

3.0 Methodology

This chapter details the researchers' chosen methodology that was applied to the capstone project, which is the Rapid Application Development Methodology, under the umbrella of Agile Methodologies. Each phase of the methodology is discussed and further associated to specific activities accomplished throughout the entire project duration.

3.1 Description

Rapid Application Development (RAD) is a type of agile software development methodology which is focused on application prototyping and iterative development throughout its stages. This methodology is best for small and highly motivated teams that will handle a small project with a defined scope that may be broken down into smaller parts.

The five (5) core elements needed for proper implementation of RAD are, prototyping, iterative development, time boxing, team members, and management support. Prototypes are used to jumpstart the process. This is where the developers create an image of the project that would be aligned with client requirements. It is often a lightweight version of the entire project, ensuring that it captures all necessary features and functions. It is also used as a tool to further refine the client requirements and to identify the areas to prioritize. Prototypes are continuously enhanced by adding a more intensive execution of the certain functions, which is also called an iterative development. During the development phase, it is important to avoid scope creeps or the tendency for the project scope to become bigger and bigger (Schwalbe, 2011), which is the main purpose of time boxing. Time boxing is the process of limiting the features of the application for future versions in order to focus and deliver the main features settled with the client. Without time boxing, it defeats the purpose of having a fast-paced agile methodology wherein the project will continue to extend as the scope extends. To put the project altogether, a small, highly skilled and experienced team is needed to materialize the project. In addition, high involvement and support of the management and/or client is needed to avoid extended iterations.

The RAD Methodology was chosen for this project due to its ability to cater to the fast pace of the project at hand while at the same time not affecting the quality of the project deliverable. RAD allows the developers to iteratively go back to the client for feedback all the while being able to move in other phases of the project. This also permits the users of the actual system to be as involved as possible with the development of the system to ensure it is up to their standards.

3.2 Steps and Tools/Techniques/Inputs and Outputs

The RAD Methodology Framework encompasses four (4) steps: Requirements Planning, User Design, Construction, and Cutover (*Refer to Appendix D for RAD Methodology Framework*).

3.2.1. Requirements Planning

Requirements planning is the first phase in RAD. Understanding the organization's current problems, familiarizing with the organization's current system, and proposing a solution are the general objectives of this phase. The team secured and signed a contract, specifically a Non-Disclosure Agreement (NDA), with the organization to ensure a safe information exchange that would only be used for academic purposes (*Refer to Appendix L for Non-Disclosure Agreements of Team Members*). In order to meet these objectives, the development team and the client of the project have regularly met and discussed until everyone on the team agrees with the objectives, functionality, and scope of the project that they were able to come up with. All exchanges, may it be via a face-to-face meeting or an email questionnaire, are recorded, granted that the team receives appropriate permission from the interviewees (*Refer to Appendix I for Interview Transcripts and Appendix J for Certificate of Interview*). Some companies use tools like IBM's Rational Rose or Microsoft's Visio to help them in requirements planning. The three main activities for this phase are:

- **Research Current Situation.** Researching the current environment and the current situation of the company initiates the requirements planning phase. The developers reviewed the current system, existing CASE repositories, and similar systems.
- **Define Requirements.** The scope of the proposed system, its module, functions, and features are all developed in this activity.
- **Finalize Requirements.** An estimate duration of the implementation of the system is made, along with a formal document that includes the scope of the proposed system.

3.2.2. User Design

During the second phase, the main objective is to analyze business activities and data with regards to the proposed system, to develop proposed screen layouts, and to plan for the transition from the current to the proposed system. The steps for doing the User Design are producing detailed system area model, developing an outline of the system design, refining the system design, preparing implementation strategies, finalizing the system design, and obtaining approval for construction. In this phase, the developers used a Joint Application Development (JAD) technique and hand drawn screens to serve as storyboards for the flow of the system. This phase also saw the developers creating test scenarios to test the certain modules they would be creating in the next phase. JAD sessions were done between the customer and the project team in order to discuss and reach an agreement regarding the main scope, objectives, and specifications, and limitations of the project. JAD follows six (6) guidelines to be followed in each session; simplify the agenda into a workshop, identify participants and concerns, quantify information, clarify requirements, unify outputs, and satisfy the customer/s. This was also in-line with the concept of using a participative design that ensured that the customer was included during the designing face and their comments and suggestions were taken into account in the final output.

3.2.3. Construction

This is also considered as the Development Stage. As the name says, this is where the developers completed the construction and development of the system by converting the prototype into the final output and adding the necessary functionalities with actual coding. Within this phase, there was an iterative cycle for development, testing, and evaluating or requirements refining. Each cycle lasted between one day and three weeks. The cycle began once a prototype has been successfully developed. The developers then tested the prototype guided by the test plan and test scripts produced in the User Design stage (*Refer to Appendix S for the Case Scenario and Appendix U for the Test Cases*). After testing, a focus group meeting was done in order to determine the necessary changes and an update of requirements for the next iteration. The focus group meeting discussed the concerns that arose during testing and what the necessary action should be with the help of producing a list of issues to be addressed rated by priority, and a new set of test scripts. The development of the prototype then started

again, and so as the cycle. The cycle lasted up to three (3) weeks. The management served an important role wherein they had to ensure that the project was on track and continuously progressing according to the plan, whilst also keeping the client informed and the team motivated. In addition, corresponding user documentation, such as the technical manual, user manual, conversion plan, and installation plan are also produced after the completion of the system.

3.2.4. Cutover

This is also considered as the Deployment or Implementation Stage. Final User Testing or also known as User Acceptance Testing (UAT) is done in this stage where a complete system review is done with the stakeholders to assess if it meets their expectations and requirements. If the system failed the expectations of the key stakeholders, it will return to the iterative design loop for modification. On the other hand, if the system meets the expectations of the stakeholders, system installation, data conversion, and user training of the system is done. The developers had conducted their UAT with the actual users of the system and they had signed off their approval of the system (*Refer to Appendix V for the UAT Forms*). The developers have also observed the operation of the system to lookout for potential enhancements and were on standby for troubleshoot concerns. The developers will then handover all related assets of the system which includes, the working system and all data used in creating the system. The objective of this stage is to integrate and align the system to the business of the key stakeholders.

After successfully implementing the system, the organization plans on integrating it with other systems such as, Task Management System, Integra, and systems that are currently under development, for their goal of *One Data*. The organization will be responsible for creating a shared dashboard for the integrated systems. The data from the Project Management System will offer an overview of all projects currently in progress and to be initiated, that may be broken down into separate departments, depending on the logged user.

3.3. Gantt Chart

The project will run through three (3) academic terms, roughly 3-4 months each, which will last for approximately a year. The first term will solely be occupied with the requirements planning phase to ensure that there is a clear image before moving forward with the project. During the transition to the second term, it will slowly begin the user design phase. The database design for the project management system is the most important activity in this phase as the overall design should be based on the database structure, which will occur hand in hand with TEI. The second term, the cycle of the user design and construction phases will reiterate until a proper output is achieved. The earlier part of the third term will consist of further construction of the phase and will move on with testing and user acceptance. The rest of the term is reserved for the implementation phase and further documentation (*Refer to Appendix E for the Gantt Chart*).