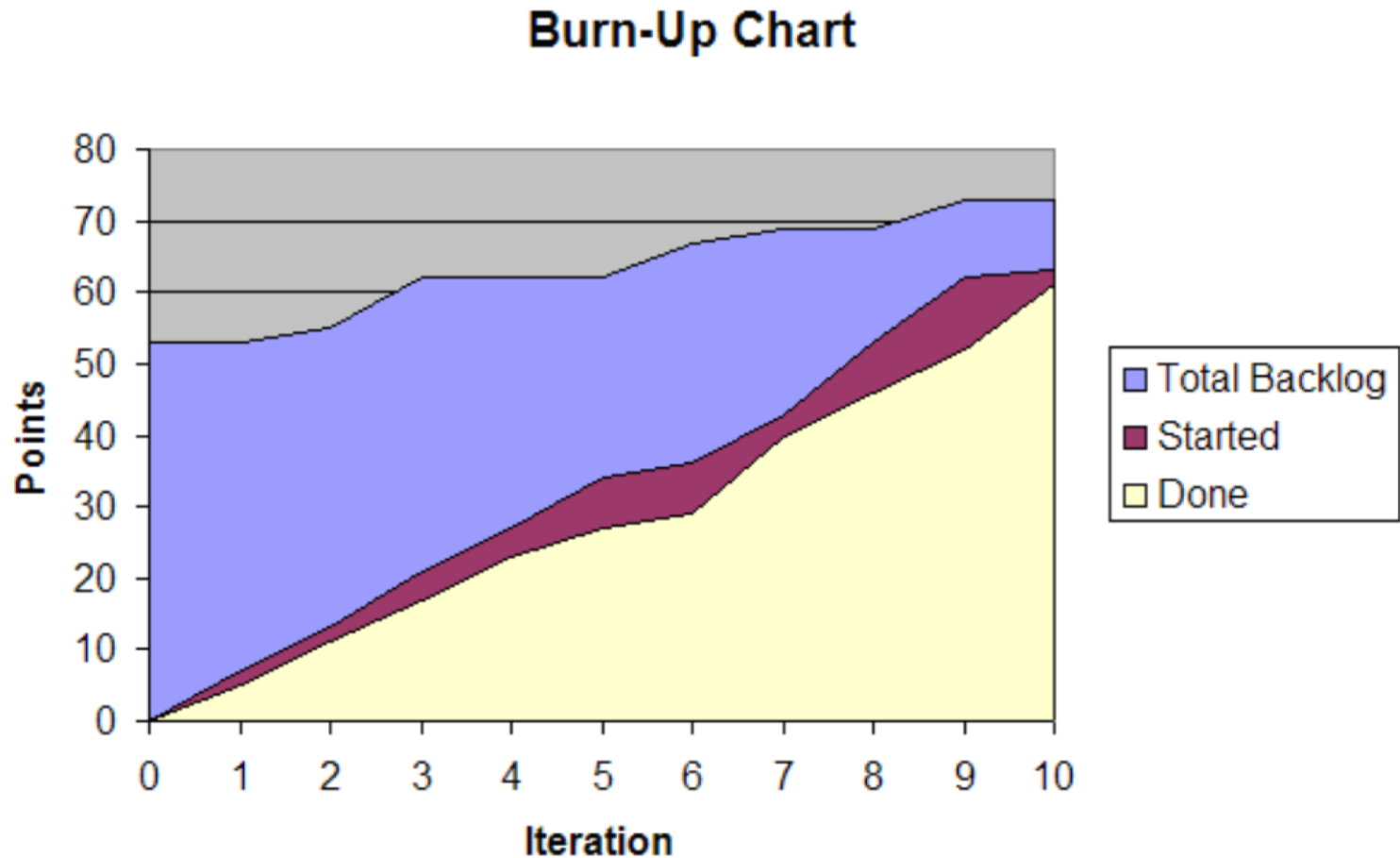
# Velocity Chart Example



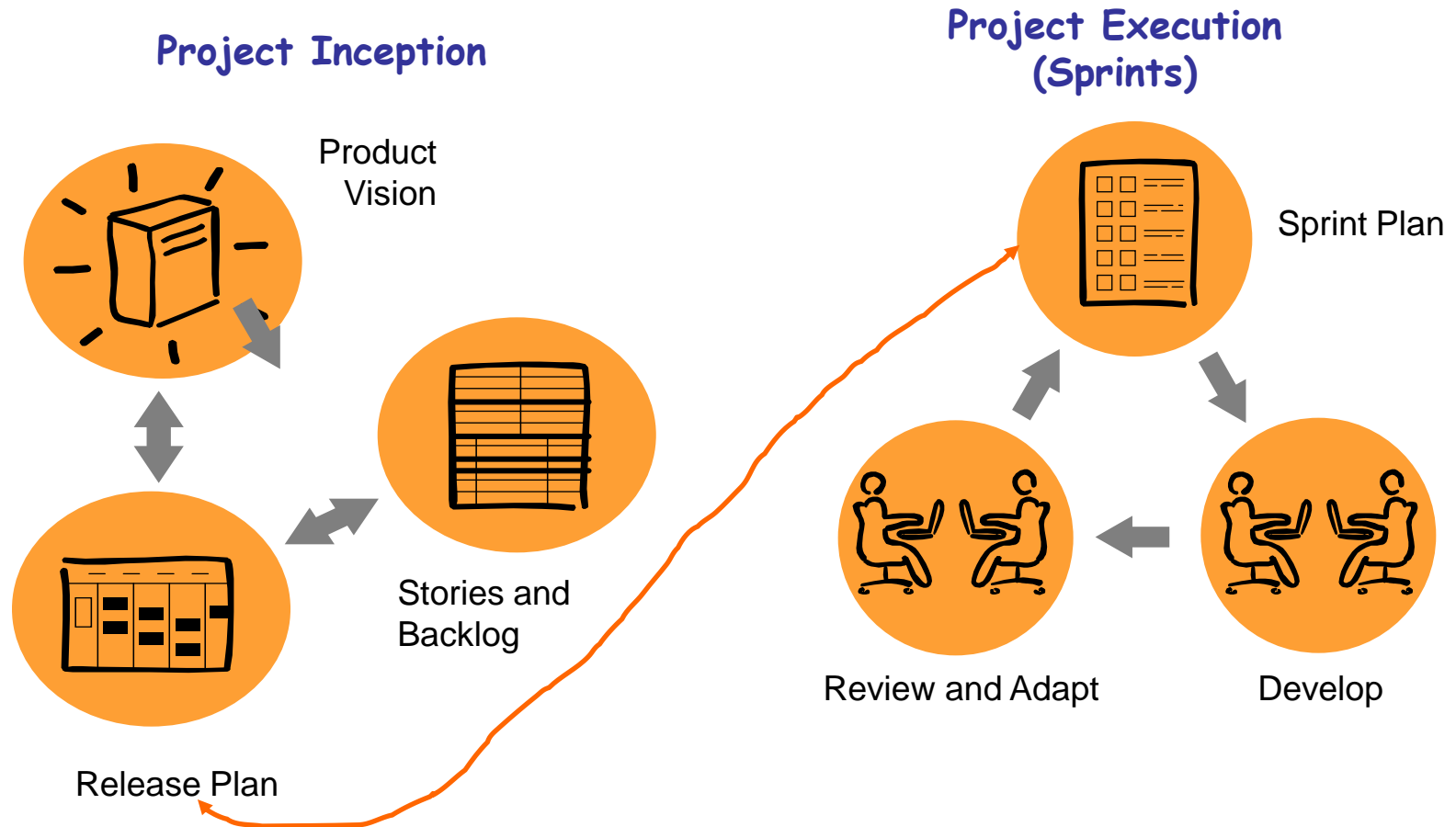
# Burndown Chart Example



# Burnup Chart Example



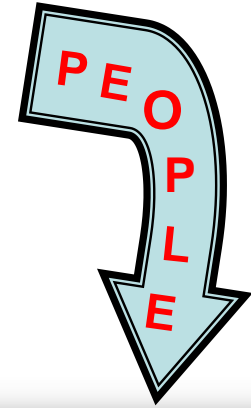
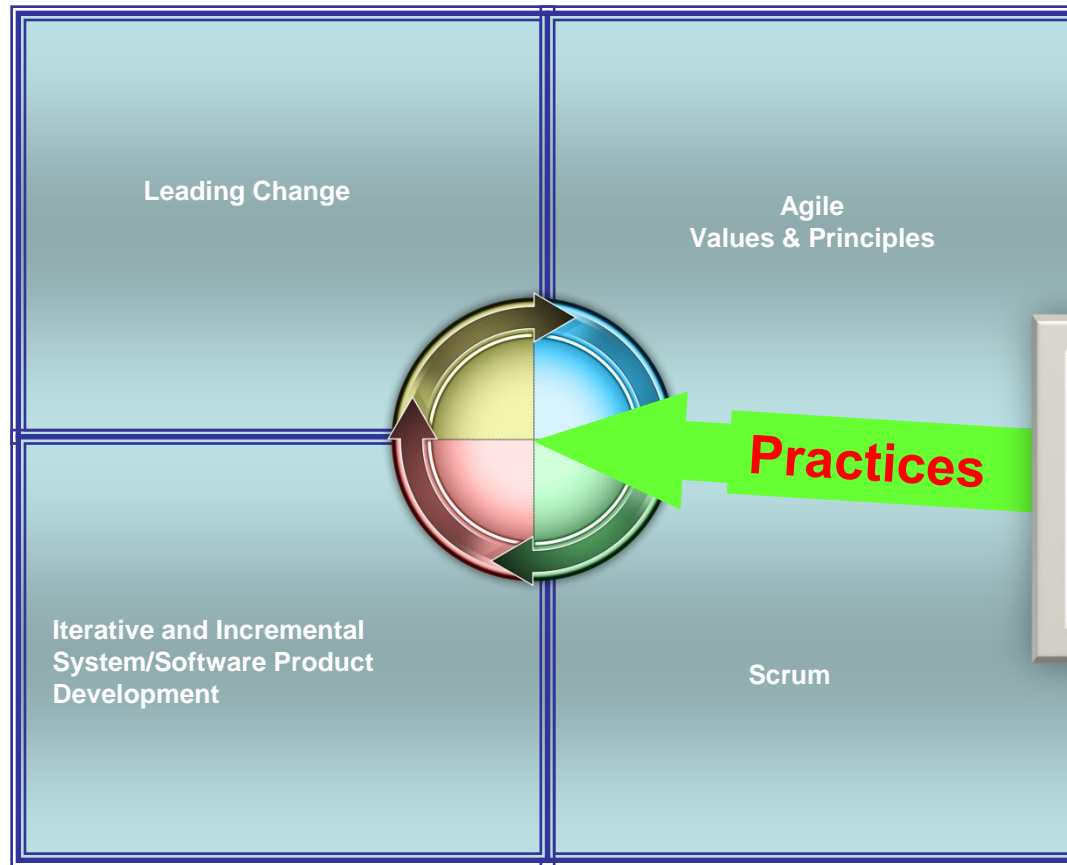
# The Release Plan is an integral part of Agile Product Development



From "Agile Project Management" Jim Highsmith Copyright 2004



# The 5<sup>th</sup> and 6<sup>th</sup> Elements of Being Agile



**A practice is a common approach  
for doing something  
with a specific purpose in mind**

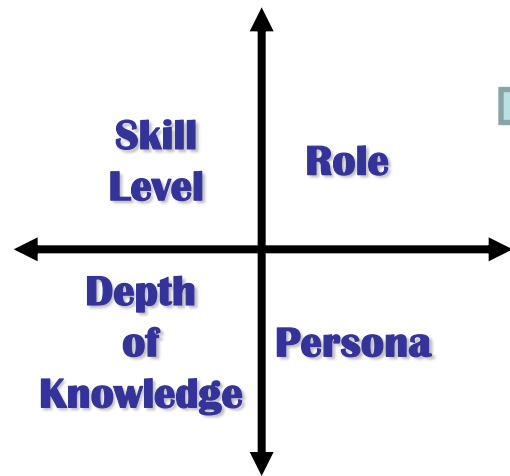


**Skill  
Level**

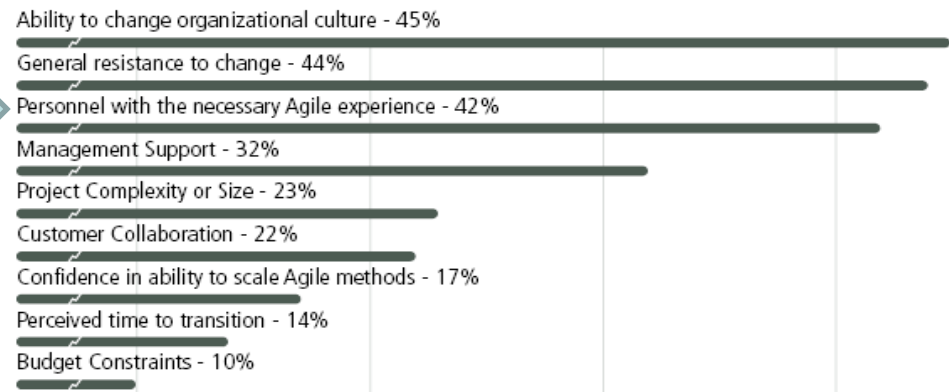
**Role**

**Depth  
of  
Knowledge**

**Persona**



Barrier to Becoming Agile



## Your Competency Assessment

	Executive		Development		Support	
Business Unit						
Information Services And Technology						



# Candidate Practices

**Three Level  
Planning**

**Mastering  
the  
Iteration**

**Continuous  
Integration**

**Team  
Collaboration**

**Reflection  
and  
Adaptation**

**Value/Goal  
Based  
Requirements**

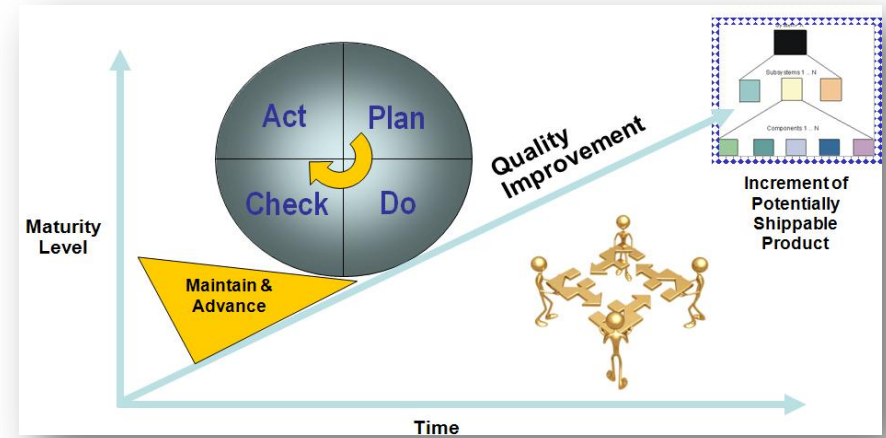
A practice is a  
common  
approach  
for doing  
something  
with a specific  
purpose in mind

**Concurrent  
Testing**

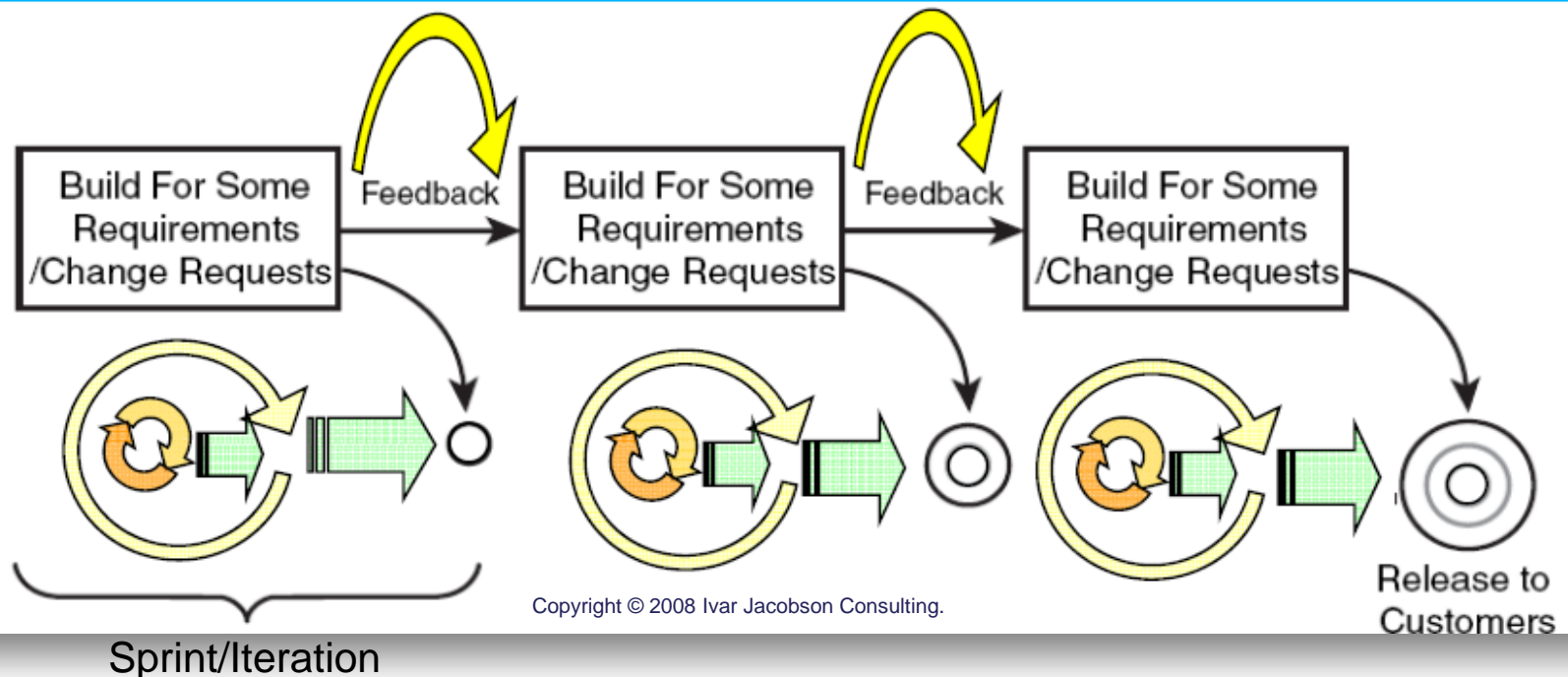
**Shared  
Vision**



# Mastering the Iteration



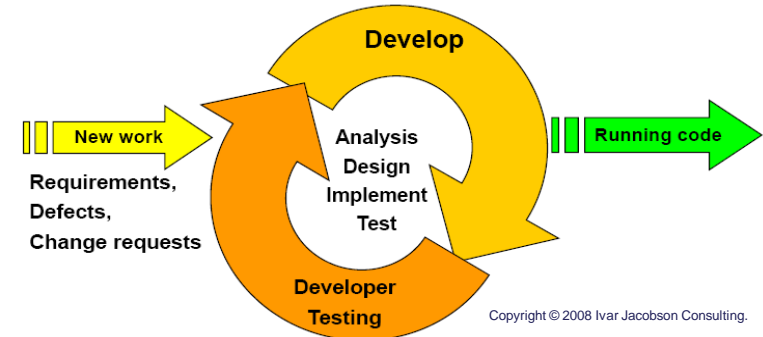
## The Big Picture



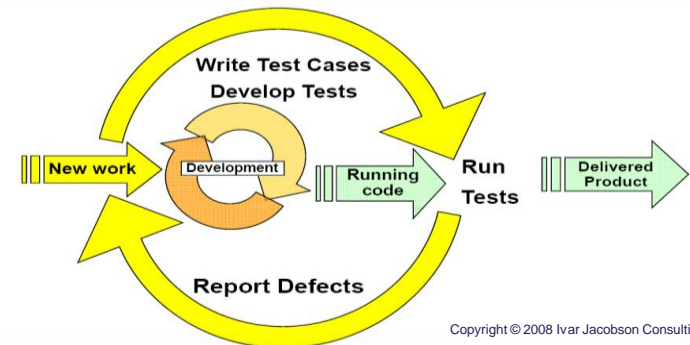


Depends on Your  
Point-of-View

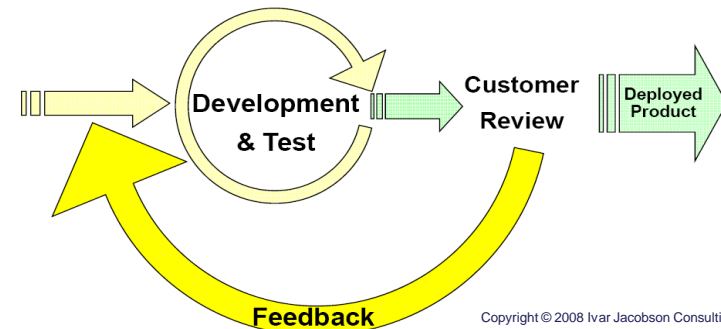
### Iteration from the Developer's Perspective



### Iteration from the Tester's Perspective



### Iteration from the Line of Business' Perspective

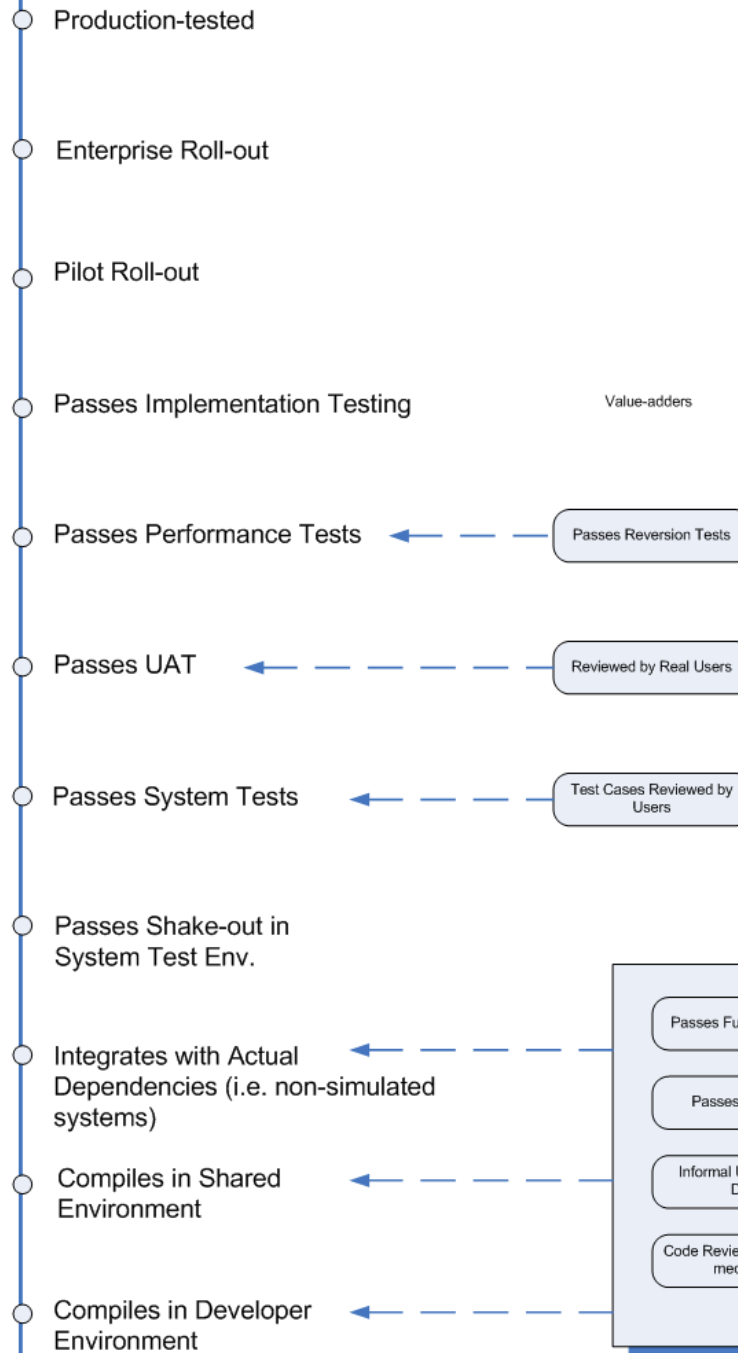






## Degrees of Done-ness

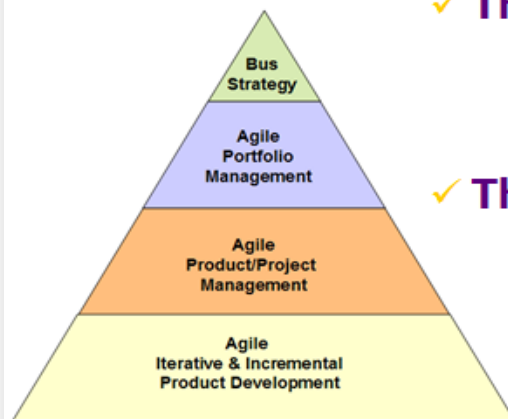
More



Less

# Three Level Planning

This practice focuses on planning the project at three levels: first the governance/program-level, next the release level, and then at a more detailed level, the iteration/sprint level, for the most immediate project iteration.



## ✓ The governance/program-level plan: the PMO plan

- Involves pulling together a number of different projects into a single initiative reflecting a broad business goal.

## ✓ The coarse-grained plan: the release plan

- Provides the “roadmap” for the product.
- Focuses on “cohesive sets” or “packages” of features to provide greater business value.

## ✓ The series of fine-grained plans: the iteration/sprint plan

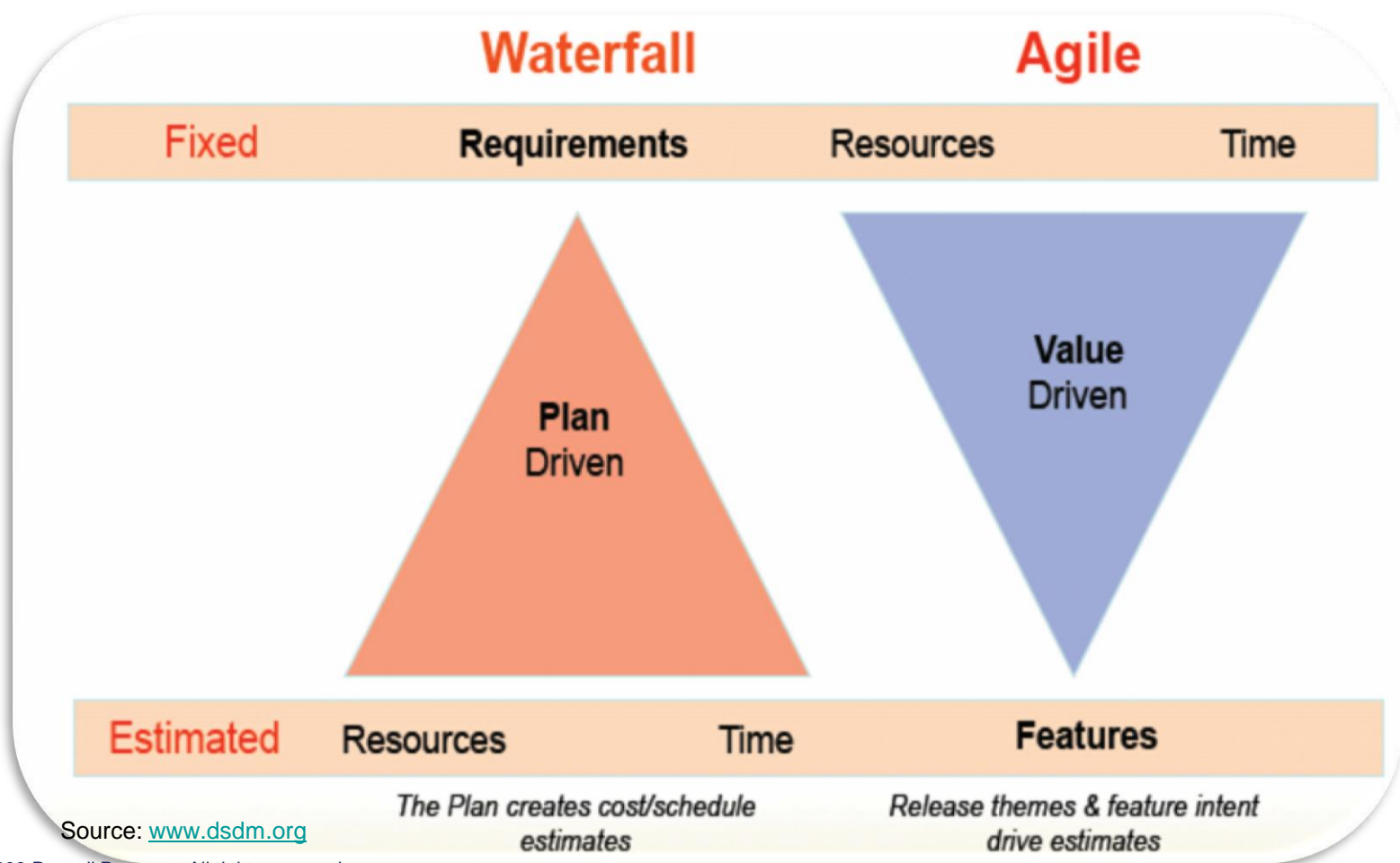
- The goal with iteration planning is to establish a few high-level objectives for what to accomplish during the iteration, produce a sufficiently detailed plan outlining who needs to do what to accomplish those objectives, and define how to assess that you accomplished what you set out to accomplish



# A Paradigm Shift



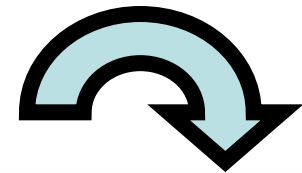
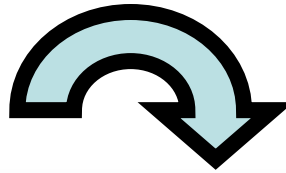
How is Agile Planning Different from Traditional Approaches?



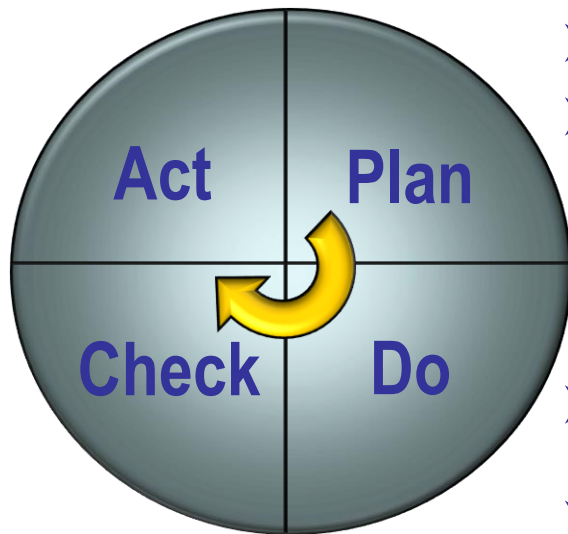
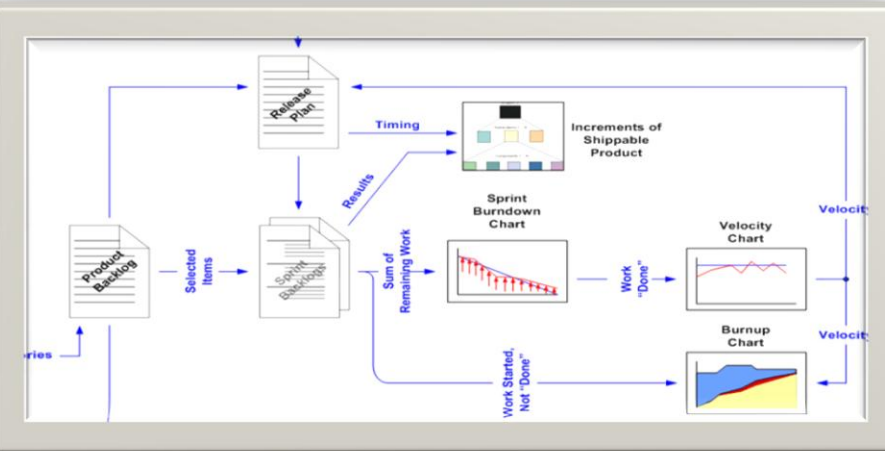
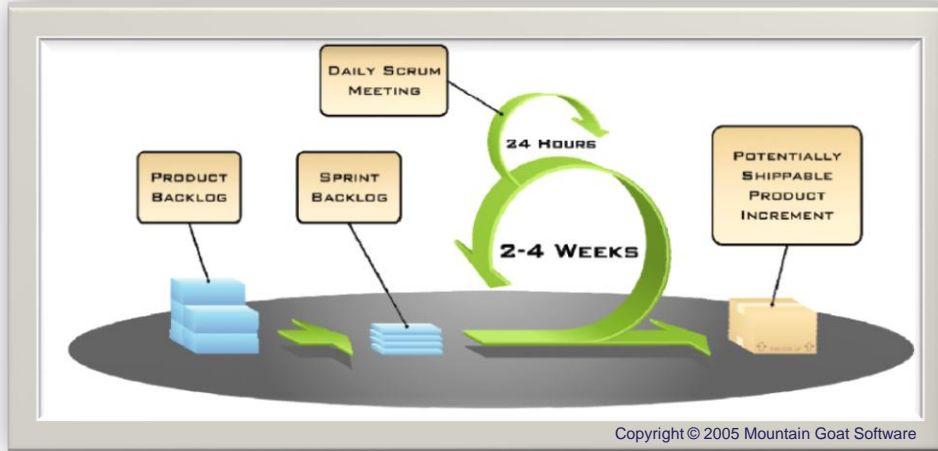
Source: [www.dsdm.org](http://www.dsdm.org)

# When Being Agile, Where Does Quality Management Fit?

***“It Depends”***



# Ensure We Are Doing the Right Things



- **Plan** – Release and Sprint Planning
- **Do** – Team and customer collaboration by elaborating on requirements, doing some design, doing some coding, creating builds and doing some integration, and doing some testing, in one time-boxed pass through a product development lifecycle
- **Check** – Daily stand-ups and end of Sprint Showcase and Retrospective
- **Act** – Adapt the way the team works based on what was learned from the Retrospective



# Ensure We Are Doing the Right Things

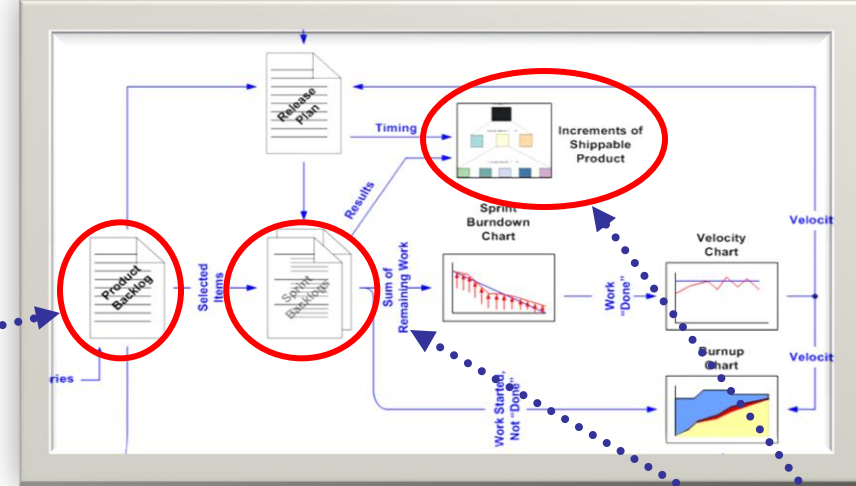
(continued from previous page)

- **Quality Analyst integrated into every team** – A member of the QA team is now an active participant in an sprint
- **Quality Analyst & Testing role** - QA is accountable for helping the development team identify how they know when a story or task is “done”. They help define “done” by co-developing tests with the team that ANYBODY can run including the QA person. They also help determine how best to implement that test (manual or automated, which tools, etc.)
- **Quality Analyst & Team Adaptation** - QA leads retrospectives at the end of sprints and releases. They will ensure that there is just enough process for the team to ensure quality, but not too much, so the team doesn't see value in the process. They also ensure all action items from the retrospective are reflected in future sprints and releases.





# Ensure We Are Doing Things Right



## Product Backlog

- ✓ The owner of the items in the product backlog is known
- ✓ The items in the product backlog are prioritized by the product owner and are assigned relative commercial or operational business value using story points

## ➤ Sprint Backlog

- ✓ Items in the sprint backlog have been derived from the product backlog
- ✓ The items in the sprint backlog have been prioritized by the sprint team and have been assigned an estimate to complete using story points, ideal days, or hours.

## Product Increment

- ✓ A fully tested demo-able or production ready build (product increment) has been delivered

# What is the Value Proposition Associated with Being Agile

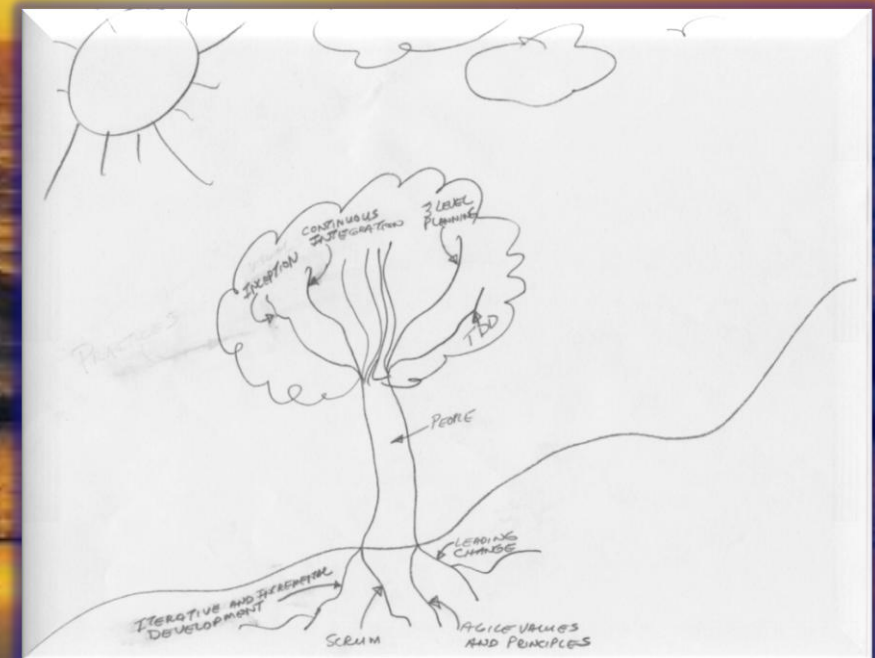
**Value = Positive Results Over Time**



1. Highly motivated and results driven individuals and teams
2. Faster time to market because of faster development cycles, resulting in responding quickly to today's competitive business environment
3. Reduced cost of systems/solution development
4. Much higher business stakeholder satisfaction with the end result
5. Early mitigation of risk & uncertainty

**Looking at the Big Picture  
Quality  
is Everyone's Responsibility**

**Thank You**  
**Any Questions?**



# Back-Up Slides



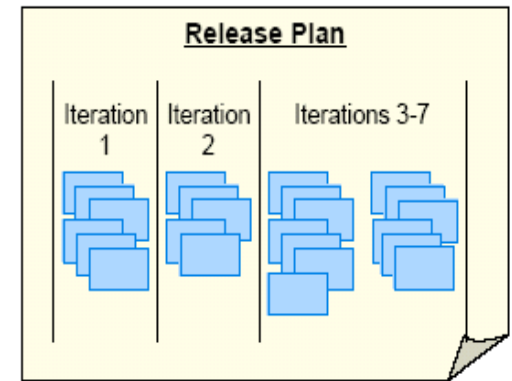
# Agile Methods/Frameworks/Processes

- SCRUM
- Dynamic Systems Development Method (DSDM)
- Crystal Methods
- Feature Driven Development
- Lean Development
- Extreme Programming (XP)
- Adaptive Software Development



# The Release Plan

- ❖ The Release Plan is determined from the team's velocity; initially this is an estimate, a best guess until the team's actual velocity can be determined from historical data
- ❖ We create the Release plan from
  - ❖ The size estimate
  - ❖ The velocity ("size per iteration")
- ❖ The Release plan shows what will be worked on in each iteration
  - ❖ Each iteration contains approximately the same number of story points and is time boxed by iteration length



From "User Stories Applied" by Mike Cohn  
Copyright 2004, Addison-Wesley



# Components of the Release Plan

The Release Plan is comprised of:

1. The Release timeline
2. The Release Content
3. Business Value statement for each release

# Creating the Release Plan

(continue from previous slide)

If our velocity is 14 story points and our sprints are three weeks how long until this product is released:

User Stories	Business Priority	Story Points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2
Story H	8	8
Story I	9	5
Story J	10	1

# Creating the Release Plan

(continue from previous slide)

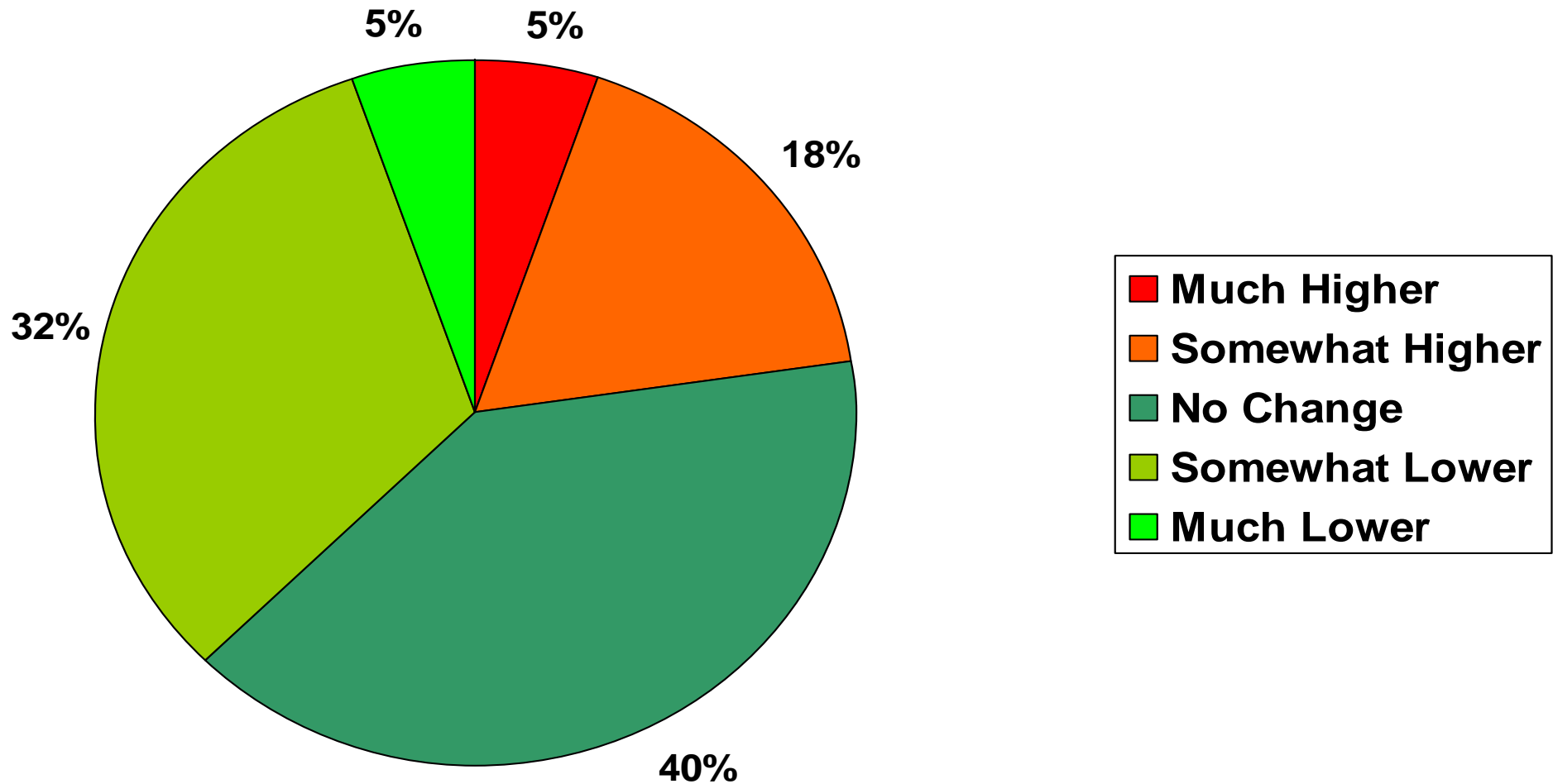
Once we have identified the Release Content for each release, we can prepare a brief summary of the Business Value to be delivered at each release

Example:

Release 1- This release implements ..... and allows users to .....



## How Has Being Agile Affected the Cost of System Development?



Results from Scott Ambler's February 2008 Agile Adoption Survey posted at <http://www.ambysoft.com/surveys/agileFebruary2008.html>