**Chapter 1**

Spike – Rapid experiment used as a way to look for which best path to take. Spikes are encourages, so team is dynamic and regularly looking for best way to proceed.

**Chapter 3**

**Agile Manifesto**

* Individuals and interactions over process and tools
* Working software over comprehensive documentation
* Customer Collaboration over Contract Negotiation
* Responding to change over following a plan.

Agile Principles

1. Satisfy customer through continuous delivery of valuable software
2. Welcome changing requirements to harness customer competitive advantage
3. Deliver working software frequently with preference on short time scales
4. Business and Tech must work together daily
5. Build project around motivated individuals, support them.
6. Face to face conversation is the best
7. Working software is the primary measure for progress
8. Agile promotes sustainable development
9. Attention to technical excellence and good design enhances agility
10. Art of maximizing amount of work not done is essential – SIMPLICITY
11. Self organizing teams are the best
12. Retro is important to adjust and reflect on how well they did.

**Declaration of Interdependence**

* Increase return on investment through continuous flow of value
* Reliable results by constant interaction and shared ownership with customers
* Expect uncertainty and manage it through iterations
* Individuals are the ultimate source of innovation – create their environment to support them
* Group accountability and shared responsibility is good for team effectiveness
* Situational specific strategies, process, and practice improve efficiency.

**Chapter 4**

Present Value/NPV

* Bigger Present value and Net present value makes a project more attractive.
* NPV have already factored in the cost of the project

Internal Rate of Return(IRR) - Project return as an interest rate.

Return on Invesment(ROI) – Return you make by investing in something. (Benefit – Cost) / Cost

Chartering – Act of pulling together important documents to create the charter

**Chapter 5 – Optimal Team Size**

* Scrum teams should be 7 +/- 2
* Colocation puts the teams together in a single room.
* Agile Tooling – Increase the sense and the bond of team to encourage participation/ communication/ interaction/ motivation.
* **Groundrules** – Unwritten rules about expectations of the project. Need to be communicated clearly so everyone knows expectations

**Chapter Six – Agile Planning**

* Stakeholder – Anyone with interest in the project
* Customer – Product Owner acts as single voice for customer

Analysis – Pre work involved to get ready for sprint

Innovation games – Another way to gather ideas and requirements.

* 20/20 Vision – All Features are prioritized one at a time from least important to most important.
* The Apprentice – One person interacts with system, that person makes comments or notes about what could be improved in the product
* Buy a feature – Stakeholders are shown features and estimate prices for their development. Product owner uses finite amount of cash to buy features
* Prune the Product tree- Map products and large/small components to a tree
* Root Cause Analysis – Looking beyond the symptoms to get to core issue causing the problem
* Force Field Analysis – ways to analyze change and understand forces for it and against it

Parking lot – Way to keep discussions focused and productive by posting extra topics on the wall.

**Planning & Estimates with Agile**

1. Cone of Uncertainty – Estimates should get sharper over time. Estimates early in the project before details are understood will not be as accurate.
   1. Measures level of variability and time
2. Product Vision Statement – Elevator statement for the product –typically product owner develops this first
3. Product Roadmap – Scheduled milestones for the product. Communication tool – this does not lock down the plan
4. Personas and Extreme Personas – Technique used to understand a product by putting yourselves in the shoes of the customer “persona”
5. Wireframe – Quick and easy to make, gives the user and developer a starting point for discussions
6. Agile Themes – Themes are assigned to iteration to high level document what is happening
7. Epic Stories – Capability is a large block of functionality that can span multiple iterations

**Xtreme Programming**

1. ***Independent , negotiable, valuable, estimateable, small, testable***

**Story Maps** – Broken down by disaggregation, initial node is a large user story or an “epic”

Feature – Something that adds value to the customer

Minimally marketable feature (MMF) – Smallest grouping of functionality that can add value to the user.

Tasks – Small tasks that build up to a user story.

Value Stream Mapping – Analyzing chain of process with goal of eliminating waste

Progressive Elaboration – You will not know all the characteristics about a project until you are already in

Rolling Wave Planning – Technique used to plan in stages instead of doing so early in the project.

Test-First Development – Have an idea of how a software should be used before you start developing. Test cases are written before hand so developer can structure routine more easily.

Definition of Done – Common understanding of what is rules need to be defined so that the project is considered “Done” for that iteration

**Agile Estimation**

Relative Sizing – Arrangement of stories ranked roughly in relative order of difficulty.

Wideband Delphi Estimating – Allows people to estimate in the absence of any group pressure while still leveraging wisdom and experience of entire group

Planning Poker – Use cards to show all at once the level of difficulty for each user story

Affinity Estimating – Rapid sizing of user stories using T-Shirt sizes. Usually used when backlog is huge, and too many user stories to plan.

Ideal Time – How much time a task would take with all the necessary resources at hand with ZERO interruptions or distractions.

Agile Modeling – Mapping out processes so that team/stakeholders can review before they are implemented.

Timeboxing – Fixed/reasonable amounts of time to complete work on features or user stories. Allows team to focus on important features while maintaining sense of urgency.

Continuous Integration – All code changes checked in is built and tested at the end of each day or even more often.

**Chapter 7 – Working with Agile**

Velocity – Story points, different for each scrum team. Can be used to see at a quick glance how the teams performance is

Cycle Time - Time it takes for a feature from inception to completion. Start is when it enters the backlog, completion depends on teams definition of done.

Burn Rate – Cost of a team and how much a certain release would cost using iteration days

Product Backlog – List of all functionality planned for projected sorted by value to customer. Product owner maintains the product backlog, it is his or her job to add new user stories as they are generated. **Backlog Grooming** is the process where product owner cleans up the product backlog to ensure it remains straightforward and relevant.

D.E.E.P. – Detailed Appropriately, Estimable, Emergent, Prioritized

Iteration Backlog – Functionality that will be completed during the current iteration. Functionality is broken down Into engineering tasks and team self organizes to decide who will work on what. This is maintained by the team and kept visible for all to see.

Product Quality - Quality Is essential and irreplaceable, if there is an issue with quality, root cause analysis should be performed to understand underlying factors.

Escaped Defects – Errors that escaped testing by developers and quality control and in to the customers hands. Important metric to track and keep track of.

Verification and Validation – Verification is conferenced with functionality meeting specification. Validation ensures that it does what it was intended to do.

Agile Smells – Ways to describe and quickly recognize and diagnose common problems so that a remedy may be pursued.

Refactoring – From time to time, reorganize code to organize it more cleanly and straightforward. DOES NOT CHANGE CODE BEHAVIOR. Complete regression is performed after to ensure streamlined code behaves same way.

Information Radiators – Information that is posted publicly on a wall that is highly visible. Gives stakeholders information they seek.

Osmotic Communication – Colocation helps team members pick up information without actually listening in on it. Distributing culture, best practices, and helping members acclimate to projects.

Burn-Down Charts and Burn up Charts - Burn down charts are update daily, burn-up charts are updated at the completion of each iteration.

Kanban – Represents WIP and puts team limits to help WIP. Kanban board is a component of an information radiator.

Earned Value Management – Measured at the iteration level, measure of how work is progressing against the plan.

Cumulative Flow Diagrams – Getting the optimum level of flow of WIP through the system.

Kaizen - Creating a plan for improvement, making very small changes, and measuring the impact and repeating. Plan-DO-Check-Act in smaller cycles.

Spikes – Experiment that can be performed quickly to help teams decide which fork in the road to take.

Risk Adjusted Backlog- Risk has to be considered when it comes to customer value. Risk is always tied to value.

**Chapter Eight – Coaching with Agile**

Daily Standup – What did you do yesterday, what do you plan on working on today, are there any obstacles or impediments in your way?

**Iterations/Sprints**

1. Short Planning session
2. Operate for a period of time
3. Work on agreed upon tasks
4. Ends cycle with deliverable working software and brief retro

**Retrospective** – What worked well and what didn’t work well, how can we improve? Customer/Product owner usually NOT in attendance.

**Stakeholder Management** – Typically its always best to involve your stakeholders as you want to build a product closest to their expectations.

Stakeholder Analysis – A way to identify all your stakeholders and their interests to gauge what the product should be doing

Emotional Intelligence – Self awareness and social awareness

**Conflict Resolution**

1. Level 1 A problem to Solve – Conflicting goals, values, or needs. Not personal yet
2. Level 2 Disagreement – Problem cannot be defined, starts to become personal, self protection becomes important. Trust becomes low
3. Level 3 Contest – Different sides are In a win or lose situation. Recruit allies for their cause. Generalizations rise about each side.
4. Level 4 Flight or Fight – Us or them scenario. Only one winner. Language becomes about principles rather than issues
5. Level 5 Intractable Situation – Issues no longer the point. Conflict must be won at all costs. Win-win is no longer acceptable. Separation of team should occur.

**Servant Leadership**  - Serve first and lead second. Focus on needs of the team rather than managing them. Helps if they are collocated/active in daily team meetings and activities to help team bond.

**Adaptive Leadership** – Adapt to environment to lead effectively. Focus on value adding activities versus doing things the way they have always been done.

**Chapter 9 – Agile Methodologies**

**SCRUM** – Three pillars – Visibility, Inspection, Adaptation. Highly iterative and always looking to improve little by little over time.

1. Product Owner will present overall vision for the project. Communicated using business terms. “What” and “How” the product and system will work.
2. Product Owner will then state product requirements in a prioritized list called the **product backlog**
3. Product Backlog is divided in to releases which are grouping of requirements that go together
4. Sprint Begins with a Planning Meeting.
   1. Planning meetings can last a maximum of eight hours that is divided in half. First half has the product owner discussing highest priority features with the team. Team interacts and asks questions about featured and ultimately picks functionality they think they can deliver within 30 day sprint.
   2. Second half of meeting, the team maps out the sprint and defines things down to a task level. Team plans to complete sprint and placed stories are placed in to Sprint Backlog.
5. Daily Stand up – 15 minute meeting every morning.
6. At the end of a 30 day sprint, team meets for a half-day Sprint review. Team and key stakeholders are in attendance and purpose is to show new product functionality while discussing features for the next sprint.
7. Every sprint ends with “shippable” functionality. Product Owner decides whether this functionality will deploy right away or be rolled with a future release.
8. Team retro at the end of a 30 day sprint.

Roles in Scrum

1. Product Owner – Stakeholder representative and voice of the customer, they control the product backlog and prioritize the features. They work closely with the team and usually sit with them. Product Owner writes user stories.
2. Team – Self organizing and self-managing. Primary job is to develop the product. Ideally team members are interchangeable and ultimately responsible for result.
3. Scrum Master – Follow scrum process, follow agile principles, encourage team members to facilitate daily standup and other meetings.

**Xtreme Programming (XP) –** Disciplined agile focuses where team and customer are HIGHLY focused.

1. Customers define application features with user stories
2. Software developers work in pairs
   1. One works writing the code
   2. One works on higher-level issues such as integration and functionality
3. Iterations are typically shorter than scrum – 1-3 weeks.
4. At the start of the iteration, team and the customer will meet to discuss user stories and features customer wants to see.
5. Team breaks down user stories in to tasks and assess level of effort by assigning story points.
   1. Rule of thumb is that team should NOT take more story points in the current iteration than he or she was able to complete in the last iteration.
6. After planning, teams work in pairs to write test cases – this is know as **test-driven development**
   1. Each user story has acceptance criteria that must be passed.
7. Twelve Practices make up XP
   1. Planning Game – Maximize value by writing user stories, estimating effort, and agreeing upon functionality for an iteration
   2. Small Releases – Get functionality in to customer hands with MMF(Minimally marketable features
   3. Metaphor – Encourage broad and creative thinking
   4. Simple Design
   5. Testing – Test cases are written before actual code to ensure program will pass.
   6. Refactoring – Optimizing Source code
   7. Pair Programming
   8. Collective Code Ownership
   9. Continuous Integration
   10. Sustainable pace
   11. On-site Customer
       1. Write user stories and define his or her acceptance tests for the functionality
8. XP Roles
   1. XP Coach – Same as scrum coach
   2. XP Customer – Product Owner
   3. XP Programmer – Dev Team
   4. XP Tracker – Evaluates and communicated progress against the plan. Release plan(user stories), iteration plan(tasks), and the acceptance tests.
   5. XP Tester – Helps customer take acceptance criteria and translate them to acceptance tests. Responsible for executing tests and communicating the results to team.

**LEAN**

1. Eliminate Waste – Ruthless in eliminating non-value adding activities(planning, meeting, etc)
2. Amplify Learning – Continuous learning and improvement
3. Decide as late as possible – Embrace adaptive change, by pushing decision out it keeps more options open
4. Deliver as fast as possible
5. Empower the team
6. Built Integrity in
7. See the whole