INSTITUTE OF INFORMATION TECHNOLOGY

JAHANGIRNAGAR UNIVERSITY

Number of Assignment **:** 02

Submission Date **:** 01/03/2023

Course Tittle **:** Software Engineering

Course Code **:** ICT - 3201

Submitted To

Fahima Tabassum

Professor

IIT – JU

Submitted By

Md. Shakil Hossain

Roll – 2023

3rd year 2nd Semester

IIT – JU

Agile development emphasizes flexibility and responsiveness to change, which can make software scheduling and estimation challenging. However, there are several techniques and approaches that can be used to effectively schedule and estimate software development work in an agile context:

1. **User stories:** Agile development relies heavily on user stories, which are brief descriptions of a software feature from the perspective of the user. These stories can be used to estimate the size of a feature, prioritize work, and create a backlog of work to be done.

2. **Time-boxing:** Agile development typically uses time-boxing to manage work. This means that a set amount of time is allocated for a particular task or feature, and the team works within that time frame to complete the work. This can help with scheduling and estimation by providing a clear time frame for completing work.

3. **Relative sizing:** Rather than trying to estimate the exact amount of time a feature will take to complete, agile development often uses relative sizing. This means that each feature is compared to other features in terms of complexity and size, using techniques such as story points. This can help with estimation by providing a rough idea of the relative effort required for each feature.

4. **Continuous refinement:** Agile development emphasizes continuous refinement of the work to be done. This means that the backlog is constantly reviewed and refined, with features being added, removed, or reprioritized as needed. This can help with scheduling by allowing the team to adjust priorities and focus on the most important work.

5. **Retrospectives:** Agile development also emphasizes regular retrospectives, where the team reflects on their work and identifies areas for improvement. This can help with scheduling and estimation by providing feedback on how well the team is doing and where improvements can be made.

Overall, agile development requires a flexible and adaptable approach to software scheduling and estimation. By using user stories, time-boxing, relative sizing, continuous refinement, and retrospectives, agile teams can effectively manage their work and respond to changing requirements and priorities.

Scheduling and estimation for time and effort in a software project can be a complex task that requires a thorough understanding of the project's requirements, scope, and team capabilities. Here is a general framework for developing a schedule and effort estimation for a software project:

1. **Define the project scope:** Develop a web-based e-commerce platform for a small retail business. The platform should allow customers to browse and purchase products, track their orders, and manage their accounts.

2. **Break down the project into tasks:**
   - Requirement gathering and analysis
   - Designing the user interface
   - Developing the front-end and back-end of the application
   - Testing and bug fixing
   - Deployment and launch of the application

3. **Estimate task duration and effort:** Here's an example of time and effort estimation for each task:
   - Creating wireframes: 2 weeks, 40 hours
   - Designing the user interface: 3 weeks, 60 hours
   - Developing the front-end: 8 weeks, 160 hours
   - Developing the back-end: 12 weeks, 240 hours
   - Testing: 4 weeks, 80 hours
   - Deployment: 1 week, 20 hours

4. **Create a project schedule:** Once you have estimated the time and effort for each task, you can create a project schedule. Here's an example:

Task	Start Date	End Date	Duration	Dependencies
Wireframes	Week 1	Week 2	2 weeks	None
UI Design	Week 3	Week 5	3 weeks	Wireframes
Front-end Dev	Week 6	Week 13	8 weeks	UI Design
Back-end Dev	Week 6	Week 17	12 weeks	Front-end Dev
Testing	Week 18	Week 21	4 weeks	Back-end Dev
Deployment	Week 22	Week 23	1 week	Testing

5. **Factor in project risks:** Identify potential risks such as changes in project requirements, technical challenges, and delays in resource availability. Factor in some buffer time to the schedule and estimation to accommodate these risks.

6. **Re-evaluate and adjust the schedule and estimation:** Monitor progress throughout the project and adjust the schedule and estimation as necessary. This helps in keeping the project on track and ensuring that it is delivered on time.