Project Milestone 1

21K-3153

21K-3372

21K-4529

**Project Description:**

Our project is a "Customizable Medical Quotations File Reader Software". It aims to read medical documents of various file formats, extract information like medicine name, dispatchers, price, current discounts etc and compile them into an Excel file.

**Process Model Analysis:**

**Waterfall Model:**

Requirements: This phase would entail holding a meeting with our teacher (primary stakeholder), asking them for their requirements and finalizing those requirements. These requirements would be considered fixed and would not undergo any changes afterwards.

Design: After requirements finalization, we would analyze the requirements one by one and create a detailed structure for the implementation of the requirements. This structure would become our roadmap for subsequent phases.

Implementation: In the implementation phase, we would start translating our structure and the requirements defined into working code, build testing modules to check the code in the testing phase and write documentation to footnote our development. For eg, implementing the HTML scraper, configuring Excel compilation, implementing scrapes of other formats and so on.

Testing: In the testing phase, we would meticulously recheck our code for errors, run the unit tests we wrote in the implementation phase and conduct unit, integration and system testing. Comprehensive documentation is maintained for clarity.

Operation and Maintenance: In the operation and maintenance phase, we would monitor performance, provide user support, manage data, address bug fixes, and keep documentation up-to-date.

**Plan driven incremental model:**

Incremental Planning: Identify and prioritize features for each increment based on stakeholder and user feedback. Ask our teacher (primary stakeholder) for their requirements.

Incremental Development: Analyze requirements and choose the most important increments first (ie implementing the HTML document scraper) and start building them.

Incremental Testing: Conduct testing for our increment to ensure they work correctly. Test and fix bugs. Address and resolve any issues identified during testing.

Incremental Deployment: Deploy our increments, show them to our primary stakeholder and observe interaction of users with software.

Feedback Integration: Use feedback from users and stakeholders to identify positives, negatives, difficulties faced and bugs. Use feedback to make subsequent increments better. Repeat with more increments (Implementing Excel compilation, scraping from different file formats etc)

Sample Increments:

Increment 1: Implement HTML doc scraping.

Increment 2: Configure data storage and categorization to an Excel Doc

Increment 3: Implement scraping of other file formats

Increment 4: Configure data storage and categorization to an Excel Doc for subsequent file formats.

Increment 5: cleaning up data gathered, cleaning up Excel files.

**Spiral:**

Objective Setting: Specific objectives for this phase of the loop will be identified, along with possible risks. As an example, the first objective would be to implement scraping the HTML documents sent by our stakeholder and potential risks could be incomplete data extraction or performance bottlenecks when scanning large HTML documents.

Risk Assessment and Reduction: Each risk would be analyzed, and steps would be taken to reduce the risk. Validation checks could be implemented to verify correctness of extracted data and code could be optimized to scan large documents without putting too much load on the PC.

Development and Validation: According to our main risk, a development model would be chosen. Since our main risk is the variability of different file formats, an agile method is best. This is due to:

Adaptability to Changes: Since we’re going to be working on multiple file formats, the agile format will allow us to adapt to new file formats as they come.

Continuous Improvement: Since the model involves iterative cycles, each iteration provides an opportunity to refine the parsing algorithms. Regular feedback loops allow for continuous improvement based on real-world usage.

User Involvement: By continuously checking with our teacher (primary stakeholder), we can observe how the software works from another perspective, identify risks, and adjust.

Planning: After reviewing the work done in the current loop, decisions will be made on what to do in the next loop. The loops will continue until the software is finished.

**Integration and Configuration:**

Since a similar open-source Medical Quotation project cannot be found on the web, the integration and configuration model is not possible for the time being.

**SCRUM:**

Product Backlog: After talking to our main stakeholder, one member of the team would become the Product manager and maintain the Product Backlog, containing all the features/tasks the other members must accomplish.

Sprint Planning: At the beginning of each sprint, the development team, along with the Product Owner, participates in a Sprint Planning meeting. The team selects a set of items from the Product Backlog to work on during the sprint, considering priorities and dependencies.

Sprint Execution: The selected work items are then implemented during the sprint, typically lasting two to four weeks. Daily Scrum meetings are held for brief status updates and to discuss any challenges the team is facing. The number of tasks accomplished during the sprint is recorded, becoming the velocity and giving a good idea of the team’s capabilities.

Incremental Development: At the end of each sprint, potential deployable software has been built, which can be shown to our stakeholder.

Sprint Review: At the end of each sprint, the team conducts a Sprint Review meeting. During this session, the increment is demonstrated to stakeholders. Stakeholders provide feedback on the implemented features, including the software's ability to parse and extract information from HTML documents.

Sprint Retrospective: Following the Sprint Review, the team holds a Sprint Retrospective to discuss what went well, what could be improved, and any adjustments needed for the next sprint. The retrospective provides an opportunity to continuously enhance the team's approach to handling HTML document structures.

Backlog Refinement: The Product Owner continually refines the Product Backlog, adding new items or adjusting priorities based on evolving requirements or feedback. Handling different HTML document structures can be prioritized based on their impact and importance.

**Advantages and Disadvantages:**

**Waterfall:**

**Advantages:**

Simple and easy to understand.

Well-structured and easy to manage due to its linear nature.

Each phase has specific deliverables and milestones.

**Disadvantages:**

Lack of flexibility as changes cannot be easily accommodated once a phase is completed.

High risk of project failure if requirements are not well-defined initially.

Limited user involvement until the later stages of development.

**Plan Driven Incremental:**

**Advantages:**

Allows for the development of core features first, followed by incremental additions.

Early delivery of partial functionality.

Feedback from users can be incorporated into subsequent increments.

**Disadvantages:**

Requires careful planning to determine the order of features to be developed.

Integration of increments may pose challenges.

Limited scalability for large projects.

**Spiral:**

**Advantages**:

Iterative approach allows for early identification and mitigation of risks.

Flexibility to accommodate changes at any stage of the development process.

Emphasizes risk management throughout the project lifecycle.

**Disadvantages**:

Can be complex and difficult to manage for small projects.

Requires extensive documentation and monitoring of risks.

Costly and time-consuming due to its iterative nature.

**SCRUM:**

**Advantages**

Flexibility to Adapt: SCRUM allows for flexibility and adaptability to changing requirements, making it suitable for projects with evolving needs, such as handling different file formats.

Continuous Feedback: Regular feedback from stakeholders ensures that the software meets user expectations and can be adjusted.

Incremental Delivery: Incremental delivery of product increments allows for early value delivery and faster response to user needs, enabling early validation of HTML parsing capabilities.

**Disadvantages**:

Complexity in Planning: Continuous reprioritization and planning can be challenging.

Dependency on Stakeholder Availability: The success of SCRUM relies on active stakeholder involvement. Limited availability or unclear feedback from stakeholders can hinder progress.

Risk of Scope Creep: Without careful management, the iterative nature of SCRUM can lead to scope creep, potentially impacting project timelines and budget. More features = less time, less usability.

**DECISION:**

Considering the scope of the project, the size of our team(3 members), the fact that this is the first time any of the team members will work on a project of this type, and time constraints, we believe that the best model for us is the Plan Driven Incremental Model.

We would start by implementing core increments, getting feedback from our teacher (primary stakeholders), and incorporate that feedback into subsequent increments. This will prevent us from making software that does not meet requirements or accomplishes something unnecessary. By incrementally making our software and checking with our teacher, we can ensure we stay on the correct path.