Agile Software Project Management

Instructor – Muhammad Sudais

Agile Project Management Process

Chapter 8

Introduction to Agile Project Management

- Ensures flexibility, collaboration, and continuous improvement.
- Focuses on value through incremental development.
- Agile processes are iterative, adaptive, and customer-driven.
- Traditional vs. Agile: Agile focuses on rapid iterations and customer feedback.

Project Feasibility

- Determines whether the project is viable and valuable.
- Steps:
  - Create Business Case: Justifies project selection.
  - Define Project Vision: Establishes long-term objectives.
- Considerations: cost, technology, business impact.

Agile Project Initiation

- Steps:

1. Create Agile Charter (Project goals and constraints).
2. Assign Project Staff (Roles and responsibilities).
3. Develop Project Backlog (Prioritized features list).
4. Create Estimates (Time and cost assessments).
5. Develop Roadmap with Story Mapping (High-level planning).

Creating an Agile Charter

- A guiding document defining project objectives.
- Differences from traditional charters:
  - More flexible and adaptive.
  - Authorizes entire Agile team, not just the manager.
  - Focuses on collaboration and high-level goals.
- Includes mission statement, success criteria, risk assessment.

Assigning Project Staff

- Agile teams are self-organizing and cross-functional.
- Key roles:
  - Product Owner: Defines priorities, manages backlog.
  - Scrum Master: Facilitates Agile processes.
  - Development Team: Builds and tests the product.
  - Stakeholders: Provide input and validate outcomes.

Assigning Project Staff

- *Self-managed:* The agile team should be able to function effectively in a self-regulating environment.
- *Colocated:* Agile teams work better when everyone is physically in the same location.
- *Small team:* The agile team should "typically" not have more than 6-10 members.
- *Single backlog:* The agile team should work from only one backlog.
- *Commitment:* The agile team should be committed to building and delivering a high-value product.
- *Communication:* The team must be effective with face-to-face communications.
- *Accommodate change:* The agile team should be able to embrace changes and accommodate unplanned work.
- *Create reasonable estimates:* The team must be able to create reasonable estimates for the work that they agree to complete and must deliver what they agree upon.
- *Continuous improvement:* The team must continuously find ways to improve its performance.
- *Cross-functional:* The team members must be cross-functional and have the capability to perform in multiple roles as needed.
- *Sustainable pace:* The agile project team must be able to maintain a sustainable pace throughout the project.

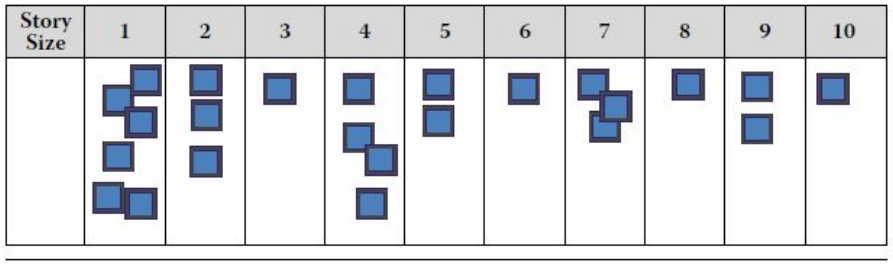
Developing the Project Backlog

- A prioritized list of features and tasks.
- Includes:
  - User stories (feature descriptions from user perspective).
  - Technical tasks (engineering needs, dependencies).
  - Bug fixes (defects to be addressed).

Estimation Techniques

- Agile teams use relative estimation techniques.
- Common approaches:
  - Affinity Estimation: Grouping tasks by complexity.
  - Planning Poker: Collaborative estimation.
  - Cost Estimation: Assessing overall effort.
    - Calculate the size of the project based on the total number of story points.
    - Calculate the amount of work to be completed on the project in person-days.
    - Determine the project schedule based on the calculated person-days.
    - Calculate the total costs of the project with labor rates and other associated costs.

Affinity Estimation



Total Project Costs

Name	Rate ($)	Total Hours	Total Costs ($)
Tom	75	1,903	142,725.00
Bruce	89	1,903	169,367.00
Donna	87	1,903	165,561.00
Mary	72	1,903	137,016.00
Total Project Costs			614,669.00

Roadmap with Story Mapping

- Helps visualize the product journey.
- Breaks work into epics, stories, and tasks.
- Prioritizes features based on business value.
- Ensures alignment with customer needs.

Release Planning

- Defines how and when product increments will be delivered.
- Steps:
  - Break down Epics into smaller stories.
  - Estimate stories using Planning Poker.
  - Create a Release Plan outlining timelines.

Breakdown Epics

- *Epics* are large user stories that need to be further broken down.
- The breakdown process results in smaller, more manageable, user stories.
- There is no right or wrong way to break down an epic
- A hierarchical approach is recommended with the end result being a task
- It is estimated by the development team and used to build a product feature

Estimate Stories with Poker Planning

- This estimation technique is based on obtaining team consensus to estimate the size of user stories.
- Planning poker starts with all team members being assigned a single deck of poker cards.
- The cards in each deck are numbered using the Fibonacci* sequence (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...).
- The numbers represent the estimated level of complexity based on effort and time.
- The Product Owner leads the estimation game and presents each user story to the development team.
- Each agile team member evaluates the user story and attempts to understand the complexity associated with it.
- Each developer then selects a card from his or her deck that describes his or her estimation for the user story being discussed.

Create Release Plan

- Determine what needs to be accomplished in terms of goals and objectives.
- Creating the release plan is not done blindly;
- The business goals and the team's velocity (capacity) is very relevant.
- The priority of the user stories is also an important factor regarding what goes into the release plan.
- A consensus must be reached in terms of the goals of the release plan.
- The development team must make a realistic commitment as to the amount of work that can be completed for each planned release.

Iteration 0 (Preliminary Phase)

- The preparation phase before the first sprint.
- Key activities:
  - The project is initiated.
  - Support and funding are established for the project.
  - Stakeholders are active and participating in the project.
  - The team is established.
  - The agile environment is created.
  - The initial system architecture is modeled.

Iteration Execution (1–N)

- Agile iterations include:
  - Daily Standups: Short team meetings.
  - Coding: Developing product features.
  - Testing: Automated & acceptance testing.
  - Burndown/Burnup Charts: Track progress.
  - Iteration Retrospective: Continuous improvements.

Iteration R

- The purpose of iteration R is to deploy the product to the production environment.
- This iteration is separated from iterations 1–*N*.
- Why does this separation occur?
  - Development team can continue to build functionality for the next release
  - Current release to be moved to production.
- The following activities occur during iteration R:
  - Complete project documentation.
  - Formal testing (security, performance, integration, or regression).
  - Deploy the software to production.
  - Celebration, Celebration, Celebration!!!!!

Coding

- The Agile Manifesto and its guiding principle state: "*Build projects around motivated individuals; give them the environment and support that they need and trust them to get the job done*."
- According to Ambysoft, Inc. (2013), a common best practice on the agile team is to provide the development team with *sandboxes* in which to develop code.
- What is a sandbox? It's just a fancy name for a respected technical development environment with a "well-defined scope."
- The sandbox reduces risk because it protects against technical errors.
- There is less worry on the agile development team because access is limited, which results in fewer mishaps.

Coding - Sandboxes

- Specific categories of sandboxes include:
  - *Development Sandbox:* Development environment where only the development team has access to develop the code for the product.
  - *Integration Sandbox:* Build environment from which each project team works. Code is integrated from all of the development activities to ensure it will work well when integrated.
  - *Demo Sandbox:* Working code environment where demos can be carried out for the client.
  - *Preproduction/Test Sandbox:* Staging environment that is generally a replication of the live production environment. Systems testing is executed in this environment prior to the release to production.
  - *Production:* This is the live environment for the completed product.

Execute Acceptance Tests

- Ensure that the product meets the requirements from the customer's perspective.
- These tests are designed to verify the functionality of the product based on actual demonstrations.
- The customer or their representative performs the actual testing to ensure that the product performs as expected.
- Acceptance testing (or user acceptance testing) is executed during development iterations.
- There are four steps to the acceptance test case and demonstration process.
  1. The requirements are discussed and the customer is asked questions so that the proper acceptance test can be developed.
  2. The acceptance test cases are entered into an acceptance testing tool.
  3. Code is developed and acceptance test cases are attached and executed. If the code is attached to the correct tests, the expectation is that the test will pass; otherwise, if not attached correctly, the tests will fail.
  4. The development team demonstrates the software to the customer using the automated acceptance tests.

Create Test Cases

- Agile projects use test-driven development to create test cases.
- Test-driven development requires that the development team envision how the product's functionality works prior to the creation of the code.
- Tests are then typically written in a unit testing language such as JUnit.
- The initial test will fail merely because the code would not have been created yet for the functionality.
- The development team writes code and runs the test case until the code has successfully passed the test.

Execute Automated Testing

- Automated tests are used for test-driven development and acceptance test-driven development.
- TDD, an XP concept, is used to implement coding and testing cycles to ensure that the code functions as intended.
- The test is developed before the code.
- When the code actually passes the test, the functionality is considered to be completed.
- On the first run of the test, failure occurs because the code has not yet been developed.
- A cycle of test execution and writing code is repeated until the test passes.
- Development efforts continue in the case where additional functionality is needed
- Otherwise when all tests pass, the code is then refactored.
- ATDD focuses on the business requirement rather than the code.

Definition of “Done”

- Agile methods place great emphasis on defining what “done” means
- A simple but effective way to define what “done” means is to document the criteria that define what is meant so that everyone is clear on its meaning.
- *Done* could describe the following conditions or a combination of conditions such as:
  - All stakeholders are in agreement that everything is done.
  - All issues have been resolved.
  - Customer signoff has occurred on all user stories for an iteration and/or release.
  - Testing is completed.
  - To summarize, the criteria for done should be agreed upon, recorded, and approved by all stakeholders.

Answer Client’s Questions for Sign-Off

- The customer and project stakeholders gather for a meeting to review the prioritized backlog items.
- The product’s functionality is validated and participants ask questions and provide feedback.
- The customer has the final say as to the acceptance or rejection of a
- Decision is based on the previously established acceptance criteria.
- The customer does not have the authority to change the agreed-upon acceptance criteria once the iteration is underway.
- Once the customer agrees that the user story is done, the prioritized backlog item is signed off.

Prepare Stories for Next Iteration

- The preparation of user stories is simply to outline requirements for the product's functionality. In the case of Scrum, for example, a user story should provide the following information:*
- *Who?:* Represents the user role executing the functionality
- *What?:* Represents what the user role should be able to perform in the system
- *Why?:* Represents the reason why the user needs to perform the desired function in the system

Daily Standup Meeting

- The daily standup meeting is time-boxed to a maximum of 15 minutes.
- Each team member (typically just the development team) is asked three questions:
  1. What have you worked on since we last met?
  2. What do you plan to complete today?
  3. Are there any obstacles in the way of completing your work?

Update Burndown/Burnup Charts

- Both the Burndown and Burnup Charts are used to show how much progress has been made on the agile project.
- The Burndown Chart reveals the amount of remaining work that needs to be completed
- The Burnup Chart reflects what has been delivered for the product.
- Can be tailored to show progress based on time or the number of iterations.
- When using iterations as a measure of progress, story points show scope levels per iteration.
- When using the amount of time as a measure, then these charts show the approximate amount of time remaining.
- Important to note is that these charts can be created in Microsoft Excel.

Iteration Retrospective

- Many of the agile methods conduct retrospectives as
- A way to learn, to reflect, and to provide the foundations for making improvements in the agile process.
- Occurs after a single iteration
- The agile team members discuss ways to improve their techniques.
- It occurs during the agile project
- There is time to implement the lessons learned information on the current project.
- The team has the opportunity to:
  - Improve its performance by increasing productivity.
  - Improve on knowledge attainment by sharing information.
  - Improve on product quality by undermining the cause of defects.
  - Improve on making processes more efficient which in turn will increase team capacity.

Iteration Retrospective - Phases

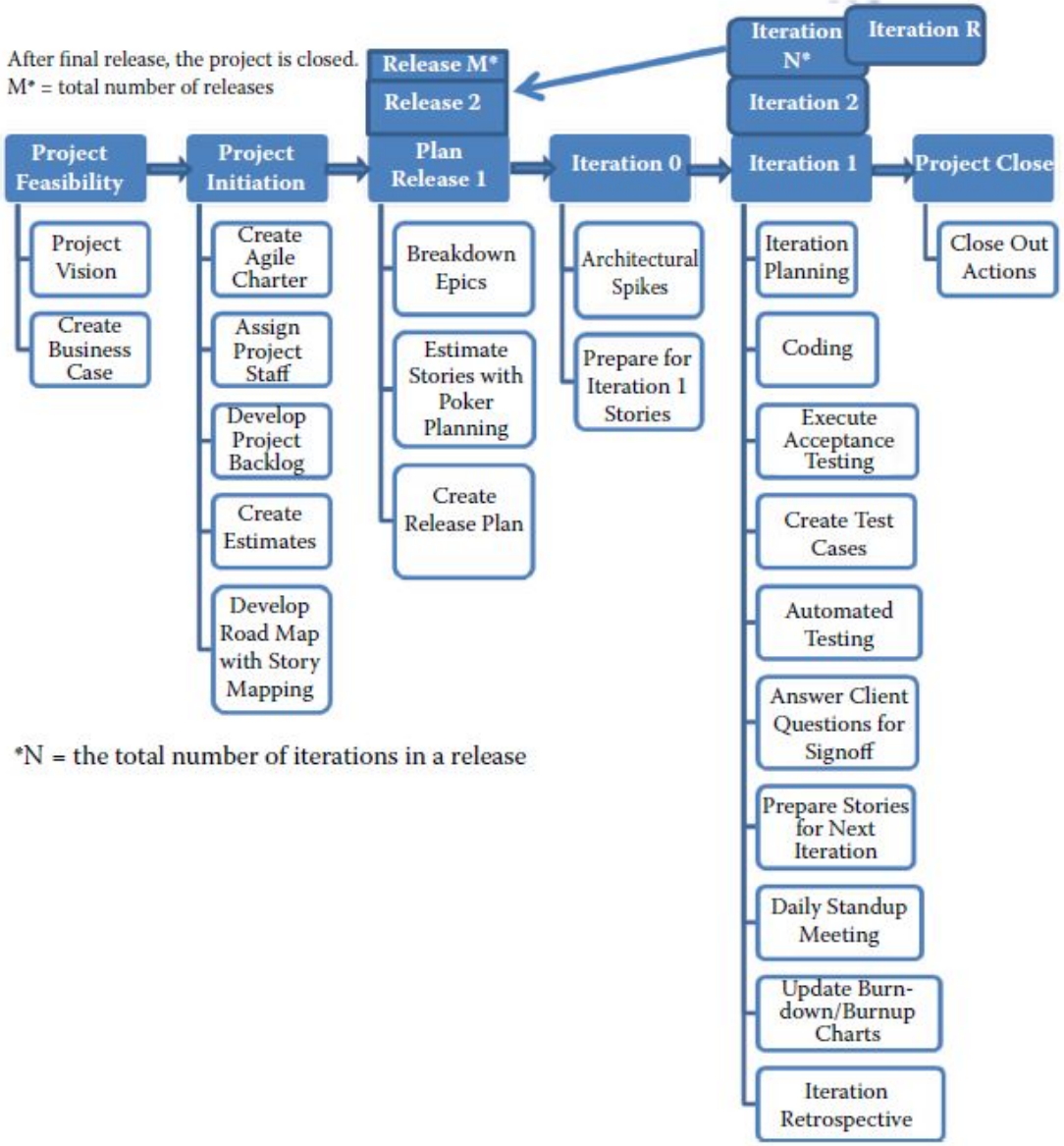
- The iteration retrospective is typically time-boxed for two hours, and it is process driven with five major phases.*

1. *Establish the stage:*
   - Focusing the participants on discussing the events of the last completed iteration.
   - Encourage the team to speak up and remain engaged throughout the meeting.
   - Give everyone a chance to contribute.
   - Create a master list of issues for the team to work from.
   - Prepare for the next step in the process.
2. *Compile the data:* Compiling the data based on the events that occurred during the iteration.
3. *Create insights:* Giving the participants an opportunity to evaluate the information gathered (brainstorming, etc.).
4. *Decide upon what needs to be done:* Giving the team an opportunity to decide on what they need to change for the next iteration based on the events from the last completed iteration.
5. *Close the retrospective meeting:* Reflections, expressions of thanks to each other, document and validate any team ideas as appropriate. Discuss what went well, what went wrong, and so on.

Close-Out Actions

- Final steps before project completion:
  - Validate product with stakeholders.
  - Gather lessons learned.
  - Archive documentation.
  - Ensure user stories meet Definition of Done (DoD).

Agile Project Management



*N = the total number of iterations in a release

Chapter Summary

- Agile Project Management follows an iterative approach.
- Focus on business value, flexibility, and collaboration.
- Effective Agile teams emphasize continuous delivery & feedback.
- Ensures customer satisfaction through incremental improvements.