Project Handbook

Senior Systems Development Project
ACS-4901 (6)

Department of Applied Computer Science
University of Winnipeg

The Project Handbook functions as a departmental guide specifically designed for all projects within the capstone course ACS-4901 (6). Complementary to the Standards Handbook and Course Outline, this document incorporates information on team roles, course regulations, significant events, logistics, and a comprehensive completion checklist. It is essential to utilize this handbook in conjunction with the Standards Handbook and Course Outline, as the combined resources elucidate work standards, control procedures, review processes, and reporting requirements.

Table of Contents

1. Preambles

1.1. Introduction

This is the capstone course in the Applied Computer Science program. The capstone project is an iterative systems development process where student teams work collaboratively through cycles, starting from gathering initial user requirements and progressing to the installation of a fully tested system. Students work in cross-functional teams, taking on the responsibilities of project management. They prepare estimates, create schedules, monitor progress, and conduct regular status reviews, embodying Agile principles of collaboration and shared responsibility.

The course includes Agile-inspired lectures on project management principles and techniques, covering aspects such as estimating, scheduling, resource allocation, change control, and conflict resolution. This aligns with Agile's emphasis on practical application of project management skills. The Agile-inspired approach requires three formal presentations of the project work. This provides an opportunity for iterative feedback, promoting continuous improvement, and allowing students to develop presentation skills, aligning with Agile values of transparency and communication. The course is structured to provide a challenging real-world experience in systems development. Students are immersed in the complexities of developing a system for a real end-user, aligning with Agile's focus on delivering tangible value to stakeholders. Students are exposed to all aspects of the system development process, moving beyond academic exercises. This Agile approach ensures a comprehensive understanding of the entire development lifecycle, fostering adaptability and holistic thinking.

The Agile process of the course extends beyond technical aspects. It focuses on developing skills in communication, teamwork, and time management, acknowledging the importance of these soft skills in the professional world. The real-world features of the course, including self-management, cross-functional collaboration, and exposure to the complete development process, make it appealing to prospective employers. This mirrors Agile's emphasis on delivering value and adapting to changing needs. The Agile-inspired iterative nature of the course, with regular status reviews and presentations, encourages feedback-driven improvement. This aligns with Agile principles of inspecting and adapting to enhance the overall learning experience.

In summary, the capstone course in the Applied Computer Science program follows an Agile-inspired process, emphasizing iterative development, cross-functional collaboration, project management principles, regular feedback, and a holistic approach to skill development. This approach prepares students for real-world systems development challenges and resonates with prospective employers seeking graduates with practical experience in Agile methodologies.

1.2. Course Materials

To have a complete description of this course, there are three documents you should have:

- The Course Outline gives specific information regarding the running of the course for the current academic year.

- The Project Handbook (this document) describes the course in more general terms than the course outline.

- The Standards Handbook describes the content of the deliverables produced as part of a project.

2. Project Roles and Responsibilities

We have listed below the roles that faculty and students take on in the Project Course. The roles and responsibilities listed for team members are meant as a guideline only. The team must decide for itself, with the advice of the IS Director, the organizational structure to adopt and the duties that each team member will perform on the team. For each team member, you may want to designate a "primary" and a "secondary" role. Teams may also designate further specialist roles if they feel it is beneficial to do so.

2.1 Course Coordinator

The Course Coordinator is a faculty member responsible for the overall coordination of teams, projects, and facilities.

2.2 System Administrator

The System Administrator is a department staff member, who provides technical advice and support to the teams.

2.3 IS Director

- The Information Systems Director is a faculty member assigned to your project.
- The IS Director meets with you regularly, reviews your work, attends some team meetings (especially phase-end meetings), provides consultation throughout the project, and assists with problem resolution.
- The IS Director recommends grades.

2.4 User/Project Sponsor

- Sponsors the project.
- Provides user requirements.
- Approves project proposal, reviews deliverables.
- Attends PPR, MPR and PCR,
- Signs acceptance for finished system.
- Provides access to their organization's infrastructures that are needed for the project.

3. Agile Team Roles

In an Agile team, various roles play a crucial part in ensuring the project's success. The specific roles may differ based on the Agile methodology employed, and there might be some overlap in their responsibilities. The primary aim of an Agile team is to work together in these roles to accomplish the project's overall objectives, which is to deliver products that fulfill users' requirements. Here are some common Agile team roles:

3.1 Product Owner:

The Product Owner is responsible for defining and prioritizing the product backlog, which is a list of features, user stories, and requirements for the project. They work closely with stakeholders, customers, and the development team to ensure that the product is meeting the needs of the users. The Product Owner has the final authority on what features are included in each iteration or release. In a small Agile team, this role can be taken over by the Scrum Master or the Technical Lead.

3.2 Agile Coach:

In some organizations, an Agile Coach may be present to provide guidance and support in implementing Agile practices effectively. They help teams and individuals understand and embrace Agile values, principles, and methodologies. The Agile Coach may work with multiple Agile teams across the organization, offering training, mentoring, and coaching to ensure successful adoption of Agile practices. This role can be taken over by the IS Director of an Agile team.

3.3 Scrum Master/Team Leader/Project Manager

The roles of a Scrum Master, Team Leader, and Project Manager in an Agile software project can vary based on the specific Agile framework being used (e.g., Scrum, Kanban) and the organization's structure. Here's an overview of the roles in the context of a Scrum framework:

1. **Scrum Master:**
   - **Servant Leadership**: Act as a servant leader to facilitate the Scrum process and uphold Agile principles.
   - **Scrum Framework Expertise**: Possess in-depth knowledge of the Scrum framework and its practices.
   - **Facilitate Scrum Events**: Organize and lead key Scrum events, such as Sprint Planning, Daily Standups, Sprint Review, and Sprint Retrospective.
   - **Remove Impediments**: Identify and eliminate obstacles that impede the team's progress.
   - **Protect the Team**: Shield the team from external interruptions, allowing focused delivery of value.

2. Team Leader:

- **Team Empowerment**: Encourage team members to take ownership and make collective decisions.
- **Skill Development**: Support and promote skill development within the team.
- **Foster Collaboration**: Cultivate a collaborative and open team culture for effective communication and information sharing.
- **Conflict Resolution**: Address conflicts promptly and facilitate resolutions within the team.
- **Performance Feedback**: Provide constructive feedback and acknowledge team members' achievements.

3. Project Manager (in Agile context):

- **Collaborate on Product Backlog:** Prioritize and manage the product backlog in coordination with the Product Owner.
- **Plan and Prioritize Work**: Work with the team to plan and prioritize tasks for upcoming iterations (Sprints).
- **Budget and Resource Allocation**: Manage project budgets, timelines, and efficiently allocate resources.
- **Facilitate Stakeholder Communication**: Ensure effective communication between the team and stakeholders to align with project goals.
- **Proactive Risk Management**: Identify and manage project risks, taking proactive measures to mitigate potential issues.

It is essential to note that these roles can sometimes be combined or adapted based on the organization's specific needs and the nature of the project. In Agile, the emphasis is on collaboration, adaptability, and delivering value iteratively, and the roles should align with these principles.

3.4 Development Team:

The Development Team consists of professionals who are responsible for delivering the product. This typically includes developers, testers, designers, and other specialists who contribute to the project. The team is self-organizing and cross-functional, meaning they have all the necessary skills to complete the work required for each iteration or sprint.

- Comprises cross-functional members responsible for delivering increments of a potentially shippable product.
- Includes roles such as developers, testers, analysts, designers, etc., depending on the skills needed for the project.
- Self-organizes and collaborates to plan, execute, and deliver work within iterations (Sprints).
- Works closely with the Product Owner to understand and implement user stories from the product backlog.

3.4.1 Lead Developer

In an Agile software project, a Lead Developer plays a crucial role in guiding the development team, ensuring the successful delivery of high-quality software, and fostering a collaborative and productive development environment. Here are the key roles and responsibilities of a Lead Developer in an Agile context:

1. **Coding and Development:**
   - Lead the team to write high-quality, maintainable code to implement user stories and features.
   - Collaborate with the team members to deliver working software during each iteration.
   - **Pair Programming:** Engage in pair programming sessions to share knowledge and improve code quality.

2. **Code Reviews and Quality Assurance:**
   - Conduct regular code reviews to ensure adherence to coding standards and best practices.
   - Contribute to the development of automated tests for code validation.
   - Collaborate with team members to identify and address code quality issues.

3. **Collaboration with Product Owner:**
   - Work closely with the Product Owner to clarify requirements and align technical solutions with business goals.
   - Provide input on technical aspects of the product backlog.

4. **Sprint Planning, Participation, and Estimation:**

- Actively participate in Agile ceremonies, including sprint planning, daily stand-ups, and retrospectives.
- Provide input on effort estimation and technical feasibility of user stories.
- Provide input on process improvements and team dynamics.

5. **Documentation:**
   - Lead the team in creating and maintaining documentation for code, APIs, and technical processes.
   - Document key decisions, design patterns, and system components.
6. **Continuous Integration and Deployment:**
   - Implement and maintain continuous integration and deployment pipelines.
   - Ensure that the team follows best practices for automated testing and deployment.

The Lead Developer plays a crucial role in ensuring the technical success of the project and maintaining a collaborative and efficient development environment.

3.4.2 DevSecOps Engineer :

A DevSecOps Engineer plays a critical role in integrating security practices into the DevOps pipeline to ensure the development and deployment of secure software. In an Agile software project, the DevSecOps Engineer contributes to the overall security posture of the application by addressing security concerns throughout the entire software development lifecycle. Here are key roles and responsibilities of a DevSecOps Engineer in an Agile context:

1. **Security Integration:**
   - Integrate security practices into the Agile development process.
   - Collaborate with development and operations teams to embed security checks in the continuous integration/continuous deployment (CI/CD) pipeline.
2. **Security Automation:**
   - Implement automated security testing and scanning tools.
   - Ensure that security checks are part of the automated build and deployment processes.
3. **Threat Modeling:**
   - Conduct threat modeling to identify potential security risks in the application.
   - Collaborate with the development team to address security concerns during the design phase.
4. **Vulnerability Management:**
   - Monitor and manage vulnerabilities in the software components.
   - Work with development teams to remediate identified vulnerabilities in a timely manner.

5. **Incident Response:**
   - Develop and implement an incident response plan for security events.
   - Collaborate with the incident response team to investigate and respond to security incidents.
6. **Security Training and Awareness:**
   - Provide security training and awareness programs for development and operations teams.
   - Foster a security-conscious culture within the Agile project team.
7. **Compliance and Standards:**
   - Ensure that the Agile software development process aligns with relevant security standards and compliance requirements.
   - Conduct regular security assessments to verify compliance with security policies.

By performing these roles and responsibilities, a DevSecOps Engineer ensures that security is not an afterthought but an integral part of the Agile development process, contributing to the creation of secure and resilient software products.

3.4.3 Technical Leader

A Technical Leader in an Agile software project plays a crucial role in guiding the technical aspects of the project and ensuring that the development team delivers high-quality software that meets the customer's requirements. Here are key roles and responsibilities of a Technical Leader in an Agile context:

1. **Technical Guidance:**
   - **Architecture Design:** Collaborate with the team to design a scalable and maintainable architecture that aligns with the project's goals.
   - **Technology Selection:** Guide the team in choosing appropriate technologies and tools based on project requirements and constraints.
2. **Team Collaboration:**
   - **Facilitate Communication:** Act as a bridge between the development team and other stakeholders, ensuring clear communication and understanding of technical aspects.
3. **Coding Standards and Best Practices:**
   - **Enforce Coding Standards:** Define and enforce coding standards to maintain consistency and readability of code.
   - **Promote Best Practices:** Advocate for and implement best practices related to coding, testing, and deployment.
4. **Continuous Learning:**
   - **Training and Development:** Encourage continuous learning within the team, and support team members in enhancing their skills.
   - **Stay Informed:** Keep abreast of industry trends, emerging technologies, and best practices to bring valuable insights to the team.
5. **Problem Solving and Troubleshooting:**

- **Address Technical Issues:** Help the team overcome technical challenges and roadblocks by providing guidance and solutions.
- **Root Cause Analysis:** Conduct root cause analysis for technical issues and implement preventive measures.

6. **Collaboration with Product Team:**
   - **Requirements Clarification:** Work closely with the product owner and business stakeholders to clarify technical aspects of requirements.
   - **Estimation:** Assist the team in providing accurate estimates for user stories and tasks.

By effectively fulfilling these roles, a Technical Leader contributes to the success of Agile software projects by ensuring technical excellence, team collaboration, and the delivery of valuable and high-quality software products.

3.4.4 Lead UX (User Experience) Designer

In an agile software development project, the role of a Lead UX (User Experience) Designer is crucial for creating a positive and effective user experience. Here are the important roles and responsibilities of a Lead UX Designer in an agile project:

1. **User Research:**
   - Conduct user research to understand the target audience, their needs, behaviors, and pain points.
   - Collaborate with stakeholders, product owners, and end-users to gather insights that inform the design process.

2. **User Personas and Scenarios:**
   - Develop user personas representing the characteristics and behaviors of the target users.
   - Create user scenarios to illustrate how users might interact with the software in various contexts.

3. **Wireframing:**
   - Design low-fidelity wireframes to outline the basic structure and layout of the user interface.
   - Iterate on wireframes based on feedback and align with Agile development cycles.

4. **Prototyping:**
   - Build interactive prototypes that provide a tangible representation of the user interface.
   - Use prototyping tools to simulate user interactions and flows for usability testing.

5. **Usability Testing:**
   - Conduct usability testing sessions to gather feedback on prototypes or early designs.
   - Analyze usability test results and incorporate insights into design improvements.

6. **Visual Design:**
   - Develop high-fidelity visual designs that incorporate branding elements and create a visually appealing interface.
   - Ensure consistency in design elements, such as color schemes, typography, and iconography.
7. **Collaboration with Development Team:**
   - Work closely with developers to ensure the feasibility of design implementations.
   - Collaborate in Agile ceremonies, such as Sprint Planning, Daily Standups, and Sprint Review, to align design work with development cycles.
8. **Iterative Design:**
   - Embrace an iterative design approach, making adjustments based on continuous feedback and evolving requirements.
   - Adapt designs to accommodate changes during Agile sprints and releases.
9. **Cross-Functional Collaboration:**
   - Foster collaboration with product owners, developers, QA engineers, and other team members to deliver a cohesive product.
   - Advocate for user-centered design principles within the Agile team.

The Lead UX Designer plays a critical role in shaping the user experience and ensuring that the product meets the needs and expectations of its users.

3.4.5 System Analyst

In an Agile software project, the role of a System Analyst is integral to ensuring the effective gathering and analysis of requirements, facilitating communication between stakeholders, and contributing to the overall success of the project. Here are the key roles and responsibilities of a System Analyst in an Agile context:

1. **Requirements Gathering:**
   - Collaborate with stakeholders, including product owners, users, and development teams, to elicit and document user stories and requirements.
   - Participate in Agile ceremonies, such as sprint planning and backlog refinement, to contribute to the creation and prioritization of user stories.
2. **Analysis and Documentation:**
   - Analyze user needs, business processes, and system functionalities to create comprehensive and clear requirements documentation.
   - Ensure that user stories are well-defined, with acceptance criteria that guide development and testing.
3. **Communication Facilitation:**
   - Act as a communication bridge between technical teams and non-technical stakeholders.
   - Facilitate effective communication and collaboration, ensuring that all team members have a shared understanding of requirements.

4. **Backlog Management:**
   - Contribute to the management and grooming of the product backlog by providing input on the priority and complexity of user stories.
   - Work closely with product owners and Agile teams to refine and adjust backlog items.

5. **User Story Refinement:**
   - Engage in ongoing refinement of user stories, ensuring they are detailed, feasible, and aligned with the project goals.
   - Break down larger features into smaller, manageable user stories that can be delivered within a single iteration.

6. **Acceptance Testing:**
   - Collaborate with quality assurance (QA) teams to define acceptance criteria for user stories.
   - Participate in acceptance testing to validate that delivered features meet the specified requirements.

The System Analyst's role in an Agile environment is dynamic, requiring a balance of technical acumen, effective communication, and adaptability to contribute to the successful delivery of valuable software increments.

3.4.6 QA (Quality Assurance) Engineer

In an Agile software project, a QA (Quality Assurance) Engineer plays a crucial role in ensuring the quality of the delivered software. Their responsibilities encompass various aspects of testing and quality control throughout the development lifecycle. Here are the key roles and responsibilities of a QA Engineer in an Agile context:

1. **Collaboration with the Team:**

   - Work closely with developers, product owners, and other team members to understand user stories, acceptance criteria, and overall project requirements.
   - Participate in Agile ceremonies, such as Sprint Planning, Daily Standups, Sprint Review, and Sprint Retrospective.
   - Act as a quality advocate, promoting the importance of testing and quality assurance practices within the team.

2. **Test Planning and Strategy:**

   - Contribute to the creation of the overall test strategy for the project.
   - Collaborate with the team to define test objectives, scope, and entry/exit criteria for each user story or feature.
   - Develop and maintain test plans that outline the testing approach for different types of testing (e.g., functional, non-functional).

3. **Test Case Design:**

- Create detailed and comprehensive test cases based on user stories and acceptance criteria.
- Ensure test cases cover positive and negative scenarios, edge cases, and relevant user interactions.
- Use test design techniques to maximize test coverage.

4. **Test Automation:**

- Identify opportunities for test automation to improve efficiency and effectiveness.
- Develop and maintain automated test scripts for regression testing and repetitive tasks.
- Integrate automated tests into the continuous integration/continuous deployment (CI/CD) pipeline.

5. **Manual Testing:**

- Execute manual test cases to validate functionality and identify defects.
- Perform exploratory testing to uncover unexpected issues and ensure a positive user experience.
- Conduct usability testing to assess the user interface and overall user satisfaction.

6. **Performance Testing:**

- Plan and execute performance testing to assess the system's responsiveness, stability, and scalability.
- Identify and address performance bottlenecks and potential issues related to load and stress.

7. **Documentation:**

- Document test cases, test results, and any relevant information related to testing activities.
- Maintain test documentation to ensure it remains up-to-date with evolving project requirements.

By fulfilling these roles and responsibilities, a QA Engineer contributes to the development of high-quality software in an Agile environment, helping to meet customer expectations and deliver value iteratively throughout the project.

4. Agile Project Plan, Schedules, and Milestones

An Agile Software Development Project lifecycle can be divided into three distinct milestones for reviews and assessments as follows: (i) Project Planning Review (PPR), (ii) Mid Project Review (MPR), and (iii) Project Completion Review (PCR). These milestones provide structured opportunities for project reviews and assessments at different stages of an Agile Software Development Project.

4.1 Project Planning Review (PPR)

Objective: Assess and validate the project plan, objectives, and initial preparations before the start of development iterations.
Activities:
- Initial Planning:
  - Define project goals, scope, and initial requirements.
  - Create the initial product backlog.
  - Set up the project team and assign roles.
- Planning Review:
  - Hold a PPR meeting to review and validate the project plan.
  - Evaluate the feasibility of the project goals and scope.
  - Ensure alignment between stakeholder expectations and the project plan.
- Review/Assessment:
  - Assess the completeness and clarity of the initial product backlog.
  - Validate that roles and responsibilities are well-defined.
  - Ensure that the project plan aligns with Agile principles and is feasible.

4.2 Mid Project Review (MPR)

Objective: Evaluate the progress made during the midpoint of the project, ensuring alignment with goals, and adapting plans for the remaining iterations.
Activities:
- Iterative Development:
  - Execute development tasks across multiple iterations.
  - Conduct regular sprint reviews and gather stakeholder feedback.
- Continuous Planning:
  - Adapt the product backlog based on evolving requirements.
  - Monitor team velocity and adjust plans as needed.
- MPR Meeting:
  - Hold an MPR meeting to assess progress against initial goals.
  - Review and adapt the release plan for the remaining iterations.
- Review/Assessment:
  - Evaluate the quality and completeness of work completed in the first half of the project.
  - Assess the team's ability to adapt to changes and meet project objectives.

15

      o   Validate that the project is on track to deliver the expected value.

4.3 Project Completion Review (PCR)

Objective: Review the project in its entirety, ensuring all goals are met, and identify areas for improvement in future projects.

Activities:

- Final Iterations:
    - Complete the remaining iterations and deliver the final product.
    - Conduct a final sprint review and obtain stakeholder acceptance.
- PCR Meeting:
    - Hold a PCR meeting to assess the overall success of the project.
    - Review achievements, challenges, and lessons learned.
    - Discuss opportunities for improvement in future projects.
- Review/Assessment:
    - Evaluate the project's overall success against initial goals and expectations.
    - Identify areas for improvement in processes, team collaboration, or tools.
    - Gather insights and lessons learned for continuous improvement.

5. Detailed Description of Agile Project Milestones

5.1 Project Planning Review (PPR)

Objective: The Project Planning Review (PPR) is a crucial phase in an Agile project aimed at assessing and validating the project plan, objectives, and initial preparations before the start of development iterations. The primary goal is to ensure that the project is well-defined, feasible, and aligned with Agile principles.

Activities:

1. **Initial Planning:**

   - **Define project goals, scope, and initial requirements:**
     - Establish a clear understanding of the project's overarching objectives.
     - Define the scope by outlining the boundaries and limitations of the project.
     - Capture initial high-level requirements to guide the development process.

   - **Create the initial product backlog:**
     - Develop a preliminary list of features and functionalities required for the project.
     - Prioritize backlog items based on initial business value and dependencies.

   - **Set up the project team and assign roles:**
     - Assemble a cross-functional team with members possessing diverse skills.
     - Define and assign specific roles and responsibilities to team members.
     - Foster a collaborative team culture to promote effective communication.

2. **Planning Review:**

   - **Hold a PPR meeting to review and validate the project plan:**
     - Organize a comprehensive meeting involving key stakeholders and the project team.
     - Review the project plan in detail, including timelines, milestones, and resource allocation.

- Seek input and feedback from team members and stakeholders to identify potential gaps.

- **Evaluate the feasibility of the project goals and scope:**

    - Conduct a feasibility analysis to ensure that project goals are realistic and achievable.

    - Assess potential risks and challenges that may impact project success.

    - Consider resource constraints, technological feasibility, and time constraints.

- **Ensure alignment between stakeholder expectations and the project plan:**

    - Confirm that stakeholder expectations are accurately reflected in the project plan.

    - Address any discrepancies or misunderstandings to maintain alignment.

    - Validate that the project plan aligns with organizational strategies and objectives.

Review/Assessment:

3. **Assess the completeness and clarity of the initial product backlog:**

    - Review the product backlog to ensure it adequately represents the project requirements.

    - Confirm that backlog items are well-defined, prioritized, and accessible to the development team.

    - Address any ambiguities or gaps in the backlog that may hinder the development process.

4. **Validate that roles and responsibilities are well-defined:**

    - Ensure that each team member understands their role and responsibilities.

    - Confirm that roles are defined in a way that promotes collaboration and efficiency.

    - Address any confusion or overlapping responsibilities to prevent future conflicts.

5. **Ensure that the project plan aligns with Agile principles and is feasible:**

    - Evaluate the project plan against Agile principles, emphasizing adaptability and collaboration.

- Confirm that the plan allows for iterative development and responsiveness to change.

- Verify that the project plan is feasible within the given constraints, considering time, budget, and resources.

The Project Planning Review sets the stage for a successful Agile project by establishing a solid foundation through clear planning, effective team organization, and alignment with stakeholder expectations. This phase ensures that the project is well-positioned for the upcoming development iterations.

Deliverables for Project Planning Review (PPR):

1. **Project Goals, Scope, and Initial Requirements Document:**

    - A document outlining the overarching goals and objectives of the project.

    - Clear definition of the project scope, including boundaries and limitations.

    - Initial high-level requirements capturing key functionalities.

2. **Initial Product Backlog:**

    - A comprehensive list of features and functionalities required for the project.

    - Prioritized backlog items based on initial business value and dependencies.

    - Documentation of backlog items accessible to the development team.

3. **Project Team Setup and Roles Assignment:**

    - Documentation of the assembled cross-functional team.

    - Clear definition and assignment of roles and responsibilities for team members.

    - Documentation promoting a collaborative team culture.

4. **PPR Meeting Documentation:**

    - Minutes or summary report from the Project Planning Review meeting.

    - Detailed review of the project plan, including timelines, milestones, and resource allocation.

    - Input and feedback received from team members and stakeholders.

5. **Feasibility Analysis Report:**

    - Document outlining the results of the feasibility analysis.

    - Assessment of potential risks and challenges impacting project success.

- Consideration of resource constraints, technological feasibility, and time constraints.

6. **Stakeholder Alignment Confirmation:**

- Documentation confirming that stakeholder expectations are accurately reflected.

- Addressing any discrepancies or misunderstandings to maintain alignment.

- Validation of the project plan alignment with organizational strategies and objectives.

7. **Completed Initial Product Backlog Assessment:**

- Evaluation of the completeness and clarity of the initial product backlog.

- Confirmation that backlog items are well-defined, prioritized, and accessible.

- Addressing any ambiguities or gaps hindering the development process.

8. **Roles and Responsibilities Validation:**

- Confirmation that each team member understands their role and responsibilities.

- Documentation ensuring roles are defined to promote collaboration and efficiency.

- Addressing any confusion or overlapping responsibilities to prevent conflicts.

9. **Agile Alignment and Feasibility Assessment:**

- Assessment of the project plan against Agile principles.

- Verification that the plan allows for iterative development and responsiveness to change.

- Confirmation that the project plan is feasible within given constraints.

The deliverables from the Project Planning Review (PPR) provide a comprehensive foundation for the subsequent phases of the Agile project, ensuring clarity, alignment, and feasibility for successful development iterations. For a detailed template for PPR report, please see the **Standards Handbook**.

5.2 Mid Project Review (MPR)

Objective: The Mid Project Review (MPR) in Agile project management serves as a crucial checkpoint to evaluate the progress made during the midpoint of the project. The primary objectives include assessing alignment with goals, adapting plans for the remaining iterations, and ensuring that the project is on track to deliver expected value. This stage involves reviewing updated deliverables from the planning stage and introducing additional deliverables to provide insight into the detailed design of the project's components.

Activities:

1. **Iterative Development:**

   - **Execute development tasks across multiple iterations:**

     - Conduct iterative development based on the Agile principles, breaking the project into manageable sprints.

     - Implement features and functionalities incrementally, allowing for continuous feedback and improvement.

   - **Conduct regular sprint reviews and gather stakeholder feedback:**

     - Host sprint reviews at the end of each iteration to showcase completed work.

     - Gather feedback from stakeholders to validate that the developed features meet their expectations.

     - Use stakeholder input to inform adjustments for subsequent iterations.

2. **Continuous Planning:**

   - **Adapt the product backlog based on evolving requirements:**

     - Continuously refine and adjust the product backlog in response to changing priorities and requirements.

     - Prioritize backlog items based on emerging business needs, ensuring maximum value delivery.

   - **Monitor team velocity and adjust plans as needed:**

     - Track the team's velocity by analyzing the amount of work completed in each iteration.

     - Adjust future plans and expectations based on observed velocity, addressing any impediments.

- Use velocity as a key metric to enhance predictability and planning accuracy.

3. **MPR Meeting:**

   - **Hold an MPR meeting to assess progress against initial goals:**

     - Organize a comprehensive meeting involving key stakeholders and the project team.

     - Review progress against the goals set during the initial planning phase.

     - Identify achievements, challenges, and opportunities for improvement.

   - **Review and adapt the release plan for the remaining iterations:**

     - Revisit the release plan, considering any changes in priorities or scope.

     - Adjust the release plan to reflect the refined product backlog and updated project goals.

     - Communicate any modifications to stakeholders to manage expectations.

Review/Assessment:

4. **Evaluate the quality and completeness of work completed in the first half of the project:**

   - Assess the delivered work in terms of quality, completeness, and alignment with acceptance criteria.

   - Identify any areas requiring additional attention or improvement.

   - Validate that completed work meets the defined Definition of Done (DoD).

5. **Assess the team's ability to adapt to changes and meet project objectives:**

   - Evaluate the team's responsiveness to changing requirements and its ability to adapt accordingly.

   - Consider the team's collaboration, communication, and problem-solving skills.

   - Assess the team's overall effectiveness in meeting project objectives.

6. **Validate that the project is on track to deliver the expected value:**

   - Confirm that the project's trajectory aligns with the intended value and objectives.

- Validate that the delivered features align with stakeholder expectations and organizational goals.

- Identify and address any factors that may impact the project's ability to deliver expected value.

The Mid Project Review serves as a critical checkpoint, ensuring that the Agile project remains adaptable, aligned with goals, and on track to deliver optimal value. This phase emphasizes continuous improvement and strategic adjustments to enhance project success.

Deliverables for Mid Project Review (MPR):

1. **Iterative Development Documentation:**

    - Records of iterative development activities conducted across multiple sprints.

    - Implementation details of features and functionalities completed incrementally.

    - Documentation showcasing adherence to Agile principles in development.

2. **Regular Sprint Review Records:**

    - Documentation from regular sprint reviews showcasing completed work.

    - Stakeholder feedback collected during sprint reviews for each iteration.

    - Adjustments or improvements made based on stakeholder input.

3. **Continuous Planning Documents:**

    - Updated versions of the product backlog reflecting evolving requirements.

    - Documentation of adjustments made to prioritize backlog items.

    - Monitoring records of team velocity and corresponding adjustments.

4. **MPR Meeting Minutes:**

    - Comprehensive meeting minutes capturing discussions during the Mid Project Review.

    - Summary of progress against initial goals discussed during the meeting.

    - Identification of achievements, challenges, and opportunities for improvement.

5. **Adapted Release Plan Documentation:**

    - Revised release plan reflecting changes in priorities or scope.

- Communication records detailing any modifications to stakeholders.

- Alignment of the release plan with the refined product backlog and updated project goals.

6. **Quality and Completeness Assessment Report:**

- Evaluation report assessing the quality and completeness of work in the first half of the project.

- Identification of areas requiring additional attention or improvement.

- Validation that completed work meets the defined Definition of Done (DoD).

7. **Team Adaptability and Effectiveness Evaluation:**

- Assessment report on the team's adaptability to changes in requirements.

- Evaluation of collaboration, communication, and problem-solving skills.

- Overall assessment of the team's effectiveness in meeting project objectives.

8. **Project Trajectory and Value Validation Documentation:**

- Confirmation that the project trajectory aligns with intended value and objectives.

- Validation that delivered features align with stakeholder expectations and organizational goals.

- Identification and documentation of factors that may impact the project's ability to deliver expected value.

Additional Deliverables for MPR

Please note that the details of the following deliverables depend on the type of project and level of design details required to execute the project.

Deliverables:

System Architecture, Database Design, User Interface Design, Detailed System Design, API Design, Security Design, Deployment Architecture, Testing Strategy, Dependencies, Release Planning, and Documentation. For a detailed template for MPR report, please see the **Standards Handbook**.

5.3 Project Completion Review (PCR)

Objective: The Project Completion Review (PCR) represents the final phase in Agile project management, aiming to comprehensively review the project in its entirety. The primary objectives include ensuring that all project goals are met, obtaining stakeholder acceptance of the final product, and identifying areas for improvement in future projects.

Activities:

1. **Final Iterations:**

   - **Complete the remaining iterations and deliver the final product:**

     - Execute the final iterations, focusing on the completion of any outstanding work.

     - Ensure that all user stories and backlog items are addressed according to the acceptance criteria.

   - **Conduct a final sprint review and obtain stakeholder acceptance:**

     - Host a final sprint review to showcase the completed product.

     - Obtain formal acceptance from stakeholders, ensuring alignment with their expectations.

     - Address any outstanding concerns or feedback before finalizing the project.

2. **PCR Meeting:**

   - **Hold a PCR meeting to assess the overall success of the project:**

     - Organize a comprehensive meeting involving key stakeholders, project team members, and relevant parties.

     - Review the entire project lifecycle, including initial planning, execution, and completion.

   - **Review achievements, challenges, and lessons learned:**

     - Reflect on project achievements, acknowledging successful outcomes and goals met.

     - Discuss challenges encountered during the project and how they were addressed.

     - Extract valuable lessons learned from both positive and challenging experiences.

- **Discuss opportunities for improvement in future projects:**
  - Identify areas where processes, collaboration, or tools could be enhanced.
  - Engage in open discussions about potential improvements for future Agile projects.
  - Encourage feedback from team members and stakeholders to gather diverse perspectives.

Review/Assessment:

3. **Evaluate the project's overall success against initial goals and expectations:**
   - Assess the project's overall success, considering whether it met its intended goals and objectives.
   - Review adherence to the defined acceptance criteria and whether stakeholders' expectations were fulfilled.
   - Examine the project's impact on the organization and its alignment with strategic objectives.

4. **Identify areas for improvement in processes, team collaboration, or tools:**
   - Identify specific areas where processes can be refined or streamlined for increased efficiency.
   - Evaluate the effectiveness of team collaboration and communication throughout the project.
   - Assess the suitability and effectiveness of tools and technologies used during the project.

5. **Gather insights and lessons learned for continuous improvement:**
   - Collect insights and lessons learned from project team members, stakeholders, and relevant contributors.
   - Document valuable experiences and knowledge gained during the project.
   - Utilize gathered insights to inform continuous improvement initiatives for future projects.

The Project Completion Review serves as a crucial reflection point, providing a holistic assessment of the project's success, lessons learned, and opportunities for ongoing improvement. This phase emphasizes the Agile principle of continuous improvement, contributing to the refinement of processes and practices in subsequent projects.

Deliverables for Project Completion Review (PCR):

1. **Final Iterations Documentation:**

   - Records of the completed final iterations, detailing the execution of remaining work.

   - Documentation ensuring that all user stories and backlog items are addressed per acceptance criteria.

   - Formal documentation of the final sprint review and stakeholder acceptance.

2. **PCR Meeting Minutes:**

   - Comprehensive meeting minutes capturing discussions during the Project Completion Review.

   - Review of the entire project lifecycle, including initial planning, execution, and completion.

   - Addressing achievements, challenges, and lessons learned during the meeting.

3. **Achievements and Challenges Report:**

   - Detailed report acknowledging project achievements and goals met.

   - Discussion of challenges encountered throughout the project and strategies employed for resolution.

   - Analysis of the impact of challenges on the project's overall success.

4. **Opportunities for Improvement Documentation:**

   - Identification of specific areas where processes can be refined or streamlined.

   - Evaluation of team collaboration and communication effectiveness, highlighting areas for improvement.

   - Assessment of the suitability and effectiveness of tools and technologies used.

5. **Project Success Evaluation Report:**

   - Evaluation report assessing the project's overall success against initial goals and expectations.

   - Review of adherence to acceptance criteria and fulfillment of stakeholders' expectations.

   - Examination of the project's impact on the organization and alignment with strategic objectives.

6. **Areas for Improvement Action Plan:**

- Action plan outlining specific improvements in processes, collaboration, or tools.

- Strategies for implementing identified improvements in future Agile projects.

- Timeline and responsibilities for implementing the proposed improvements.

7. **Insights and Lessons Learned Compilation:**

- Compilation of insights and lessons learned from project team members, stakeholders, and contributors.

- Documented experiences and knowledge gained during the project.

- Utilization of gathered insights to inform continuous improvement initiatives for future projects.

8. **Continuous Improvement Recommendations:**

- Recommendations for continuous improvement initiatives based on gathered insights.

- Proposed strategies for incorporating lessons learned into the organization's Agile practices.

- Suggestions for fostering a culture of continuous improvement within the project team.

The deliverables from the Project Completion Review (PCR) represent a comprehensive assessment of the project's overall success, lessons learned, and actionable insights for continuous improvement. This phase emphasizes the importance of reflection and learning to enhance future Agile project endeavors. For a detailed template for PCR report, please see the **Standards Handbook**.

6. Logistics

The following guidelines are set up so that your team can operate smoothly during the project year.

1. **Photocopying**
   - Each team is permitted to use University facilities to photocopy up to 1,000 pages during the project. These accounts are monitored for excessive use.
   - An account will be set up for each team on the departmental photocopier in the Duckworth building. This account is for project-related documents only. Any use of it to do non-project work (e.g. copying class notes or material related to other courses) will result in the account being disabled.
   - Photocopying must be used for short documents only. Longer documents such as reports and manuals must be submitted to Printing Services. See Printing below.
   - Once you have been assigned a project, you must arrange to use your end-user's photocopying services, if at all possible.
   - Please note that faculty members have priority to use the photocopier in the staff room, i.e. you must suspend your photocopying if a faculty member queues up behind you.
2. **Printing**
   - The University's Printing Services department (in the basement of Bryce Hall) may be used for reasonable printing requests in the course.
   - Printing Services (as opposed to the photocopier) must be used for long documents, and for shorter documents when many copies are required.
   - Any document to be printed must first be reviewed and approved by your IS Director, who will sign a "Duplicating Requisition" form for you.
   - Deliver the form and your originals to Printing Services. If your printing job has special requirements, you must discuss them with Printing Services.
   - Printing Services has a five-working-day lead time.
3. **Lab Access**
   ID01 (Project Lab):
   - Teams will receive the combination at the appropriate time.
   - Use of the Applied Computer Science labs is governed by the Applied Computer Science Project Labs Code of Conduct.
   3C13 (Computing Lab):
   - Teams may also request access to 3C13. Check the department web site www.acs.uwinnipeg.ca for Lab Procedures.

4. **Team Accounts**
   - Each team will be given an account on the ACS network.
   - Requests for special accounts should be approved by the IS Director and then forwarded to the System Administrator.

5. **Room Bookings**
   - The Department will reserve a room for each team for team meetings throughout the year.
   - To obtain a room booking, the Project Leader should email to the Project Coordinator stating the team's section number, team number, and desired meeting time. Include the day, start time, and duration.
   - Room bookings for PPR, MPR and PCR Seminars should be directed to Project Coordinator.
6. **Equipment Bookings and Room Set-Up**
   - The Department will provide equipment required for formal project reviews.
   - To obtain equipment, email to the project coordinator stating clearly the details of your request. Include date, start time, duration, and location as soon as possible.

If your team requires further services or facilities, see the Project Coordinator.

7. Completion Checklist

All of the following items must be completed before your final grade in ACS-4901/6 will be submitted to Student Records. If you fail to complete all of these items **by the date specified in the Course Outline**, your grade will be negatively affected.

1. Complete the hard copy of your team's **Project Repository** with your IS Director. Ensure that your repository contains *final actuals*.

2. Submit to the System Administrator the following items:

   · The electronic project repository. See Standards Handbook for details.

   · Your complete system. See the System Administrator for details.

3. Remove all software that you installed on any computer in the project lab or on the network.

4. Delete all sundry project files (EXCEPT THE WORKING COPY OF THE DEVELOPED APPLICATION) from all computers in the Project Lab and from the network.

5. Return all items (software, books, etc.) you have received from your user or the Applied Computer Science Department.

6. Remove all personal/team items from the Project Lab.

7. Remove the contents of your project locker, if there is any, and remove the lock.

NOTE:
· Items 3 to 7 require sign-off from the System Administrator.
· Your IS Director may also impose additional requirements that are specific to your team. Please check with him/her well in advance.