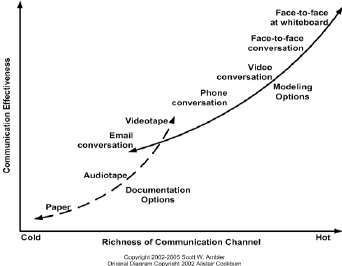
**ACP Notes**

Project: Temporary endeavor undertaken to create unique product, service or result

**Richness of Communication**

****

Best form of communication is face to face at whiteboard, while worst form of communication is email conversation.

Best form of documentation is videotape, while worst form of communication is paper.

**Project Selection Methods**

1. Payback period: Number of time units to recover initial amount invested
2. Net Present Value (NPV): Value of future inflow due to business in today’s value
3. Return on Investment (ROI): Amount gained from investment
4. Internal Rate of Return (IRR): ROI in percentage form

For NPV, ROI and IRR, higher value is preferable.

For payback period, lower value is preferable.

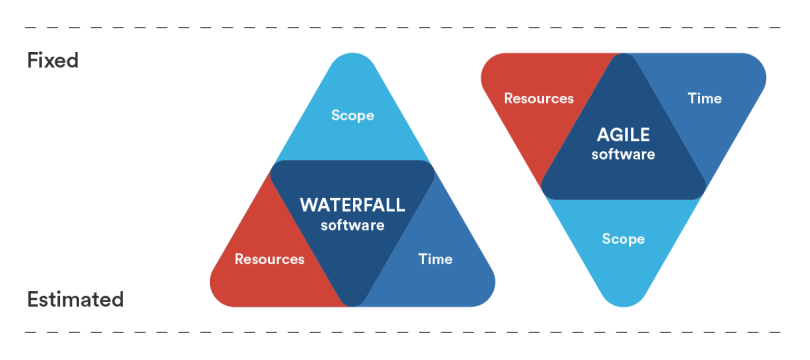
**Waterfall Methodology**

1. Define business requirements
2. Create project architecture
3. Programming
4. Testing
5. Debug errors

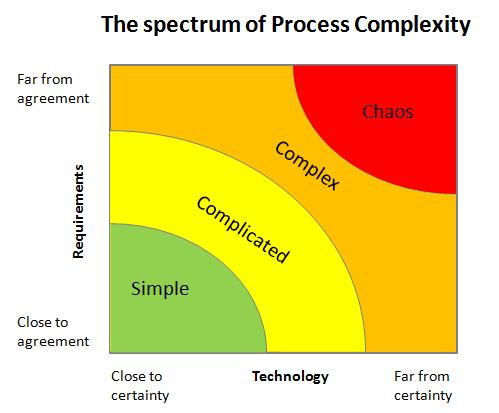
Issues related to waterfall methodology:

1. Assumes business requirements correctly captured at start
2. Changes to requirements may be risky and costly
3. Increase risk of miscommunication due to handoffs between different teams
4. Working software only available at later stage of the process

**Iron Triangle – Waterfall vs Agile**



**Spectrum of Process Complexity**



For simple, waterfall methodology is preferred.

For complex, agile methodology is preferred.

For complicated, either waterfall or agile methodology would work.

**Empirical Process Control**

Empirical process is implemented where progress is based on observation and experimentation, instead of detailed, upfront planning and defined process.

Three components of empirical process control:

1. Transparency: Provide visibility to aspects of process to those responsible for the outcome
2. Inspection: Timely checks on progress towards an iteration goal
3. Adaptation: Adjust process as soon as possible to minimize further issues

**Agile Manifesto**

Agile manifesto is a document that sets out the values and principles behind the Agile philosophy and serves to help development teams work more efficiently and sustainably.

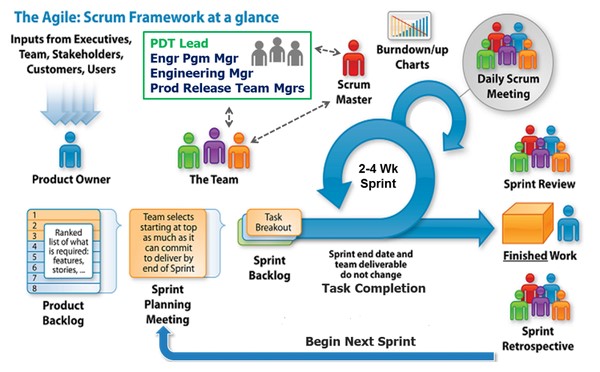
Values of Agile Manifesto:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Principles of Agile Manifesto:

1. Satisfy customer through early and continuous delivery of valuable software
2. Welcomes changes in requirements, even in later stages of development
3. Deliver working software frequently in shorter time scale
4. Business people and developers work together throughout the project through colocation
5. Build projects around motivated individuals with the environment and support required to succeed
6. Most effective method of conveying information is face-to-face communication
7. Working software is primary measure of progress
8. Agile process promotes sustainable development
9. Continuous attention to technical excellence with enough upfront design
10. Maximizing amount of work done is essential
11. Self-organizing teams is most preferable
12. Teams reflect on becoming more effective on regular basis

**SCRUM methodology**



Scrum is a structured framework for product development that is frequently used by agile software development teams.

Quick summary of scrum methodology:

1. Product owner is responsible for capturing business requirements, which are stored in product backlog.
2. A sprint planning meeting is conducted at the start of every sprint, where product owner explains user stories from the product backlog and the team pulls the work from product backlog to complete in every sprint.
3. The team is responsible to decompose business requirements into technical tasks, which are stored in sprint backlog.
4. On a regular basis (2 to 4 weeks), technical tasks need to be completed 100% before sent for sprint review with customer.
5. Throughout the sprint, a daily scrum meeting is conducted for 15 minutes under the direction of the scrum master.
6. After review is completed, any user stories with defects or improvements required is placed back into the product backlog by the product owner.
7. At the end of sprint, a sprint retrospective is conducted for the entire scrum team to reflect on their observations during the sprint and plan for changes to be made for future improvements.

In scrum, pig-chicken concept is involved. Chicken refers to parties that are not suppose to be actively involved in a certain project phase, while pig refers to parties that are suppose to be actively involved.

**Responsibilities of various parties in SCRUM team**

1. Product owner

* Ensures optimal business value is achieved based on product backlog prioritization
* Determines schedule, scope and cost of project
* Maintains product backlog
* Understands team capacity and capability
* Communicate changes in business with the team

|  |  |
| --- | --- |
| Do’s | Don’t |
| Say what needs to be done | Say how to do it or time duration to complete |
| Challenge the team | Bully team |
| Build high performance team | Focus on short term deliveries |
| Practice business value driven thinking | Stick to original scope |
| Protect team from outside noise | Worry team with uncertain changes |

1. Scrum Master

* Servant leader that helps the team and product owner to practice Agile methodology
* Not the team’s boss, but either a team member or ex-project manager
* Process coach, problem solver and protector
* Focus on coaching entire team at start and end of sprint
* Focus on coaching individuals at middle of sprint

1. The team

* Responsible for implementing the product
* 5 to 9 people per team recommended (bigger projects would require multiple teams working in parallel)
* Self-organizing: Working together within team to set realistic targets for sprint
* Cross-functional: Team with diverse set of skills

**Phases involved in SCRUM**

1. Release planning
2. Sprint planning
3. Daily scrum meeting
4. Product backlog refinement
5. Sprint review
6. Sprint retrospective
7. Hardening Sprint

**Release Planning**

Before the start of project, the team reviews strategies and vision shared by customer and determines method to map work from product backlog into iterations using product backlog, release backlog and iteration backlog.

Key tasks involved in release planning:

1. Establish project goals
2. Create user stories
3. Prioritize user stories
4. Estimate stories
5. Grouping stories
6. Setting release date

In addition, the team is responsible for setting up development and test environments with enough up-front design and architecture to get started

**Project Goals**

For establishing project goals, documents like project charter and business case required, which is signed by customer.

Business case breakdown:

1. Opportunity
2. Goals
3. Strategy
4. Product vision
5. Milestones
6. Investment
7. Expected payback

**Creating User Stories**

User stories are short, plain-language description of a feature, centered on who needs it and why.

User stories usually require defining personas, which represents different types of potential customers involved.

Three main components of user stories:

1. Card: User stories are visibly shown
2. Conversation: Not all details needed to be included in user stories to encourage detailed conversation between product owner and the team
3. Confirmation: The team confirms with product owner on acceptance criteria for every user story

Guidelines for user stories (INVEST):

* Independent
* Negotiable
* Valuable
* Estimable
* Sized appropriately
* Testable

Guidelines for tasks (SMART):

* Specific
* Measurable
* Achievable
* Relevant
* Time boxed

Epics represent large user stories that are usually located at the bottom of product backlog. Epics can be decomposed into smaller user stories until it is small enough that 1-2 person can complete the given user story within 3-4 days.

**Techniques for prioritizing user stories**

1. 100 points method
2. Multi-voting
3. CARVER (Criticality, Accessibility, Return, Vulnerability, Effect, Recognizability)
4. Monopoly money
5. MoSCow (Must have, Should have, Could have, Would have)
6. KANO analysis (Must have, Performance, Exciters)
7. Priority matrix (Based on criteria like urgency and importance)

Note that user stories are usually prioritized based on business value and risk. High business value and high business risk is prioritized first, while low business value and high business risk is prioritized last.

**Techniques for estimating user stories (time duration)**

1. Planning poker: Arbitrary relative estimates that requires consensus between team members without large discrepancy between smallest and largest estimates.
2. Program evaluation review technique (PERT): Average between estimates that are optimistic, pessimistic and most likely.
3. Affinity estimation: Assigns user stories according to size (XS, S, M, L and XL)
4. Delphi: Estimation of user stories is done anonymously and changes are made sequentially until no further changes are made
5. Wide-band Delphi: Similar to Delphi method, but initial meeting with all team members related to user stories in product backlog 1st.

**Setting release date**

Product owner will able to set release date based on estimates made by team from all user stories, velocity of the team (Number of points completed per iteration) and buffer for uncertainty and work improvements.

**Product and Story Roadmaps**

Product roadmap is a high-level representation of features or themes to be delivered in each release.

Most important features from every functionality is usually prioritized on earlier releases.

First release known as “Walking Skeleton” is the minimum working version that creates value for stakeholders, while helping to prioritize product backlog.

**Definition of Done**

Definition of Done defines meaning of product increment to be done at the end of iteration.

The entire agile team is responsible for defining definition of done before start of 1st iteration.

Definition of Done can change over time but most ideal to be consistent

**Last Responsible Moment Decision Points**

Last responsible moment decision points refer to instances at which failing to make decision eliminates important alternative

**Wireframes**

Wireframes are sketch blueprints that represents looks of potential end product

**Sprint Planning**

At the very start of every sprint, product owner, scrum master and the team plan the sprint accordingly.

Meeting is usually time-boxed to 2 hours per week in sprint.

Scrum master explains the goal of the sprint to the entire team.

Product owner explains to team members about the highest priority user stories in the product backlog, explaining details and answering questions from the team.

The team makes an initial estimate on how many user stories can be completed, creating an iteration backlog and confirms with product owner on their target of number of user stories to complete in a given sprint.

**Daily Scrum Meeting**

Purpose of daily scrum meeting is to enable the team to provide brief updates to one another and make challenges visible to everyone.

Meeting is usually time-boxed to 15 minutes with no discussion allowed.

Further discussions with individuals involved only after the daily scrum meeting

Daily scrum meeting requires all team members to explain what they have completed yesterday, what they will do today and any challenges identified. Team members will also require to replan on time duration remaining for pending user stories to be completed.

Note that product owner is not required to be involved in this meeting and can play the role of an observer instead.

**Product Backlog Refinement**

During product backlog refinement, the entire agile team looks at upcoming product backlog items that will be worked on in the next 2 to 3 iterations and start decomposing bigger user stories into more refined ones.

There is no fixed timing or format for conducting product backlog refinement.

This phase usually happened in the middle of every sprint and during release planning phase.

**Sprint Review**

The team aims to have “Done” increment at end of every iteration for review.

Done increment represents useful subset of functionality that meets agreed acceptance criteria and built to agreed definition of done, which is potentially shippable.

Purpose of sprint review is for inspecting and adapting to the product.

Meeting is time boxed to 1 hour per week in sprint.

Product owner, scrum master, the team and stakeholder will get hands on with finished functionalities of the product.

Any features that failed to meet definition of done or features that failed during integration testing or features that require improvements are noted by the product owner and its corresponding user stories are added back into the product backlog.

Integration testing involves testing functionalities completed from different sprints to check for compatibility issues.

**Sprint Retrospective**

Purpose of sprint retrospective is for inspecting and adapting to work practices.

Meeting is time boxed to 1 hour per week in sprint.

Product owner, scrum master and the team talk about their experiences during the spring, both positive and negative, while creating a suitable plan of action to improve practices in future sprints.

**Hardening Sprint**

Hardening sprint involves improving quality of existing completed user stories, without adding on additional user stories to complete.

This phase usually occurs before the end of release date, but can also be done for every sprint, provided there’s buffer of time remaining before end of given sprint.

**Osmotic Communication**

Osmotic communication involves individuals who are not directly involved in discussion between active members, but able to listen in close proximity.

This allows other members to be able to learn new knowledge from them during the discussion process.

**Large Scale Agile**

Characteristics of large-scale Agile teams:

1. Every cross functional and self-organized team will work on smaller portion of product backlog
2. One chief product owner oversees entire product backlog and multiple product owners will be responsible for their own product backlogs.
3. Every team will have one scrum master and all scrum masters will meet up together daily for scrum of scrums meeting.
4. All teams have extra buffers in every sprint (i.e. feature buffer, project buffer, feeding buffer and resources buffer)
5. Direct communication between different teams is highly encouraged

Note that dependencies involved between different teams can be resolved with one of the methods:

1. Synchronize priorities of user stories
2. Handover user stories to other teams
3. Create dummy user story for integration testing first before replacing with finished version by other teams at later stage.

**Distributed Agile**

Scenario 1: Product owner and team in two different locations

* Scrum master should be located with team members
* After daily scrum meetings, scrum master emails list of challenges to product owner for assistance clearing if relevant
* Product owner and team members should have weekly video calls to clear pending challenges and answering questions
* Product owner should meet up with team at start of 1st iteration for face-to-face interactions

Scenario 2: Single team split into different locations

* All team members should be collocated for the first iteration
* Encourage ongoing ambassadorship
* Multiple forms of continuous live communication
* Agile artifacts maintained electronically

**Earned Value Management**

Earned value management is a technique of performance measurement that integrates project scope, cost and schedule.

Formulas and terminologies:

1. **BAC (Budget at Completion)**: Budget planned at completion
2. **PV (Planned Value)**: Budgeted cost of work scheduled

* PV = BAC \* % work planned
* % work planned = [[Total story point planned/Total num. sprints planned] \* Num. sprints completed]/Total story point planned

1. **EV (Earned Value)**: Budgeted cost of work performed

* EV = BAC \* % work performed
* % work performed = Story points completed/Total story points

1. **AC (Actual cost)**: Actual budget at completion
2. **CV (Cost variance)**: Identifies if project is under or over budget

* CV = EV – AC
* +CV: Under budget; -CV: Over budget

1. **SV (Schedule variance)**: Identifies if project is ahead or behind schedule

* SV = EV – PV
* +SV: Ahead of schedule; -SV: Behind of schedule

1. **CPI (Cost performance index)**: Value of work completed compared to actual cost

* CPI = EV/AC
* CPI>1: Under budget; CPI<1: Over budget

1. **SPI (Schedule performance index)**: Measure of progress achieved compared to progress planned

* SPI = EV/PV
* SPI>1: Ahead of schedule; SPI<1: Behind of schedule

1. **TCPI (To complete performance index)**: Efficiency level to achieve in remaining work to meet remaining budget

* TCPI = (BAC – EV)/(BAC – AC)

**Other Agile Methodologies**

1. Extreme Programming (XP) principles

* Simple design
* Refactoring
* Test-driven development
* Pair programming
* Code standard
* Sustainable pace
* Collective ownership
* Continuous integration
* Metaphor (Wireframe)
* Small releases
* Customer test
* Whole team
* Planning game

1. Lean principles

* Eliminate waste
* Amplify learning through osmotic communication
* Decide as late as possible
* Deliver as fast as possible
* Empower team
* Build integrity
* See the whole

1. Kanban principles

* Visualize workflow
* Limit work in progress
* Focus on flow
* Continuous improvement