PMI-ACP Exam Prep Seminar

About This Seminar: Passing the PMI-ACP Exam!

Updated 6.8.2016

Goals of this Course

- Pass the PMI-ACP examination
- Learn about Agile project management
- Practice questions and review
- Pass the PMI-ACP examination

Contact Hours for the PMI-ACP

- 21 contact hours for the exam
- 21 PDUs if you're a PMP or other PMI credential holder

Who's Teaching the Course?

- Joseph Phillips
  - PMI-ACP
  - PMP
  - Project+
  - Certified Technical Trainer
- Author of several books, including:
  - *The PMP Study Guide*
  - *The PMP Lab Manual*
  - *The CAPM/PMP All-in-One Book*
  - *Project Management for Small Business*
  - *Certified Technical Trainer+ All-in-One Book*
  - *Vampire Management: Why Your Job Sucks*
  - *The Lifelong Project*
  - *IT Project Management: On Track from Start to Finish*
  - *Software Project Management for Dummies*

Will PMI Accept this Course?

- Yes!
- PMI Registered Education Provider #4082
- Instructing.com, LLC
- ccrs.pmi.org – PMIACP416

How to Claim the PDUs

- Quick demonstration
- Ccrs.pmi.org
- PMIACP416

Certificate of Completion

- Udemy certificate of course completion
- Instructing.com certificate of completion
  - If you're audited, use this certificate

What's in this course?

- Agile principles and mindset
- Value-driven delivery
- Stakeholder engagement
- Team performance
- Adaptive planning
- Problems, issues, detection and resolution
- Continuous improvement

Practice Tests

- 10-question quiz for each exam module
- 50-question final exam for the course
- PDF document format
- Take each exam until you score 100 percent

Introducing the PMI-ACP Exam

What Must You Know to Pass the Exam?

About the PMI-ACP Exam

- Agile approaches – Scrum, XP, Lean, Kanban
- Test-driven development (TDD)
- Collaboration skills
- PMI's fastest growing certification

Exam Prerequisites

- 2,000 hours of general project experience
  - PMPs and PgMPs qualify for this requirement
- 1,500 hours working on agile project team or agile methodologies
- 21 contact hours of training in agile practices
  - Yes, this course satisfies this objective

What's the Cost?

- PMI Member: $435
- Non-member: $495
- Cost to join PMI: $129 plus $10 application fee

Complete the Application Process

- 90 days to complete the application once started
- PMI reviews your application (up to 10 days)
- Pay for the exam
- Potential audit – 90 days max to complete
- Exam qualification – 1 year to complete exam
- Certification cycle
  - 3-year cycle
  - 30 PDUs needed on agile

Random Audits

- You'll mail to PMI:
  - Copies of your diploma/global equivalent
  - Signatures from your supervisor(s) or manager(s) from the project(s) recorded in the experience verification section of the application
  - Copies of certificates and/or letters of registration, from the training institute(s) for each course recorded on the application to meet the required contact hours of training in agile practices

PMI-ACP Exam Details

- 100 scored test questions
- 20 unscored test questions
- 120 total questions
- Three hours to complete the exam
- Pre-test tutorial/post-test survey

PMI-ACP Exam Blueprint

- Domain I. Agile Principles and Mindset 16%
- Domain II. Value-driven Delivery 20%
- Domain III. Stakeholder Engagement 17%
- Domain IV. Team Performance 16%
- Domain V. Adaptive Planning 12%
- Domain VI. Problem Detection and Resolution 10%
- Domain VII. Continuous Improvement 9%

Testing Center Details

- Computer-based test
- Paper-based test
  - At least 300 km from a Prometric CBT site
- PBTs must have at least 10 participants
- Schedule your exam ASAP!

How to Schedule the CBT Exam

- Prometric website (www.prometric.com/pmi).
- Select “Schedule an Appointment.”
- Unique PMI Eligibility and the first four letters of your last name
- Search for testing sites in your area
- Available dates will appear in blue on the calendar, and dates with no appointments available will be in grey.
- Confirm your contact information and provide a valid email address
- Your examination confirmation, along with your 16 digit unique confirmation number, will be displayed.

Rescheduling/Cancelling an Exam

- Within 30 days: $70
- Within 2 days: Forfeit fee
- No-show is a forfeiture
- Emergencies: PMI will determine if the emergency qualifies
  - Work emergencies do not qualify as an emergency

Testing Center Details

- Valid government-issued ID
  - English characters/translation
  - Your photograph
  - No photo? Need a secondary ID with your name and photo
- Your name must match the PMI-used name
- Not accepted:
  - Social security cards
  - Library cards

Testing Center Process

- Arrive 30-minutes before exam start time
- You'll be assigned a locker for your belongings
- Nothing can go into the testing center with you
- Sweaters and coats can be worn, but not removed
- The Testing Center Administrator will ask you to empty your pockets and roll up your sleeves, and will scan you with a metal-detecting wand.

Testing Center Process

- Booklet of scratch paper and two pencils, or
- Dry-erase board and marker
- You don't get to choose
- Calculator (also in exam software)
- Ear plugs or head phones available

Final Testing Center Details

- Can take a break
- Cannot access locker
- No pause in the exam timer
- You can get booted for:
  - Disruptions
  - Trying to cheat
  - Trying to keep scrap paper (even tearing)
  - Eating or drinking in testing room
  - Attempting to tamper with computer
  - Being a moron

Examination Report

- CBTs report the Pass/Fail immediately after survey
- PMI defines the levels of "proficiency" as follows:
  - Proficient – indicates performance is above the average level of knowledge in this domain.
  - Moderately Proficient – indicates performance that is at the average level of knowledge in this domain.
  - Below Proficient – indicates performance is below the average level of knowledge in this domain.
- PMI doesn't say what the passing score is

How This Course is Structured

Preparing you to pass – not just take – the PMI-ACP exam

Passing the Exam is Our Goal

- First goal of the course is you passing the exam
- Second goal, is build a deep understanding of agile methodologies
- This is not a course on:
  - How to apply agile project management
  - How to be a good project manager
  - Me helping you do your job

Course Structure Details

- Watch the lectures
- Complete the section exams
- Write down the key terms
- Repeat the section exams until you can score 100%
- Complete the assessment exam after all the module exams

Resources from this course

- PDF of the course slides
- PDF of the course exam materials
- Ask questions about the course/exam content
- Nope:
  - Downloadable lectures in video or MP3
  - PowerPoint slides
  - Me helping you manage a project

Final Lecture is the Certificate

- The last lecture of this course is a PDF certificate
- Add your name and completion date

PMI-ACP Examination Details

What's on the exam?

Domains and Tasks

- Domains are categories of knowledge
- Tasks are activities within that domain

Agile Principles and Mindset

- Nine tasks
- Explore, embrace, and apply agile principles and the agile mindset within the context of the project team and organization
- 16 percent of exam; roughly 19 questions

Agile Principles and Mindset

- Advocate for agile principles by modeling those principles and discussing agile values in order to develop a shared mindset across the team as well as between the customer and the team.
- Help ensure that everyone has a common understanding of the values and principles of agile and a common knowledge around the agile practices and terminology being used in order to work effectively.

Agile Principles and Mindset

- Support change at the system or organization level by educating the organization and influencing processes, behaviors, and people in order to make the organization more effective and efficient.
- Practice visualization by maintaining highly visible information radiators showing real progress and real team performance in order to enhance transparency and trust.

Agile Principles and Mindset

- Contribute to a safe and trustful team environment by allowing everyone to experiment and make mistakes so that each can learn and continuously improve the way he or she works.
- Enhance creativity by experimenting with new techniques and process ideas in order to discover more efficient and effective ways of working.

Agile Principles and Mindset

- Encourage team members to share knowledge by collaborating and working together in order to lower risks around knowledge silos and reduce bottlenecks.
- Encourage emergent leadership within the team by establishing a safe and respectful environment in which new approaches can be tried in order to make improvements and foster self-organization and empowerment.

Agile Principles and Mindset

- Practice servant leadership by supporting and encouraging others in their endeavors so that they can perform at their highest level and continue to improve.

Agile Principles and Mindset

- Nine tasks
- Explore, embrace, and apply agile principles and mindset within the context of the project team and organization
- 16 percent of exam; roughly 19 questions

Value-Driven Delivery

- Four subdomains; 14 tasks
- 20 percent of exam; 24 questions
- Deliver valuable results by producing high-value increments for review, early and often, based on stakeholder priorities. Have the stakeholders provide feedback on these increments, and use this feedback to prioritize and improve future increments.

Value-Driven Delivery

- Define Positive Value
  - Define deliverables by identifying units that can be produced incrementally in order to maximize their value to stakeholders while minimizing non-value added work.
  - Refine requirements by gaining consensus on the acceptance criteria for features on a just-in-time basis in order to deliver value.
  - Select and tailor the team's process based on project and organizational characteristics as well as team experience in order to optimize value delivery.

Value-Driven Delivery

- Avoid Potential Downsides
  - Plan for small releasable increments by organizing requirements into minimally marketable features/minimally viable products in order to allow for the early recognition and delivery of value.
  - Limit increment size and increase review frequency with appropriate stakeholders in order to identify and respond to risks early on and at minimal cost.
  - Solicit customer and user feedback by reviewing increments often in order to confirm and enhance business value

Value-Driven Delivery

- Prioritization
  - Prioritize the units of work through collaboration with stakeholders in order to optimize the value of the deliverables.
  - Perform frequent review and maintenance of the work results by prioritizing and maintaining internal quality in order to reduce the overall cost of incremental development.
  - Continuously identify and prioritize the environmental, operational, and infrastructure factors in order to improve the quality and value of the deliverables.

Value-Driven Delivery

- Incremental Development
  - Conduct operational reviews and/or periodic checkpoints with stakeholders in order to obtain feedback and corrections to the work in progress and planned work.
  - Balance development of deliverable units and risk reduction efforts by incorporating both value producing and risk reducing work into the backlog in order to maximize the total value proposition over time.
  - Re-prioritize requirements periodically in order to reflect changes in the environment and stakeholder needs or preferences in order to maximize the value.
  - Elicit and prioritize relevant non-functional requirements (such as operations and security) by considering the environment in which the solution will be used in order to minimize the probability of failure.
  - Conduct frequent reviews of work products by performing inspections, reviews, and/or testing in order to identify and incorporate improvements into the overall process and product/service.

Stakeholder Engagement

- 9 tasks
- 3 subdomains
- 17 percent of exam; 20 questions
- Engage current and future interested parties by building a trusting environment that aligns their needs and expectations and balances their requests with an understanding of the cost/effort involved. Promote participation and collaboration throughout the project life cycle and provide the tools for effective and informed decision making.

Stakeholder Engagement

- Understand Stakeholder Needs
  - Identify and engage effective and empowered business stakeholder(s) through periodic reviews in order to ensure that the team is knowledgeable about stakeholders' interests, needs, and expectations.
  - Identify and engage all stakeholders (current and future) by promoting knowledge sharing early and throughout the project to ensure the unimpeded flow of information and value throughout the lifespan of the project.

Stakeholder Engagement

- Ensure Stakeholder Involvement
  - Establish stakeholder relationships by forming a working agreement among key stakeholders in order to promote participation and effective collaboration.
  - Maintain proper stakeholder involvement by continually assessing changes in the project and organization in order to ensure that new stakeholders are appropriately engaged.
  - Establish collaborative behaviors among the members of the organization by fostering group decision making and conflict resolution in order to improve decision quality and reduce the time required to make decisions.

Stakeholder Engagement

- Manage Stakeholder Expectations
  - Establish a shared vision of the various project increments (products, deliverables, releases, iterations) by developing a high level vision and supporting objectives in order to align stakeholders' expectations and build trust.
  - Establish and maintain a shared understanding of success criteria, deliverables, and acceptable trade-offs by facilitating awareness among stakeholders in order to align expectations and build trust.
  - Provide transparency regarding work status by communicating team progress, work quality, impediments, and risks in order to help the primary stakeholders make informed decisions.
  - Provide forecasts at a level of detail that balances the need for certainty and the benefits of adaptability in order to allow stakeholders to plan effectively.

Team Performance

- 9 tasks
- 3 subdomains
- 16 percent of exam; 19 questions
- Create an environment of trust, learning, collaboration, and conflict resolution that promotes team self-organization, enhances relationships among team members, and cultivates a culture of high performance.

Team Performance

- Team Formation
  - Cooperate with the other team members to devise ground rules and internal processes in order to foster team coherence and strengthen team members' commitment to shared outcomes.
  - Help create a team that has the interpersonal and technical skills needed to achieve all known project objectives in order to create business value with minimal delay.

Team Performance

- Team Empowerment
  - Encourage team members to become generalizing specialists in order to reduce team size and bottlenecks, and to create a high performing cross-functional team.
  - Contribute to self-organizing the work by empowering others and encouraging emerging leadership in order to produce effective solutions and manage complexity.
  - Continuously discover team and personal motivators and demotivators in order to ensure that team morale is high and team members are motivated and productive throughout the project.

Team Performance

- Team Collaboration and Commitment
  - Facilitate close communication within the team and with appropriate external stakeholders through co-location or the use of collaboration tools in order to reduce miscommunication and rework.
  - Reduce distractions in order to establish a predictable outcome and optimize the value delivered.
  - Participate in aligning project and team goals by sharing project vision in order to ensure the team understands how their objectives fit into the overall goals of the project.
  - Encourage the team to measure its velocity by tracking and measuring actual performance in previous iterations or releases in order for members to gain a better understanding of their capacity and create more accurate forecasts.

Adaptive Planning

- 10 tasks
- 3 subdomains
- 12 percent of exam; 14 questions
- Produce and maintain an evolving plan, from initiation to closure, based on goals, values, risks, constraints, stakeholder feedback, and review findings.

Adaptive Planning

- Levels of Planning
  - Plan at multiple levels (strategic, release, iteration, daily) creating appropriate detail by using rolling wave planning and progressive elaboration to balance predictability of outcomes with ability to exploit opportunities.
  - Make planning activities visible and transparent by encouraging participation of key stakeholders and publishing planning results in order to increase commitment level and reduce uncertainty.
  - As the project unfolds, set and manage stakeholder expectations by making increasingly specific levels of commitments in order to ensure common understanding of the expected deliverables.

Adaptive Planning

- Adaptation
  - Adapt the cadence and the planning process based on results of periodic retrospectives about characteristics and/or the size/complexity/criticality of the project deliverables in order to maximize the value.
  - Inspect and adapt the project plan to reflect changes in requirements, schedule, budget, and shifting priorities based on team learning, delivery experience, stakeholder feedback, and defects in order to maximize business value delivered.

Adaptive Planning

- Agile Sizing and Estimation
  - Size items by using progressive elaboration techniques in order to determine likely project size independent of team velocity and external variables.
  - Adjust capacity by incorporating maintenance and operations demands and other factors in order to create or update the range estimate.
  - Create initial scope, schedule, and cost range estimates that reflect current high level understanding of the effort necessary to deliver the project in order to develop a starting point for managing the project.
  - Refine scope, schedule, and cost range estimates that reflect the latest understanding of the effort necessary to deliver the project in order to manage the project.
  - Continuously use data from changes in resource capacity, project size, and velocity metrics in order to evaluate the estimate to complete.

Problem Detection and Resolution

- 5 tasks
- 10 percent of exam; 12 questions
- Continuously identify problems, impediments, and risks; prioritize and resolve in a timely manner; monitor and communicate the problem resolution status; and implement process improvements to prevent them from occurring again.

Problem Detection and Resolution

- Create an open and safe environment by encouraging conversation and experimentation, in order to surface problems and impediments that are slowing the team down or preventing its ability to deliver value.
- Identify threats and issues by educating and engaging the team at various points in the project in order to resolve them at the appropriate time and improve processes that caused issues.
- Ensure issues are resolved by appropriate team members and/or reset expectations in light of issues that cannot be resolved in order to maximize the value delivered.

Problem Detection and Resolution

- Maintain a visible, monitored, and prioritized list of threats and issues in order to elevate accountability, encourage action, and track ownership and resolution status.
- Communicate status of threats and issues by maintaining threat list and incorporating activities into backlog of work in order to provide transparency.

Continuous Improvement

- 6 tasks
- 9 percent of exam; 11 questions
- Continuously improve the quality, effectiveness, and value of the product, the process, and the team.

Continuous Improvement (Product, Process, People)

- Tailor and adapt the project process by periodically reviewing and integrating team practices, organizational culture, and delivery goals in order to ensure team effectiveness within established organizational guidelines and norms.
- Improve team processes by conducting frequent retrospectives and improvement experiments in order to continually enhance the effectiveness of the team, project, and organization.
- Seek feedback on the product by incremental delivery and frequent demonstrations in order to improve the value of the product.

Continuous Improvement (Product, Process, People)

- Create an environment of continued learning by providing opportunities for people to develop their skills in order to develop a more productive team of generalizing specialists.
- Challenge existing process elements by performing a value stream analysis and removing waste in order to increase individual efficiency and team effectiveness.
- Create systemic improvements by disseminating knowledge and practices across projects and organizational boundaries in order to avoid re-occurrence of identified problems and improve the effectiveness of the organization as a whole.

PMI-ACP Exam Blueprint

- Domain I. Agile Principles and Mindset 16%
- Domain II. Value-driven Delivery 20%
- Domain III. Stakeholder Engagement 17%
- Domain IV. Team Performance 16%
- Domain V. Adaptive Planning 12%
- Domain VI. Problem Detection and Resolution 10%
- Domain VII. Continuous Improvement 9%

PMI Agile Certified Practitioner

Official Examination Content Outline

Official Exam Objectives

- https://www.pmi.org/certification/agile-management-acp.aspx

PMI-ACP Handbook

- https://www.pmi.org/~/media/PDF/Certifications/handbooks/agile-certified-practitioner-handbook-acp.ashx

PMI Exam Guidance

- https://www.pmi.org/certification/agile-management-acp/pmi-acp-exam-prep.aspx

PMI-ACP Reference list

- https://www.pmi.org/~/media/PDF/Certifications/ACP_Reference_list_v2.ashx

Agile Principles and Mindset

Overview of Domain 1: PMI-ACP

Agile Principles and Mindset

- Think agile and value for the exam
- 16 percent of the exam content
- 19 questions
- Agile fundamentals
- Values for agile projects
- Agile principles and methodologies
- Serving as an agile leader

Agile Principles and Mindset Tasks

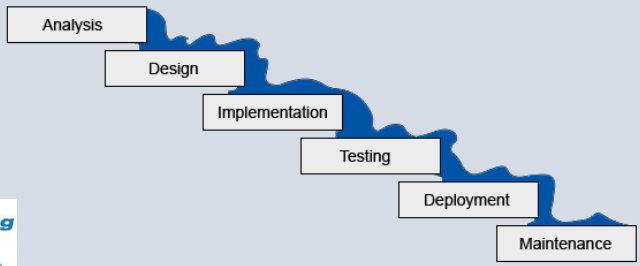
- Advocate for agile principles and values in the organization
- Ensure common understanding of agile principles
- Educate and influence agile
- Transparency equates to trust
- Safe environment for experimenting

Agile Principles and Mindset Tasks

- Experiment with new techniques and processes
- Share knowledge new collaboration
- Emergent leadership
- Practice servant leadership

Utilizing agile

- Why use agile?
- Different types of projects use different approaches
- Predictive project plan up front
- Knowledge work has many unknowns



Knowledge work projects are special

- Industrial work requires up-front planning
- Knowledge work expects change
- Knowledge work is invisible work
- Agile is best suited for software development projects

Contrasting industrial and knowledge work projects

Industrial projects
- Visible
- Stable
- Running things
- Structure
- Correct answers
- Task driven
- Command and control
- Standards
- Performance measurement
- Cost of workers for a task

Knowledge work projects
- Invisible
- Lots of changes
- Changing environment
- Less structure
- Lots of questions
- Value-driven
- Autonomy driven
- Innovation
- Learning and teaching
- Workers are an asset not a cost

Defined processes vs. empirical processes

- Industrial work relies on defined processes
- Knowledge work relies on empirical processes
- A defined process defines all steps in advance
- Empirical processes are interactive, incremental, change often, adapt, and pass through the reviews
- Empirical processes are change-driven

The Agile Mindset

- The exam expects you to have an agile mind set
- Can you be agile or just do a job?
- An agile mind set is based on agile values and principles
- An agile mindset is different than just recalling agile terms

Declaration of Interdependence

- We increase return on investment by making continuous flow of value our focus
- We deliver reliable results by engaging customers infrequent interactions and shared ownership
- We expect uncertainty and manage for it to iterations anticipation and adaption

Declaration of Interdependence

- We unleashed creativity and innovation by recognizing that individuals are the ultimate source of value and creating an environment where they can make a difference
- We boost performance through group accountability for results and shared responsibility for team effectiveness
- We improve effectiveness and reliability through situationally specific strategies processes and practices

Exam Tip

- You must be agile not only know agile
- Embrace empirical processes
- Knowledge work is invisible work
- Communication and collaboration is paramount

Creating an Agile Mind Set

Having an agile mind set is key to exam success

Summing up Agile Core Principles

- Welcome and expect change
- Create small value-added increments
- Utilize feedback loops
- Constantly learn through discovery and creativity
- Focus on developing a value for stakeholders
- It's okay to fail fast
- Deliver value throughout the project
- Constantly improve upon the project and the project work

Being Agile Versus Doing Agile

Being agile

- Possessing an agile mind set
- Choose correct practices
- Implement correct practices
- Tailor agile processes

Doing agile

- Doing a job without embracing agile
- Forcing agile practice
- Command and control
- Understand agile

Influencing Organizational Agility

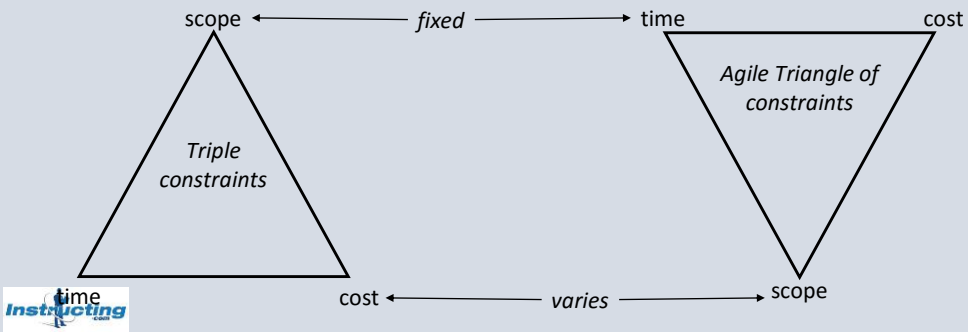
- An individual with an agile mind set will feel frustrated if the remainder of the organization doesn't embrace agile
- A single team using agile may feel isolated if the rest of the organization doesn't understand agile practices
- An entire organization adapting agile help everyone to work together and improve the organizational agility

Creating Organizational Agility

- Think and embrace agile as an individual
- Doing an understanding agile helps us to begin to influence others to embrace agile
- Influence others to embrace agile through education and demonstration

Inverted Triangle Model

- Time cost scope
- What's a variable and what is fixed



Exam Tip

- Don't cram for the PMI – ACP exam
- Embrace the mindset of agile
- Understand agile from a leaders perspective

The Agile Manifesto

Created by the Agile Alliance

Four Values of the Agile Manifesto

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

Individuals And Interactions Over Processes and Tools

- Individuals and interactions are most important
- Processes and tools will be needed on projects
- Projects are completed by people not processes and tools
- Agile projects are people driven

Working Software Over Comprehensive Documentation

- Agile project need to deliver value
- Value is about the purpose or business need the project aims to deliver
- Documentation is barely sufficient
- Documentation is done just in time – as the last responsible moment
- Documentation might also be just because
  - Industry requirements
  - Organizational requirements

Customer Collaboration Over Contract Negotiation

- Agile is flexible, accommodating, and willing to change
- Contracts are often rigid and uncooperative
- Agile contracts must accommodate change
- There's a difference between being right and doing the right thing

Responding to Change Over Following A Plan

- Agile welcomes change
- Predictive projects plan everything in advance
- Agile projects have lots and lots of many changes
- Agile projects have uncertainty up front

12 Principles Behind The Agile Manifesto

- 12 guiding principles
- These support the Agile Manifesto
- For your PMI-ACP exam be familiar, but no need to memorize these 12 guiding principles

Principle One

- Our highest priority is to satisfy the customers through early and continuous delivery of valuable software

Principle Two

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

Principle Three

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

Principle Four

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

Principle Five

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

Principle Six

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

Principle Seven

- Working software is the primary measure of progress.

Principle Eight

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

Principle Nine

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

Principle Ten

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

Principle 11

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

Principle 12

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

Four Values of the Agile Manifesto

- **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 Methodologies: Scrum

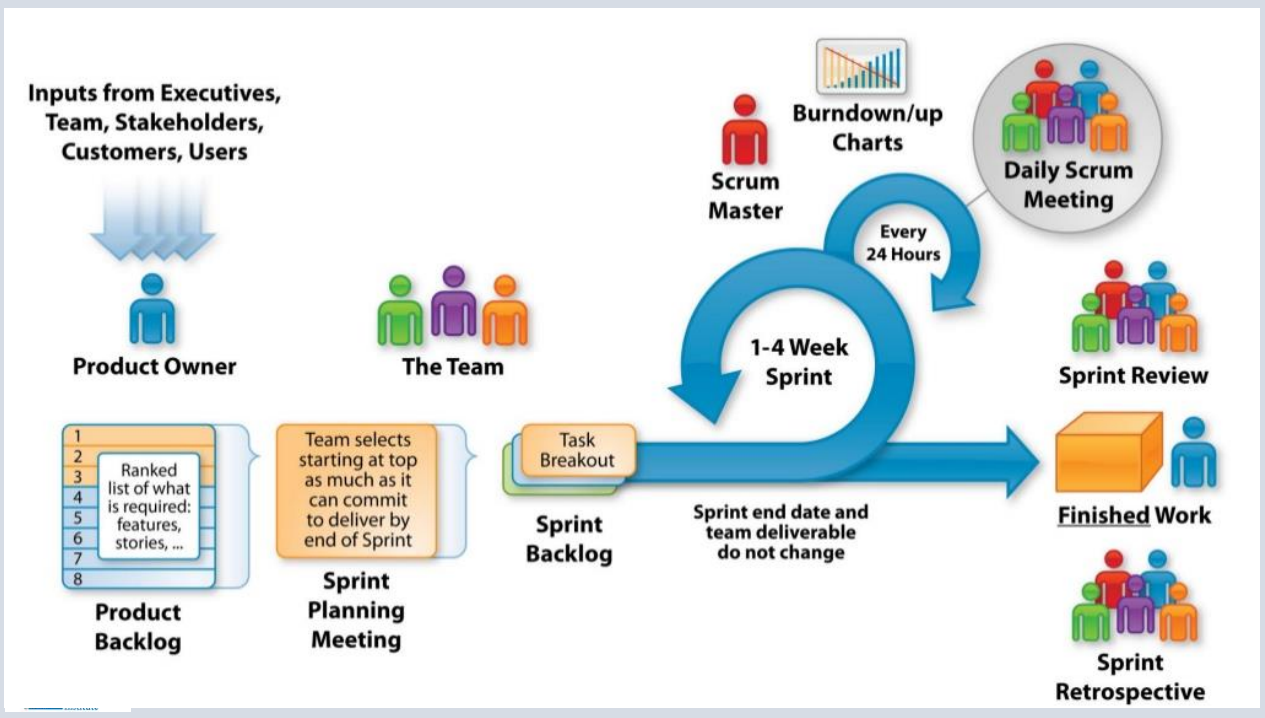
Examining the Agile Methodologies for the PMI-ACP exam

Reviewing Scrum

- Scrum is easy to understand, but can be difficult to master
- Scrum is a rugby term
- Scrum uses a methodology called the scrum framework
- The scrum framework is a set of practices, roles and responsibilities, events, artifacts, and rules

Scrum Pillars and Values

- Transparency – a common understanding of what done means in a scrum project
- Inspection – reviewing the project to determine the completeness of the project; finding root cause of variances from the project goals
- Adaptation – making adjustments to the scrum process to mitigate problems or bad trends



Introducing Scrum Sprints

- Timebox iteration for project work
- A timebox is a predetermined duration
- Scrum sprints are between one and four weeks in duration
- During a sprint no changes are made that would affect the goal of the sprint
- A sprint can be cancelled if they change in the project goals make the sprint goals obsolete
- Only the product owner may cancel a sprint

More About Scrum Sprints

- If a sprint is cancelled, uncompleted backlog items are returned to the product backlog
- Within a sprint there are several activities:
  - Sprint planning meeting
  - Development
  - Daily scrubs
  - Sprint review meeting
  - Sprint retrospective meeting

Introducing Scrum Team Roles

- Scrum Master – responsible for communicating the scrum methodology and ensuring the methodology is used effectively
- Product owner – prioritizes the product backlog to ensure value from each sprint
- Development team – the software developers who create the product through the sprint

Looking At The Scrum Activities

- Scrum activities are also known as events or ceremonies
- There are five scrum ceremonies:
  - Product backlog refinement
  - Sprint planning meetings
  - Daily scrum
  - Sprint reviews
  - Sprint retrospective

Grooming the Backlog

- The product owner owns the backlog
- Backlog refinement is the prioritization backlog items
- The entire project team may participate in the backlog grooming

Sprint Planning Meeting

- Project team needs to discuss the goals of the upcoming sprint
- Team discusses how the work will be accomplished
- Product owner reviews with the team items in the updated backlog
- Development team defines how the work will be done in the goals of the sprint will be achieved
- The development team is self-organized

Participating In a Daily Scrum

- The daily scrum is also known as a stand-up meeting
- This is a 15-minute timeboxed meeting
- The daily scrum is held every day at the same time and location
- The daily scrum is for the development team only

Daily Scrum Meeting Questions

- What have I done since the last daily scrum?
- What do I plan to do today?
- Are there any impediments to my progress?

Working With Large Scrum Teams

- Scrum of scrums
- Scrum of scrums of scrums
- Four questions are answered:
  - What has your team done since we last met?
  - What will your team do before our next meeting?
  - Are there any roadblocks in your team's way?
  - Will your team put anything in another team's way?

Introducing A Sprint Review

- Hosted at the end of every sprint
- Attendees will be the development team, the product owner, scrum master, and sometimes other project stakeholders
- The development team will demo the work created in the increment
- The group will decide if “Done” has been achieved
- The development team and the product owner will discuss the sprint and the remaining items in the product backlog

Introducing a Sprint Retrospective

- The development team meeting posted after the sprint review, but before the next sprint planning meeting
- This is a meeting to inspect an adapt
- Lessons learned and opportunities for improvement
- Review of the product owner’s feedback about the last iteration
- An opportunity to improve on their approach based on the retrospective and the last sprint

Scrum Artifact – Product Increment

- The product increment is the outcome of an iteration
- The product increment is a chunk of the project work
- The development team and the product owner must be an agreement of what done means for an increment

Scrum Artifact – Product Backlog

- The product backlog is the source for all product requirements
- The product owner sorts and prioritizes the backlog items
- The development team always works on the most important items based on the prioritized items in the product backlog
- The backlog is always prioritized before the current sprint
- Backlog refinement is done by both the product owner and the development team working in harmony
- The team estimates their capacity to attack the items in the product backlog

Scrum Artifacts – Sprint Backlog

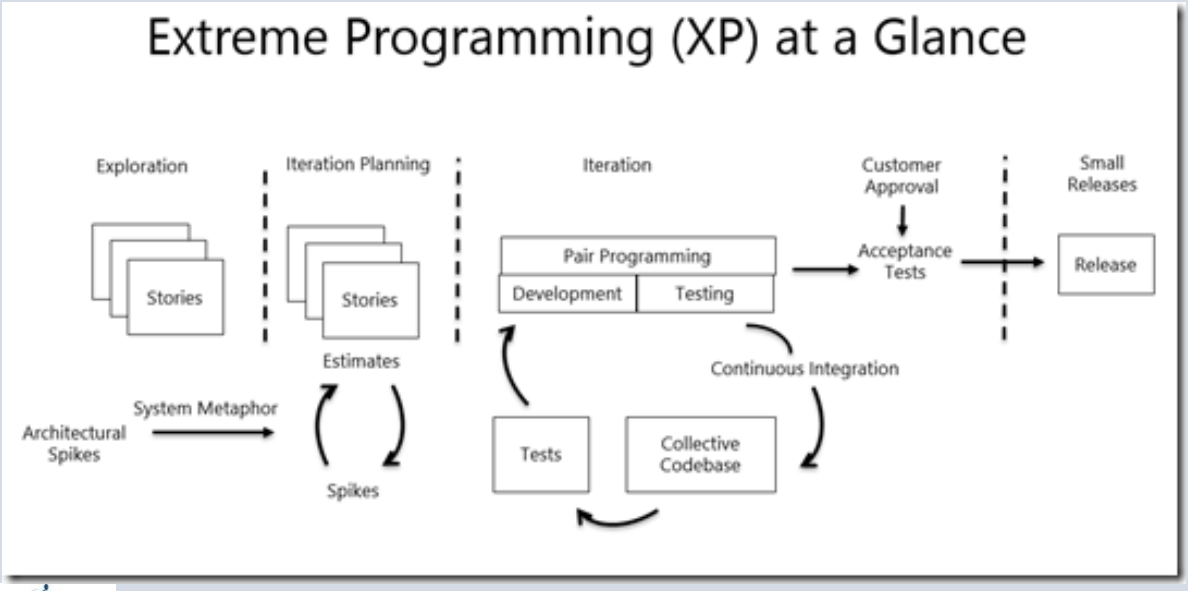
- Like the product backlog the sprint backlog is a prioritization of the product
- The sprint backlog is a subset of the product backlog
- The sprint backlog serves as the goal for the current iteration
- The sprint backlog is a view into the work to be accomplished in the current sprint
- The sprint backlog is updated and refined by the development team

Agile Methodologies: XP

Examining the Agile Methodologies for the PMI-ACP exam

eXtreme Programming

- Extreme programming is also known as XP
- XP is all about software development best practices
- The PMI-ACP exam will lightly test your knowledge of XP



XP Core Values

- Simplicity – reducing complexity, extra features, and waste
- Find the simplest thing that could possibly work

Extreme Programming Core Values

- Communication
  - Ensuring that the project team knows what is expected of them
  - Ensuring the project team knows what other people are working on
  - The daily standup meeting is an excellent communication tool

Extreme Programming Core Values

- Feedback:
  - The development team needs feedback early in the project
  - Failing fast is a way to get feedback early
  - Feedback gives the team an opportunity to improve the project

Extreme Programming Core Values

- Courage
  - Developers' work is entirely visible to others on the project team
  - Team members share code and correct each other's code
  - XP uses pair programming

Extreme Programming Core Values

- Respect
  - Team members must respect one another
  - Everyone is responsible for the success and or failure of the project
  - Everyone works differently but must work together

XP Team Roles

- Coach – mentor/guide/facilitator/communicator similar to the ScrumMaster
- Customer – the individual who provides requirements priorities and direction for the project similar to the product owner
- Programmer – the developers who write the code
- Testers – Define and write the acceptability test

XP Core Practices – Whole Team

- XP team members are collocated
- Generalizing specialist not role specialist
- Efficient and sharing of information

XP Core Practices – Planning Games

- Planning games are just planning activities
- Release planning is the release of new functionality
  - No more than one or two releases per year
  - The customer outlines the functionality required in the release
  - Developers estimate the difficulty to build the functionality

XP Core Practices – Planning Games

- Iteration planning is similar to sprint planning
  - Iteration planning happens at the start of every iteration
  - The customer defines what functionality they want to see by the end of the iteration
  - The development team estimates the difficulty to build the functionality

XP Core Practices – Small Releases

- Small releases to a test environment are part of the XP practices
- Increases visibility to the customer
- Helps to deploy working software to the end users

XP Core Practices – Customer Tests

- Definition of the required functionality
- Description of one or more test criteria for the software to be working

XP Core Practices – Collective Code Ownership

- Any pair of developers can improve or amend the code
- Multiple people will work on all the code
- Improve defect resolution in discovery
- Knowledge is shared not isolated

XP Core Practices – Code Standards

- A coding standard is defined
- The team adheres to the standard
- Provides for consistency in writing the code

XP Core Practices – Sustainable Pace

- Productivity is optimized through a sustainable pace
- Consistent overtime and long hours are not sustainable

XP Core Practices – Metaphor

- Metaphors and similes are used to explain designs
- Metaphors help communicate the software to the customer

XP Core Practices – Continuous Integration

- Compiling the code frequently throughout the day
- Programmers check-in code to the code repository
- Integration test run automatically for immediate feedback

XP Core Practices – Test-driven Development

- Acceptance test are written prior to developing new code
- Initial tests will fail because the code has not been fully developed yet
- When the code has been written correctly it will pass the test

XP Core Practices – Refactoring

- Cleaning up the code
- Removing duplicated code
- Lowering coupling
- Increasing cohesion

Exam Tip

- Embrace the idea of refactoring
- Cleans up the code to make it:
  - Simpler
  - Easier to understand
  - Adhere to standards
  - Future work

XP Core Practices – Simple Design

- What is the simplest thing that could work?
- Simple does not mean easy
- Simple design is a risk mitigation approach

XP Core Practices – Pair Programming

- One person writes the code while the second person reviews the code
- The two people change roles frequently
- The pair will catch mistakes and speed up productivity

Agile Methodologies: Lean

Examining the Agile Methodologies for the PMI-ACP exam

Lean Product Development

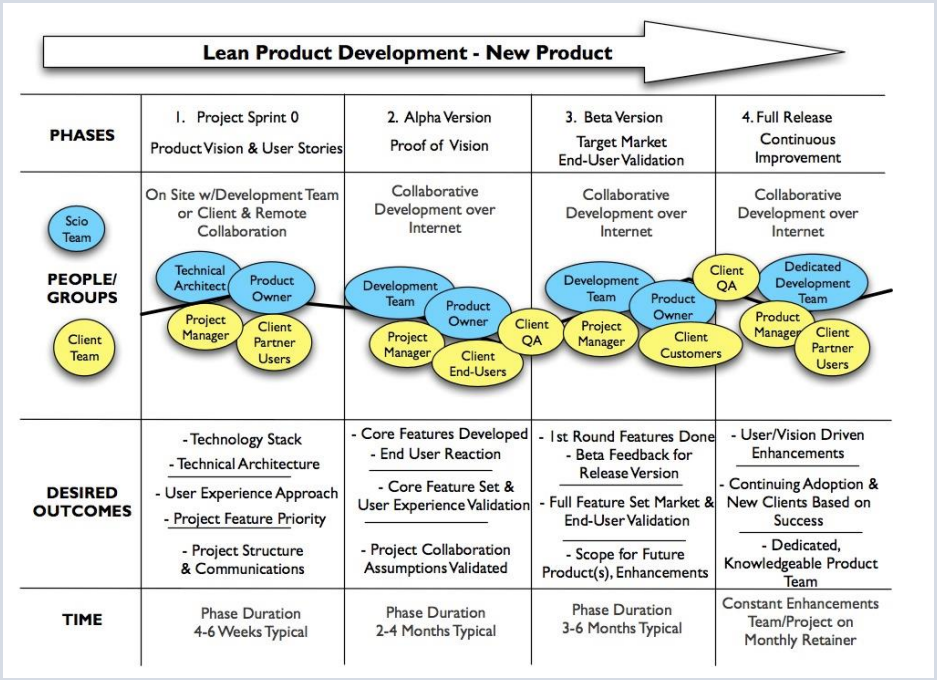
- Toyota production system
- Visual management tools
- Customer to find value
- Learning and continuous Improvement

Seven Lean Core Concepts

- Eliminate waste
- Empower the team
- Deliver fast
- Optimize the whole
- Build quality in
- Defer decisions
- Amplify learning

Seven Wastes Of Lean

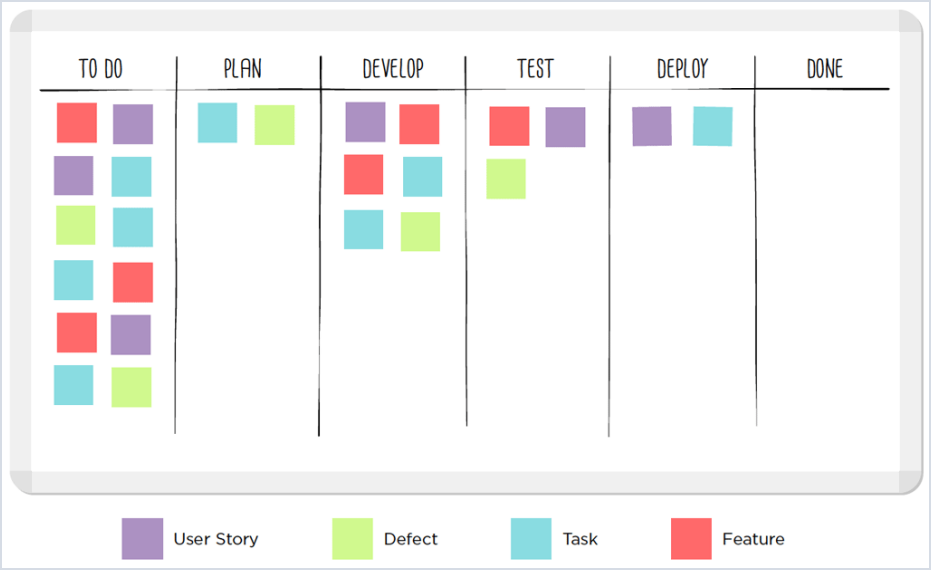
- Partially done work
- Extra processes
- Extra features
- Task switching
- Waiting
- Motion
- Defects



Introducing Kanban

- Japanese word that means sign board
- The signboard has categories of work for each stage of the production process

Kanban Board



Five Principles Of Kanban

- Visualize the workflow
- Limit work in progress
- Manage flow
- Make process policies explicit
- Improve collaboratively

Kanban Pull System

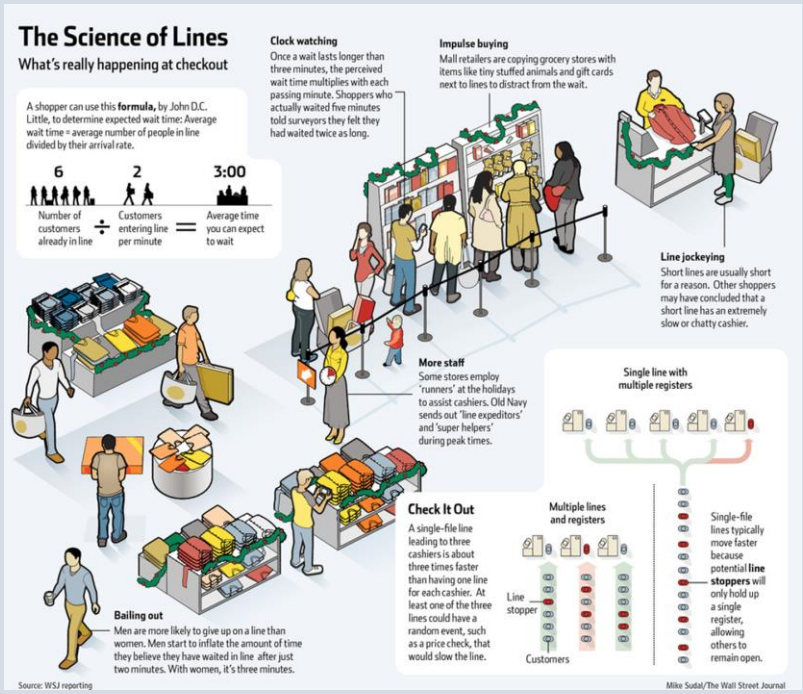
- A pull system moves work through development
- The development team completes an item; the next item in queue is pulled into the next stage of the process
- Kanban does not use timeboxed iterations
- Only so many items can be in each stage of the project
- Work moves from left to right

Exam Tip

- KanBan Boards are pull systems
- They visualize the project
- KanBan boards are sign boards
- KanBan in Japanese means “cards you can see”

Little’s Law

- The duration of a queue is proportional to its size
- By limiting the work-in-progress teams complete work faster



Agile Methodologies: Other Agile Approaches

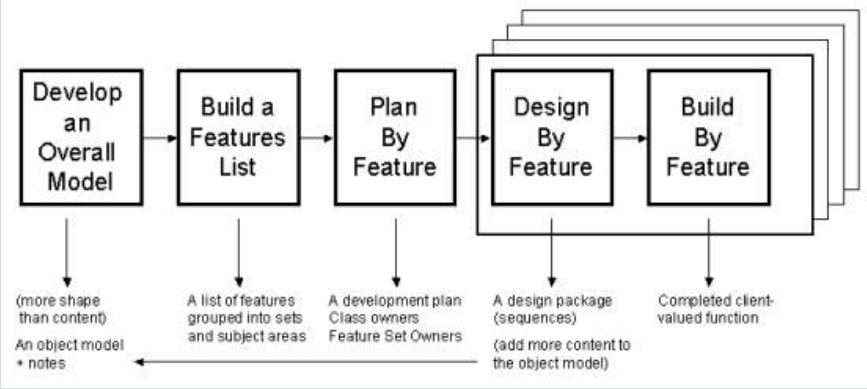
Examining the Agile Methodologies for the PMI-ACP exam

Feature-Driven Development

- The development team creates a model for the product
- They will build a feature list and a plan for the work
- The team moves through the design and build the directions for the product features
- The team designs by features and builds by features

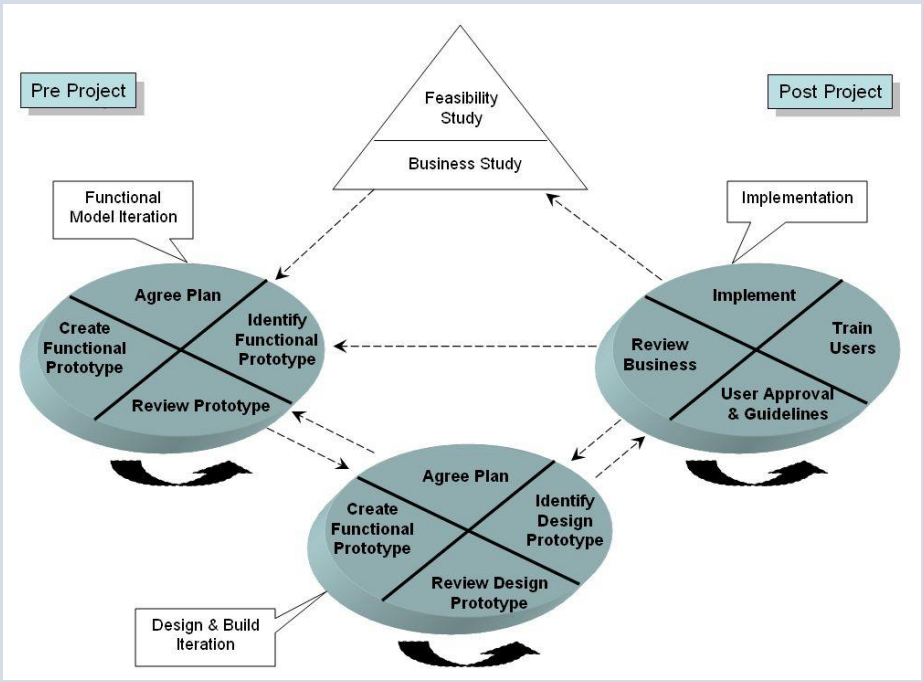
Feature-Driven Development Characteristics

- Domain object modeling
- Developing by feature
- Individual class code ownership
- Feature teams
- Inspections
- Configuration management
- Regular builds
- Visibility of progress and results



Dynamic Systems Development Method

- Focus on the business need
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control



Crystal

- Customized methodologies coded by color names
- Methodologies are appropriate for different criticalities and team sizes
- Criticality is about the impact of a product defect design

Agile in Action

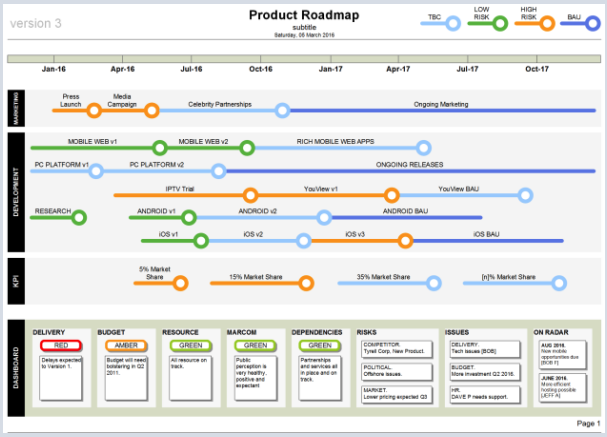
A quick look at how agile works.

Product Vision

- A product vision is created
- The product must be in alignment with the company's strategy
- This is done by the product owner

Creating the Product Roadmap

- The visualization of product features
- The product roadmap equates to the product division as a whole
- This is done and owned by the product owner



Release Planning

- The release timing for specific product functionality
- Priorities are assigned to the product features from most important to least important
- These features become the product backlog
- Owned by the product owner

Iteration Planning

- Goals are established for the current Sprint or iteration
- Goals are based on the product backlog
- Product owner in the development team work together to prioritize features
- This is done at the start of each Sprint

Daily Scrum

- The development team meets each day throughout the Sprint
  - What have you done since the last daily scrum?
  - What will you do today?
  - Are there any impediments in your way?

Sprint Review

- The development team provides a demo of the product
- Happens at the end of each Sprint
- Product owner and development team own the stage

Sprint Retrospective

- A look back at the last sprint
- Opportunity to optimize efficiency
- Opportunity to refine environment and processes

Released Product

- The product is released according to the project's release plan

Agile Leadership

Reviewing the specifics of leadership and agile projects.

Management vs. Leadership

- Management is about getting things done
- Leadership is about getting people to want to do what needs to be done
- Management is more concerned with task control command

Comparing Management and Leadership

Management

- Task and things
- Control
- Efficiency
- Speed
- Command
- Doing things right

Leadership

- People
- Empowerment
- Being effective
- Direction
- Principles
- Doing the right things

Servant Leadership

- The recognition is on the team members not the project manager or coach or ScrumMaster
- The team gets the work done
- The team adds value
- Servant leaders provide what team members need

Shield the Team From Interruptions

- Isolate and protect the team
- Keep business partners from interrupting the team design
- Ensuring people communicate to the designated channels
- Protect the team from the diversions

Remove Impediments To Progress

- Clear obstacles for the team
- Remove things that may cause delays or waste
- Consider documentation compliance activities or anything that does not directly add value
- Daily scrum give insights for the servant leader

Communicate The Project Vision

- Ensures that stakeholders have a clear image of what the team is creating
- Ensures that all stakeholders have a common vision of what done means
- Communicate and recommunicate the project Vision to reinforce the vision

Carry Food and Water

- Ensuring that the team has the resources they need to be productive
- This includes items like proper tools compensation encouragement and other resources
- Training and professional development may be included as well

12 Principles For Agile Leadership

- Learn the team members' needs
- Learn the project requirements
- Act for the simultaneous welfare of the team and the project
- Create an environment of functional accountability
- Have a vision of the completed project
- Use the project vision to drive your own behavior

12 Principles For Agile Leadership

- Serve as the central figure in successful project team development
- Recognize team conflict as a positive step
- Manage with an eye towards ethics
- Remember that ethics is not an afterthought but an integral part of our thinking
- Take time to reflect on the project
- Developed the trick of thinking backwards

Exam tip

- You *do not* need to memorize these leadership principles
- You *do* need to adapt the leadership principles and mind set

Agile Leadership Practices

- Honesty
- Forward looking
- Competent
- Inspiring

Leadership Tasks

- Give transparency through visualization
- Create a safe environment for experimentation
- Experiment with new techniques and processes
- Share knowledge through collaboration

Value-Driven Delivery

Maximizing business value through prior authorization incremental delivery testing and validation

Value-Driven Delivery for the PMI ACP Exam

- 20 percent of the exam
- Roughly 24 exam questions
- Know this topic for exam and agile success

14 Tasks for Value-driven Delivery

- Plan work incrementally
- Gain consensus on just in time acceptance criteria
- Tune process to organization team and project
- Release minimal viable product
- Work in small batches
- Review often
- Prioritize work
- Refactor code often
- Optimize environmental operational it infrastructure factors
- Review and checkpoint often
- Balance value and risk
- Reprioritize to maximize value
- Prioritize nonfunctional requirements
- Review and improve the overall process and product

Value-driven Delivery Definition

- Projects exist to create business value
- The project manager's goal is to increase value and reduce risk as early as possible
- Value-driven delivery has the most weight of all PMI ACP exam domains

Deliver Value Early in the Project

- Based on prior translation value is delivered first
- The longer a project lasts the more opportunity for risk
- By delivering high-value items early, the team demonstrates an understanding of the customers needs
- Early value help stakeholders maintain synergy an interest in the project

Minimize Waste is a Constant Goal

- Waste reduces value
- Poppendieck's Seven Areas of Waste:
  - Partially done work
  - Extra processes
  - Extra features
  - Task switching
  - Waiting
  - Motion
  - Defects

Assessing Value In Agile Projects

Value is expressed in financial terms

Exam Tip

- Formulas and calculations are unlikely in the PMI ACP exam
- Be topically familiar with the topics in this lecture

Return On Investment – ROI

- Return on investment is the profitability in a project
- Return on investment is the value of the project minus the investment in the project
- A higher return on investment means you are getting a better return then a lower return – bigger is better
- Return on investment is not the best approach to discovering business value in a project

Present Value

- The calculation of a future amount in today's terms given and assumed interest rate and inflation rate

Net Present Value

- The present value of a revenue stream over a series of time periods
- Higher net present values are good

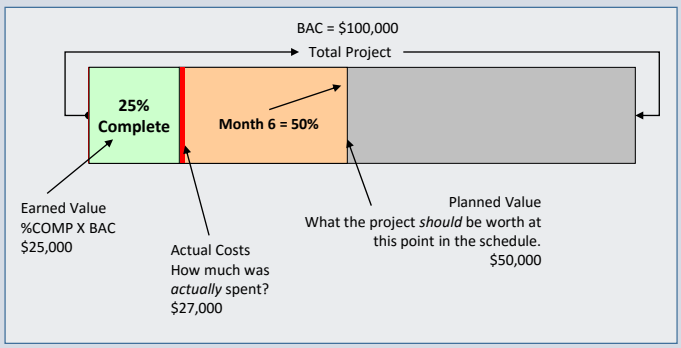
Internal Rate Of Return

- Calculates the NPV of the cost of the project and when the NPV of the project meets or exceeds the NPV of the benefits of the project
- The higher the IRR the more valuable the project is

Earned Value Management For Agile Projects

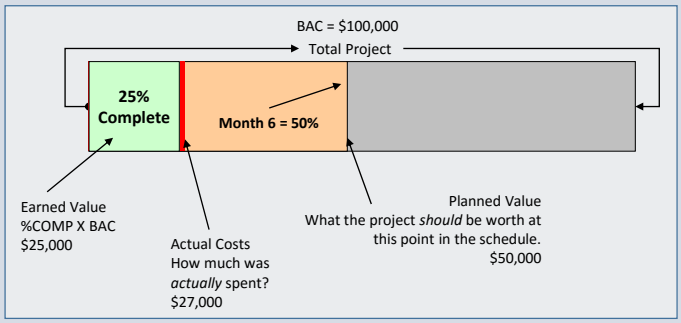
- It's unlikely you'll see earned value management on the exam
- EVM is a suite of formulas to show performance
- Earned value compared to actual performance to planned performance

EVM Foundation



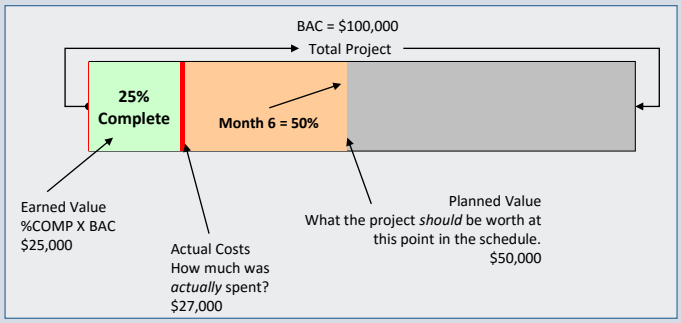
Finding the Variances

- Cost variance= EV-AC
- Schedule variance=EV-PV



Measuring Performance

CPI= EV/AC	SPI=EV/PV
CPI=25,000/27,000	SPI=25,000/50,000
CPI=.93	SPI=.50

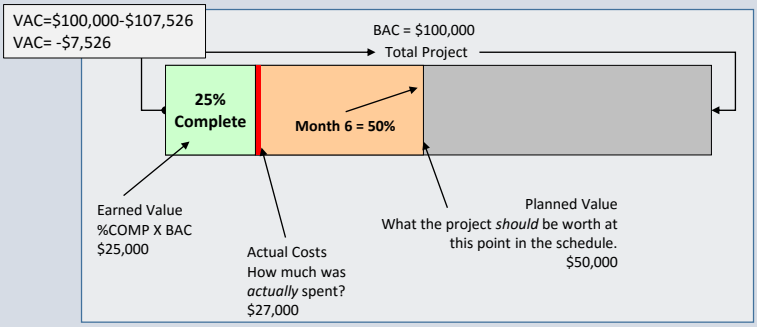


Predicting the Future

- Estimate at Completion=BAC/CPI
- Estimate to Complete=EAC-AC

EAC=$100,000/.93
EAC=$107,526

ETC=$107,526-$27,000
ETC=$80,526



EAC in Action

EAC Formulas						
CPI will remain the same	BAC	CPI	Result			
BAC/CPI	1,000,000	0.89	$1,125,000.00			
Trends will continue	AC	BAC	EV	Result		
AC+BAC-EV	450,000	1,000,000	400,000	$1,050,000.00		
Future work estimate is no longer valid	AC	ETC	Result			
AC+ETC	450,000	675,000	1,125,000			
Weight values for SPI or CPI	AC	BAC	EV	CPI	SPI	Result
AC+[(BAC-EV)/(CPIxSPI)	450,000	1,000,000	400,000	0.89	0.80	$ 1,293,750.00

To-Complete Performance Index

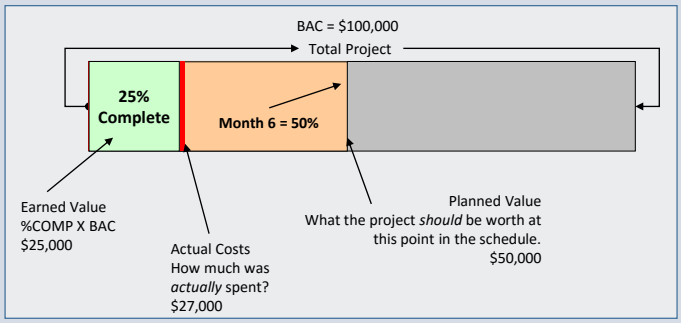
- Efficiency needed to meet BAC:
- **TCPI=(BAC-EV)/(BAC-AC)**

- Efficiency needed to meet EAC:
- **TCPI=(BAC-EV)/(EAC-AC)**

- *Great than 1, hard to accomplish*
- *Exactly 1, same level of efficiency*
- *Less than 1, easier to accomplish*

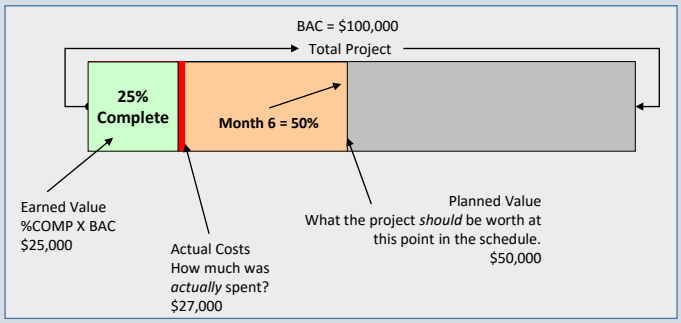
TCPI=(BAC-EV)/(BAC-AC)

- TCPI=($100,000-$25,000)/($100,000-$27,000)
- TCPI=75,000/73,000
- TCPI=1.0273



TCPI=(BAC-EV)/(EAC-AC)

- TCPI=($100,000-$25,000)/($107,526-$27,000)
- TCPI=75,000/80,526
- TCPI=.93



Story Points and EVM

- Can use Story Points instead of dollars
- Planned to complete 20 story points
- Completed 18 instead
- SPI=EV/PV
- SPI=18/20
- SPI=.90

Five EVM Rules

- EV is first
- Variance means subtract
- Index means division
- Less than one is bad in an index
- Negative is bad in a variance

Agile Project Accounting

- Agile accounting defines the economic models of agile projects
- Project work and smaller chunks of a larger project
- Smaller chunks of work are less risky
- Agile project accounting is accountability of what was invested in relation to the value of the return on investment

Key Performance Indicators

- Key performance indicators are metrics to show how well the project is performing
  - Rate of progress
  - Remaining work
  - Likely completion date
  - Likely cost remaining

Managing Risk In An Agile Project

- Risk in an agile project is anything that threatens the project's goals
- Risk is considered an anti value
- Risk must be managed in a project
- Risk identification is an iterative activity
- Risk a recorded in a risk along

Addressing Features With High-risk

- Features that have high levels of risk can be addressed early in project iteration
- High areas of risk need to be addressed sooner rather than later
- A risk-adjusted backlog brings risk features into an early portion of the project
- A risk burndown chart tracks risk as they move down in priority and elimination

Regulatory Compliance For Agile Projects

- Regulations are requirements
- Regulatory compliance is one instance for documentation where just because is utilized

Prioritizing Value In Agile Projects

Welcoming changing requirements is key to Agile projects

Customer Value Prioritization

- Agile teams work on the items that yield the highest value to the customer first
- The product owner is responsible for keeping items in the backlog prioritized by business value
- When changes added to backlog, they must be prioritized for value
- The customer is the person who will declare what success looks like
- The team will discuss with the customer at the end of each iteration the priority of the remaining work items

Prioritization Schemes

- How the work is prioritized
- The team agrees on the prioritization scheme
- The prioritization scheme is communicated and agreed upon by the entire agile team

Simple Scheme for Prioritization

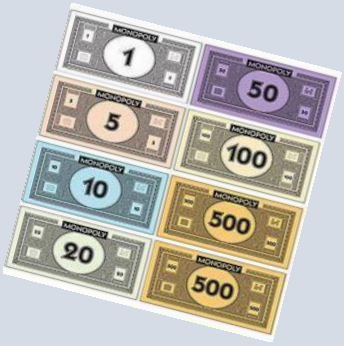
- Items in the product backlog are ranked:
  - Priority one (high)
  - Priority two (medium)
  - Priority three (low)
- This approach has a risk that everything is ranked as priority one

MoSCoW Prioritization Scheme

- Made popular by DSDM
- Must have
- Should have
- Could have
- Would like to have, but not at this time

Monopoly Money

- Stakeholders receive monopoly money equal to the amount of the project budget
- The monopoly money is distributed among the system features
- This approach is most effective when it's limited to prioritizing business features



100 Point Method

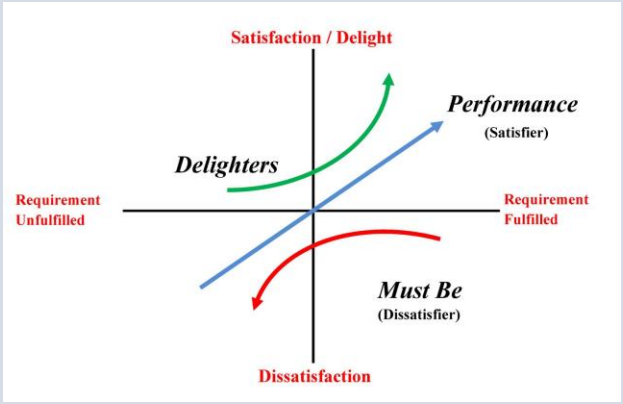
- Each stakeholder is allotted 100 points
- The points are assigned to the most important requirements

Dot Voting Or Multi Voting

- Stakeholders gets a predetermined amount of dots
- Dots are assigned to the business features
- Dots could be check marks or stickers

Kano Analysis

- Delighters exciters
- Satisfiers
- Dissatisfiers
- Indifferent



Requirements Prioritization Model

- Uses a scale of 1 to 9
- Benefit penalty cost and risk of every feature is rated

Relative Prioritization Ranking

- Priority of features
- Simplest the features from most important to least important
- Determination made to meet budget and schedule
- Changes may change the prioritize list
- Changes may bump some priorities from the list

Incremental Delivery

Optimizing the value of delivery

Incremental Delivery

- The team regularly deploys working increments
- Usually to a test environment for evaluation
- This is an opportunity for an early return on investment

Minimum Viable Product

- Complete enough to be useful
- Small enough that it does not represent the entire project
- Also known as the minimal marketable feature
- Barebones essentials of a product

Agile Tooling

- Agile teams prefer low-tech high-touch tools over-sophisticated computerized models
- Technical tools can exclude team members from interacting
- Consider high-tech tools for scheduling:
  - Data accuracy perception increases
  - A bad estimate is a bad estimate
  - Barriers for stakeholder interaction are created

Examples Of Low-tech High-touch Tools

- Cards
- Charts
- Information radiator
- Tools that promote communication and collaboration
- Tools that promote learning and knowledge transfer

Scheduling Software Vs. Kanban Board

- Also known as a task board
- Help teams monitor the work in progress

Work In Progress (WIP)

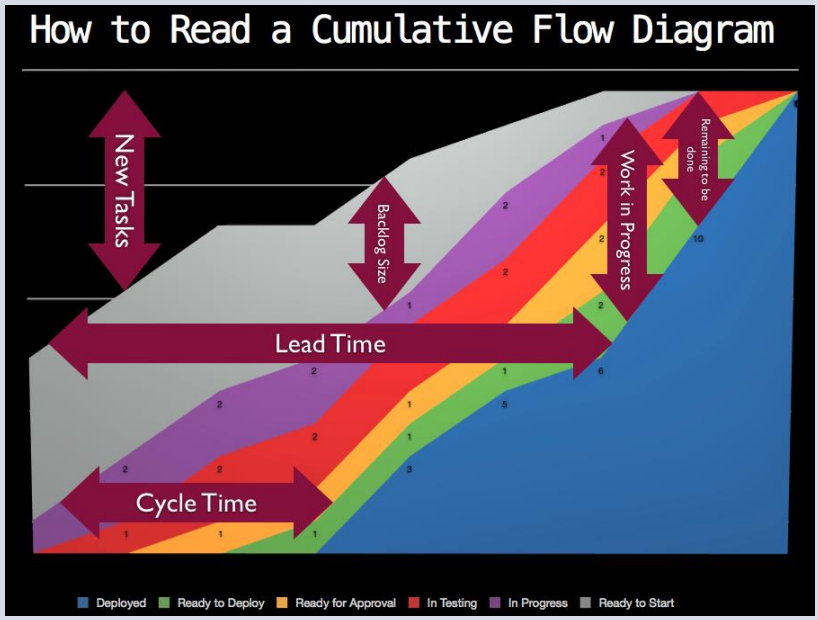
- Also known as a work in process and work in play
- WIP is risk
- WIP hides bottlenecks
- WIP requires investment but delivers no return until the work is complete
- WIP needs to be limited

Limiting Work In Progress

- Agile attempts to limit WIP
- Kanban boards can have WIP limits
- WIP limits keeps the team from taking on too many pieces of work
- WIP limits reveal bottlenecks

Cumulative Flow Diagrams

- Cumulative flow diagrams help tracking and forecasting the delivery of value
- Reveals the total in progress and completed work



Bottlenecks And The Theory Of Constraints

- A constraint is anything that limits your options
- Time, cost, and scope are typical constraints
- Constraints can be throughput of capacity
- A thin line in a cumulative flow diagram can reveal a bottleneck

Five Focusing Steps Of Goldratt's Theory Of Constraints

- Identify the constraint
- Exploit the constraint
- Subordinate all other processes to exploit the constraint
- If after steps two and three is done more capacity is needed to meet demand elevate the constraint
- If the constraint has not moved go back to step 1

Contracting in Agile Projects

Exploring Agile project contracts

Request for Proposal

- A request for proposal is from the buyer to the seller
- If the seller is to use a agile practices it must be defined in the request for proposal
- The buyer may need to educate the vendor about agile practices
- Agile project welcome change so this may be hard to do with contracts

Agile Constraints And Contracts

- Agile is flexible
- Contracts are not flexible
- Contracts are a form of a constraint
- Contracts are constrained by an offer and a consideration
- Collaboration over contracts

Agile Constraints And Contracts

- Agile projects constrain time and cost
- Agile project allow the scope to change
- Contracts typically constrain and balance time cost and scope

Considerations For Contracts

- Scope changes
- Priorities
- Time and cost

Exam Tip

- Customer collaboration over contract negotiation
- Customers are more involved than traditional projects

Graduated Fixed-price Contract

- Both parties share some of the risk and reward
- If a vendor delivers on-time they get paid for their work at the hourly rate
- If the vendor delivers early they get paid for their work but at a higher hourly rate
- If the vendor delivers late they get paid for their work but at a lower hourly rate

Fixed Price Work Packages

- The price of the work remains constant
- Individual work packages are estimated for cost
- Changes to the scope reflect a new estimate for those work packages

Customize Contracts

- The buyer and the seller can make any agreement they want
- Procurement is always tricky with agile projects

Value, Verification, and Validation

Ensuring value with agile project

Gulf of Evaluation

- The difference between what is said and what is understood
- Intangible projects often experience this gulf
- What does done looks like?

Frequent Verification and Validation

- Testing checkpoints and reviews
- Frequent verification and validation happened throughout the project
- Build consensus between the project team and the project stakeholders

Examples Of Verification and Validation In XP

- Pair programming
- Unit testing
- Customer collaboration
- Stand up meetings
- Acceptance testing
- Iteration demonstrations
- Product release

Exploratory Testing

- The tester aims to discover issues and unexpected behavior
- The tester explores the software
- This is in addition to scripted testing

Usability Testing

- How will a user respond to the system under realistic conditions?
- How easy is it to use the system?
- What improvements need to be made for usability?

Continuous Integration in Agile Projects

- Incorporate new and changed code into the code repository
- Small code commits
- Frequent integration
- Relies on automated tools to integrate code when new code is checked in

Continuous Integration System

- Source code control system – version control
- Build tools – build tools compile the code
- Test tools – unit test to ensure functionality operates as expected
- Scheduler or trigger – builds are launched on a schedule or based on conditions
- Notifications – an email or instant message reporting on the results of a build

Why Continuous Integration?

- Early warning a broken conflicting or incompatible code
- Problems are fixed as they occur
- Immediate feedback
- Frequent unit testing define issues quickly
- Easy to reverse the code back to the last known good

Disadvantages Of Continuous Integration

- Set up time is lengthy
  - Often called Iteration 0
- Cost of a dedicated server
- Time required to build a suite of automatic tests

Exploring Test-driven Development

- Also called test first development
- Test are written before the code is written
  - Nunit
  - Junit
- Code is developed and edited until the code passes all tests
- Refactoring is the final step to clean up the code

Test-driven Development

- Red – green – re-factor
- Red – green – clean
- Focus on the test first
- Early testing helps catch defects early in development
- Beware of developers writing their own test

Acceptance Test Driven Development

- Testing focus is on business requirements
- Test represent the functionality the software is to have
- It's all about the desired behavior
- FIT – framework for integrated testing
  - Also called FitNessee
  - http://www.fitnesse.org/

Acceptance Test-driven Development Cycle

- Discuss the requirements – developers as the product owner questions that are designed to gather acceptance criteria
- Distill test in a framework friendly format – gets the test ready to be entered into the acceptance test tool
- Developed the code and run the test – test initially fail because the code hasn't been written completely
- Demo – with automated acceptance testing scrips and demonstrations of the software

Stakeholder Management in Agile Projects

Managing and engaging the stakeholders

Stakeholder Engagement in the PMI ACP Exam

- 17 percent of the exam
- 20 exam questions

Stakeholder Engagement Objectives

- Working with the project stakeholders
- Establishing a shared vision
- Creating collaboration
- Communicating with project stakeholders
- Using interpersonal skills

Stakeholder Engagement Tasks

- Engage and empower business stakeholders
- Share information frequently with all stakeholders
- Form working agreements for participation
- Assess organizational changes to maintain a stakeholder engagement
- Used collaborative decision-making and conflict resolution

Stakeholder Engagement Tasks

- Establish a shared vision for project stakeholders
- Maintain a shared understanding of project success
- Provide transparency for better decisions
- Balance certainty and adaptability for better planning

Who are Project Stakeholders?

- Are impacted by the project
- Can impact the project
  - Customers
  - Project sponsor
  - Project leaders
  - Development team
  - Vendors
  - End users

Identify Stakeholders

- Stakeholder analysis
- Expert judgment
- Meetings
- Create the stakeholder register

Plan Stakeholder Management

- Expert judgment
- Meetings
- Analytical techniques
- Create the stakeholder management plan
- Update project documents

Manage Stakeholder Engagement

- Communication methods
- Interpersonal skills
- Management skills
- Issue log
- Change request
- Project management plan updates
- Project document updates
- Organizational process assets updates

Control Stakeholder Engagement

- Information management systems
- Expert judgment
- Meetings
- Work performance information
- Change request
- Project management plan updates
- Project document updates
- Organizational process assets updates

Keeping Stakeholders Engaged

- Agile project worked with stakeholders
- Not command and control
- Consider servant leadership
- Identify project stakeholders as early as possible

Educating Stakeholders for Agile Projects

- People new to agile projects need some basic education
- Address concerns directly with project stakeholders
- Explain the approach that will be used

Managing Stakeholder Engagement

- Short iterations keep stakeholders involved
- Reviews and demos show the results of the work
- Agile places value of work that is done
- Agile is naturally visible for project stakeholders

Incorporating Stakeholder Values

- Work is based on what the stakeholders value
- Engage the product owner to prioritize the backlog
- Work is executed by priorities
- The development team creates the highest priority items
- The development team delivers early value to the business
- Stakeholders are invited to planning meetings and retrospective

Incorporating Community Values

- Agile teams must share the values of their broader community
- Respect
  - Agile works for consensus
  - Don’t judge suggestions
  - Respect differing opinions
- Courage
  - Agile teams display courage through demonstrations
  - Pair programming
  - Product owner prioritizing requirements
  - Retrospective

Principles of Stakeholder Engagement

- Get the right stakeholders
- Insure a stakeholder participation
- Manage stakeholder interest
- Frequently discussed what done looks like
- Show progress to project stakeholders
- Openly discuss project estimates and projections

Creating a shared vision

Understanding what is requested and delivering what was requested

Failing Fast

- Failing fast means failing early and cheaply
- Good way to discover misunderstandings
- Ensures the project team understands what stakeholders want

Creating an Agile Charter

- Agile charters authorize the project and the project manager
- Agile charters are from the project sponsor
- Could be lightweight or very detailed
- Acknowledge change is likely in the actual project

Agile Project Charters are Different

- Traditional charters are very specific
- Agile charters are broad and high level
- Agile charters define:
  - Who will be engaged
  - What is the project about
  - Where will the project take place
  - When will the project start and end
  - Why this project being chartered
  - How the goals of the project be achieved

Creating A Project Tweet

- Project customers in the project team can work together to create a project tweet
- Describe the goal of the project in 140 characters or less
- This exercise defines a high-level description of the project
- Elevator statement

What Does Done Mean

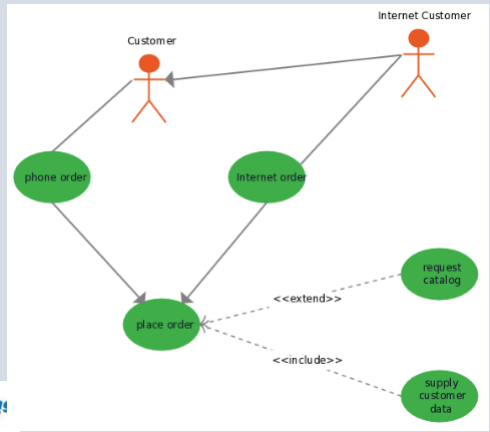
- Defining done is important for everyone
- An example of a shared vision
- User stories – done will mean developed documented and tested
- Releases – done means there are no large defects or remaining change requests
- Final project deliverables – priority features are implemented three months of trouble-free operation and satisfactory scores

Working With Agile Modeling

- Modeling techniques for agile projects
- The value is in discussion and creation of the model
- Often treated on whiteboards and photographs for a record
- Lightweight and barely sufficient

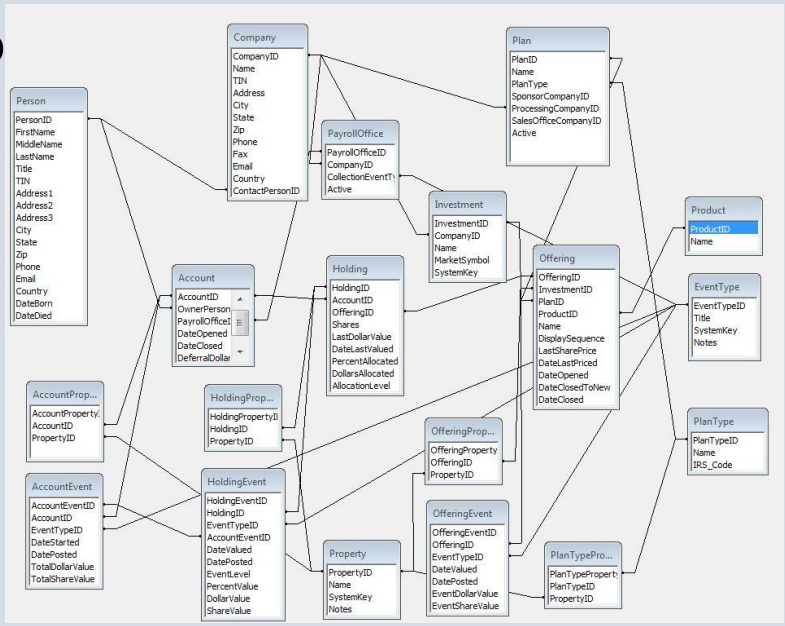
Use Case Diagram

- How will the user use the solution



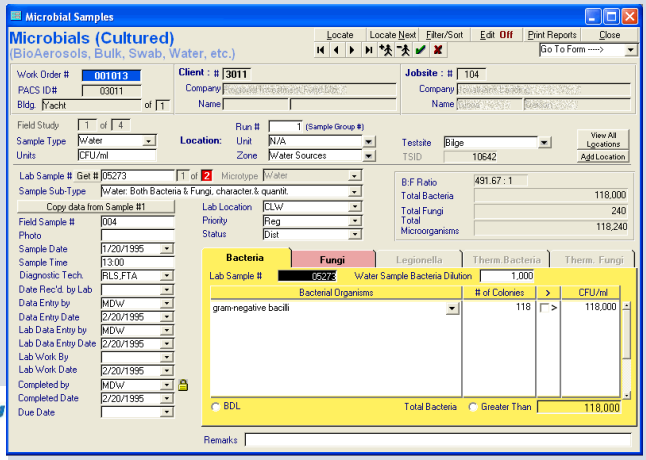
Sample Data Mo

- How data flows through the system



Screen Designs

- What does the user interface look like



Wireframes

- A quick mock-up of a product
- Could be screens and data flows between screens
- Ensures that everyone has the same understanding of the product
- A form of low fidelity prototyping
- Quick way to get feedback

User Personas

- Biographical sketches of key stakeholders
- Description of product users
- Somewhat grounded in reality
- Goal oriented
- Show tangible and actionable outcomes
- Focus on the users and who the users will be

Communications

Managing communications in agile projects

Plan Communications Management

- Communication requirements analysis
- Communication technology
- Communication model
- Communication methods
- Meeting
- Create the communications management plan
- Update project documents

Manage Communications

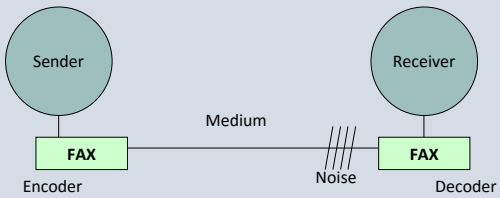
- Communication technology
- Communication models
- Communication methods
- Information management systems
- Performance reporting
- Communicate with stakeholders
- Update the project management plan project document organizational process assets

Control Communications

- Information management systems
- Expert judgment
- Meetings
- Work performance information
- Change request
- Update the project management plan project documents and organizational process a set

Communication Model – Sender Receiver

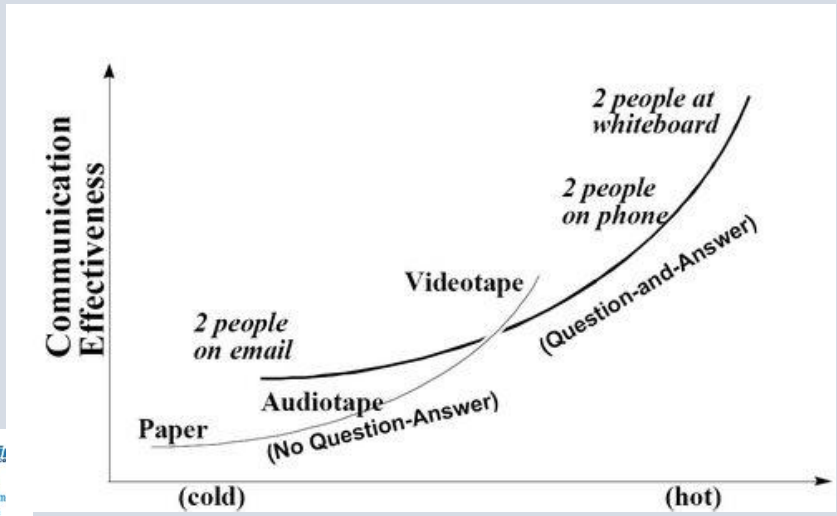
- Sender
- Encoder
- Medium
- Decoder
- Receiver
- Barrier
- Noise



Face-to-face Communication

- Face-to-face communication is preferred
- Highest bandwidth of all communication types
- Most effective communication

Effectiveness Of Different Communication Channels



Two-way Communication

- Dispatching model – top down communication
- Collaborative model – interactive communication between sender and receiver

Knowledge Sharing

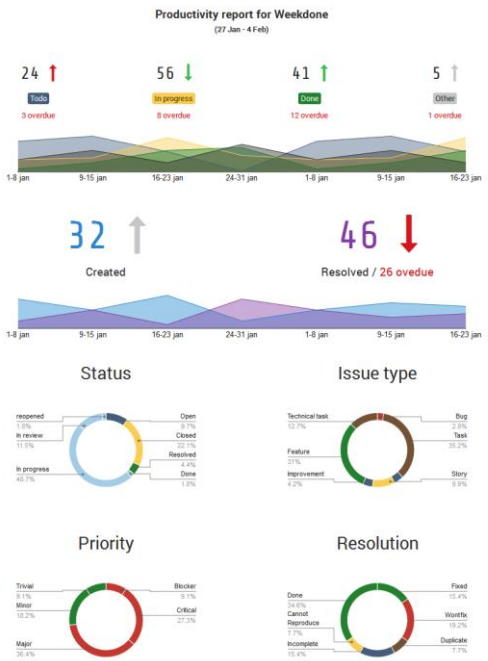
- Knowledge sharing is critical for agile projects
- Share information with everyone
- Collective code ownership means any developer can edit any code at any time
- Agile practices promote knowledge sharing:
  - Kanban boards
  - Information radiators
  - Personas
  - Wireframes

Communicating And Agile Projects

- Low-tech high-touch tools
- Stand up meetings
- Osmotic communications
- Tacit knowledge

What is an Information Radiator?

- Highly visible displays of information
- Large graphs or charts that summarize project data
- Out in the open and easily accessible
- Also known as visual controls



What's on an Information Radiator?

- Features delivered versus features remaining
- Who is working on what
- Current iteration features to be created
- Velocity and defect measurements
- Retrospective outcomes
- Threats and issues for the project
- Burn up and burndown chart
- Story maps

Social Media and Agile Projects

- Remote workers staying in touch
- Facebook
- Twitter
- Web collaboration tools
- Non co-located teams
- Consider the sensitivity of the project information

Collaborative Approaches in Agile

Collaboration is key in agile projects

Collaboration is Key in Agile Projects

- Customer collaboration over contract negotiation
- Business people and developers must work together daily throughout the project

Benefits of Collaboration

- Generates wiser decisions
- Promotes problem solving
- Promotes action
- Build social capital
- Ownership of collective problems

Engaging People in Agile Projects

- Engagement creates better ideas and put some conversations
- Active problem solving instead of command and control
- Taking action rather than being passive
- Collective ownership of ideas
- Motivates and engages the project team
- Shifts the power downward

Green Zone versus Red Zone

- Takes responsibility
- Responds non-defensively
- Not easily threatened
- Build mutual success
- Seeks solutions
- Uses persuasion
- Firm, but not rigid
- Thinks both short-term and long-term
- Considers other points of views
- Welcomes feedback
- Considers conflict to be natural
- Speak calmly and directly about difficult issues
- Accept responsibility
- Seeks excellence

Green Zone Versus Red Zone

- Blames others
- Responds defensively
- Feels threatened or wrong
- Triggers defensiveness
- Holds grudges
- Shame, blame, and accusations
- Binary thinking
- Short-term advantage
- Feel victimized
- Doesn't seek feedback
- Must win at any cost
- Is rigid and reactive
- Creates a climate of antagonism
- Disapproval and content
- Sees others as the enemy
- Does not listen effectively

Hosting Workshops

- Meetings for participants to get work done
- Clear goals and a schedule
- Retrospectives
- Planning meetings
- Estimating sessions

Tips for a Great Workshop

- Have a diverse group of people
- Facilitated for involvement
- Get people involved early

User Story Workshops

- Preferred approach for candidate user stories
- Also known as story writing workshops
- Optimize the workflow by understanding user needs
- Engage stakeholders in the design process

Brainstorming

- Collaborative technique too rapidly generate lots of ideas
- Maximize number of suggestions
- No stupid ideas
- Will sort through the ideas later

Brainstorming Methods

- Quiet writing
- Round robin
- Free for all

Collaboration Games

- Also known as innovation games
- Remember the future
- Prune the product tree
- Speed boat
- Buy a feature
- Bang for the buck

Remember the Future

- Collaboration game
- Stakeholders look back at the project
- 20 minutes to write a future report about how the project went
- Includes what was created; written on sticky notes
- Notes are moved into associated clusters and duplicates removed
- This game define success

Prune The Product Tree

- Drawing of a big tree
- The trunk is what we already know or have built
- The branches are new functionality and what needs to be designed
- Participants add features on sticky notes to the tree
- Closer to the trunk represents higher priority

Speed Boat Games – Sailboat

- Imagine it's a boat
- What winds are pushing the sailboat
- What anchors are holding the sailboat back
- What direction is the sailboat going
- What rocks are in the way

Interpersonal Skills for Agile Success

The soft stuff is the hard stuff and the hard stuff is the easy stuff

Interpersonal Skills for Agile Projects

- Emotional intelligence
- Active listening
- Facilitation techniques
- Negotiation
- Conflict resolution
- Participatory decision making

Emotional Intelligence

- The ability to identify and influence our emotions and the emotions of others

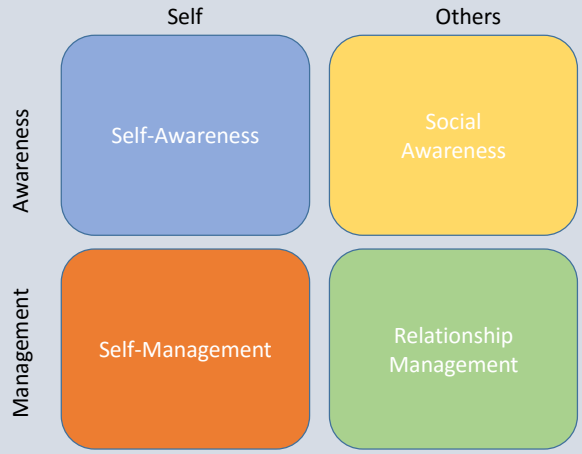
Quadrants Of Emotional Intelligence

- Self-management
  - Self control
  - Conscientiousness
  - Adaptability
  - Drive and motivation
- Self-awareness
  - Self confidence
  - Emotional self-awareness
  - Accurate self-assessment

Regulate and recognize
Page 179

- Social skills
  - Influence
  - Inspirational leadership
  - Developing others
  - Teamwork and collaboration
- Social awareness
  - Empathy
  - Organizational awareness
  - Understanding the environment

Quadrants of Emotional Intelligence



Active Listening

- Hearing what someone is really trying to say
- Level 1 – internal listening
- Level 2 – focus listening
- Level 3 – global listening

Level 1 – Internal Listening

- Words are heard, but we're not very attentive
- We interpret the meaning – how is this going to affect me

Level 2 – Focused Listening

- The speaker's perspective
- We empathize with the speaker
- We look for emotional indicators such as voice and tone
- Facial expressions and words

Level 3 - Global Listening

- We build on level 2
- A higher level of awareness
- Subtle clues about meeting such as the speaker's posture and energy
- Helps us to develop a fuller context of the message

Facilitation

- Running effective meetings and workshops
- Goals – ensuring that meetings are not a waste of time by promoting participation
- Rules – establishing ground rules and holding people accountable to these rules
- Timing – the duration of the meeting is established ahead of time
- Assisting – making the meeting effective and assuring that everyone may contribute

Negotiation

- Negotiations happen throughout the project
- Consider the priorities of user stories
- Avoid a zero-sum games where only one person wins
- Healthy negotiations allow for give and take

Conflict Resolution

- Differences of opinions and competing interest
- Some conflict is healthy

Level	Name	Characteristic	Language
One	Problem to solve	Information sharing and collaboration	Open in fact based
Two	Disagreement	Personal protection Trump's resolving the conflict	Garden and open to interpretation
Three	Contest	Winning Trump's resolving the conflict	Includes personal attacks
Four	Crusade	Protecting one's own group becomes the focus	Ideological
Five	World War	Destroy the other	Little or non-existent

Participatory Decision Making

- Engaging stakeholders for decision making
- Communication and decision-making are critical to keep everyone informed then engaged
- Involves stakeholders when making decisions
- Stakeholder involvement increases as they commit to the project

Convergent And Shared Collaboration

- Convergent – participating decision making models in for conversions for collective agreements
- Shared collaboration they share the decision-making process fairly

Simple Voting

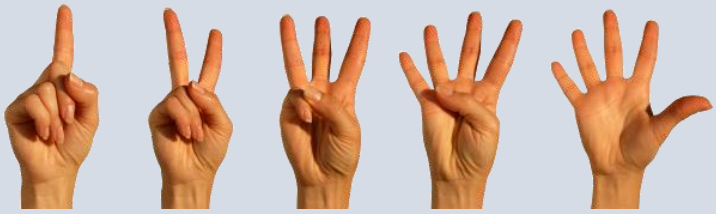
- Participatory decision approach
- The team votes for or against an idea by a show of hands

Thumbs Up, Down, or Sideways

- Thumbs up and individual is for the decision
- Thumbs down and individual is against the decision
- Thumb sideways the individual is neutral or undecided

Fist of Five Voting

- The number of fingers shown indicates degree of support



Highsmith Decision Spectrum

- Participants place a checkmark on a spectrum
- In favor
- Okay but with reservations
- Mixed feelings
- Not in favor but will commit
- Veto

Team Performance

PMI-ACP domain overview

Team Performance Overview

- 16 percent of the PMI-ACP exam
- 19 exam questions
- Building high-performing teams

Team Performance Tasks

- Develop team rules and processes to foster buy in
- Help grow team interpersonal and technical skills
- Use generalizing specialist
- Empower and encourage emergent leadership
- Learn team motivators and demotivators

Team Performance Tasks

- Encourage communication via collocation in collaboration tools
- Shield team from distractions
- Align team by sharing project vision
- Anchor team to measure velocity for capacity and forecast

Develop and Support Self-organizing Teams

- Self-organizing
- Self-empowered
- The project team are stakeholders
- Team leaders
- Scrum masters
- Coaches

People Over Processes

- People are more important and processes
- Focus on the people
- Servant leadership

Development Team / Delivery Team

- Coders
- Writers
- Analyst
- Testers
- People can perform multiple jobs in switch from role-to-role

Development Team / Delivery Team

- Build the product increments
- Update information radiators
- Self organize and self direct
- Share progress through daily standup meetings
- Right acceptance test
- Test and revise the product increments
- Demonstrate completed increments
- Hold iteration retrospective
- Estimate the stories and task

Business Representatives

- Product owner
- Customer
- Proxy customer
- Value management team

Business Representatives

- Prioritize product features
- Manages the product backlog
- Ensures a shared understanding
- Provides the acceptance criteria
- Makes change request
- May change the product features and priorities
- Facilitate engagement of external project stakeholders
- Provides due date for the project
- Attends planning meetings reviews and retrospectives

ScrumMaster

- Coach or team leader
- Servant leader
- Helps the delivery team self-govern and self organize
- Facilitator and communicator
- Coach and mentor to the delivery team
- Guides agile processes
- Helps the product owner manage the product backlog
- Helps the product owner communicate
- Facilitates meetings
- Follows up on issues

Project Sponsor

- The main advocate for the project
- Provides direction to the product owner
- Determines value on time and on budget
- May attend iteration review meetings
- Authorizes the project

Building Agile Teams

How to build an agile team

Team Characteristics

- 12 or fewer members
- Team members have complementary skills
- Team members are generalizing specialist
- Team members are committed to a common purpose
- Team members hold themselves mutually accountable
- Team members have shared ownership for the project outcome

Defining Generalizing Specialist

- Team members can serve in multiple roles
- Team members can easily switch between rules
- Helps to resolve bottlenecks

Characteristics Of High-performing Teams

- Create a shared vision for the project team
- Set realistic goals
- Limit team size to 12 or fewer people
- Build a sense of team identity
- Provide strong leadership

Eight Characteristics of High-Performance Teams

- Self-organizing
- Empowered
- Believe that as a team they can solve any problem
- Committed to team success
- Owns its decisions and commitments
- Motivated by trust
- Consensus driven
- Participate in constructive disagreement

Self-organizing Teams

- Not command and control
- Can use their own knowledge to organize work
- Structure that work based on iteration goals
- Responsibility delegated to the team

Self-directing Teams

- Empowered to work collectively
- Make local decisions
- Estimate and decide the project work
- Make mistakes and learn from mistakes

Exam Tip

- Self-organizing and self-directing teams are goals of agile projects
- Teams usually do not to start as self-organized and self-directed

Defining Emergent Leadership

- Different people lead different initiatives
- High-performing teams allow multiple leaders
- No power struggle when leaders change roles
- Leaders are self-selected not assigned

Experimenting and Failing Safely

- The team should experiment and try new approaches
- It's okay to fail
- Learn from failure and move forward
- An engagement culture rewards people for problem solving collaboration and sharing ideas

Encourage Constructive Disagreement

- Debate and conflict is natural and healthy
- Constructive conflict leads to better decisions and buy-in
- Divergence means the team will argue and debate
- Convergence means the team will agree on the best solution

Five Dysfunctions Of A Team

- Absence of trust
- Fear of conflict
- Lack of commitment
- Avoidance of accountability
- Inattention to results

Shu-ha-ri Model Of Skill Mastery

- Shu: obeying the rules to keep protect or maintain
- Ha: consciously moving away from the rules; ha means to detach or break free
- Ri: unconsciously finding an individual path; ri means to go beyond or transcend

Shu-ha-ri Model Of Skill Mastery

- Shu – start by following the rules
- Ha – once the team has mastered the guidelines they can move away from them and work intuitively
- Ri – the team reaches full mastery and can transcend the rules

Dreyfus Model Of Adult Skill Acquisition

- ***Novice*** follow the rules they've been given and make analytical decisions
- ***Advanced beginner*** still following the rules but based on experience better understand the context of the rules
- ***Competent*** determining which rules are best for each situation
- ***Proficient*** actively choosing the best strategy rather than simply relying on the rules
- ***Expert*** decision-making becomes intuitive

Tuckman Model Of Team Formation And Development

- ***Forming*** – the team comes together and learns about each other
  - Also known as a working group
- ***Storming*** – conflict and struggle for the approach and leadership
  - Also known as a pseudo team
- ***Norming*** – the team works with each other and conflict has settled
  - Also known as a potential team
- ***Performing*** – the team hits their stride
  - Also known as a real team and becoming a high-performing team

Adaptive Leadership

- Directing – happens during team forming
  - Team members may have low competence but high commitment
  - Leaders hi directive and low supportive behavior
- Coaching – happens during storming
  - Team members have some competence and low commitment
  - Leaders high directive and high supportive behavior

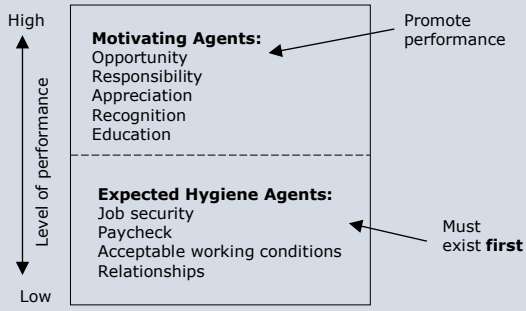
Adaptive Leadership Continued

- Supporting – happens during norming
  - Team members have moderate to high competent and variable commitment
  - Leaders offer low directive and high supportive behavior
- Delegating – happens during team performing
  - Team members have high competence and high commitment
  - Leaders offer low directive and low supportive behavior

Motivating The Project Team

- Continuum of net contribution

Herzberg's Theory Of Motivation



Training Coaching And Mentoring

- Training is teaching a skill or knowledge through practice and instruction.
- Coaching is a facilitated process to help individuals develop and improve performance
- Mentoring is a professional relationship more free flowing. The mentor offers advice

Guidelines for One-on-one Coaching

- Meet them a half a step ahead
- Guarantee safety
- Partner with managers
- Create positive regard

Collaborative Team Spaces

Creating a collaborative team space

Co-located Teams are Preferred

- All the team works in one location
- Ideally with in 33 feet or ten meters of each other
- No physical barriers like walls are doorways
- Distributive teams is collaborative software

Creating a Team Space

- Location for team members in the same space
- Also known as the war room
- Visible information radiators for project metrics
- Lots of white boards and task boards

Caves and Commons

- Caves are private spaces for phone calls or one-on-one conversations
- Commons is the primary work area

Tacit Knowledge

- The unwritten information collectively known by the group
- How to restart a server
- How to turn on the lights
- Larger groups have more difficulty with tacit knowledge

Osmotic Communication

- Learning by overhearing each other's conversation
- Benefit of a collocated team

Managing Virtual Teams

- Consider different time zones
- Consider different cultures
- Different communication styles
- Different native languages

Managing Distributed Teams

- Distributed teams are virtual teams
- Short iterations help collaboration in coordination
- Distributed teams is not the same as outsourcing
- Distributed teams faced more of a challenge it was storming and norming
- The project manager may need to introduce controversial or difficulties of the work early in the project
- A face-to-face kickoff is often needed

Exam tip

- Maintain a metaphor
- Use frequent communication
- Intensified facilitation
- Use best practices for conference calls
  - Keep on track
  - Keep on time
  - Keep track of who is on the call
  - Keep the answers coming
  - Keep it fair
  - Keep it documented

Digital Tools For Distributed Teams

- Video conferencing it like that
- Interactive whiteboard
- Instant messaging
- Presence based application
- Electronic task for the storyboards
- Web-based meeting facilitators
- Virtual card wall
- Wiki site

Tracking Team Performance

Monitoring progress and performance

Utilizing Burndown Chart

- Track the work that remains to be done on a project
- Measures the team progress in completing the project work

Utilizing Burn Up Chart

- Track the work that has been completed
- As work is done the line moves upward
- Provides additional insight into the project status

Understanding Team Velocity

- Velocity is the measure of a team's capacity for work per iteration
- Measured in the same unit that the team estimates the work
- Velocity very early and then stabilizes
- Velocity tends to plateau

Calculating Completion Time

- The team's velocity has been 20 story points per iteration
- There are 200 story points left
- Each iteration is two weeks
- 200 divided by 20 is 10
- 10 times 2 is 20
- There are 20 weeks left in the project

Adaptive Planning Overview

Planning an agile project

Adaptive Planning Exam Domain

- 12 percent of the PMI-ACP exam
- 14 exam questions
- Sizing estimating planning

Key Tasks for Adaptive Planning

- Progressive elaboration and rolling wave planning
- Transparent planning and key stakeholders
- Managing expectations by refining plans
- Adjusting planning cadence based on project factors and results
- Inspect and adapt the plans to changing events

Key Tasks for Adaptive Planning

- Size items first independently of team velocity
- Adjust capacity for maintenance and operations demands to update estimates
- Start planning with high-level scope schedule and cost range estimates
- Refine ranges as the project progresses
- Use actuals to refine the estimate to complete

Define Adaptive Planning

- Planning is an ongoing activity
- Agile planning is different than predictive planning
- Plan for early delivery business value, risk reduction, visibility

Interactive Planning Examples

- Daily standup
- Backlog prioritization
- Sprint retrospective
- Iteration planning

Agile Planning Concepts

Planning an agile project

Beginning with Adaptive Planning

- Agile projects are value-driven
- Minimize non-value-added work
- Plan to replan
- Early plans are necessary, but they're likely flawed
- Uncertainty requires replanning

Agile Versus Non-agile Planning

- Trial and demonstration uncover true requirements
- Less up front and more iterative planning
- Mid-course adjustments are normal

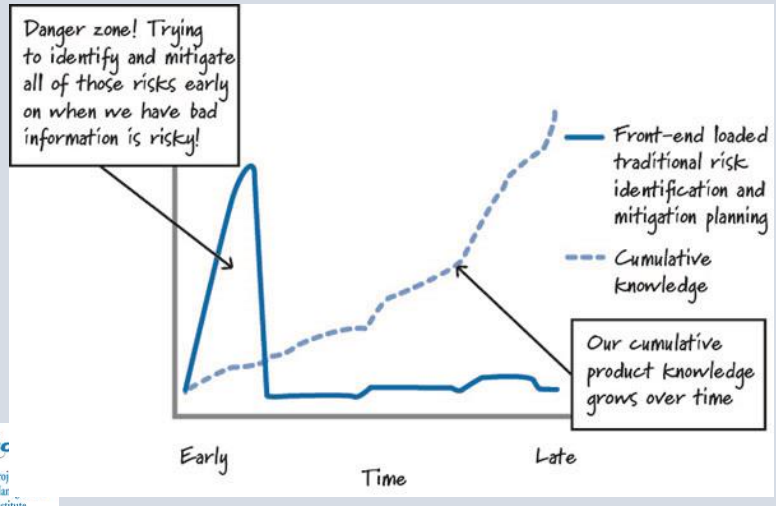
Trial and Demonstration

- Prototypes help initial planning
- Helps to avoid the gulf of evaluation
- Communicate agile planning practices

Iterative Planning

- The planning effort is distributed throughout the project life cycle
- Agile projects are risky
- Consider planning efforts over the project life cycle
- Agile projects typically do more planning

Planning Risk Graph



Changes to Plans are Normal

- Knowledge work doesn't follow a predictive plan

Principles of Agile Planning

- Plan at multiple levels
- Engage the team and customer in planning
- Demonstrate progress in velocity
- Taylor processes for the project
- Priorities will cause the plan to be updated
- Account for risk distraction and team availability
- Utilize estimate ranges
- Base projections on completion rate
- Factor in diversion from outside work

Agile Discovery

- Emergent plans and designs versus predictive plans and designs
- Pre planning activities together consensus
- Backlog refinement – grooming
- Estimating uncertain work forces certain work
- New product development vs. Repeatable project

Progressive Elaboration

- As more information becomes available more planning can happen
- Continuing steadily in small increments
- Progressive elaboration examples
  - Plans
  - Estimates
  - Risk assessments
  - Requirements definition
  - Software design
  - Test scenarios

Progressive Elaboration and Rolling Wave Planning

- Rolling wave planning is planning a multiple points
- Plan and execute iteration
- Progressive elaboration is incorporating new information into the plans
- Progressive elaboration is the implementation of rolling wave planning

Value-based Analysis

- Assessing and prioritizing the business value of work items
- Business benefit – cost
  - Business benefit equals $8,000
  - Cost equals $5,500
  - Value is $2,500

More Value-based Analysis

- Will the item generator business value every week or month
- A high business value item may be dependent on a low business value item

Value-based Decomposition

- Requirements elicitation
- Grouping of like features
- Breaking down of features
- Ranking of requirements
- Prioritize requirements into development

Creating A Product Box

- Imagine a product box or software
- Top three features
- Major functional elements
- Prioritization of features
- This is a visualization exercise

Coarse Grained Requirements

- Keeps the overall design balanced
- Delays decision on implementation details until the last responsible moment

Timeboxing an Agile Project

- Fix duration period of time
- Define set of activities
- Daily scrum or stand up
  - Daily standup meetings - 15 minutes
  - Retrospective – 2 hours
  - Iterations and sprint's – 1 to 4 weeks

Timeboxed Sprints

- 12 work items
- Team completes eight
- Remaining four items returned to the backlog

Parkinson's Law

- Work expands to fill the time allotted to it
- Student syndrome – students wait to the last possible minute to start working

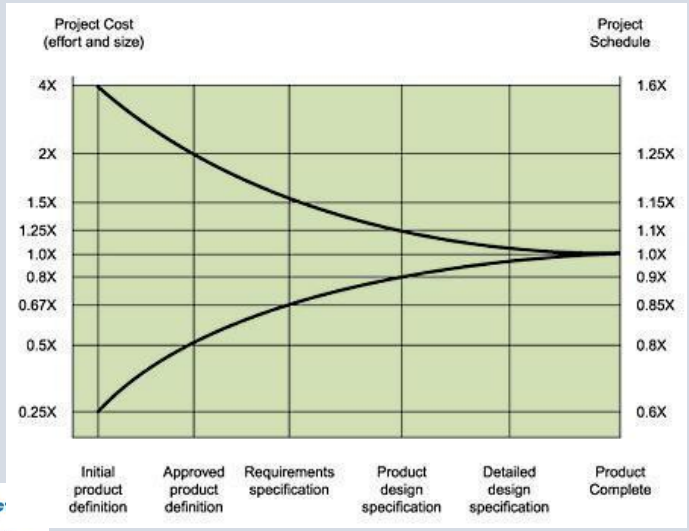
Tools for Project Sizing and Estimating

Estimating agile projects

Providing Estimate Ranges

- Not as precise as predictive planning
- More uncertainty and agile projects
- Include a range of variance
  - Between $500,000 and $550,000
  - Plus or minus 10%

Estimate Convergence Graph



Creating Agile Estimates

- Why is an estimate needed – to create a schedule and budget
- When does estimating happen – the last responsible moment and throughout the project
- Who does the estimating – team members estimate their own work
- How are estimates created – stages of sizing and planning; roll out an estimate cost may also be included
- How are estimates stated – always include a degree of uncertainty

Factoring Ideal Time

- Estimate as if there would be no interruptions
- Ideal time assumes all time in the estimate is for project work

Assumptions for Sizing and Estimating

- Details emerged as the project moves forward
- Plans are adjusted based on feedback
- Privatization happens throughout the project

Decomposing Project Requirements

- Breakdown of the project work
- Epics – large user stories that span one or more iterations
- Feature – attributes of the product
- User story – decomposition of a feature
- Task – smallest element of the decomposition

What is a User Story?

- Small chunk of business functionality within a feature that involves roughly 1-3 days work
- Also defined as for 40 hours of work
- User stories are written on index cards or sticky notes
- User stories are the items in the product backlog

Creating A User Story

- Potential stories are called candidate stories
- Perspective of the user or customer
- Often written in the following format
  - As a *role* I want *functionality* so that *business benefit*
- Answers two questions:
  - Who was asking for this?
  - Why are we doing this?

User Story Formats

- Given – the scenario for the story
- When – the action that takes place
- Then – the result of the action

Three C's Of User Stories

- Card – just enough text to identify the story
- Conversation – details are communicated via a conversation between the customer and the development team
- Confirmation – customer confirmed the story has been implemented correctly

INVEST In User Stories

- **C**haracteristics of effective user stories follow INVEST:
- **I**ndependent – stories can be prioritized in any order
- **N**egotiable – the team can discuss the user story with the product owner and make trade-offs based on the cost and function
- **V**aluable – the user story must have obvious value
- **E**stimate – the user story can be estimated for effort
- **S**mall – small user stories are easier to create and test that large user stories; 4 to 40 hours work
- **T**estable – the story results must be testable

Defining the User Story Backlog

- Also known as the product backlog
- User stories are listed and sorted
- User stories are prioritized in the backlog
- There is only one backlog

Grooming the Backlog

- The backlog needs to be kept continuously updated
- Prioritizing or refining the backlog is called grooming

Changes to the Backlog

- New stories can be added
- Existing stories maybe reprioritized or removed
- Stories can be decomposed into smaller chunks

Changes to the Backlog

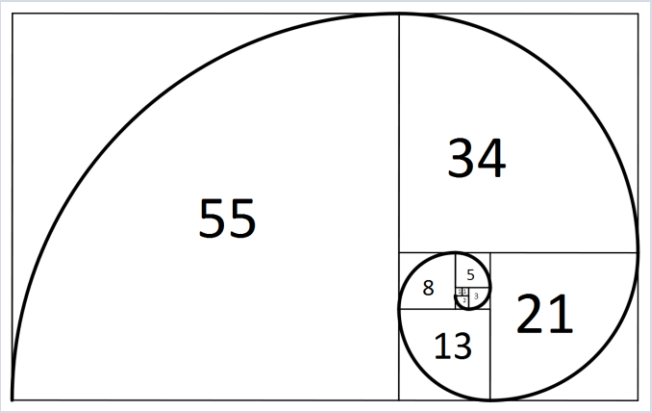
- Customer or value management team can add new story or reprioritize existing stories
- Decomposing stories – also called slicing – is typically done by the development team

Relative Sizing for Story Points

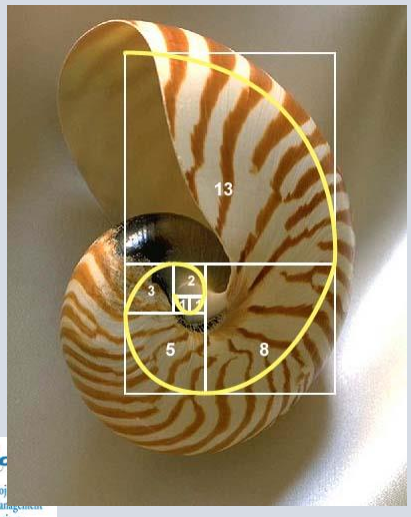
- It's difficult to make absolute estimate
- Story points are points assigned to stories size
- Relative sizing assigns points to stories on a relative scale
- The team then decides how many points can be done in our generation

The Fibonacci Sequence

- Starting with zero the two numbers in the sequence are added together to get the next number
- 0 plus 1 equals one
- One plus one equals two
- 2 plus 1 equals 3
- 1, 2,3,5,8,13,or 21 points
- Only these numbers are assigned to user stories



Fibonacci Sequence



Guidelines for Storing Points

- The team owns the definition of their story points
- Story point estimate should be all inclusive
- Point sizes should be relative
- When disaggregating estimates the totals don't need to match
- Complexity work effort and risk are all included in the estimate

Affinity Estimating

- Grouping items into similar categories or collections
- Group items based on story points
- Affinity estimating is like triangulation
- Allows the team to see the collection of user stories by points assigned

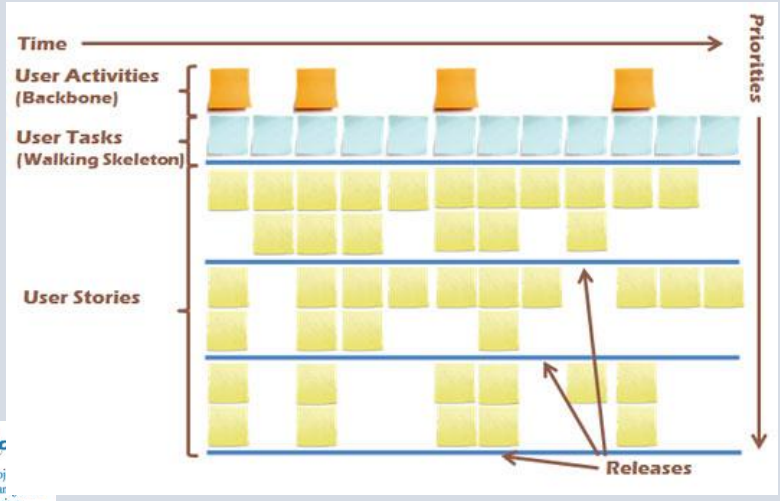
T-shirt Sizing

- User stories are assigned to t-shirt sizes

Story Maps

- High-level planning tool to map out project priorities
- Prioritized matrix of the features and user stories for the product being built
- Backbone – top level of the story map; essential functions for the system to work
- Walking skeleton – second row of the story map; smallest version of the system that will beat the customers both basic needs

Story Map Example



Creating Product Roadmap

- Visual depiction of product releases
- Primary items that will be included in each release
- Helps to check risk and viability

Wideband Delphi

- Rounds of anonymous estimates
- Helps to build consensus
  - Bandwagon effect – gathering around common viewpoint
  - Highest-paid person's opinion – HIPPO
  - Groupthink – making decisions to maintain group harmony

Planning Poker

- Cards with the Fibonacci sequence
- User stories review
- Participants show their cards at the same time the score the user story

Planning for Releases and Iterations

Agile projects are divided into releases and iterations

Defining Releases and Iterations

- Iterations are short, timeboxed periods of development
- Usually last two to four weeks
- Releases are the publishing of the software

Defining Iteration Types

- Development timeboxes
- Sprints
- Time for development

Iteration Zero

- Sets the stage for development
- Typically no deliverables for the customer
- Prepares to do the work

Iteration H

- Hardening sprint
- Wrap up sprint
- Used to stabilize the code
- Document the product
- Compile final assembly
- Final testing

Architectural Spikes

- Proof of concept
- Timeboxed effort to test the approach

Risk-based Spike

- Short effort to investigate risk
- Reduce or eliminate through mitigation
- Good for new technology and early in the project

Fast Failure

- Testing of different approaches for viability
- Good result before wasting time and money

High-level Planning – Visioning

- Prior to planning the first release
- Mapping out the overall effort of the project
- Product owner and sponsor
- Key team members
- Other major stakeholders

Outputs Of High-level Planning

- Updated prioritize backlog
- Coarse grained relative estimates for each user story
- Goals of the release
- Release date

Hosting A Release Planning Meeting

- All stakeholders represented
- Happens before each release
- The goal is to find which stories will be done in which interations for the release
- Also defines future iterations for future releases

Hosting A Release Planning Meeting

- Assess the prioritize backlog
- Reviews story sizing
- Sort the stories by release
- Define the initial outline or road map for the release
- Slice the stories as needed for the plan release

More about Slicing Stories

- Compound stories – includes other independent stories within
- Stories – one large complicated story; usually can't fit in one iteration

Iteration Planning

- Meeting for and facilitated by the delivery team
- Confirms goal for the current iteration

Iteration Planning, continued

- Discuss the user stories in the backlog
- Select the user stories for the iteration
- Define the acceptance criteria and write the acceptance test for the stories
- Breakdown the use for stories in tasks
- Estimate the task

Velocity for Finding Estimates

- Base of the teams velocity more accurate estimates can be created
- Burn up charts
- Burndown chart

Hosting the Daily Standup

- Also known as the daily scrum
- Call the stand up because the team's stand through the meeting
- 15-minutes duration
- Answer three questions
  - What have you worked on since our last meeting?
  - What will you finish today?
  - Are there any problems or impediments to your progress?

Rules for the Daily Standup

- People with tasks must attend
- Only people who have tasks can talk
- Address the entire team not the scrum master
- No side conversations
- Add new task to sticky notes if they are started
- Discuss issues after the stand up
- Solve problems offline

Problem Detection and Resolution

Overview of domain six for the PMI ACP exam

Domain 6 Summary – Problem Detection and Resolution

- 10 percent of the PMI-ACP exam
- 12 questions
- Practices for prevention identification and resolution of threats in issues

Problem Detection and Resolution Tasks

- Create a safe an open environment to surface problems
- Engage team in resolving threats and issues
- Resolve issues or reset expectations
- Maintain a visible list of threats and issues
- Maintain a threat list and add threat remediation efforts to the backlog

Four Themes In Problem Detection and Resolution

- Understanding problems
- Detecting problems
- Managing threats and issues
- Solving problems

Risk Management

- Risk are uncertain events or conditions
- An agile projects risk are always negative
- Probability and impact
- Risk identification and tracking

What's the Problem?

Problems and agile projects

Issues and Risk

- Risk are uncertain events they've not yet happened
- Issues are risk events that have occurred

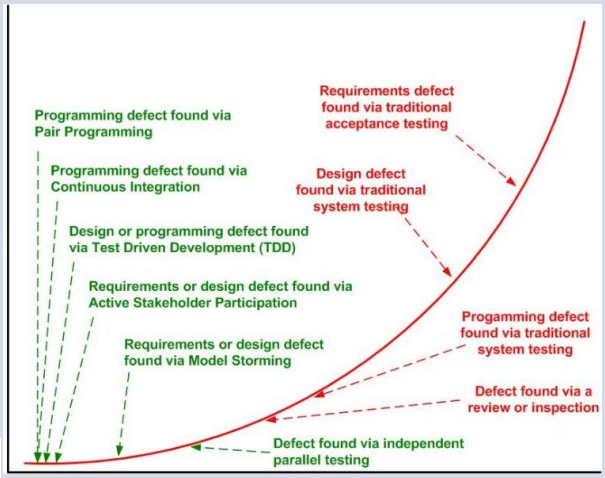
Truly Understand Problems For Resolution

- Problems can mushroom
- Problems can have ripple effects
- What happens in one area of the project affect all other areas of the project

Financial Impacts Of Problems

- The longer a defect is left unaddressed, the more expensive it will be to fix
- It is like going to the dentist
- The longer you wait the more expensive it will be

Where Issues Are Discovered



Reviewing Technical Debt

- Backlog of work caused by lack of regular clean-up maintenance and standardization
- Refactoring solves technical debt
- Red green refactor
- Refactoring cleans up and standardize is code to make it easier to support
- Refactoring should be included in estimates

Creating a Safe Environment

- PMI-ACP calls for safe and open environment
- People should feel comfortable experimenting for solutions
- When people get stuck they should share the problem with her teammates
- Safe environments are coaching opportunities

Understanding Failure Modes

- We make mistakes – mistakes happen
- We prefer to fail conservatively
- We prefer to invent rather than research
- We are creatures of habit
- We are inconsistent

Understanding Success Modes

- We are good at looking around
- We are able to learn
- We are malleable
- We take pride in our work

Creating Success Strategies

- Balance discipline with tolerance
- Start with something concrete and tangible
- Copy and alter
- Watch and listen
- Support both concentration and communication

Creating Success Strategies – Continued

- Match work assignments with a person
- Retain the best talent
- Use rewards that preserve joy
- Combine rewards
- Get feedback

Detecting Problems in Agile Projects

Finding problems and defects

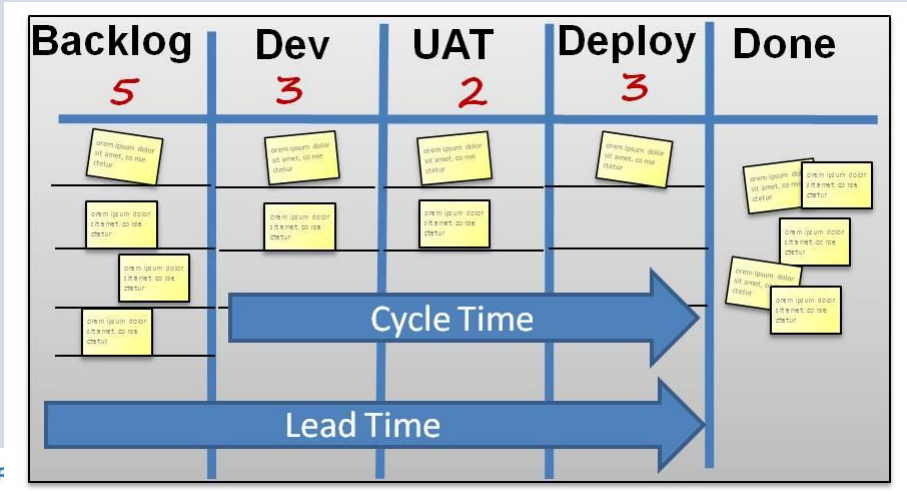
Start With the Daily Standup

- Are there any problems or impediments to the team members?
- This is the first step to detecting problems

Reviewing Lead Time and Cycle Time

- Lead time is how long something takes to go through the entire process
- For example from concept design to shipping
- Cycle time is a subset of lead time
- Cycle time is how long something takes to go through a part of the process
- For example from coding to testing

Lead Time and Cycle Time on a Kanban Board



Cycle Time WIP and Throughput

- Understanding the team's throughput allows for forecasting future capabilities
- Cycle time equals WIP / throughput

Project Cycle Time

- The project duration is the cycle time for the entire project
- Average amount of work the team can get done in a time.
- Productivity is the rate of efficiency at which the work is done
- Productivity could be the amount of work done per team member

Looking at Defect Cycle Time

- Defect cycle time is the amount of time between when the defect was discovered and when the defect was fixed
- The longer the defect cycle time typically the more expensive the defect

Defect Rates

- Defects that slip by the testing team are called escaped defects
- Escape defects are the most expensive to fix
- The defect rate measures the frequency of defects found
- An increase in escape defects signals a problem with a process

Variance Analysis

- The difference between what was planned and what was experienced
- Cost variance
- Schedule variance
- Other tracking items

Causes Of Variance

- Average day to day differences
  - Good days and bad days
  - Ups and downs
- Special causes of variance
  - Something unusual that causes a problem
  - Power went out for 2 days
  - Three team members caught the flu

Relying on Trend Analysis

- Measurements of past experiences are called lagging metrics
- Leading metrics provide a view into the future
- Trend analysis aims to predict performance or problems

Control Limits

- Upper and lower control limits
- Set boundaries and expectations for performance
- WIP and Kanban are a form of control limits

Managing Threats and Issues

Removing anti value in agile projects

Working With the Risk-adjusted Backlog

- Risk is considered anti value
- Goal is to attack high-risk items early in the project
- Items with the greatest value in greatest risk move to the top of the backlog

Backlog Grooming for Risk

- Stories are ranked based on business value and risk level
- The ranking is subjective often based on gut feeling or preference
- The return on investment for the project can be broken down per item
- The business representatives assign the ROI to each item in the backlog

Finding the Expected Monetary Value

- Expected monetary value is the worth of a risk event
- EMV =risk impact x risk probability
- EMV = $45,000 x 30%
- EMV = $13,500
- This is done for each risk in a probability impact matrix

Probability-Impact Matrix

Risk event	Probability	Impact	Ex$V
A	.60	-10,000	-6,000
B	.20	-75,000	-15,000
C	.10	25,000	2,500
D	.40	-85,000	-34,000

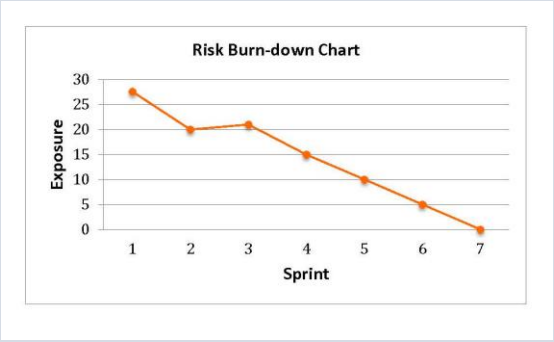
Contingency reserve = $52,500

Risk Severity

- Instead of risk probability and dollar amounts
- Probability and impact uses a simple scale
- For example low medium high
- Risk severity =risk probability x risk impact
- Risk severity = 3 x 1

Risk Burndown Chart

- Stacked area graphs of cumulative project risk severity
- Visual communication of risk events
- Severe these for each risk are plotted on top of one another to show the cumulative severity of the project
- Overtime risk should diminish so the chart diminishes as well



Solving Problems

Solving problems for agile projects

Problem Solving is Continuous Improvement

- Problem solving games to fix the problem before it happens
- Consider daily standup
- Iteration reviews and retrospectives
- Sprint planning sessions

Engage the Team in Problem Solving

- By asking the team for a solution we inherit consensus for the proposal
- Engaging the team excesses a broader knowledge base
- Team solutions are practical
- When consulted people work hard to generate good ideas
- Asking for help shows confidence
- Seeking others ideas models desired behavior

Considerations for Team Engagement

- Involve the team where it can be most helpful
- Solve real problems only
- Team cohesion is necessary
- Check in after project changes
- Be sure to follow through

Some Problems Can't be Solved

- Even with team engagement some problems won't find a solution
- Work around these problems
- Track and monitor the problems

Continuous Improvement for the PMI ACP exam

Reviewing Domain Seven: Continuous Improvement

Continuous Improvement - Exam Domain 7

- 9 percent of the PMI ACP exam
- 11 exam questions
- People, processes, and product

Continuous Improvement Tasks

- Periodically review and tailor the process
- Improve team processes through retrospectives
- Seek product feedback via frequent demonstrations
- Create an environment for continued learning
- Used values dream analysis to improve processes
- Spread improvements to other groups in the organization

Lessons Learned

- Lessons learned are captured in each iteration
- Allows lessons to be applied in next iterations
- Lessons are repeated until they are learned

Continuous Improvement and Quality

- Quality assurance is prevention driven
- Quality assurance is planning for quality
- Quality control is inspection driven
- You cannot inspect quality into a product

Kaizen

- Japanese word meaning change for the better
- Small incremental steps for improvement
- Plan Do Check Act
- Plan, develop, evaluate learn

Continuous Process Improvement

Improving processes in an agile project

Process Tailoring in Agile Projects

- Adapting agile for your environment
- There is some risk with tailoring
- Better to create processes for each project as needed
- Consider risk and reward

Risk and Reward of Process Tailoring

- First embrace traditional agile processes before attempting to change
- Second examine the motivation for changing processes

Using a Hybrid Model

- You can use elements from different agile models
- Combinations to best fit your environment
- There's no right or wrong hybrid

Systems Thinking

- Understanding the system's level environment
- Classifying projects by their level of complexity and uncertainty
  - Project requirements
  - Technical approach
- Looking for a balance between simple and highly chaotic

Process Analysis

- Reviewing and diagnosing issues with agile methods
- Trying to discover what does and does not work
- Often done after process tailoring

Looking for Anti Patterns Of Methodologies

- One-size-fits-all projects
- Intolerant processes
- Heavy methodologies
- Embellished methodologies
- Untried methodologies
- Used only once

Three Success Criteria

- The project got shipped
- The leadership remained intact
- The team would work the same way again

Methodology Success Patterns

- Interactive face-to-face communication is the cheapest and fastest channel for exchanging information
- Excess methodology weight is costly
- Larger teams need heavier methodologies
- Projects with greater criticality require greater ceremony

Methodology Success Patterns

- Feedback and communication reduce the need for intermediate deliverables
- Discipline skills and understanding counter process formality and documentation
- Efficiency is expendable in on bottleneck activities

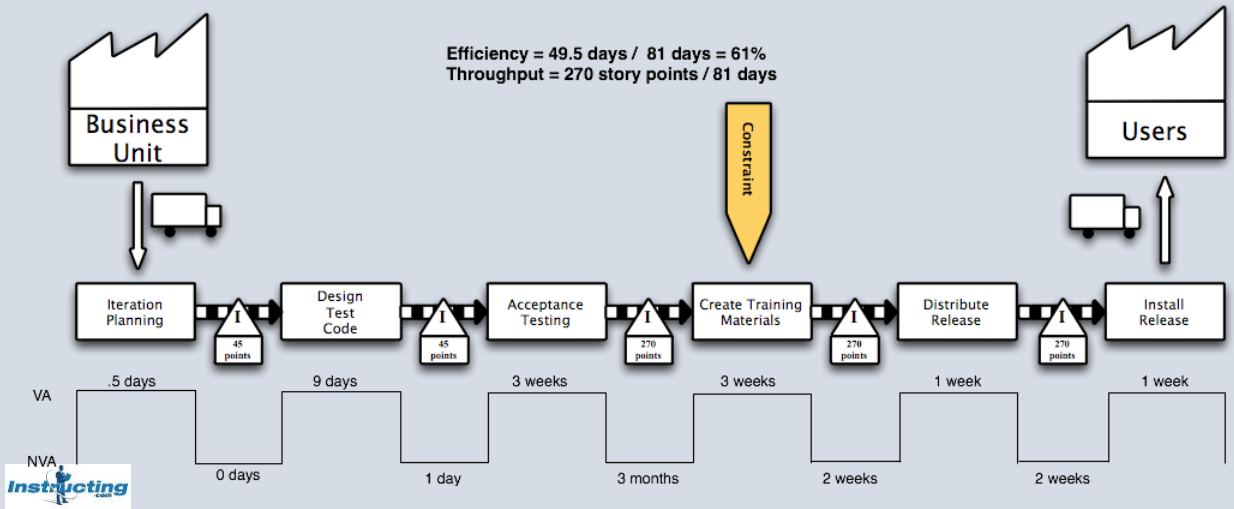
Value Stream Mapping

- Lean manufacturing technique adopted by agile
- A visual map of a process flow to identify delays waste and constraints

How To Create a Visual Stream Mapping

- Identify the product or service to be analyzed
- Map current processes steps, queues, delays, and information flows
- Review the map for delays waste and constraints
- Create a new value-stream map of the desired future state for the process
- Develop a road map for creating the optimized state
- Plan to revisit the process in the future to continually refine and optimize

Value Stream Mapping Example



Project Pre-mortem

- Aims to find failure points before they happen
- Imagine the failure
- Generate the reason for failure
- Consolidate the list
- Revisit the plan

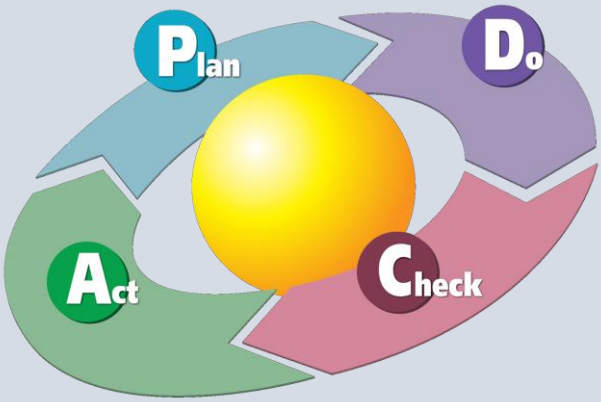
Continuous Product Improvement

Products are iteratively and incrementally improved

Product Reviews

- Product feedback
- Retrospective and demos

PDCA Cycle



Agile Review Rules

- Let the data speak for itself
- Respect individuals
- Diverge and then converge

Product Feedback Loops and Learning Cycles

- Does it meet the customers needs and expectations
- Does it work and all conditions
- Did we break anything while building this
- How can we improve efficiency
- How can quality be improved
- How can we share lessons learned

Product Feedback Methods

- Prototypes
- Simulations
- Demos

Approved Iterations

- Iterations and sprint reviews
- Held at the end of a sprint
- Demonstrates the new incremental build
- Business partner approved work
- Next sprint may begin

Continuous People Improvement

Helping people to improve in agile projects

Retrospective for People Improvement

- What is going well
- What areas could use improvement
- What should we be doing differently

Why You Need Retrospectives

- Improve productivity
- Improved capabilities
- Quality improvement
- Capacity improvement

Setting the Stage For a Retrospective

- Encourage participation
- Set the ground rules
- Define what people want from the retrospective
- Have people checking in with one or two words
- Ask the team to commit to focus
- Explorer shopper vacationer prisoner
- Working agreements for the retrospective

Gather Data in the Retrospective

- Used techniques to extract data
- Get people involved in contributing
- Facilitate the meaning for contributions

Methods For Gathering Data in the Retrospective

- The team creates a timeline
- Triple nickels – five groups spending five minutes on 5 ideas 5 times
- Color-coded dots – used color-coded dots to track your energy was high and low throughout the duration
- Mad sad or glad – track emotions throughout the timeline

Methods For Gathering Data During the Duration

- Satisfaction histogram – a bar chart showing satisfaction about particular areas or issues
- Team radar – an assessment of performance improvement
- Locates strengths - what went well, or not well, in the iteration
- Like to like – compares reactions to different iteration events

Generating Insights

- Evaluate the data
- Create insights based on the data
- Done with the project team

Brainstorming

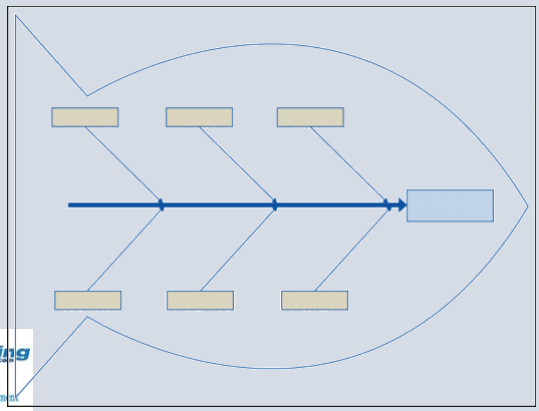
- Quiet writing
- Round robin
- Free for all

Five Whys

- Cause and effect exercise
- Small groups
- Ask why five times
- The goal is to find root cause

Fishbone Analysis

- Also known as a cause and effect diagram
- Also known as an Ishikawa diagram



Decide What To Do

- Create an action plan
- Short subjects – keep drop add
- Smart goals – specific measurable attainable relevant timely
- Circle of questions – each person ask a question on how to improve one of the issues in the next person of the circle answers the question
- Retrospective planning game – play in the task to reach process improvement goals

Close the Retrospective

- Plus / delta – what to do more of and what to do less of
- Help hindered hypotheses – feedback on the retrospective
- Return on time invested – the team discuss the benefits of the retrospective and gives a great
- Appreciation – the team gives appreciation to other team members based on efforts from the last iteration

Team Self Assessments

- Shores team self assessment scoring model
  - Thinking
  - Collaborating
  - Releasing
  - Planning
  - Developing

How to Use This Course

Working Smart to be Smarter

Creating A SMART Goal

Smart goals are

Specific
Measurable
Attainable
Realistic
Timely

Your PMI-ACP Goal

- Pass the PMI-ACP
- Pass or fail exam
- You can do this
- Qualifications
- Set a deadline

Create a Strategy

- 9 modules of learning – watch at least twice
- Study the flashcards daily through exam day
- Focus on the most important exam domains
  - Value-driven delivery, 20 percent
  - Stakeholder engagement – 20 percent
  - Agile Principles and mindset – 16 percent
  - Team performance – 16 percent
  - Adaptive planning – 12 percent
  - Problem detection and resolution – 10 percent
  - Continuous improvement – 9 percent

Create a Place to Learn

- Emulate the exam environment
- Quiet, no distractions
  - Emails
  - Texts
  - Visitors
- Two hours of quiet time
- Immerse yourself in the PMI-ACP

Complete the Course

- Take the course in any order you like
- Take notes as you move through the course
- Review your notes before leaving a module
- Ask questions! Use the discussion feature

Prepare to Pass the PMI-ACP Exam

- Complete your application as part of this course
- Create a deadline
- If you don't understand a concept or term – Ask!