Sri Lanka Institute of Information Technology

Methodology, Team Structure and Deliverables Plan

SE5100 - Rapid Application Development

2023-03-16

Table of Contents

Group Details

- MS23003132 - Akila Kasun Siriwardana
- MS23006966 - Malaka Silva
- MS23001602 - Yanuka Deneth
- MS23046320 - Tavish Perera
- MS23001534 - Eesha Wickramasinghe

Methodology

The RAD methodology is a flexible approach to efficiently create and implement software applications. It is designed to accommodate changes and new requirements such as updates, features, and functions throughout the development process unlike traditional models like Waterfall, where it is difficult to change functions, features once the software is built.

The term "Rapid Application Development" was initially introduced by James Martin in his book of the same name. Martin developed this approach in the 1980s, and in 1991 he formally presented it as a concept that was built on the groundwork of individuals such as Barry Boehm.

While both RAD Methodology and Agile Development Process share many similarities, there are some differences as well.

i. Approach: RAD methodology is a linear sequential approach that follows the traditional Waterfall mode where each phase should be completed before the next. Whereas, Agile is an incremental approach that uses an iterative cycle of planning, designing, building, testing, and delivering in a flexible manner.

ii. Time Period: RAD is a time-bound approach where the development cycle is completed within a specific time while Agile is an ongoing process that focuses on delivering small increments of working software.

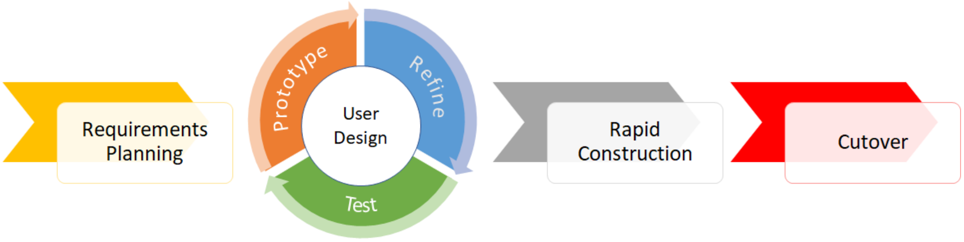
iii. Involvement: RAD involves the customer in initial phase of the development and their feedback will be incorporated throughout the development process. In Agile on the other hand, customer will involve in every phase of the development and their feedback will be incorporated continuously.

iv. Focus: RAD is more focused on rapid prototyping and quick delivering of software while Agile focuses on developing working software that meets customer needs and delivers values.

In summary, RAD is more suitable for projects that have a well-defined scopes and tight deadlines, whereas Agile is more suited for projects that require ongoing development and continuous improvements.

There are four main phases in RAD methodology,

- Requirements Planning
- Design
- Rapid Construction
- Cutover



These phases are iterative, meaning that each phase can be revisited as necessary to make changes based on feedbacks and test results. This allows the development process to be flexible and accommodate changes in requirements.

Phase 1 – Requirements Planning

In contrast to other software development models, the RAD model places a significant emphasis on the flexibility and adaptability of the project requirements. The first phase of the RAD methodology known as requirement planning phase, and it is critical phase for success of the project, and it lays the foundation for the rest of the development.

In this phase developers, clients and team members define the scope of the project and identifies the requirements of the project as well as current potential issues that would need to be addressed during the development. This broad nature of the requirement, the RAD model provides the project team with a foundational understanding of the project's goals and objectives, also enables for the inclusion of the specific requirements at different points in the development cycle.

Throughout the RAD development cycle, specific requirements are fed into the project as they are identified or refined. This approach enables the project team to respond effectively to change in the project environment and to deliver a final product that meets the needs and expectations of stakeholders.

This phase is already completed by the team, and the work performed is summarized below.

During the requirements planning phase of the RAD methodology for the proposed CV generator for engineers and developers, the primary focus was on the need and expectations of the end users, I.e., engineers and developers. The project team identified and prioritized the key requirements and features of the application.

Furthermore, the project team paid significant attention to the design of the interface, user experience and identify the essential features of the application during the requirements planning phase.

Phase 2 – Design

The second phase of the RAD model is the user design. During this phase, the project team will focus on designing the user interface and user experience of web-based application for the CV generator. Team will work on creating wireframes and prototypes and will ensure that the design of the application is consistent with the requirement identified in the previous phase.

User design phase involves gathering feedback and collaborating with developers to build prototypes. This is an iterative process in which users interact with the prototype and provide feedback and work with the project team to ensure the final product aligns with the initial requirements. This iterative process allows for continuous improvement and refinement of the application's design.

Phase 3 – Rapid Construction

In phase 3, the prototype and beta systems developed in the design phase are transformed into a functional model. Since the design phase has addressed most of the issues and changes, the final model can be created more efficiently than with a traditional project management approach.

The phase involves several smaller steps such as preparing for rapid construction, developing programs and applications, coding, and testing at various levels. The software development team consists of coders, programmers, developers, and testers collaborate during this phase to ensure that the end result meets the client's requirements and expectations.

This phase is crucial because clients can provide feedback throughout the process, suggesting changes or new ideas to overcome any challenges that may arise.

In summary, main tasks of Rapid Constructions are,

i. Creating application architecture

Application's overall structure and design are finalized including the system hardware and software requirements.

ii. Rapid prototyping

Developers create series of small, working prototypes of the final product to validate system requirements and to test its functionality. This approach helps to figure out potential problems and makes it easier to modify or refine the product.

iii. Component integration

Various software components and modules are integrated to create a functioning system.

iv. Coding and testing

Once components and modules are integrated, developers write actual code for the system and do extensive testing to make sure that the system is working properly.

v. User feedback

Users are encouraged to test the system and provide feedback, which will then be used to improve the final product.

Phase 4 – Cutover

Cutover is a critical stage in the Rapid Application Development (RAD) methodology that involves moving the software from the development environment to the production environment. It is the process of switching from the old system to the new system, which can be a challenging and complex task.

The cutover stage in RAD methodology includes several activities that are designed to ensure a smooth transition from the old system to the new system. These activities typically include:

1. Final testing

Before the cutover, the system is tested thoroughly to ensure that it is stable and ready for production. Any last-minute issues are identified and addressed during this stage.

2. Data migration

Data from the old system is transferred to the new system. This process may involve data cleansing, formatting, and validation to ensure that the data is accurate and complete.

3. User training

Users are trained on how to use the new system. This training may include classroom instruction, online tutorials, or other forms of training to ensure that users are comfortable with the new system.

4. Parallel processing

During the cutover, both the old and new systems may run in parallel for a period of time to ensure that the new system is functioning properly and to minimize any disruption to business operations.

5. Post-cutover support

After the cutover, developers provide ongoing support to address any issues that may arise and to ensure that the system is working as expected.

Overall, the cutover stage in RAD methodology is a critical step in the software development process. It involves several activities that are designed to ensure a smooth transition to the new system while minimizing disruption to business operations

Team Structure

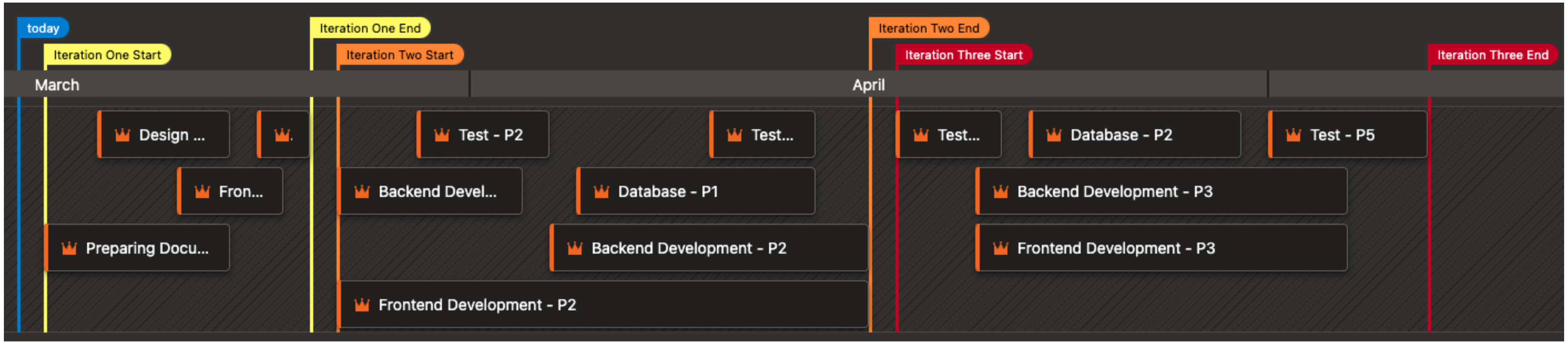
Our team consists of five members as follows:

ID	Name	Role
MS23003132	Akila	Frontend Developer, UI-UX
MS23006966	Malaka	Frontend Developer, UI-UX, Project Manager
MS23001602	Yanuka	Frontend Developer, Database, QA
MS23046320	Tavish	Backend Developer, Database, QA
MS23001534	Eesha	Backend Developer, Database

Deliverables Plan

There will be three Iterations on the below mentioned dates. We have broken down the individual tasks that will be done per iterations. The tasks will be maintained in our Azure DevOps Boards.

- Iteration 1 – 16th March 2023 to 26th March 2023
- Iteration 2 – 27th March 2023 to 16th April 2023
- Iteration 3 – 17th April 2023 to 7th May 2023



First Iteration

Task	Start Date	End Date	Person(s)	Description
Documentation	16-03-2023	22-03-2023	All	Quick note down of all the requirements for the application
Design	18-03-2023	22-03-2023	Akila, Malaka	Designing the basic looks of the application on Figma
Frontend Dev – P1	21-03-2023	24-03-2023	Akila, Malaka, Yanuka	Working the basic website with no functionality
Test – P1	24-03-2023	25-03-2023	All	Testing the basic website and getting ready for the next phase

Second Iteration

Task	Start Date	End Date	Person(s)	Description
Frontend Dev - P2	27-03-2023	15-04-2023	Akila, Malaka, Yanuka	Adding functionality to the Frontend application
Backend Dev – P1	27-03-2023	02-05-2023	Eesha, Tavish	Working on getting the basic backend and APIs setup
Test – P2	30-03-2023	03-04-2023	All	Testing for any issues on the Frontend and Backend
Backend Dev – P2	04-04-2023	15-04-2023	Eesha, Tavish	Implementing the backend and APIs
Database – P1	05-04-2023	13-04-2023	Eesha, Tavish, Yanuka	Designing and implementing the basic Database
Test – P3	10-04-2023	13-04-2023	Yanuka, Tavish	Making sure the Database is following the written documentation

Third Iteration

Task	Start Date	End Date	Person(s)	Description
Test – P4	17-04-2023	20-04-2023	All	Pre-Testing to see what we can build better and optimise more.
Frontend Dev – P3	20-04-2023	03-05-2023	Yanuka, Akila, Malaka	Optimising the frontend application for better performance and efficiency.
Backend Dev - P3	20-04-2023	03-05-2023	Eesha, Tavish	Optimising the backend code and adding security layers in.
Database – P2	22-04-2023	29-04-2023	Eesha, Tavish, Yanuka	Optimisations on the Database
Test – P5	01-05-2023	06-05-2023	All	Final testing phase to check everything.