TechLauncher 2025 Sprint Checklist

Overview

This Checklist provides examples of good practice to meet the sprint rubric outcomes. **It is neither prescriptive nor exhaustive.** Teams are encouraged to adopt and expand the items relevant to their own project.

Stakeholder Engagement

- Schedule and attend regular stakeholder meetings in The Hive
- Share detailed updates including progress and challenges with stakeholders
- Identify and clarify stakeholder expectations, and communicate any potential risks or misalignments early
- Record meeting minutes, including required action items
- Assign and manage tasks to deal with feedback and action items
- Report the status of meeting action items to stakeholders
- Develop practices to improve stakeholder relationships and address conflicts, if any

Planning and Organisation

- Establish a Project Vision in collaboration with key stakeholders
- Use a tool such as GitLab to plan and manage your project
- Record project goals and milestones in your chosen tool
- Develop and maintain a backlog. Include items for creating personas, scenarios, user stories, testing, reviews, etc.
- Prioritise backlog items based on project goals, user needs, and stakeholder input
- Identify and manage risks
- Map Product Backlog Items (PBI) to applicable goals and milestones
- Use Planning Poker to create (and revise) estimates for PBIs. Teams may use simple deadlines during early stages of their project
- Assign PBIs to appropriate team members, set clear responsibilities and feasible deadlines (based on estimates)
- Monitor team velocity and feed it into planning/estimation activities for subsequent sprints

Execution and Quality

- Use a project management tool (e.g. GitLab, Trello) for task management
- Keep milestones and PBIs up to date to reflect actual progress
- Apply standard version control practices (e.g. appropriate commits, branching, and merge requests), where applicable
- Implement CI/CD pipelines, where applicable

- Conduct unit, integration, and other testing as relevant to the project
- Conduct peer reviews of code, documentation, or designs
- Actively monitor and respond to technical or project challenges (e.g. delays, scope changes, blockers)
- Summarise the impact of work completed to date (e.g. user value, stakeholder benefit, relevance to real-world use)

Reflection and Improvement

- Seek feedback from tutors and stakeholders regularly
- Conduct sprint review at the end of each sprint to demonstrate completed work, gather feedback, and adjust priorities as needed
- Conduct sprint retrospective at the end of each sprint to reflect on team processes, identify issues, and agree on improvements for future sprints

- Create PBIs to implement all actions resulting from each retrospective and review
- Develop and record clear responses to each item of feedback
- Optional: Record individual reflections at the end of each sprint. These notes can then be collated to support your *Individual Reflection* assessment at the end of the semester

Recommended Resources

Students may refer to the resources below for practical guidance and examples related to key areas of sprint execution. These materials are intended to support understanding of best practices and help strengthen project outcomes. Use them as needed to supplement your team's approach.

Stakeholder Engagement

- Engaging Stakeholders and Managing Expectations: *Atlassian Guide on Stakeholder Communication*

- Documenting Meetings and Minutes: *Sample Meeting Minutes Templates*

Planning and Organisation

- Product Backlog Item (PBI) Management: *Guide on Backlog Structure and States*

- How to Write a User Story: *Guide on User Story and Acceptance Criteria*

- Creating and Using Personas in Agile: *Creating Agile User Personas*

- Task Estimation and Planning: *Mountain Goat video on Planning Poker*

- Story Points, Velocity, and Estimation Techniques: *Atlassian Guide on Agile Estimation*

Execution and Quality

- Guidelines on Code Quality, Peer Reviews, and Documentation: *Google Engineering Practices*

- Practices for Version Control and Issue Tracking: *GitHub Docs*

Reflection and Improvement

- Understanding the Differences Between Sprint Reviews and Retrospectives: *The 2020 Scrum Guide*

- Retrospective Techniques: *Five Fun Retrospective Ideas*

Additional Resources

- Software Engineering Best Practices: *Software Engineering at Google (O'Reilly)*

- How to Facilitate an Effective Daily Scrum: *Mountain Goat video on Daily Scrum*

- Team Communication: *Ten Smart Ways to Work in Slack*