Process Report

Date: 17/12/2023

Location: Fontys Hogeschool

Members:

- Danila Solovenko
- Claudiu Badea
- Angel Rusev
- Nazim Ahmedov

Tutor: Michiel Koehorst

Contents

Work Division

During the course of this project, the team members were assigned specific roles and responsibilities to ensure effective collaboration and task completion. The division of work among team members is outlined below:

- Danila – back-end implementation, documentation
- Claudiu - back-end implementation, documentation, UI/UX
- Angel – back-end implementation, documentation
- Nazim - back-end implementation, documentation

Personal Reflections

1. **Danila:**
    - Individual Contribution: Implemented shift automation and seamlessly integrated it into the UI, enhancing the application's efficiency. Resolved issues in employee and shift addition functionalities, ensuring a smoother workflow. Overhauled the website announcement system for improved communication. Conducted significant code refactoring to enhance the application's maintainability and scalability. Delved into handling heavy algorithms, optimizing code performance, and learned strategies to improve overall code quality.
    - Challenges Faced: Encountered complexities in shift automation and UI integration, requiring careful debugging and problem-solving. Addressed issues in employee and shift addition functionalities, working towards a more robust solution. Tackled challenges in the website announcement system to ensure reliable communication.
    - Lessons Learned: Gained valuable experience in handling complex algorithms, contributing to a deeper understanding

of algorithmic efficiency. Developed skills in code optimization, identifying areas for improvement, and implementing strategies to enhance overall code quality.

2. **Claudiu:**

- Individual Contribution: Full Front-End Design for the WinForms Application, as well as full implementations of announcements and vacations functionality, plus the extension of employee functionality and forget password. Built up the base documentation and extended upon it (Test Plan, Test Report, Activity diagrams, UML and more)
- Challanges Faced: WinForms custom elements acting up and not working, database migration issues
- Lesson Learned: more GDI+ knowledge and some UI animation assets

3. **Angel:**

- Individual contribution: Implemented shifts functionality of the application - worked on the UI and Logic, created algorithm for looking up employees by partial name and shifts correlated to the partial name. I also worked on improving the code logic for the website functionalities and fixed bugs related to them. Worked on improving the security of the application by hashing the sensitive user data as required by client and other security features as well.
- Challenges faced: Encountered difficulties with the performance of the algorithm at first and had to optimize it to work faster and more efficiently. Group work integration through GIT was sometimes causing minor challenges due to some cache being pushed initially as well. which we resolved afterwards but still was a challenge.
- Lessons learned: The group project definitely helped me hone my soft skills in terms of teamwork and also allowed me to get a close sense of what a real working environment would be like when developing a software product. I learned also a lot about UI from some of my teammates and this helped me get a better understanding of modern UI concepts. Additionally I learned about optimizing algorithms and security in the applications.

4. Nazim:

- Individual Contribution: Implemented Email Notifications functionality, Ensured Salary encryption for enhanced security, Forgot Password Functionality, Stock Change Amount Functionality, Hot Fix on Vacation Table, UML, Project Plan, URS, Process Report

- Challenges Faced: Faced difficulties during the salary encryption process, necessitating database refactoring for resolution

- Lessons Learned: Recognized the critical importance of thorough planning and testing, particularly in areas involving sensitive data encryption. This experience underscored the need for a proactive approach to anticipate and address potential challenges during the development process.

Reflection on Applying Iterative

- **Strengths:**
  - Development experience and group communication
- **Weaknesses:**
  - Client and tutor feedback processing

Reflection on Differences Between Waterfall and Iterative

Waterfall Model:

Advantages:

1. **Clear Phases:** The Waterfall model provides a structured and sequential approach with distinct phases, ensuring a well-defined process from requirements to implementation.
2. **Initial Stability:** Requirements are fixed at the beginning, offering stability in terms of project scope and deliverables.
3. **Documented Processes:** Extensive documentation is produced at each stage, facilitating comprehensive project understanding and future maintenance.

Disadvantages:

1. **Rigidity:** The fixed scope lead to rigidity when dealing with changes or evolving requirements during the project.
2. **Late Feedback:** Stakeholder feedback is typically gathered at the end of the project, which may result in late adjustments and increased costs.
3. **Long Delivery Time:** Due to its sequential nature, the Waterfall model often requires a longer delivery time before a functional product is available.

Agile Model:

Advantages:

1. **Flexibility:** Iterative development allows for flexibility in accommodating changing requirements throughout the project.
2. **Early Prototypes:** The iterative model facilitates the creation of early and evolving prototypes, enabling stakeholders to visualize and interact with the product sooner.
3. **Continuous Feedback:** Regular feedback loops provide opportunities for stakeholders to review and provide input, enhancing collaboration and reducing the risk of misunderstandings.

Disadvantages:

1. **Complexity:** Managing multiple iterations can introduce complexity, requiring careful planning and coordination.
2. **Documentation Challenges:** The emphasis on working software may sometimes result in less comprehensive documentation, requiring a robust communication strategy.
3. **Potential Scope Creep:** The flexibility of iterative development can lead to scope creep if

not properly managed, impacting timelines and resources.