Agile:
An Overview

CS-250

Mark Webster

Say Hello to the Agile Team!

- Product Owner
  - Communicates with clients to determine needs
  - Translates needs into user stories
  - Maintains Product Backlog of user stories
- Scrum Master
  - Leads Scrum events with team
  - Facilitates communication between team members
  - Helps Product Owner groom Product Backlog as team progresses and/or client requirements change
- Testers
  - Translate user stories into acceptance criteria
  - Design and deploy means of testing systems, modules, and units either manually or automated
- Developers
  - Plan and execute user stories, turning them from ideas into software solutions
  - Follow acceptance criteria to ensure quality
  - Work together to ensure timeliness and quality of software deliverables

Steps of Agile

- Initiation and Analysis
  - Product Owner meets with clients to determine project scope, high-level requirements, timelines, et cetera
  - Requirements are translated into user stories, which are sorted into a Product Backlog based on priortity

Steps of Agile

- Sprint Planning
  - The team meets daily during stand-ups to discuss what they are working on, what problems they've encountered, et cetera
  - The team estimates time necessary for upcoming tasks, delegation of responsibilities, and builds a Sprint Backlog
  - Testers generate acceptance criteria

Steps of Agile

- The Sprint
  - Typically 2 to 4 weeks
  - The tasks of the Sprint Backlog are worked through, with the goal of deliverable a potentially viable feature at the end
  - Changes should be kept to a minimum during the sprint
  - Testers continuously test new software as it is produced to ensure quality

Steps of Agile

- The Sprint Review
  - Meant to look at what was completed during the Sprint
  - The team will go over the resulting new software features, bugs, et cetera
  - A look at the concrete results of a Sprint, not necessarily the process as a whole

Steps of Agile

- The Sprint Retrospective
  - A more expansive look at the Sprint process itself as a whole team
  - What worked during the Sprint? What didn't work? What are the areas of improvement?
  - The idea is that each Sprint should get better in terms of communication, efficiency, and finished product quality as time goes on

Waterfall

- Waterfall follows a much more linear, phase-based model
- Focuses on less-complex projects where much more is known up front and planning can be done with a fair degree of absolution
- In Agile, many steps are done simultaneously, such as development and testing. In waterfall, these steps are separate, leading to the potential of problems down the line causing larger problems due to the time-cost and complexity of going back to fix them.
- In a Waterfall model, we often don't go to the next phase until the current one is complete—this can lead to inefficiencies in time usage

Waterfall or Agile?

- When deciding between a Waterfall or Agile approach, several questions should be asked:
  - How much of the project requirements will we know upfront?
  - How complex is the project?
  - How involved does the client wish to be?
- For our project, we just had a lot of high-level client desires without technical specifications, most likely subject to change, so the Agile approach worked well
- However, if the full scope of the project had been known from the beginning, with the client very hands-off, a traditional Waterfall approach may have worked very well
- Not every project is a good fit for Agile or Waterfall entirely; rather, there is a spectrum we can choose from depending on each project's needs

References

- Koelblen, Rene. “Waterfall Project Management (Waterfall 101): Software Development.” Software Hut, softwarehut.com/blog/it-project-management/waterfall-101.
- Lotz, Mary, “Waterfall vs. Agile: Which Methodology Is Right for Your Project?” Segue Technologies, 20 Nov. 2018, www.seguetech.com/waterfall-vs-agile-methodology/.
- West, Dave. “Agile Scrum Roles.” Atlassian, West, Dave. “Agile Scrum Roles.” Atlassian, www.atlassian.com/agile/scrum/roles.