Starrs Team SQA Plan

Standards

- Before every sprint, a set of standards and requirements is established to be adhered to by all team members throughout the process of design and development to ensure conformance to a professional look and feel of significant project components.
    - Analysis and Design Documents: Diagrams are to be made with a white or gray background to signify unity within the project.
    - Build: UI colors, fonts, and themes are to remain consistent across every screen within each stakeholder category. Naming conventions will follow camel case for code variables and snake case for screen names to simplify identification. Source control methods will follow that at least one other team member will review work before it is merged into the main branch, and branch names will coordinate with screen names.

Review Process/Discipline

- The Starss team values consistent, effective, and clear communication in all aspects of design and developmental work that are of significant project value.
- For every phase (analysis, design, build), the following procedure is to be applied for project review and team communication:
- Communication is facilitated through informal text channels, as well as daily scrum meetings ranging in length from 15 minutes to several hours, in which project progress, individual statuses, questions, and issues are discussed.
    - For preparation purposes, in order to ensure every team member is on the same page, the scrum master releases a preliminary list of information, tasks, and the meeting agenda to be reviewed prior to the meeting.
    - Scrum meeting notes are recorded for every meeting in which any significant information is exchanged/discussed that must be recorded, or there are ongoing issues that must be tracked. This is to ensure that spoken communication is recorded so there are no miscommunications moving forward.
- Informal peer reviews are available on a daily basis during scrum meetings, in which at least 2 other team members are available to look over and provide feedback for any project work that is in progress.
    - Peer reviews are to be conducted when any team member has made sizeable progress that needs to be reviewed prior to integration and continuation, or if an issue has occurred that requires aid from other team members.
- Group reviews are available on a weekly basis during scrum meetings, in which at least one day out of a week, all team members are present to look over and provide feedback for any project work that is in progress, or is complete and to be submitted.
    - Mandatory group reviews are to be conducted prior to the start of every sprint to review, and adjust if necessary, analysis and design documents/diagrams pertaining to the sprint, as well as before completion of any sprint to review team efforts and products, before submission of any documentation or deliverables, and when any significant project management changes are made and need to be discussed with the full team.

- Build: Formal technical reviews are conducted as a whole group meeting weekly. Project demos and code walkthroughs are done as a technical assessment and to facilitate discussion on the extent of quality dimensions being present within the project.
  - During the formal technical review meetings, inspections are conducted on the drafts and finished work of peers by all members before submission/integration to ensure there are no inconsistencies with other work and/or errors. Work is scrutinized to ensure that the software is uniform and consistent. In addition, contributions are reviewed to ensure they follow the set of standards and requirements outlined. Project organization is also discussed to ensure it is organized and readable.
  - Software safety is enacted during formal reviews by a group discussion on potential hazards to the software.

Testing

- Analysis: Determine a set of standards to be adhered to within the test case environment and tracker. Ensure that user stories are drafted in a manner that allows for ease of composition of corresponding test cases.
- Design: Ensure that diagrams and development drafts are produced in a way that allows for ease of testing, as well as have alternate routes depending on test outcomes.
- Build: All facets of development of the software product are to be done in a way that adheres to the aforementioned standards, as well as in a way that allows for ease of readability, traceability, error/defect tracking, and scalability.
  - At the end of each sprint, unit testing will be conducted during a group review session led by the development coordinator to ensure that the software acts as intended.
  - At the end of each sprint following sprint 1, integration testing will be conducted during a group review session led by the development coordinator to ensure that the new software developed interacts with the previous code correctly and without issue, as well as does not disrupt the functionality of either product release.
  - Following the final sprint and release, system testing will be conducted during a group review session led by the development coordinator to ensure each function and feature works as intended and errors/defects have been identified and ideally, resolved.
  - Development should begin and end with the understanding that each feature and functionality will be tested.
  - Testing follows the use cases outlined in the test plan tracker, as well as checks that implicit requirements have been met (ease of use, feel of software, response time, etc.).
  - All tests can be traced back to a stakeholder requirement and the user stories.
  - Testing will also include cases in which users act outside of anticipated courses of action to ensure the safety, reliability, security, dependability, and availability of the software.

Error/Defect Tracking

- For every phase (analysis, design, build), the following procedure is to be applied for project error/defect tracking:
  - Any errors/miscommunications encountered during each phase are submitted for peer review and analysis through informal means of communication, as well as are also

discussed in the daily scrum meetings, which are then either resolved or a plan of action is outlined to resolve/address the issue.
    - Any ongoing problems are tracked through the scrum meeting notes, in which current complications and their ongoing statuses are recorded.
    - Self-imposed team deadlines are enforced to ensure that issues are resolved to an acceptable degree within a reasonable time frame.

Change Management

- For every phase (analysis, design, build), the following procedure is to be applied for project change management:
    - Any significant project changes (documents, requirements, diagrams, code, tools) made are to be discussed with the entirety of the group during weekly full group meetings. Changes must be met with unanimous acceptance in order to move forward.
    - Any changes to the project management system are discussed prior and, if approved, are recorded within Jira to ensure that any changes in roles, tasks, requirements, and management are set for official reference.
- Build: GitHub is utilized in order to enforce version control and change management of the code.

Deployment Support Plan Post-Production

CI/CD Pipelines and Deployment:

- CI/CD Pipelines (2 weeks before deployment):
    - Implement automated testing for vending machine software.
    - Establish continuous integration pipelines for efficient code deployment.
    - Conduct regular code reviews to ensure quality before deployment.
- Deployment Plan:
    - Initial release: Deploy 10 vending machines for testing and feedback (first month after deployment).
    - Subsequent release: Scale up to 500 vending machines based on successful pilot.

Training and Onboarding (1 week before deployment):

- Restocker Training:
    - Conduct comprehensive training sessions on stocking procedures.
    - Provide a detailed restocking manual with visual aids.
    - Implement online training modules for continuous learning.
- Management Training:
    - Train managers on system administration and monitoring.
    - Provide documentation on troubleshooting common issues.
    - Conduct simulated scenarios to enhance decision-making skills.

Employee Training and Support:

- Online Manual Training:
    - Develop an online manual training program for employees.
    - Provide access to a digital manual library for reference.
    - Conduct periodic refreshers to keep employees updated on manual changes (annual).
- Employee Allocation and Schedule (during the first month after deployment):
    - Allocate 15 employees for deployment and initial operations.
    - Implement a rotating schedule to cover all shifts efficiently.

        - Ensure cross-training for flexibility in employee roles.

Customer Interaction and Support:

- Customer Schedule (business hours-ongoing):
    - Create a schedule for customer access to vending machines.
    - Consider peak hours and adjust staffing accordingly.
    - Implement feedback mechanisms for continuous improvement.
- Software Support Rotation:
    - Establish a 24/7 support rotation for the first month.
    - Gradually transition to a standard support schedule (business hours) based on usage patterns (takes about 3 months).
    - Develop a tiered support system for efficient issue resolution.

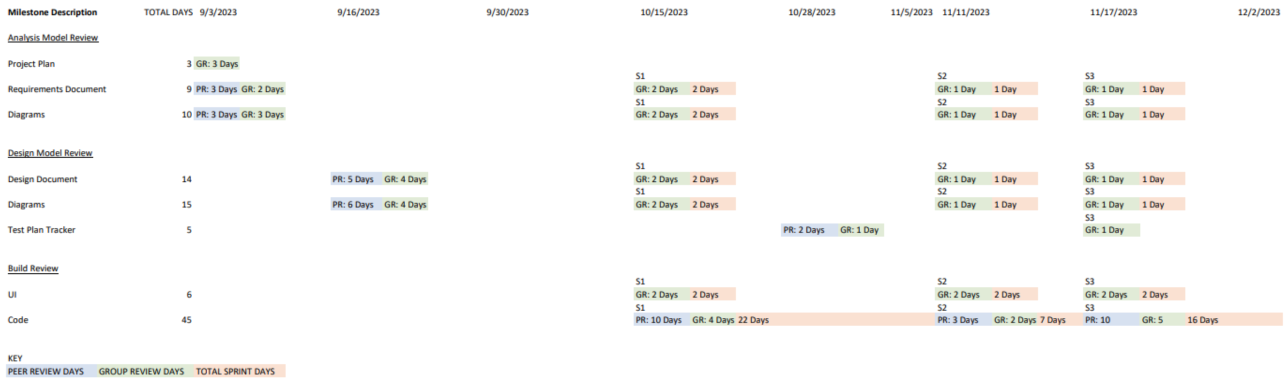
Post-Deployment Monitoring and Evaluation (after the first month):

- Performance Monitoring:
    - Implement tools for real-time monitoring of vending machine performance.
    - Set up alerts for any anomalies in sales, inventory, or machine status.
- Customer Feedback:
    - Collect feedback from customers through various channels.
    - Conduct regular post-deployment reviews to identify areas for improvement.

Continuous Improvement (annual update):

- Iterative Deployment:
    - Plan for regular updates and improvements to the vending machine software.
    - Utilize feedback loops from customers and employees for enhancement.
- Employee Training Updates:
    - Keep employee training programs up-to-date with software changes.
    - Conduct periodic training sessions for ongoing skill development.

<u>Gantt Chart</u>



Milestone Description	TOTAL DAYS	9/3/2023	9/16/2023	9/30/2023	10/15/2023	10/28/2023	11/5/2023	11/11/2023	11/17/2023	12/2/2023
<u>Analysis Model Review</u>										
Project Plan	3	GR: 3 Days								
Requirements Document	9	PR: 3 Days GR: 2 Days			S1 GR: 2 Days 2 Days			S2 GR: 1 Day 1 Day	S3 GR: 1 Day 1 Day	
Diagrams	10	PR: 3 Days GR: 3 Days			S1 GR: 2 Days 2 Days			S2 GR: 1 Day 1 Day	S3 GR: 1 Day 1 Day	
<u>Design Model Review</u>										
Design Document	14		PR: 5 Days GR: 4 Days		S1 GR: 2 Days 2 Days			S2 GR: 1 Day 1 Day	S3 GR: 1 Day 1 Day	
Diagrams	15		PR: 6 Days GR: 4 Days		S1 GR: 2 Days 2 Days			S2 GR: 1 Day 1 Day	S3 GR: 1 Day 1 Day	
Test Plan Tracker	5					PR: 2 Days GR: 1 Day			S3 GR: 1 Day	
<u>Build Review</u>										
UI	6				S1 GR: 2 Days 2 Days			S2 GR: 2 Days 2 Days	S3 GR: 2 Days 2 Days	
Code	45				S1 PR: 10 Days GR: 4 Days 22 Days			S2 PR: 3 Days GR: 2 Days 7 Days	S3 PR: 10 GR: 5 16 Days	

KEY

PEER REVIEW DAYS	GROUP REVIEW DAYS	TOTAL SPRINT DAYS