About Me, Chris Zarbock

• Certified: PMI-ACP, PMP, CSM

• One of the first to earn PMI-ACP internationally

• Masters degree in Technology Mgt

• 15 years as a Project Manager
THE EXAM
What you need to know
Why Take The Exam?

Gartner predicts that by the end of 2012, Agile will be used on 80% of all software development projects.

PMI’s research has shown that the use of Agile has tripled from December 2008 to May 2011.

The PMI-ACP is positioned to recognize and validate knowledge of this approach.
## 11 Reference Books

<table>
<thead>
<tr>
<th>Book</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Agile Retrospectives – Making Good Teams Great</td>
<td>Esther Derby, Diana Larsen, Ken Schwaber</td>
</tr>
<tr>
<td>2. Agile Software Development: The Cooperative Game</td>
<td>Alistair Cockburn</td>
</tr>
<tr>
<td>3. The Software Project Manager’s Bridge to Agility</td>
<td>Michele Sliger, Stacia Broderick</td>
</tr>
<tr>
<td>4. Coaching Agile Teams</td>
<td>Lyssa Adkins</td>
</tr>
<tr>
<td>5. Agile Project Management: Creating Innovative Products</td>
<td>Jim Highsmith</td>
</tr>
</tbody>
</table>
## Reference Books, con’t

<table>
<thead>
<tr>
<th>Book</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Agile Estimating and Planning</td>
<td>Mike Cohn</td>
</tr>
<tr>
<td>8. The Art of Agile Development</td>
<td>James Shore</td>
</tr>
<tr>
<td>9. User Stories Applied</td>
<td>Mike Cohn</td>
</tr>
<tr>
<td>10. Agile Project Management with Scrum</td>
<td>Ken Schwaber</td>
</tr>
<tr>
<td>11. Lean Agile Software Development</td>
<td>Allan Shalloway, Guy Beaver, James R. Trott</td>
</tr>
</tbody>
</table>

*All exam questions come from these books.*
Exam – Test & Timing

• 100 scored items and 20 un-scored (pre-test) items, 3 hours to complete

Note: The un-scored items are not identified and are randomly distributed throughout the exam
## Exam Content

<table>
<thead>
<tr>
<th>Content</th>
<th>% of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Tools &amp; Techniques</td>
<td>50%</td>
</tr>
<tr>
<td>Agile Knowledge &amp; Skills</td>
<td>50%</td>
</tr>
</tbody>
</table>

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
### Tools and Techniques (50% of Exam)

<table>
<thead>
<tr>
<th>Agile Tools &amp; Techniques</th>
<th>Soft skills negotiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning, monitoring and adapting</td>
<td>Value-based prioritization</td>
</tr>
<tr>
<td>Agile estimation</td>
<td>Risk management</td>
</tr>
<tr>
<td>Agile analysis and design</td>
<td>Metrics</td>
</tr>
<tr>
<td>Product quality</td>
<td>Value stream analysis</td>
</tr>
</tbody>
</table>

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
# Exam Content, con’t

## Knowledge and Skills (50% of Exam)

<table>
<thead>
<tr>
<th>Level</th>
<th>% of Knowledge and Skill Content / % of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>65% / 33%</td>
</tr>
<tr>
<td>(18 knowledge/skills)</td>
<td></td>
</tr>
<tr>
<td>Level 2</td>
<td>25% / 12%</td>
</tr>
<tr>
<td>(12 knowledge/skills)</td>
<td></td>
</tr>
<tr>
<td>Level 3</td>
<td>10% / 5%</td>
</tr>
<tr>
<td>(13 knowledge/skills)</td>
<td></td>
</tr>
</tbody>
</table>

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
### Knowledge and Skills – Level 1

<table>
<thead>
<tr>
<th>Level</th>
<th>% of Knowledge and Skill Content / % of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>65% / 33%</td>
</tr>
<tr>
<td>(18 knowledge/skills)</td>
<td></td>
</tr>
</tbody>
</table>

33% of total examination questions

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
# Exam Content, con’t

<table>
<thead>
<tr>
<th>Level</th>
<th>% of Knowledge and Skill Content / % of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2</td>
<td>25% / 12%</td>
</tr>
<tr>
<td>(12 knowledge/skills)</td>
<td></td>
</tr>
</tbody>
</table>

12% of total examination questions

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
## Knowledge and Skills – Level 3

<table>
<thead>
<tr>
<th>Level</th>
<th>% of Knowledge and Skill Content / % of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 3 (13 knowledge/skills)</td>
<td>10% / 5%</td>
</tr>
</tbody>
</table>

5% of total examination questions

Source: PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline – July 2011
Exam Eligibility

• **Education:** High school diploma, associate’s degree or global equivalent

• **General Project Experience:** 2000 hours (12 months) working on project teams, earned in the last 5 years

Exam Eligibility, con’t

• **Agile Project Experience:** 1500 hours (8 months), in the last 3 years, working on project teams using Agile methodologies

• Hours are in addition to the 2,000 hours required in general project management

• **Training:** 21 contact hours of training earned in Agile practices

# Exam Cost

<table>
<thead>
<tr>
<th>Exam Type</th>
<th>PMI Status</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Based</td>
<td>Member</td>
<td>$435 US</td>
</tr>
<tr>
<td>Computer Based</td>
<td>Non-member</td>
<td>$495 US</td>
</tr>
<tr>
<td>Reexamination</td>
<td>Member</td>
<td>$335 US</td>
</tr>
<tr>
<td>Reexamination</td>
<td>Non-member</td>
<td>$395 US</td>
</tr>
<tr>
<td>Certification renewal</td>
<td>Member</td>
<td>$90 US</td>
</tr>
<tr>
<td>Certification</td>
<td>Non-member</td>
<td>$130 US</td>
</tr>
</tbody>
</table>

AGILE OVERVIEW

Why Agile?
Project teams are expected to deliver business value in shorter timeframes
Deliver or Perish

In this economic climate, when project teams don’t deliver quickly, they are often replaced.

The Agile Framework can get the project team to success.
The PMI-ACP certification shows that you have the Agile skills and experience to get the job done.
Research Shows

Agile Usage

2010 – 35% of projects

2012 – 80% of projects

Sources:
PMI Network Magazine, May 2010
Gartner research, PMI website

Copyright Chris Zarbock, 2012
WHO USES AGILE
Who doesn’t?
Companies using Agile

Bank of America

Amazon.com

PayPal

Microsoft

HP

Adobe

Google

Oracle

LexisNexis

Agile Czar
Charlotte companies

Bank of America
Time Warner Cable
PREMIER
TIAA CREF Financial Services
Wells Fargo
Lowe's
Lendingtree
WATERFALL METHOD
SDLC, DMAIC, etc.
Traditional SDLC Waterfall

Requirements determined early

- Requirements
- Design
- Implementation
- Verification
- Maintenance
## Microsoft Project Plan

<table>
<thead>
<tr>
<th>% Complete</th>
<th>WBS</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>New Website Project Plan</td>
<td>47.31 days</td>
<td>Wed 3/17/10</td>
<td>Fri 5/21/10</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>1</td>
<td>Project Kickoff and Initiation</td>
<td>11 days</td>
<td>Wed 3/17/10</td>
<td>Thu 4/1/10</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>0.1</td>
<td>Acceptance of SOW by Client and Vendor</td>
<td>0 days</td>
<td>Thu 4/1/10</td>
<td>Thu 4/1/10</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>1.2</td>
<td>Conduct Project Kickoff Meeting</td>
<td>0.31 days</td>
<td>Wed 3/17/10</td>
<td>Wed 3/17/10</td>
<td></td>
</tr>
<tr>
<td>25%</td>
<td>2</td>
<td>Project Management</td>
<td>47 days</td>
<td>Wed 3/17/10</td>
<td>Fri 5/21/10</td>
<td></td>
</tr>
<tr>
<td>67%</td>
<td>2.1</td>
<td>Baseline Project Plan</td>
<td>3 days</td>
<td>Wed 3/17/10</td>
<td>Mon 3/22/10</td>
<td>3</td>
</tr>
<tr>
<td>25%</td>
<td>2.2</td>
<td>Project Monitoring &amp; Control</td>
<td>47 days</td>
<td>Wed 3/17/10</td>
<td>Fri 5/21/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2.2.1</td>
<td>Contact List</td>
<td>1 hr</td>
<td>Wed 3/17/10</td>
<td>Wed 3/17/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2.2.2</td>
<td>Project Resourcing</td>
<td>3 hrs</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>9</td>
</tr>
<tr>
<td>25%</td>
<td>2.2.3</td>
<td>Weekly Status Reports</td>
<td>42 days</td>
<td>Wed 3/24/10</td>
<td>Fri 5/21/10</td>
<td>3FS+5 days</td>
</tr>
<tr>
<td>25%</td>
<td>2.2.4</td>
<td>Project Administration (daily consultant status, weekly status)</td>
<td>47 days</td>
<td>Wed 3/17/10</td>
<td>Fri 5/21/10</td>
<td>3</td>
</tr>
<tr>
<td>25%</td>
<td>2.3</td>
<td>Knowledge Transfer</td>
<td>40 days</td>
<td>Wed 3/17/10</td>
<td>Wed 5/12/10</td>
<td>3</td>
</tr>
<tr>
<td>25%</td>
<td>2.3.1</td>
<td>Knowledge Transfer &amp; Monitoring (not to replace formal training)</td>
<td>37 days</td>
<td>Wed 3/17/10</td>
<td>Wed 5/12/10</td>
<td>3</td>
</tr>
<tr>
<td>57%</td>
<td>3</td>
<td>Planning / Design</td>
<td>12.94 days</td>
<td>Wed 3/17/10</td>
<td>Mon 4/5/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3.1</td>
<td>Review current system architect</td>
<td>0.69 days</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3.1.1</td>
<td>Schedule Web-ex to Review Current Environment</td>
<td>1 hr</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>95S</td>
</tr>
<tr>
<td>0</td>
<td>3.1.2</td>
<td>Review Hardware Environment</td>
<td>0.5 hrs</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>3.1.3</td>
<td>Review Supporting Software Environment</td>
<td>1 hr</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>3.1.4</td>
<td>Review Existing Component</td>
<td>1 hr</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>21</td>
</tr>
<tr>
<td>0</td>
<td>3.1.5</td>
<td>Review Existing Workflows</td>
<td>1 hr</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>22</td>
</tr>
<tr>
<td>0</td>
<td>3.1.6</td>
<td>Review Existing Data Capture Templates</td>
<td>1 hr</td>
<td>Mon 3/22/10</td>
<td>Mon 3/22/10</td>
<td>23</td>
</tr>
<tr>
<td>35%</td>
<td>3.2</td>
<td>Documentation</td>
<td>6.25 days</td>
<td>Wed 3/17/10</td>
<td>Thu 3/25/10</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>3.2.1</td>
<td>Requirements Document</td>
<td>3.25 days</td>
<td>Wed 3/17/10</td>
<td>Mon 3/22/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3.2.2</td>
<td>System Test Plan</td>
<td>6.13 days</td>
<td>Wed 3/17/10</td>
<td>Thu 3/25/10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3.2.2.1</td>
<td>Create Test Plan</td>
<td>1 day</td>
<td>Wed 3/17/10</td>
<td>Thu 3/19/10</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>3.2.2.2</td>
<td>Test Cases</td>
<td>3 days</td>
<td>Thu 3/16/10</td>
<td>Tue 3/23/10</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>3.2.2.3</td>
<td>Test Readiness Review</td>
<td>1 day</td>
<td>Tue 3/23/10</td>
<td>Wed 3/24/10</td>
<td>30, 31</td>
</tr>
<tr>
<td>0</td>
<td>3.2.2.4</td>
<td>Capture &amp; Maintain Test Results</td>
<td>1 day</td>
<td>Wed 3/24/10</td>
<td>Thu 3/25/10</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>3.2.2.5</td>
<td>Schedule Customer Resources for QA</td>
<td>1 hr</td>
<td>Thu 3/25/10</td>
<td>Thu 3/25/10</td>
<td>30</td>
</tr>
</tbody>
</table>
AGILE HISTORY
Some history and a manifesto
We are uncovering better ways of developing software by doing it and helping other 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.
12 Principles

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

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

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

4. Business people and developers must work together daily throughout the project.
12 Principles, con’t

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

6. The most efficient and effective way of conveying information to and within a development team is face-to-face communication.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers and users should be able to constant pace indefinitely.
12 Principles, con’t

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity – the art of maximizing work not done - is essential.

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

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

- **Scrum**
  Focused on Project Management. Iterative process, open to change. *This is the most popular “flavor” of Agile.*

- **Lean Software Development**
  Create visible customer value rapidly, build change-tolerant software. Create only necessary functionality and no more.

- **Kanban**

- **XP – Extreme Programming**
  Process is visible and accountable. Developers commit to what they will accomplish, provide deployable software. Developed by Kent Beck.
XP Activities

- Coding
- Designing
- Listening
- Testing

Reference
http://www.selectbs.com
for more information
XP Values

- Simplicity
- Communication
- Feedback
- Respect
- Courage

Reference [http://www.extremeprogramming.org](http://www.extremeprogramming.org) for more information
XP Practices

Fine Scale Feedback:
- Pair Programming
- Planning Game
- Test Driven Development
- Whole Team

Continuous Process
- Continuous Integration
- Design Improvement
- Small Releases

Shared Understanding
- Coding Standards
- Collective Code Ownership
- Simple Design
- System Metaphor

Programmer welfare
- Sustainable Pace
XP Roles – The Customer

- Drives the project
- Defines the project and sets goals
- Answers the “What” question
XP Roles – The Coach

➢ Guides and monitors the team

➢ Acts as a mentor

➢ Resolves conflicts
XP Roles – The Programmer

- Define coding standards
- Estimate the work
- Turn Customer stories into working code
XP Roles – The Tester

- Ask the questions to generate good tests
- Writing acceptance tests
- Executing tests
- Modifying tests over iterations
LEAN SOFTWARE DEVELOPMENT
Lean Software Development

- Lean principles are rooted in manufacturing
- First used by Toyota
- Focuses on eliminating waste, delivering value
- Doing high priority work only
KANBAN
Limit WIP
Kanban – Limit WIP

- Comes from Lean principles
- Means “Visual Card” 看板
- Visualize the workflow
- Limit Work in Process (WIP)
- Measure and optimize flow
Kanban – Limit WIP, con’t

- Pull system
- Move work left to right
- **Limit** amount of WIP in different steps at one time
- Not time-boxed
Kanban Board

Source: http://blogs.mulesoft.org/
VALUE STREAM MAPPING
Value Stream Mapping

- Tool for organizations to capture and document process steps of a value stream
- Way to identify value added and non-value added steps
- Eliminate waste
SCRUM
A better way of working
Iterative process - Welcomes change

Source: Mountain Goat Software
What is a Sprint?

- Iteration (normally 2 or 4 weeks)
- A time-boxed period
- Team is focused on delivering a “potentially shippable increment” of product functionality
- Sprint has a goal - (i.e. Develop the user interface)
Scrum Artifacts

- **Product Backlog** – Product Owner’s Vision

- **Sprint Backlog** – Work to be done in the Sprint, tasks from previous Sprints

- **Burn down Chart** – Shows remaining hours or story points, we don’t care what hours or story points have been spent, we want to know what is remaining
TEAM DYNAMICS
Let’s all work together
Self-organizing, Self-managing

- The product owner prioritizes the work
- Time to do the work is time-boxed
- User Stories are broken down into tasks
- Acceptance criteria is created
- The team decides how to execute
- The team communicates progress
Agile Encourages Co-location

- Ideally, co-locate the team in a single room
- Encourages osmotic communication
- Fosters chemistry and loyalty among team members

Distributed Agile, as the name implies, is a model in which projects execute an Agile Methodology with teams that are distributed across multiple geographies.
Team Decision Making

- High performance teams
  - Rely on collaboration to create and respond to change

- Collaboration
  - The team makes decisions
  - Scrum Master only guides the process
Team Maturity Model

Forming
- Team members get to know each other

Storming
- Discord, competing ideas

Norming
- Desired to be accepted, work as a team

Performing
- Teams work together for a common goal

Source: Tuckman, Bruce (1965)
Establishing a Product Vision

- Business Case
- Project Charter
- Elevator Statement
- Vision Box
UNDERSTANDING THE CUSTOMER
Customer Personas

- Fictional character
- Give them a name
- Create a backstory
- Their goals for the system
- Likes and dislikes

Creating customer personas keeps the team in a customer-centric mode
PRODUCT BACKLOG
What gets done when
What is a Product Backlog?

✓ Defined and prioritized by the Product Owner
✓ A single prioritized list
✓ Input from Dev team and Stakeholders
✓ Team provides estimates
✓ Continuous backlog grooming
Minimal Marketable Feature (MMF)

- The key to phased delivery
- Smallest possible set of functionality that has value in the marketplace
- Software could provide just one of these features and you would see some benefit.

Source: James Shore, The Art of Agile
USER STORIES
Let’s have a conversation
Some teams start out with sticky notes to:
Generate ideas, create user stories, tasks, quality checks and acceptance criteria.
User Stories - Format

As a **dog** *(who)*

I want to **order food online** *(goal)*

So that I don’t have to rely on my **owner** *(why)*

Tasks are written by and for the team
User Stories – Format con’t

As a customer (who)
I want to open an account online (goal)
So that I can bank from home (why)

Task
Design
Security

Task
Create
User Interface

Stories are used as a conversation starter
INVEST MODEL for User Stories

I = Independent
N = Negotiable
V = Valuable
E = Estimable
S = Small
T = Testable
Definition of “Done”

When is the story “Done Done”?

- Design
- Coded
- Passed Unit Tested
- Passed System Tested
- Accepted by Product Owner
ESTIMATING STORIES

Cards anyone?
## Methods of Estimating

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask the Expert</td>
<td>Expert relies on past experience, intuition</td>
</tr>
<tr>
<td>Analogy</td>
<td>How large is this story compared to other stories?</td>
</tr>
<tr>
<td>Disaggregation</td>
<td>Split a story into smaller stories</td>
</tr>
</tbody>
</table>
Estimating Using Story Points

• Story points are a method of relative sizing

• Used to determine estimated effort compared to other stories

• Not a precise estimate, like using hours
Someone with a PMP background may be used to all tasks being estimated in hours.

Comparison can break tasks down into small, medium and large which helps the developer think more abstractly about the task.
Fibonacci Scale

- Using the Fibonacci scale allows for relative sizing

- The reason for using the Fibonacci sequence is to reflect the inherent uncertainty in estimating larger items
Each team member votes on the estimate using planning poker cards, which typically represent the Fibonacci scale of numbers.
Estimating Methods and Planning Poker

“Planning poker combines expert opinion, analogy and disaggregation into an enjoyable approach to estimating that results in quick but reliable efforts.”

Source: Agile Estimating and Planning by Mike Cohn
PRIORITIZATION
What’s important now
MOSCOW PRIORITIZATION
MOSCOW

**Must Have**
*fundamental to system*

**Should Have**
*important to system*

**Could Have**
*can do without in the short term*

**Would Like To Have**
*but will have to wait until later*
KANO Analysis

Must Haves
Baseline features

Satisfiers
Value added, the more of these features, the more satisfied the customer is

Delighters
Exciting unexpected features

Indifferent
Perception is that these features are not value-added
PARETO ANALYSIS
Pareto Analysis

- 80/20 rule
- The target can be to prioritize to deliver 80% of the highest value with 20% effort.
RELEASE PLANNING

Are we there yet?
Release Planning Basics

- High level product vision
- Business objectives
- A ranked product backlog
- Known velocity
- Team capabilities
The Release Plan

- Iteration length
- Estimated velocity
- Stories mapped to iterations
- Risks/Issues
- Commitment
- **High level** estimate of delivery date
ITERATION PLANNING

Sprint Planning
Planning the Sprint

• Determine Sprint length
  • 2 to 4 weeks
  • Typically, the shorter the iteration the better

• Determine the team’s capacity

• Review work completed last Sprint
Planning the Sprint, con’t

• Review work moved to current Sprint
• Review priorities
• Discuss User Stories, break into tasks
• People sign up for tasks
• Estimation
• Team commits to Sprint
Velocity

• A measurement of how many story points the team spends in an iteration

• Can also be measured in hours or other increments

• No partial points

• Not a measure of performance
Iteration 0

- Important first step
- Understand/agree to release plan
- Set up work environment
- Remove impediments for release
- Review team practices
Backlog Grooming

• Product Owner and Team
• Adds and removes stories
• Reprioritizing of stories

Ken Schwaber, one of the originators of Scrum, advises teams to dedicate five percent of every sprint to this activity.
SCRUM ROLES
Scrum Master

- Keeps an eye on the team, shields it from outside disturbances
- Removes obstacles to progress
- Protects the Sprint
The Delivery Team

Developers, Business Analysts, UX, QA team, etc.

“Fully committed”

Getting the work done!
Daily Scrum

1. What did you do yesterday?
2. What do you plan to do today?
3. Do you have any obstacles?
Obstacles

- The team needs to raise obstacles as soon as they occur and they are unable to resolve them
- 24 hour rule
- The Scrum Master should take on responsibility to clear the obstacle if the team member cannot
- Escalation can continue if needed

Obstacles are also called Impediments
SPRINT REVIEW
Show and Tell
Sprint Review Meeting

• Done on equipment where software was developed and tested – no PowerPoint

• Presented by Scrum team to Product Owner and customers/users

• Basis for planning the next Sprint

• Must represent potentially shippable increment of product functionality
SPRINT RETROSPECTIVE
Sprint Retrospective

• The Good, The Bad and The Ugly
• Lessons Learned
• What worked? What didn’t?
• What do we want to change for the next Sprint?
• Keep getting better!
Helpful Sites

Websites/Groups/Tools:

- http://www.pmi.org
- http://www.rallydev.com/
- http://www.mountaingoatsoftware.com/

LinkedIn groups:

- PMI® Agile Certified Practitioner PMI-ACP Exam Prep Study Group

Especially helpful for practice questions:

- www.agileexams.com

- Note: There is a cost to register at this site for the practice questions, but I found the cost to be well worth it.
PMI-ACP training

Monday, Oct. 8 – Wednesday, Oct. 10th
Aloft Hotel Ballantyne

Register at:

www.leanagiletraining.com

21 PDUs, 100% Pass Guarantee

Discounts available for PMI and Agile groups, early bird discount
THANK YOU