

Frankfurt University of Applied Sciences

Masters of Engineering

Information Technology

Course: Agile Development in Cloud Computing Environment

Project: 3b Access Platform for Providers (APP)

Authors:

Ankita Talande 1427349

Abstract

"In recent years, the software development industry has been prominently shaped by the adoption of agile methodologies. Agile software development represents a collection of approaches emphasizing iterative development, wherein teams collaboratively craft solutions to problems. This process is facilitated by self-organizing, cross-functional teams. Notably, Scrum and Kanban stand out as two of the most prevalent Agile techniques. Our objective is to develop an application using agile methodologies, specifically designed to interact with other applications. This application will access master agreements and available services within the company, facilitating providers in extending offers to employees.

Keywords: Software Development, Provider Platform, Agile, Sprints, Iterative Methods, Scrum, Kanban.

1. Introduction –

"Agile software development" embodies a spectrum of iterative methodologies where requirements and solutions evolve collaboratively within self-organizing, cross-functional teams. These methods foster a structured project management approach, advocating frequent inspection and adaptation. Embracing a leadership philosophy that emphasizes teamwork, self-organization, and accountability, Agile incorporates engineering best practices to enable the rapid delivery of top-notch software. Moreover, it aligns development processes with client needs and organizational objectives.

Within Agile, Scrum stands as a prominent subset—a popular and lightweight framework for agile development. Distinguished by its specific roles, artifacts, and time boxes, Scrum offers a structured approach that sets it apart from other Agile methodologies.

Kanban boards, on the other hand, serve as visual workflow tools engineered to streamline work processes and bolster productivity by curbing work-in-progress. This visualization tool enhances transparency, enabling the team to identify and address problematic work phases promptly. By fostering increased visibility, Kanban facilitates prompt solutions, ultimately enhancing team efficiency and effectiveness.

1. **Scrum Methodology**

In Scrum, three distinct roles — the ScrumMaster, the Product Owner, and the Development Team — collectively steer the process. These roles collaborate closely, ensuring seamless information exchange and swift problem resolution among their responsibilities.

2.1 **ScrumMaster**

The ScrumMaster shoulders multiple responsibilities aimed at streamlining the process, eliminating obstacles hindering productivity, and orchestrating crucial meetings. This role involves guiding the Product Owner in leveraging Scrum effectively to achieve their objectives and maximize return on investment (ROI). Encouraging empowerment and innovation among the development team to enhance their quality of work life is pivotal. Additionally, improving the team's productivity, refining engineering practices for potentially shippable increments, maintaining everyone's awareness of team progress, and keeping stakeholders updated fall under the purview of the ScrumMaster. Notably, the ScrumMaster acts as an educator and supporter for other roles, possessing in-depth Scrum knowledge.

The ScrumMaster remains vigilant about the project's status, comparing it to anticipated progress, resolving obstacles, and adapting swiftly to emerging challenges. As a liaison between stakeholders, the ScrumMaster shields the Team from external interferences, though task allocation remains the responsibility of the Team themselves.

2.2 **Product Owner**

The Product Owner stands as the custodian of criteria and requirements. Serving as the "single source of truth," they delineate requirements and their intended execution sequence for the Team. Acting as a bridge between the Team and business stakeholders, clients, and their product-related demands, the Product Owner manages inquiries about product requirements. They collaborate closely with the team to articulate user-facing and technical needs, document requirements, determine their implementation sequence, and curate an up-to-date and detailed Product Backlog. Moreover, the Product Owner assesses the readiness of implementations for release, ensuring the requisite functionality and quality, while also setting release timelines for finished work.

2.3 **Development Team**

Comprising self-organizing, cross-functional individuals, the Development Team undertakes the hands-on creation and testing of the product. Empowered to decide how to execute tasks, the Team is responsible for delivering the final output. Hence, the Team self-organizes, determining task allocation and division during Sprints. Ideally, the team size ranges between five to nine members to ensure optimal collaboration and productivity.

2.4 **Sprint**

Sprints define Scrum's iterative work periods, fostering consistency and swift feedback cycles. The shorter duration facilitates continual inspection and adaptation, crucial for efficient workflow management and reducing risks associated with longer cycles. Sprints follow a sequential pattern, immediately commencing after the conclusion of the preceding Sprint, maintaining a cadence for frequent iterations and feedback.

End

Provider

Evaluate cycles for each provider offers

End of publish

Add the data for each position

Accept the offers

Publish master agreement for offers

Create new Master Agreement

Establish master agreement

2.4.1 Sprint Planning:

The project was planned into 4 sprints with 2week of sprint intervals as dated below:

A diagram of a agile project

Description automatically generated

Sprint 1: First major task in the sprint is Architecture design and pattern discussion. Edit profile and Logout -Frontend.

Sprint 2: Design of Login page and its sub fields on Frontend and Blackened, implementation of Nav-bar.

Sprint 3: Sharing of APIs created between group 1 and group 2 for open services agreement.

Sprint 4: Update the offered employee status with the response from project A (Profile Accepted/Rejected)

Sprint 5: Testing of data between group 2 and 4. Updating selected profile in database and fixing file path object.

1. Kanban Board

Teams can effectively visualize and control their operations using Kanban boards, employing cards to represent tasks visually. Each Kanban card includes comprehensive details about the task, such as the due date, assigned personnel, and a description of the work in progress.

A screenshot of a computer

Description automatically generated

Figure: Kanban Board

**A product backlog** is a dynamic, prioritized list of tasks, features, and enhancements that a development team plans to work on in a product. It evolves over time, reflecting changing priorities and business needs. The backlog helps prioritize work, estimate effort, and provides visibility into upcoming tasks for efficient product development. Managed by the product owner, it serves as a living document in agile development.

**Task Name:** Clarity in task labels is crucial for understanding the nature of the work. Tasks that begin with a verb, like "Add functionality to web app," provide a high-level overview of the required work.

**Key Dates:** Depending on your team's process, certain dates may hold significance for both project managers and developers. For example, a project manager might need the task's start date, while a developer would require the task's deadline.

**Task Owner:** The designated task owner is responsible for ensuring that the job is completed. Any queries regarding the task should be directed to this individual.

**Task Status:** In the Kanban system, task status is often indicated by the card's location on the Kanban board. It is essential for everyone to be familiar with the various stages of your team's Kanban board, as each team may have a distinct process for tracking work in progress.

**Task Priority:** Illustrating the significance of an activity in relation to others is crucial. Setting priorities can help maximize lead time for tasks that demand it, assisting the development team in organizing their work effectively.

**Subtasks:** Some tasks may be sufficiently labor-intensive to warrant division into smaller subtasks. These subtasks can be linked to a parent task, contributing to a more organized workflow.

1. Requirement

General requirements:

• User wants to login using authentication component.

• Provider Admin wants to edit provider credentials (name).

• Provider Admin wants to configure user management for provider (register new user, deregister user).

• User wants to see the offers of the company „future-X“ regarding master agreement types (example: technology-based).

• User wants to provide an offer to establish a master agreement.

• User wants to bid on open service requests and material request.

• User wants to upload profiles of employees for service requests and responses to material requests.

• User wants to make suggestions based on the knowledge of each offered employee.

• User accepts contract if his offer has been chosen. Negotiation can take place.

• Further specification takes place during collaboration with the customer.

• APIs need to be provided to other groups.

1. Software Achitecture
2. Database Model

Mongodb Atlas is being utilized by us. A separate database is maintained for each provider. The collections available for each provider include Agreements, Bids, Employees, Offers, OpenServiceRequests, and Users.

1. Usecase Diagram

This diagram illustrates the use case for the provider in phase 1. Following registration, the provider can log in. Once logged in, the provider can modify personal details and access the master agreement provided by the company. The provider can request daily rates and has the option to accept or reject master agreements. After accepting, the provider can then bid for active master agreements.

A diagram of a company

Description automatically generated

This use case diagram represents the provider's functionalities in phase 2. During this phase, the provider can access open service requests from the company, aligning with the accepted master agreements. The provider is equipped with the capability to create employees and extend offers to them based on the existing open requests. A diagram of a company

Description automatically generated