The Report committee for LuisLopez-Cabanas

Certifies that this is the approved version of the following report:

Improving a Management Tool through the use of Software Architecture

APPROVED BY

SUPERVISING COMITEE:

Supervisor: ___________________________________

Kathleen Barber

Co-Supervisor: ___________________________________

Herbert Krasner

Improving a Management Tool through the use of Software Architecture

by

Luis Lopez-Cabanas, BSEE, MBA

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2014

Dedication

I would like to dedicate this report first of all to God and my family, especially my wife, Joelyn, and my sons, Luis Joel and Edwin Miguel. They have been extremely supportive during my journey to obtain an Engineering Master Degree.

Acknowledgement

I would like to acknowledge Dr. Herb Krasner for being my Supervisor on this report as well as Dr. Suzanne Barber, who has always been supportive during the report process.

Abstract

Improving a Management Tool through the use of Software Architecture

By

Luis Lopez-Cabanas, MSE

The University of Texas at Austin, 2014

Supervisors: Kathleen Barber, Herbert Krasner

Architecture Design for deploying or improving a tool or application is a vital step which should be neither ignored nor avoided. The architecture will provide the framework and instructions on how the tool needs to be created in order to comply with the stakeholders' most important requirements. Utilizing data collected from the different stakeholders involved in the use of an existing tool, an effective architecture structure will be created to improve the tool and satisfy the users' needs to achieve the desired goals in it. Through the use of an effective architecture design, a toolkit will be created to improve an existing Management Tool to provide a desired outcome.

We have learned that having an architecture established prior to starting a development project or in the early lifecycle stages will positively influence the

project’s outcome, timely deliverables and financial impact associated with it. It is crucial to consider all aspects surrounding a process or software design, such as stakeholder requirements, internal and external customer feedback, and any particular feature that will guarantee a reliable structure and deployment process.

Table of Contents

List of Tables

List of Figures

Terminology	Definition
Tool	The system used to Manage the workflow of projects deliverables.
Toolkit	Components created to improve the existing tool, satisfying the stakeholders' requirements.
Application	The software that will execute the toolkit operations
System	The entire hardware and software environment the tool is currently running and will continue to run on.
Product	All the deliverables required by a customer (Equipment and Documentation).
Project	The collaboration of different interacting groups to meet the customer requirements.
Process	All the activities performed to achieve the desired goal.

Terminology

Chapter 1: Background and Literature Review

1.1 Background

The main focus of this research is to create a particular Software Architecture Design in order to improve an existing document management tool that is currently not delivering the desired results. This is accomplished by improving communication within the different groups involved in a project. At the present time, this inefficient document management tool is being used to manage all the documentation and distributed data among all the groups involved in handling various sub projects.

In this specific case presented, a Telecommunications Company, which shall be referred to as XYZ, presents the following problem: Their projects are either not being delivered to customers in a timely matter, they are not fulfilling the specifications, or they are being noncompliant with the requirements established. The company is looking to identify the root cause of why most of their projects are failing to meet deadlines. What has been found to be the main reason is the lack of effective communication within the groups involved in a specific project. In search for solutions to improve communication, they have found that there is no documentation existing to explain the sub processes or how the tool should operate. Due to the lack of detailed process documentation, each group is using a different method in order to convey the information in a manner they assume is necessary.

It is within this scenario where Software Architecture comes into play. Company XYZ has decided to improve the current tool by creating an architecture that will include the documentation of all the necessary processes and improve the communication channels between the different groups working on a project, while ensuring timely delivery and accuracy.

The creation of an upgraded and enhanced architecture that will provide a framework for the tool expected functionality and successful result is the motivation for this report. We will demonstrate how, after gathering various stakeholder requirements, an improved architecture was created and analyzed based on its structure complexity and cohesion. This will also allow us to estimate how many resources will be needed in order to deploy and maintain the tool effectively.

Following is the literature review summarizing the previous research done on the practice of creating a software architecture.

1.2 Literature Review

An important step in creating an effective application or tool is to properly translate the requirements of different users' needs into a real and functional tool. It is essential to identify which components are possible to achieve and which not. This is where the architecture process comes into play. As expressed by

Bass, Clemens and Kazman, the architecture is a bridge between those (often abstract) business goals and the final (concrete) resulting application [1]. It is taking the informal outline of the requirements and creating a formal design to be implemented [3].

It wasn't until recently that creating an architecture prior to the implementation of an application was ever considered. For decades, when an application was needed, it was based only on the technical requirements. Nowadays, with the implementation of an architecture; many other factors are taken into consideration. Such factors include the environment, stakeholders' needs, and technical aspects of the tool. With the addition of an architectural analysis, all decisions made during this stage are relatively inexpensive, making it very cost-effective. [4].

As it has been mentioned, new applications or modifications to existing applications or tools are created to satisfy different requirements, provided by stakeholders, dictating the performance, availability, security and compatibility with existing tools. They need to be able to adapt to new or upgraded configurations in the future. An architecture will improve the delivery of these requirements at the beginning of the process while providing the necessary elements for bringing a successful outcome.

As witnessed in the majority of the literature found, there is no definition of a good or bad architecture. It all depends on which structure will best fit the application tool to be deployed. To create the best architecture to fit the desired application, a structure, or set of structures, needs to represent the view of the stakeholders. These structures will follow one of the three categories [1]:

- Module Structure – Showing the functional units essential to its design and construction.
- Component-and-connector Structure – Enabling the application's required elements to be configured and structured following the necessary interactions (i.e. components and connectors).
- Allocation structure – show how the requirements will relate to the structures outside the application such as CPUs, networks, development teams, etc.

The selection of the structures to be used will be based solely on the most relevant quality attributes determined by the stakeholders' needs, as the structure will give leverage to them in the application. Furthermore, the selected structures will provide the primary source for the architecture documentation.

Based on different analyses and perspectives reviewed, we can see that selecting the correct structure to represent the application will be dependent on the quality attributes received from the stakeholders, but also on the tool to be implemented.

Consider 2 of the following different scenarios: When the tool created is completely new and there are no existing components needing interaction and when an existing tool is needed to be improved in order to satisfy the new requirements of the stakeholders. When integrating components into an existing tool, the structure or structures selected will need to display the interaction between the new and existing components.

One aspect emphasized repeatedly in the literature is the importance of having a software architecture implemented. It is essential to produce concise documentation of the application functionality and demonstrate the interactions between different components and the functions for which they are responsible. Supporting all of the requirements and providing precise documentation will promote the maintenance and manageability of the application after being deployed. The book by Bass, Clemens and Kazman represents best known practice for software architecture definition, and is heavily relied on in this project. [1]

Chapter 2: Methodology

The case study methodology that was used required collection of relevant data coming from the different stakeholders working on the current tool since they are directly involved in the utilization of the tool and are therefore dependent on its outcome. Moreover, due to their status as current users, their observations and experiences enable us to see its current state, constraints, and elements that can impact and affect the new structure. The main purpose of collecting this data is to shed light on what is causing the current tool to fail while delivering its expected functionality. This allowed us to determine how the current tool could be improved in order to generate the desired outcome.

When compiling all this data, we needed to identify and define where the main defects of the tool are. This would come from the data collected in order to better prioritize the aspects of the new design and course of the application. The survey data collected for this work is predominantly qualitative from the observations of the current tool processes as well as from the stakeholders' feedback about their expectations. Quantitative data was also gathered from the analysis of the architectural design to define its complexity, as it can impact the implementation and maintenance stages. In order to acquire the qualitative data needed, members of each stakeholder's group were interviewed to better understand their interactions with the current tool and their desires they have based on their experience. This served to promote the changes needed to improve the tool efficiencies. Supplementary qualitative data was also collected by observing the current tool processes and examining the aspects currently lacking and consequently affecting its current operating state. This is a critical component to be considered in the implementation process of added improvements.

Quantitative data originated from the evaluation of the new tool design, contingent upon the information needed to evaluate it against the requirements gathered. These quantitative data is of the "ratio type" as we will compare how the different components in the architecture structure compare to each other. It provided information on how the different components in the improvements impact the dependencies on each other and revealed how certain modifications of a given element will influence the overall implementation.

For example, if you have an interconnection between components, and you continue to upgrade one of them, all the other components dependent on their interaction with the upgraded component will require upgrading as well. By evaluating these interactions will allow the decision makers to include these features for future maintenance requirements of the implemented improvements.

Chapter 3: Description of the Toolkit

After collecting the experience data from the stakeholders involved in the current tool and observing the current tool processes we can then begin to evaluate and create an effective architecture that will help achieve the company's goals. We will now go through the process needed to create an efficient architecture. First, we begin by defining the environment where the improvements will be implemented.

3.1 Architecture Environment

When creating an architecture, we need to have knowledge of the environmental factors that affect the current operations. This will assist in identifying the best solution. In the following case, Company XYZ is a telecommunications specialty company providing equipment and services for a diverse set of industries. Their focus is on the hardware equipment in specific microwave radios. As the industry evolves, the company's focus is shifting to providing services to increase their revenue. Their goal is to attach, with every hardware transaction sale, their design and implementation services based on its high margin revenue. In order to achieve this, they need to focus on the following major areas of improvement: 1) The delivery of their services in a timely matter, 2) Accuracy and quality and 3) Lifecycle management of the product. To increase their value proposition to their clients, they also need to focus on improving their existing management and communication tools. They need to improve their current delivery times and remain compliant with both internal and external customers' requirements.

From the survey data gathered, around 65% of their design and implementation services missed their deadlines, and in addition to this, around 25% of the products delivered missed at least one of the agreed upon customer's requirements. Most of these deficiencies were due to miscommunication between the different project groups or failure to provide the most updated or accurate information. These resulted from an unstructured process in which major decisions were based on: random telephone calls that were not properly documented, random email communication, and deficient tracking of change requests and deliverables.

The company now has a Document Repository implemented with the purpose of having a single point location to access and archive all documentation needed. Supposedly each group can access the most accurate and up-to-date information. As a result of having an unstructured process, this Document Repository had not been used to its maximum potential, nor did it have visibility among the involved groups.

Another major issue brought to our attention by the Company management team was the lack of knowledge of a particular project's current status. In most cases, each group knew when their expected deadline was;

however, there was a lack of communication between the various groups regarding the deadline, thus making it difficult to schedule an overall outcome. The different groups are divided in two major subsets:

- The groups responsible for creating the required network design and documentation to be implemented. This involved the Transmission, Network and Configuration Engineer groups.
- The group in charge of the equipment's implementation and testing. This group involved the installers and technicians who configure and test the equipment before delivery.

Services to be rendered are shown below (See Figure 3.1).

Figure 3.1: Service Workflow

3.2 Envisioned Application: A Project Management (PM) Toolkit

The PM toolkit needs to operate within the existing infrastructure, complying with the constraints specified. Also, it needs to majorly satisfy the requirements of the stakeholders.

From the data collected, the main complaint from stakeholders was the lack of an online notification program between the different groups informing the team that a task has been completed, while prompting the next task and group to begin, thus continuing the workflow. One vital component of the new improvements is the implementation of an online notification component. This would notify the responsible party when an activity or task needs review, has an action item, or is completed. The PM toolkit would also allow the responsible party to update the current project status as well as assign and include reviewers or responsible contacts until the project is released and ready to be closed. Furthermore, this toolkit would provide visibility to upper management on the status of the current project's staged process and who is currently working on what. Another requirement of this toolkit would be the incorporation of a process to keep track and notify of any changes in the project requirements. With the current tool, some of the notifications were not being communicated to all the groups involved. It was not until the customer verified the implementation and configuration of the products ordered that some of the stakeholders involved in the project had access to information on where the project was at.

The PM toolkit would also include a straightforward module to notify the Project Manager and the responsible party involved when a deadline is fast approaching or is at risk of being missed, thus allowing the PM and the responsible parties to take the action needed. Another aspect of the tool is to keep track of all the information received by each group, ensuring that it is accurate and complete. These are the most important features the envisioned improvements will be composed of. Other features will be implemented as long as they do not interfere with the features mentioned above. As we continue, we will define the different stakeholders involved in operating the current tool, who will also be involved in the deployment and utilization of the new toolkit added.

3.3 Stakeholders Analysis

The stakeholders are one of the key components taken into consideration. They are responsible for the current tool operation and creation of the new additions to be deployed. They are accountable for rating the additions as a failure or a success based on their expectation of the functionality and efficiency of the improvements being built into the current tool. The stakeholders are divided into two major categories: the producers and the consumers. The producers will be in charge of producing and maintaining the toolkit and it improvements. The consumers will benefit from the advantages provided by the new added capabilities supporting the necessary data and documentation for the creation of the different projects' design and implementation. As we describe the different type of stakeholders involved, we take into consideration their expectations, vision of the expected tool, and the impact of not meeting these expectations.

Stakeholder (Consumers):

The consumers are the stakeholders who worked with the tool on a daily basis to accomplish their tasks. They are the ones benefiting of the tool's functionality to provide the expected outcome.

Transmission Engineer

Vision: As a document repository to collect and file all transmitted information

Developer Role: User and consumer

Organization: Collect, design and submit FCC license permit forms

Impact: Minimal impact, since the current tool has the functionality they need

Network Engineer

Vision: Help consolidate all the documentation required for the design of the products following customer requirements

Developer Role: User and consumer

Organization: Design the architecture of the product ordered by the customer

Impact: If their expectations are not met, they will resist using the improvements, causing it to fail and not provide the documentation required to follow the process.

Configuration Engineers

Vision: Improvements will provide a single, central point of storage for all the information required. It will also receive and provide notifications when a task is completed.

Developer Role: User, consumer, and sponsor

Organization: Design of detailed information about all the product hardware integration and configuration. They are the group responsible for providing the budget for improvements to the current tool.

Impact: If their expectations are not met, they will not utilize the improvements in the tool. It will create a disruption, causing the improved tool to fail. They also have the power to cancel the project if it doesn't meet their expectations.

Drafters

Vision: Allow easy access to the documentation required to complete their task. Receive and issue notifications when completed

Developer Role: User and consumer

Organization: Create the customer products final diagrams to be used for the installation of the customer requested products. Create the final required documentation being delivered to the customer.

Impact: If their expectations are not met, they cannot rely on the information provided on the improved tool, consequently failing to deliver accurate information to other groups and customers.

Installers

Vision: Have access to documentation

Developer Role: End-user

Organization: Implementing the final product.

Impact: They are the end users; if their expectations are not met; they can avoid and decline to use it, generating pressure within decision makers in order to return back to the current tool.

Technicians

Vision: Have access to the configuration documentation.

Developer Role: End-user.

Organization: Configure and test the implemented products.

Impact: Same as the installers, if their expectations are not met; they can avoid and decline to use it, generating pressure within decision makers in order to return back to the current tool.

Project Manager

Vision: Will be able to view and notify project changes and timelines in real time; get notification of completed tasks and project risks due to open tasks beyond their deadline.

Developer Role: User and consumer

Organization: Project management responsibilities

Impact: If the expectation is not met, they can exercise pressure on decision makers, giving visibility to the areas of failed improvements in order to provide timely and accurate project information.

Stakeholder (Producers of the tool):

The producer stakeholders are in charge of developing and testing the necessary components the tool requires. After the tool is deployed, the producers will be responsible for the maintenance of the tool.

Software Developer

Vision/Expectations/Constraints: Will develop the software functionality requirements for the toolkit to be implemented.

Contribution: They will develop the software required

Impact: If their expectations are not met, the improvements will be rejected.

System Administrator

Vision/Expectations/Constraints: Will be in charge of any changes needing to occur on the improved tool after it is deployed; also will provide the constraints on the equipment to be used.

Contribution: The one who maintains the tool and helps develop the toolkit application

Impact: If the future expectations are not met, the toolkit will stop working and if its constraints are not followed, the equipment may need to be upgraded.

Database Administrator

Vision/Expectations/Constraints: The application data will have to comply with the specific data requirements; also the best way to archive data storage must be defined.

Contribution: The one who provides the input in the database server design concerning the manipulation of data.

Impact: If the expectations or constraints are not met, the records in the tool will not be valid.

System Network Engineer

Vision/Expectations/Constraints: The improvements will contain various components interacting simultaneously. One example will have servers, desktops, and laptops operating concurrently.

Contribution: The one who designs the connections the tool will need in order to transmit data simultaneously without interruptions

Impact: If the expectations are not met, there will be no communication between components.

Tester

Vision/Expectations/Constraints: The improved tool will need to be reliable before it is deployed.

Contribution: Execute all the improved testing functions.

Impact: If they do not trust the improved tool's reliability, its delivery will be delayed, affecting the projects' workflow improvements.

3.4 Functional Requirements

The Functional Requirements for this architecture are divided in to three different sections. The first part will demonstrate the different functional specifications of the improvements. It will give a description of the different utilities required as well as determine the trigger for these functions, and the output shown in each case. Also, it will show who will be performing each task, the location of the tool function, and the conditions of the tool prior to and following implementation.

The second part will present the Scenario Specification in which each operation of the tool is defined. They will be presented as an operation taking place within the tool execution. In this step, we demonstrate the processes stakeholders are expecting the improvements to accomplish. The final step in the functional process will be to present the Essential Scenario.   Here the most important functionality of the improved tool is presented, demonstrating the intelligence behind the design and the reason for why it needs to be selected over the current tool.

For demonstration purposes only, the most important functions of the improved tool are shown in the next section.

3.4a Function Specifications

These are the essential functions that our toolkit must support:

A. Allow new Projects to be created in the toolkit.

Description: Authorized User will open a new project in the new toolkit, including services to be provided and their respective milestones.

Input: Authorized User will manually enter the data required into the toolkit.

Output: The screen will show the data entered into the tool and a report will be created.

Performers: The performer will be established as a "user," In this case, it will be the Network Engineer.

Resources: Existing PC or laptop the user employs and the Oracle Database to store the data.

Location: The operation will occur in the Oracle Database.

Preconditions: The same Project Number does not already exist in the Database.

Post conditions: The project is created and is available in the tool.

B. Allow uploading of documents into the tool.

Description: Users will need to be able to upload documents in various formats into the database to be accessed by all authorized users

Input: Document to be uploaded into the tool (i.e. Spreadsheet, Drawings, etc.)

Output: List of documents added to the project in the tool.

Performer: Drafters, PM, Network Engineer, Configuration Engineers, Transmission Engineer.

Resources: Database and the interface used by the user to upload file

Location: This will occur in the document management interface of the database

Precondition: Document doesn't exist on the Database.

Post condition: Document available from the tool

C. Allow creation of a Process Ticket

Description: Allow the creation of a process ticket to initiate the Project workflow and all tasks needed; also to change or update status of project documentation

Input: Project Number and finish date for each function

Output: Send ticket into queue, that will order the ticket order on the date expected for each function

Performer: PM will request the ticket and the toolkit will create and monitor the ticket

Resources: Ticket Tool

Location: Ticket Tool

Precondition: There is no ticket number created nor assigned to the project

Post condition: Ticket number assigned and added to the project

D. Create Status Report

Description: Users will be able to create a status report on the project.

Documentation: See if the project is on track to be completed on time.

Input: Request for ad-hoc project document status report

Output: User will receive a report with the documentation included in the project and information that still is missing.

Performer: All Authorized users

Resources: Document Module in Server

Location: Document Module

Precondition: No report on document status for a project

Post condition: Document Status Report is created

E. Allow the closure of tasks on a Process Ticket and notify users of completion

Description: Allows closing a task on a process ticket once it is completed.

Notify other users the task has been completed.

Input: Date of Task Completed and description of completion

Output: Task completion shown on Process Ticket

Performer: All authorized users

Resources: Ticket Tool

Location: Ticket Tool

Precondition: Task shown as open on process ticket

Post condition: Task shown as closed on process ticket and notification sent to users.

F. Provide notification of customer required function changes in project

Description: Allow the notification of changes requested by the customer during any step of the project

Output: Notification Notice to all users in a specific project

Performer: PM and Network Engineer

Resources: Notification Tool

Location: User interface and Notification Tool

Precondition: No new notification of changes exists

Post condition: Notification of changes sent to all users

3.4b Scenarios

After defining the different functions, scenarios are created to represent how these functions will be used within the toolkit. Each scenario will show a sequence of the different functions it uses. Some scenarios are shown as examples.

A. New workflow is created

- a. Sequence Function
  - i. Process ticket is created

ii. Process ticket is forward to a user

iii. Required Documents are uploaded

iv. Completed Task is closed in the Process Ticket and users are notified.

b. Environment Scenario:

Trigger: A new workflow and Process Ticket is open

Post Condition: A ticket is forward to the different users.

B. New Project is booked

a. Sequence Function.

i. User logs into the tool

ii. User is given access to the tool

ii. Project is created in the tool

iii. Create a Process Ticket

b. Scenario Environment

Trigger: New project is ready to be booked

Post condition: The scenario is completed when the project is created in the tool.

C. Customer requests changes on a project

a. Sequence Function

i. A Process Ticket is created

ii. Process Ticket is forward to a user

iii. Required Documents are uploaded

iv. Completed Task is closed in the Process Ticket and users are notified.

v. Notification of changes is created in the project

b. Scenario Environment

Trigger: Customer requests a change on the product ordered

Post Condition: Customer Change is conveyed to the users in the project

As shown above, each scenario will include functions that are included in the function specifications. If it is determined that a function is not included in the function specifications, one will need to be created.

3.4c Essential Scenario

Below we will show the most important scenario that demonstrates the importance of creating this toolkit to be added to the existing tool.

a. Sequence Function

i. A Process Ticket is created

ii. Process Ticket is forward to a user

iii. Required Documents are uploaded

iv. Completed Task is closed in the Process Ticket and users are notified.

v. Notification of changes is created in the project

b. Scenario Environment

Trigger: Customer requests a change on the product ordered

Post Condition: Customer Change is notified to the users in the project

The importance of this scenario is to show the missing notification function which doesn't exist in the current tool. Thus far, one of the greatest issues of the existing tool involves restriction of communication among all involved in the project.

3.5 Requirements and Constraints

3.5a Qualities and Constraints

In this section, we will present the quality requirements and constraints provided by the stakeholders of the project. We have to place special attention, focus and emphasis on these. These are priorities that must be established and incorporated. The different items will need to have a description of the request, in which this entails the category in which the request will fall. It can be a quality feature, a maintainability request, a budget concern or any other feature impacting the improvement's integration into the tool. Another important concern is to show which parts of the improvement functions will be affected and to identify which evaluation methods will be required in order to be successfully implemented into the toolkit. Below are the quality requirements and constraints from the stakeholders.

Tool Reliability – The new additions to the tool cannot cause it to be down for more than 6 consecutive hours.

Description: Any interruption in the tool or maintenance work cannot cause an interruption longer than 6 hours. It must allow access to users after an interruption occurs in a period of time under 6 hours. This requirement is based on the project needing to be smoothly worked on a tight schedule and completed in a timely manner.

Category: This quality is a reliability feature because it will affect the execution of the tool.

Stakeholder Source: This quality was requested by the Configuration Engineer Team who are providing budget to the project.

Scope: This quality impacts all functions in all areas of improvement.

Evaluation: Quality will be evaluated by a reliability report every month describing the down-time of the tool.

Budget Amount - The Configuration Engineering Group allocated a budget amount for this project of $20,000 to provide improvements to the current tool suite.

Description: Limited budget amount allocation cannot be exceeded in the creation of the improvements.

Category: This constraint is part of the budget.

Stakeholder Source: The requirement was included by the Configuration Team as they are responsible for funding the project. It is very important to stay within this budget as there are no additional funds available for it.

Scope: This constraint will touch every part of the project.

Evaluation: There are no more funds available to invest in this project.

Time to complete the project – The schedule for this project will take into consideration that full deployment needs to occur within a 3 month period.

Description: The project cannot take longer than 3 months to be fully deployed and functional. The new additions need to be available in such a short timeframe in order to strengthen and boost the Company's image amongst their customers.

Category: This constraint deals with the scheduling of the project.

Stakeholder Source: This constraint was requested by all the different stakeholders involved in the use of the tool.

Scope: This constraint will affect the whole project.

Evaluation: The Company will request to collect penalty fee compensation from the company in charge of deploying the improvements if they are not delivered in the agreed timeframe.

3.5b Deployment Environment

The Deployment Environment will consist of the entire infrastructure environment where the toolkit will be implemented. It will include all the hardware the tool will be running on or interacting with. Additionally, it will include the applications the toolkit needs in order to interact, along with the type of network infrastructure the tool will be running on. For this case in particular, here are some of the Deployment Environment Components required:

- Document Database is used to hold all the required documentation needed for each project. The tool will need to interact with the database for the documentation and communication flow between the different groups involved in the project. Specifications are provided by the System Administrator.
- An existing computer will be provided to install the new Ticket tool, as there is no available budget to deploy this tool in new equipment. Specifications of this equipment will be provided by the System Administrator. He will need to provide specific hard disk capacity, type of processor, and operating system. The Application Developer Team could

accommodate the tool with these constraints. This specification comes from the Configuration Engineer Team as there is no additional budget for new equipment.

- The tool will require a wired 100 Mbps connection into the existing Network to connect to all other existing infrastructure; a wireless connection cannot be accepted for security reasons. These specifications are given by the Network Administrator.
- The tool will need to interact with third party applications such as Microsoft Windows applications, AutoCAD®, Internet Explorer, Firefox, and Outlook among others, to let the users download, upload and transmit information through the different groups interacting in each project. This specification was given by the Software developers.

After defining the Deployment Environment requirements and affected components, the next steps to follow will be to incorporate these requirements and functions to create the components needed to implement the toolkit and obtain the essential metrics to analyze the architecture.

3.6 Stakeholders Qualities, Constraints and Priorities

From the data collected, the qualities and constraints for the improvements in the tool are determined and organized, based on their priority.

The priorities are defined based on the stakeholders' requests. These priorities are used to determine which need or quality will prevail in the event of a conflict. Table 3.1 displays the qualities, classified by category and the justification of the applied priority.

Table 3.1: Quality/Constraint Prioritization

Priority	Need/Quality	Classification	Priority Justification
1	The budget of the project is $20,000.	Cost	Is a request made by the Configuration Engineer Team who is financing the project and there is no additional capital to allocate for the project.
2	The schedule for this project will take into consideration that the full deployment should occur by in a 3 month period.	Schedule	It is required by all groups of Stakeholders, and is needed to help comply with customer project deadlines. This will uphold the Company Image.
3	The tool cannot be down more than 6 consecutive hours.	Reliability	It is requested by the Configuration Engineer Team to help deliver projects on time and not cause delays to projects.

In Table 3.1, we see a summary of the 3 top qualities prioritized for this project. From the priorities presented, we determined the budget allocation has the highest priority of all, as the project must be completed at a cost of $20,000. No additional source for funding will be available.

After we have collected and analyzed the data received from the stakeholders and their observations of the desired improvements, we can then begin to create a plan to guide the development of the improvement to the tool, beginning with the creation of the Business Blueprint.

3.7 Derivation of the Business Blueprint (BB)

In order to start creating the tool additions, we will begin by working the derivation of the Business Blueprint. First, we will show in Figure 3.2 the improved tool UML diagram. This represents the different components and their functions' interaction with the different components. The functions interactions are between the different components shown on Figure 3.2 and will be described in detail later on Table 3.3.

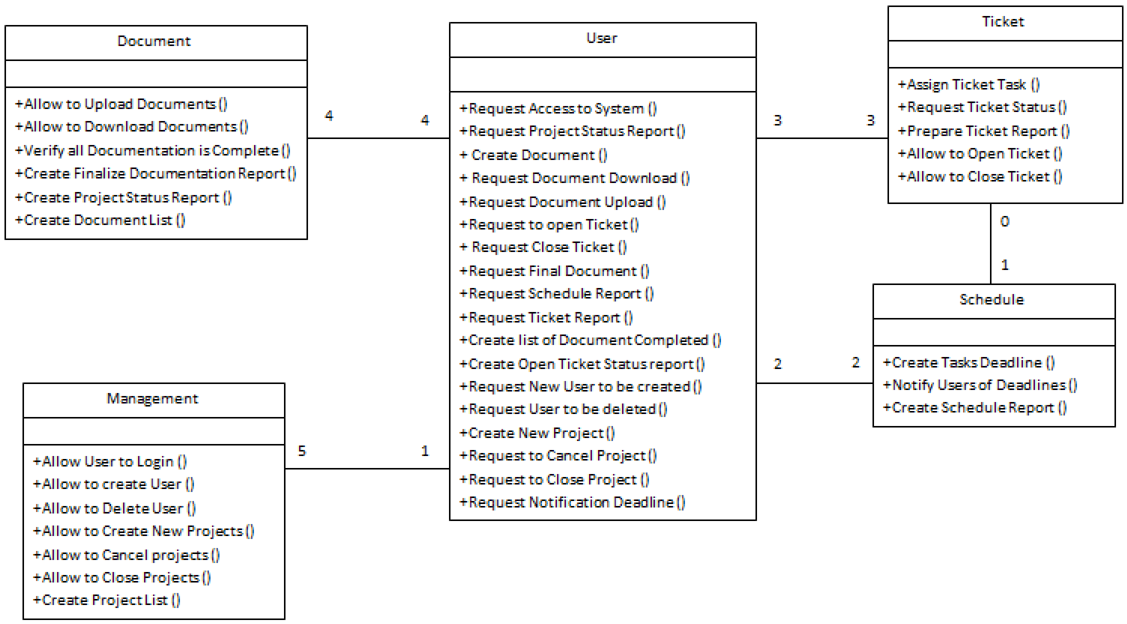


Figure 3.2: Tool UML

As we can observe from the UML diagram, we have a total of 5 components. Each component has been assigned to a set of functions or operations defined from the functional requirements. Table 3.2 presents each of the components and the functions they are assigned to perform. It also shows which data object the component will be in charge of providing. The five different components are the user, which in this case will be the entity utilizing and interfacing to complete a task, and the document, ticket, management and schedule modules to be included in the automation process to be implemented into the tool.

Table 3.2: Components Functions and Data Elements

Components	Functions and Data
User	Functions: i. Request Access to the tool (T1) ii. Request Project Status Report(T2) iii. Create Document (T3) iv. Request Document Download (T4) v. Request Document Upload(T5) vi. Request to open Ticket(T6) vii. Request Close Ticket(T7) viii. Request Final Document(T8) ix. Request Schedule Report(T9) x. Request Ticket Report (T10) xi. Create list of Document Completed(T11) xii. Create Open Ticket Status Report(T12) xiii. Request New User to be created(T13) xiv. Request User to be deleted(T14) xv. Create New Project(T15) xvi. Request to Cancel Project(T16) xvii. Request to Close Project(T17) xviii. Request Notification Deadline(T18) Data: i. Final Document Request Form(D1) ii. Completed Document Download Form(D2) iii. Completed Document Upload Form(D3) iv. Completed Ticket Request Form(D4) v. Completed Ticket Close Form(D5) vi. Ticket Status Form(D6)

Table 3.2 (Cont.)

Components	Functions and Data
User (Cont.)	vii. List of Documents Completed(D7) viii. User Record(D8) ix. Open Project Form(D9) x. Cancel Project Form(D10) xi. Close Project Form(D11) xii. Open ticket status Report(D12) xiii. Schedule Request Form(D13)
Document	Functions: i. Allow to Upload Documents(T19) ii. Allow to Download Documents(T20) iii. Verify all Documentation is Complete(T21) iv. Create Finalize Documentation Report(T22) v. Create Project Status Report(T23) vi. Create Document List(T24) Data: i. Document List(D14) ii. Status Report(D15) iii. Final Document File(D16) iv. Document Upload Form(D17) v. Document Download Form(D18)
Ticket	Functions: i. Assign Ticket Task(T25) ii. Request Ticket Status(T26) iii. Prepare Ticket Report(T27) iv. Allow to Open Ticket(T28) v. Allow to close Ticket(T29) Data: i. Ticket Status Report(D19) ii. Ticket File(D20) ii. Ticket Request Form(D21) iii. Ticket Close Form(D22)

Table 3.2 (Cont.)

Components	Functions and Data
Management	Functions: i. Allow User to Login(T30) ii. Allow to Create User(T31) iii. Allow to Delete User(T32) iv. Allow to Create New Projects(T33) v. Allow to Cancel projects(T34) vi. Allow to Close Projects(T35) vii. Create Project List(T36) Data: i. Authorized Users List(D23) ii. Project List (D24)
Schedule	Functions: i. Create Tasks Deadline(T37) ii. Notify Users of Deadlines(T38) iii. Create Schedule Report(T39) Data: i. Schedule Report(D25) ii. Notification Form(D26)

3.8 List of Inputs and Outputs between Components

Once the different modules have been defined; and the different functions have been assigned to each of the modules, we can then identify how the different modules interact with each other and which functions and data will be used to complete this interaction. In Table 3.3, the different interaction patterns between the components of the tool are represented.

Table 3.3: Inputs and Outputs Between Components

Components	Functions and Data
From: Document To: User	Request Document Download will require to allow to download and Document Download Form Request Document Upload will require allow to upload documents and Document Upload form Request Final Document will require Create Finalize Documentation Report and Final Document File Project Status Report will require Create Project Status Report and Status Report
From: Ticket To: User	Request to open Ticket will require Allow to Open Ticket and open ticket form Request Close Ticket will require Allow to Open Ticket close and ticket form Request Ticket Report will require Prepare Ticket Report and ticket Report
From: Management To: User	Request Access to the tool will require Allow User to Login and authorized user list
From: Schedule To: User	Request Schedule Report will require Create Ticket Project Timeline and Schedule Report Request Notification Deadline will Require Notify Users of Deadlines and Notification Form
From: User To: Document	Allow to Upload Documents will Require Request Document Upload and Completed Document Upload Form Request Final Document will Require Request Final Document and Final Document Request Form Allow to Download Documents will Require Request Document Download and Completed Document download Form Verify all Documentation is Complete will Require Create list of Document Completed and List of Documents Completed.

Table 3.3 (Cont.)

Components	Functions and Data
From: User To: Ticket	Request Ticket Status will require open ticket status report and Ticket Status Form Allow to Open Ticket will require Request to open Ticket and Completed Ticket Request Form Allow to Close Ticket will require Request to close Ticket and Completed Ticket Close Request Form
From: User To: Management	Allow User to Login will Require Request Access to the tool and user record Allow to create User will require Request New User to be created and user record Allow to Delete User will require Request User to be deleted and user record Allow to Cancel projects will require Request to Cancel Project and cancel project form Allow to Close Projects will require Request to close Project and close project form
From: Ticket To: Schedule	Create Tasks Deadline will require Assign Ticket Task and ticket file
From: User To: Schedule	Notify Users of Deadlines will require create open ticket report and Open ticket Status Report Create Schedule Report will require request schedule report and schedule request Form

As seen from the Inputs and Outputs list, all interactions can be considered as internal. All the functions are fulfilled internally without the need for accessing services outside the tool itself.

3.9 Derivation Plan and Rationale

Based on the quality priorities defined earlier, we can define the different heuristics to be applied to the Business Blueprint. This heuristics will help to satisfy the stakeholder's defined objectives and goals for the tool. Table 3.4 contains the most important heuristics defined based on the given priorities.

Table 3.4: BB Heuristics

1	Goal: Cost	Description
1.1	BB Heuristic: Group Based on Implementation Reality (Use existing technologies that can provide the functionality)	• Why: Instead of creating new toolkit, the components and respective functions can be done by existing technologies in the market. • Priority Justification: Will reduce the personnel needed to develop the application also reduce the time to implement.
1.2	BB Heuristic: Reduce Class Complexity - Size (Reduce the numbers of functions in the components)	• Why: It will reduce the complexity of the design in the component, which could be done by more simple components. • Priority Justification: It will reduce the time and cost of development.
2	**Goal: Reliability**	**Description**
2.1	BB Heuristic: Specify Overlapping Capabilities (performer hierarchy)	• Why: It will to reduce coupling and eliminate duplicate definitions. Will create parent components that will produce functions and data of two different components. This will reduce risk of failure. • Priority Justification: Requested by the Configuration Engineer Team that we have a reliable tool.
3	**Goal: Flexibility**	**Description**
3.1	BB Heuristic: Group Based on Task similarity (Group functions with similar data and parent function)	• Why: In case the tool needs to be modified, to be used as part of other tools needing update, simpler to change one component or move one component to a larger quantity of components. • Priority Justification: The stakeholders request that the tool can be adapted easily to modifications or other tools.

We have defined three goals needing to be achieved in order to create an effective application fulfilling the stakeholders' needs.

After defining all the elements needed to develop the business blueprint, we begin the analysis and the creation of the business blueprint deployment.

3.10 Evaluate Business Blueprint Structure

After the different components of the Blueprint have been defined, we can now begin to evaluate the future structure. As shown below in Figure 3.3, a structure of the map can be followed when creating the improved tool. Figure 3.3 demonstrates the Business Blue Print for the improved product.

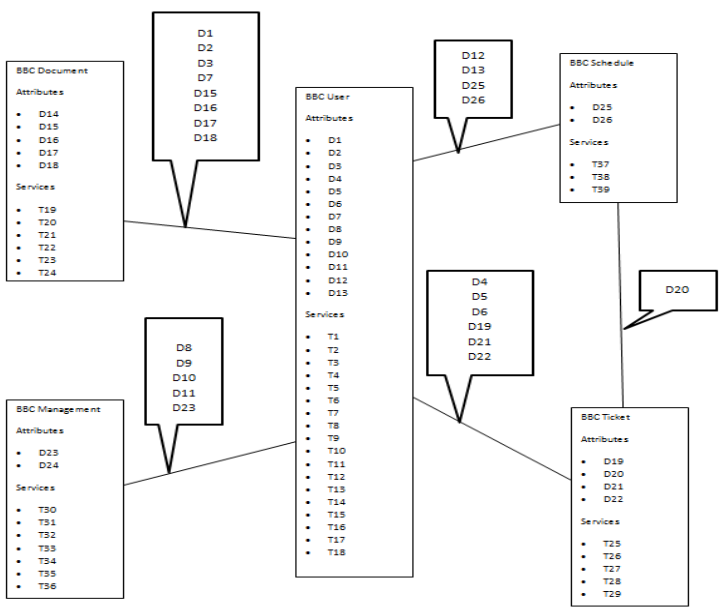


Figure 3.3: Business Blueprint Structure

As seen in Figure 3.3, all the different components of the Business Blueprint contain the attributes and services it provides. These services and attributes were presented in Table 3.2 of section 3.7 of the report. Notice the attributes are used to relate the different components to each other. For the analysis of this Business Blueprint we apply the following metrics.

First, we will begin by calculating the Inputs and Outputs between the different components of the Blueprints to see the component's interaction complexity.

Number of Inputs/Outputs between BB components (NIOBC):

NIOBC (User) = 23

NIOBC (Document) = 8

NIOBC (Ticket) = 7

NIOBC (Management) = 5

NIOBC (Schedule) = 5

From this analysis, we determined the interaction is distributed similarly in all components, except for the User Component that has nearly 3 times the amount of interactions as the other components (meaning it will be the most complex component in the tool).

After completing all the components' interactions, we will determine the dependency of each of the components to each other in the tool. This will determine how the changes will affect the improvements created.

Number of dependencies between BB Components (NDBC):

NDBC (User) = 4

NDBC (Document) = 1

NDBC (Ticket) = 2

NDBC (Management) = 1

NDBC (Schedule) = 2

As seen with this metric, the User component is at least 2 times greater than the other components, implicating that any change made in this component will have the greatest effect on the tool.

Another metric used to analyze the Blueprint Structure will be the degree of cohesion; we are measuring how each of the components will depend on each other in order to complete their desired function. It is a percentage measurement of all the Inputs and Output that occur inside the components over the sum of all the Inputs and Outputs the components pose.

Degree of Cohesion (DC):

DC (User) = 0

DC (Document) = 1/6 = 17.67%

DC (Ticket) = 0

DC (Management) = 1/7 = 14.29%

DC (Schedule) =0

As seen from the above results, the biggest degree of cohesion is in the Document component with almost 18% of all the Inputs and Outputs occurring within itself not depending on any other components. As higher cohesion is achieved, the component shows less dependency from other components when completing their operation. A top performing cohesive application will be easier to maintain as changes are made.

The next t measurement calculation used to help define the complexity of a component is the numbers functions each component is performing.

Number of Functions in a component (NFC):

NFC (User) = 18

NFC (Document) = 6

NFC (Ticket) = 5

NFC (Management) = 7

NFC (Schedule) = 3

From the results shown above regarding the number of functions in a component, we can identify the user component as a main contributor, as it carries the majority of functions. It possesses greater than 2 and a half times the number of operations as the other components. Comparing these results with the previous outcome obtained from the Number of Inputs and Outputs from components, we see can see the User component is one the most complex in the tool.

Now, calculating the number of data elements being incorporated on each of the components will provide another measurement to quantify the complexity of each of the components in the tool.

Number of Data Elements in a component (NDEC):

NDEC (User) = 13

NDEC (Document) = 5

NDEC (Ticket) = 4

NDEC (Management) = 2

NDEC (Schedule) = 2

Following the trend of the previous complexity measurements, we can see from the results the User component has a higher complexity in comparison to the other components, showing it has more than 2 times data elements than the other components.

Combining the values of the number of Inputs and Outputs, the functions and data element contained in a component will give the overall complexity value of the different components,

Component Complexity (CC):

CC (User) = 54

CC (Document) = 19

CC (Ticket) = 16

CC (Management) = 14

CC (Schedule) = 10

As expected from the previous measurement, the User component is the more complex component. Having a higher complexity will translate into this component requiring more effort to build than the other components. This will also mean it will incur in additional cost in order to maintain.

The last measurement to show is the number of components inside the Business Blueprint. It is the sum of all the components included within the structure.

The end result from the number of Components in a structure will vary as a heuristic is applied to achieve the different goals defined.

Number of Components in the Blueprint (NCB):

NCB = 5

After performing the structure analysis, we see the user component is the only one posing a higher complexity compared to the other components. One way to solve this is to apply one of the different heuristics defined in Table 3.4. In this specific case, we can use the 1.2 heuristic to achieve the cost goal. This can be accomplished by reducing the number of functions of a component. It will entail the creation of an additional component to share some of the functions of the user components. The addition of a new component will affect the different metrics calculations. For example, the number of components will increase to 6, but the complexity value of the user component may decrease, depending on how many functions the new component will be responsible for.

3.11 Blueprint Deployment

After analyzing the Business Blueprint structure, we are ready to deploy the Business Blueprint to the improved tool. Below we demonstrate the solution to solve the current issues. We then validate how to satisfy their current tool needs using the different components and functions defined.

3.11a Satisfaction of Domain Functions by Solutions

It is important to confirm that all functions requested are being satisfied by the different components of the tool. This way, the improved tool can perform effectively and comply with the stakeholders' requirements. Table 3.5 shows the different components on the solution Blueprint and how each of them suits the requested functions. It also shows how all the components combined will fulfill all the functions required by stakeholders.

Table 3.5: Solution Components

SB Solution component	BB Functions Satisfied
User	Functions: i. Request Access to the tool(T1) ii. Request Project Status Report(T2) iii. Create Document(T3) iv. Request Document Download(T4) v. Request Document Upload(T5) vi. Request to Open Ticket(T6) vii. Request Close Ticket(T7) viii. Request Final Document(T8) ix. Request Schedule Report(T9) x. Request Ticket Report (T10) xi. Create List of Document Completed(T11)

Table 3.5 (Cont.)

SB Solution Component	BB Functions Satisfied
User (Cont.)	xii. Create Open Ticket Status Report(T12) xiii. Request New User to be Created(T13) xiv. Request User to be Deleted(T14) xv. Create New Project(T15) xvi. Request to Cancel Project(T16) xvii. Request to Close Project(T17) xviii. Request Notification Deadline (T18) Data: i. Final Document Request Form ii. Completed Document Download Form iii. Completed Document Upload Form iv. Completed Ticket Request Form v. Completed Ticket Close Form vi. Ticket Status Form vii. List of Documents Completed viii. User Record ix. Open Project Form x. Cancel Project Form xi. Close Project Form xii. Open ticket Status report xiii. Schedule Request Form
Document Database Server	Functions: i. Allow to Upload Documents(T19) ii. Allow to Download Documents(T20) iii. Verify all Documentation is Complete(T21) iv. Create Finalize Documentation Report(T22) v. Create Project Status Report(T23) vi. Create Document List(T24) Data: i. Document List ii. Status Report iii. Final Document File iv. Document Upload Form v. Document Download Form

Table 3.5 (Cont.)

SB Solution Component	BB Functions Satisfied
Ticket Module	Functions: i. Assign Ticket Task(T25) ii. Request Ticket Status(T26) iii. Prepare Ticket Report(T27) iv. Allow to Open Ticket(T28) v. Allow to Close Ticket(T29) Data: i. Ticket Status Report ii. Ticket File ii. Ticket Request Form iii. Ticket Close Form
Management Module	Functions: i. Allow User to Login(T30) ii. Allow to Create User(T31) iii. Allow to Delete User(T32) iv. Allow to Create New Projects(T33) v. Allow to Cancel Projects(T34) vi. Allow to Close Projects(T35) vii. Create Project List(T36) Data: i. Authorized Users list ii. Project List
Schedule Module	Functions: i. Create Tasks Deadline(T37) ii. Notify Users of Deadlines(T38) iii. Create Schedule Report(T39) Data: i. Schedule Report ii. Notification Form

We can see now the required functions are fulfilled by all the components defined. After confirming all the functions are satisfied, we can continue to define the physical location of each component.

3.11b Allocation of Solutions to Deployment Components

When Deploying the Business Blueprint, we need to identify the physical location of each of the proposed components. As seen in Table 3.6, each of the Deployment Blueprint Components is allocated to a defined location. In this case, all the physical locations are currently in existence as there is no budget allocated to introduce new equipment.

Table 3.6: Components Allocation

Deployment Blueprint Component	Solution Blueprint Components Allocated to Deployment Blueprint Component
User	User
Document Module	Database Server
Ticket Module	Server
Management Module	Server
Schedule Module	Server

3.12 Rationale for Solution Deployment

Comparing the stakeholders' qualities against the deployed blueprint will demonstrate how the improvements added will suit the expectations of the different users involved in the current tool. Below we present the most important qualities defined by the stakeholders to demonstrate how the improved tool is in compliance with them.

Satisfaction of stakeholder qualities and constraints:

Cost: The improvement's cost has been kept within budget. This has been achieved by using all existing assets and adapting the tool to run on this equipment. .

Reliability: The documentation on the tool will be more accessible and more accurate, as the transmission of document communication between the groups involved in the project will improve significantly.

After showing compliance with two of the most important qualities the stakeholders are currently seeking, we can continue to define the design to be used that will deploy the improved tool inspired by the architecture.

Design inspired by:

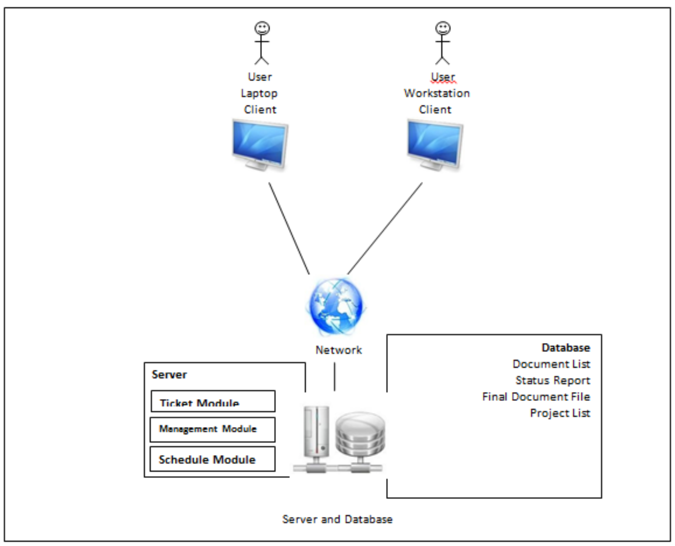
Client-Server architecture: The sharing of documents and reports, which will be handled by one server with various users' interfaces accessing it.

Database-Centric architecture: All documents will be centralized in one location for long periods of time and retrieved as requested by the different users.

Event – Driven architecture: All processes will need a status change or event occurring, enabling the process to begin and be completed. There should be a request initiated by the user or by another component in order to begin a process.

Cloud Computing Architecture: The tool can be accessed and run on the computers connected at the same time.

Figure 3.4 displays an image of the architecture to be used as a solution. Also, Figure 3.4 shows all existing equipment being used for each of the different Components.



Equipment	Minimum Required
Client computer	OS: Windows 7 RAM: 1.0 GB Hard Disk: 20 GB of free space Processor: Dual core 2.0 GHz or better Other: jre v6.0 or better
Network	BW: 100 Mbps
Server	OS: Windows Server 2012 RAM: 4 GB Hard DISK: 120 GB of free space Network: 2 ports 100 Mbps Other: SSL Certified
Ticket Module	Developed In-House
Management Module	Developed In-House
Schedule Module	Developed In-House

Figure 3.4: Solution Architecture

The solution presented fulfills the need to have all project related documentation and interactions centralized as well as create notifications to improve the communication channels between the different groups and stakeholders involved, all while achieving successful process flow, ensuring timely delivery, providing accurate content, and avoiding unnecessary risks and penalty fees.

3.13 Architecture Tradeoff Analysis Method (ATAM)

ATAM is a process in which the sensitive and trade-off points are analyzed. This will enable the incorporation of any necessary modification to the architecture in the early stages of the software lifecycle. It is useful in order to show which area will need to be prioritized over the others and where changes can be done to create a more efficient architecture. The priorities are defined in terms of High (H), Medium (M) or Low (L). This needs to happen before the improvements are implemented into the tool, which will translate as a cost saving in the project. Part of the ATAM analysis is shown below.

- Utility
    - Cost
        - <u>Ensure not to go over budget</u>
            - Objective: Use of the shelf or third party vendor's software. (H,H)
                - Metric: Keep track of the cost of implementation (H).
                - Metric: Keep track of the cost of using third party applications, in case is required. (H).
                - Metric: Calculate the Man/Hour cost of implementing the tool's improvements. (H).
        - <u>Maintain the project on schedule</u>

            - Objective: Maintain the project on the schedule less than 3 months. (H,L)
                - Metric: Calculate the time of developing and implementing the tool's improvements, counting the time to install the off the shelves software. (H).

- Reliability
    - <u>Keep Application Secure</u>
        - Objective: Use components with modules certified to be secure. (M,M)
            - Metric: Keep track of all transactions done by each of the modules and identify any security breach. (M).

    - <u>Keep Application without outages</u>
        - Objective: Use reliable modules on the tool to avoid a major outage. (H,M)
            - Metric: Count the outages by components and the duration of each. (M).

- Flexibility
    - <u>Keep Application updated</u>
        - Objective: Use a central database with all information so it can be easily adapted to an updated tool. (M, M).
            - Metric: Measure the accuracy of all records on the tool. (L).

    - <u>Enable Application Modifications</u>
        - Objective: Use easily maneuvered components so they can be modified to use on other tools(L,L)
            - Metric: count the quantity of modules that can be modified. (L).

    - <u>Modify Application</u>
        - Objective: Use components in the tool that can be used on other tools. (L,L)
            - Metric: Count the numbers of components that can be reused in similar tools. (L).

After this tradeoff analysis, it is easier to identify architecture changes that can be made without affecting the expectations of the stakeholders. With this analysis, we defined which areas are prioritized based on their importance to the stakeholders.

In this section we defined the creation and analysis process for the improved tool architecture. In the next section, we elaborate on the importance of incorporating the collected data into the improvements and how we collected it.

Chapter 4: Collection and Application of Data

Data and observations from the stakeholders and various groups currently involved in all projects were collected. These included the identification of process flows and tasks deliverables needed in addition to the analysis of tool interactions among them. This was vital in creating an improved architecture application.

The raw data received from the stakeholders was collected using a survey based on their functions and expectations of improvements to be made to the tool. We took of sample of 2 members from each group due to the small size of the company and each group possessing only an average of 6 members per group. Sample questions that were asked included “What is your group role within the organization? How do you get notified on projects you have deliverables assigned to provide? How do you think this process can be improved?” These were in addition to other questions related to their expectations of the improvements to be done. The full survey document can be found in Appendix A.

Other data were collected from the observations based on the process flow, were performed following the process from beginning to end. These observations were conducted on 2 different projects which involved the interaction of all the stakeholders. This data was analyzed and compared to the data collected from the stakeholders to confirm our observations.

After all the data were gathered, they were applied to the base design of the new architecture. Based on the analysis, we were able to identify which qualities were important to the stakeholders. This way, we were able to prioritize the components based on the stakeholders' valued deliverables. It also gave us the role and level of responsibility of each of the different groups involved in the operation of the current tool. This information provided visibility into the expected goals satisfaction upon completion of the tool improvement. It offered the required information in order to create the deployment environment, to define the stakeholders' roles and to determine which functions and data were required in the tool to operate efficiently. In summary, the data collected were the building blocks needed to create the architecture that would ultimately help the company achieve their business goals of providing accurate designs and implementation to customers in a timely and accurate manner.

After the Architecture Structure was created, the data collected from metrics was applied to identify the complexity of the different components and their dependency on each other. With the results of this complexity and dependency, the structure can be further improved to create a more effective architecture.

Chapter 5: Lessons learned and data analyzed

5.1 Lessons learned

According to the results of our research, we have learned some important lessons from the creation of an architecture process. Below is the list of lessons learned throughout our case study process.

1. There are different types of structures needed to present an architecture structure (Component to Component, allocation, module structure, etc.): The selection of the structure or structures to be used are based on the quality attributes derived by stakeholders. The type of structure selected leverages these attributes into the new or improved tool.

2. There is no good or bad architecture: The architecture is fitted into the solution given to achieve the desired goal. For example, two different architects can provide different architectures to solve the same issue, and both can solve the problem successfully, even if their approach and process were different.

3. The importance of creating an architecture process before deploying a solution: Architecture gives a vision on how the changes will be implemented; it helps to create an estimated timeline to deploy a system and determine how many resources are needed in order to deploy it. Also, any changes needed to be done at this stage can be considered as savings due to the fact that none of the applications had been implemented yet.

4. Collecting and accurately analyzing a set of data: If the data are not analyzed correctly or have not been collected from the proper sources, it will result in a failed architecture, impacting the overall operation of the company including production, resources, utilization, and financial loss.

5. You can have all the components needed, but this will not necessarily translate into achieving a successful product: As seen from the data collected, all the components needed for the design and implementation of the improvements were available, but lack of participation, trust and miscommunication from all the responsible parties involved did result in, not only missed deadlines, but also not delivering a quality product to the end customer.

6. Implementing data to create an architecture structure solution: Through the research, we have seen how implementing the collected data will help to create the architecture structure to solve a tool problem.

7. Analysis of an architecture structure: With the use of different metrics, we can define the complexity and dependency of the different components.

5.2 Data analysis

To design an architecture for the tool, we first needed to collect the necessary data from the current tool and all users/stakeholders. This allowed creating a solution that truly fits their needs. After the data have been collected, it needs to be analyzed to ensure its synergy between the “wants,” “needs,” and implementation. In the case presented for Company XYZ, as it is been mentioned above, the data were collected using a questionnaire distributed among the stakeholders, along with interviews, to receive feedback on their observations of the current processes.

Considering the data obtained from the observations regarding the current tool’s failures. During the observation, we noticed how 2 different projects run through the current tool simultaneously. In our observation, we noticed that on 2 different occasions, a request to change a design requirement was done by the customer because the current tool didn’t have a notification feature, the Engineers were not prompted to review the change, and the Project Manager failed to send the message of the change requested to the responsible groups. In this particular case, the Engineers needing to make the change were not notified until the project was ready to be released, causing unnecessary delay and a missed deadline. This resulted in the Engineers having to return to the configuration stage in order to include the previous customer’s requested changes. Another incident we observed was the lack of communication between the Network and Configuration Engineers groups. We observed another missed deadline resulting from the Network Engineer lack of notification and communication with the Configuration Engineer regarding the High Level design needed to create the specifications for installation and mandatory documentation. Not only did this create a poor customer experience, but it also impacted his productivity, resources, and project timelines. Most of all, it compromised the quality and accuracy of the diagrams for installation needed to be included in these important documentation.

The survey or questionnaire distributed among the stakeholders included five questions. The results from the survey reaffirmed the need to improve the current tool and served as the foundation for the development of the new architecture. Based on the first question, what is your role in the project process? we received the information of who the key players were along with their respective roles in the process of developing a project. What was interesting from this perspective is that each group who participated in the survey had a different idea, perception or approach on the correct flow and tasks that a project should follow. Below you will see a summary of each group's response:

Configuration Engineers: Responsible for the design of detail information about all the customer product hardware integration and configuration

Drafters: Create the customer product final diagrams to be used to install the desire product; create the final documentation being delivered to the end-customer.

Installers: Installation of the customer product

Technicians: Configuring and testing the customer product

Program Manager: Manage the resources and budget of a project; customer's point of contact

Transmission Engineer: Collect, design and submit FCC license permit forms

Network Engineer: Create the high level design of the customer product

The answer received from question #2 of the survey indicated that all involved parties working on a project were in agreement that the actual process was not effective.

Question #3, what do you think is the major cause for the process to fail if any? revealed serious concerns from all stakeholders regarding the inefficiency of the current tool. We observed that 70% of the answers received identified the current tool and its processes, or lack of it, as the main reason for not delivering projects accurately or on time. The other 30% expressed the current tool is missing a notification tool which will prompt the responsible contact or task owner of the next step or action that needs to be completed.

Question #4 what do you think must be improve in the process? and Question #5, if a new Process is created what are your expectations? consisted of collecting feedback on how the stakeholders suggested the current tool could be improved and their expectations from it as the changes were implemented.

The feedback showed approximately 80% of the participants think the current tool could be improved if a notification tool was included. They pinpointed lack of communication between the groups as a major obstacle. They were all in agreement regarding the notification tool and process as an important milestone to improve the internal and external customer experience, plus ensuring a timely and accurate deployment of all tasks involved.

The last part of the data collection stage prior to starting the architecture design process was conducting the Stakeholders' interviews. From the interviews, we gathered the necessary data, coming mostly from the environment were the improved tool will be deployed.

One pivotal piece of information received during the interview stage was that the budget available to complete the tool's improvement could not exceed $20,000 (no further funding is available.) This information was provided by the Configuration Engineer. The Network Administrator added to this budget limitation on the premise that "no new equipment could be bought for the improved tool. " In addition to these constraints, one of the high priority demands was the reliability requirement on tool failures. They were very clear in requesting that the tool cannot be out of service for more than 6 consecutive hours, as it can negatively impact the project timetables and deliverables schedule.

With the collection of the data mentioned above, we began the process of creating an architecture solution.

When the architecture structure was created, we proceeded to analyze the different components to demonstrate the complexities and dependences between each of the components within the structure. These results are presented in section 3.10. A summary of the results is presented in Table 5.1.

Table 5.1: Summary of Architecture Metrics Results

Metric	Component	Total
Number of Inputs/Outputs Between BB Components	User	23
Number of dependencies Between BB Components	User	4
Degree of Cohesion	Document	17.67%
Number of Functions in a Component	User	18
Number of Data Elements in a Component	User	13
Component Complexity	User	54
Number of Components in the Blueprint	Blueprint	5

As seen on Table 5.1, we are presenting the highest value for each of the metrics. Based on the outcome, we can see the user is the more complex and more dependent component in the architecture structure. On the other hand, the document component is the more cohesive component that is able to conduct more operations on its own.

As data was collected, requirements from the different groups of Stakeholders were gathered to help the design process. Table 5.2 show a summary of the most important requirements, it source and in which part of the design it was taken into consideration.

Table 5.2: Summary of Requirements by Originator

Requirement	Solicited By	Collection Method Used	Design Implementation
Create a Process for effective Workflow and Notification	Configuration Engineer	Information was collected from information provided through interview method	This requirement is implemented as part of the Ticket Toolkit
Maintain Accurate and up to date documentation	Various Groups: Drafting Group Configuration Engineer Technicians Project Manager Network Design Engineer	The information was gathered through the survey questions and through the interview process	This requirement is implemented in the Document Module
Maintain cost of the implementation under $20,000.00	Configuration Engineer	This requirement was collected form the information provided by the customer that is the only fund available for the project	At the moment this requirement is being on track by the use of existing equipment and resources to keep low expenses
Provide an automatic status report of a project	Project Manager Configuration Engineer	Information was collected from information provided through interview method	This requirement is implemented as part of the Ticket Toolkit
Get Notification when a task is completed	Project Manager Configuration Engineer	Information was collected from the answer on the questions in the survey	This requirement is implemented as part of the Ticket Toolkit

Table 5.2 (Cont.)

Requirement	Solicited By	Collection Method Used	Design Implementation
Keep most of the current function of the tool	Configuration Engineer Network Design Engineer	Information was collected through the interview method with various group of stakeholders	This is taking into account in all the new modules added, as they will be improvements or additions to the existing tool
Create Finalize documentation Report	Drafting Group	Information was collected from information provided through interview method, as they need to know the information is completed, so the final documentation for the customer could be created	It is implemented in the Document Module
Create a Notification Ticket and assign tasks to each of the members involve in a project	Project Manager Configuration Engineer	The requirement was gathered during the interview process	This requirement is implemented as part of the Ticket Toolkit

As seen in table 5.2 above, the requirements were traced back to each originator and which part of the design was driven by it, to create the different modules to improve the existing tool.

Chapter 6: Conclusion and Next Steps

As presented in this report, we have used the data collected from various stakeholders regarding the current tool to be fixed in order to identify the process defects which are impacting several areas of the business. Using various data collection tools such as surveys, questionnaires, interviews and observations, we determined these methods to be essential in the analysis of the data in order for the new architecture structure to be incorporated into the improved toolkit.

With the creation of new modules and modifications performed, the stakeholders' requirements are better fulfilled and the new process is aligned for success. The improved tool provides the needed notification toolkit, enabling the communication channels workflow between groups to ensure a productive, timely and accurate project completion and delivery.

One of the main takeaways from this report is that there is no "good/bad," approach to building a successful architecture. The ultimate goal is to create an architecture fulfilling the needs of the stakeholders' requirements and meeting their expectations and desired outcomes.

For future instances, the architecture created can be deployed and tested to confirm the reliability and continuity of the process. From the design standpoint, we can try and implement a component of the BB Heuristic mentioned to minimize the complexity of the tool and observe if it can provide a cost saving without generating new issues. For example, in order to fix a new issue, this might entail creating another module and bringing additional dependencies on the improved tool.

In conclusion, we have learned that having an architecture established prior to starting a project or in the early lifecycle stages will positively influence its outcome, produce timely deliverables, and improve the financial impact associated with it. It is crucial to consider all aspects surrounding an application design. This includes determining all requirements of the stakeholders, acquiring internal and external customer feedback, and investigating any particular feature that will guarantee a reliable tool and deployment process.

Appendix A

Name: ______________________ Date: ____________________

Position: __________________ Dept.:____________________

Answer the following questions of how you feel about the system process for notification and conveying of the documentation between groups working on a project.

1. What is your role in the project process?
   ____
   ____
   ____

2. Do you think the existing process is working effectively?

   Yes No

3. What do you think is the major cause for the process to fail, if any?
   ____
   ____
   ____

4. What do you think must be improve in the process?
   ____
   ____
   ____

5. If a new Process is created what are your expectations?
   ____
   ____
   ____

References

[1] Bass, Len; Clements, Paul; Kazman, Rick (2012-09-25). Software

Architecture in Practice (Third Edition), Pearson Education. Kindle Edition.

[2] Database Centric, [online], Available: http://en.wikipedia.org/wiki/Database-centric_architecture

[3] Deriving a solution which satisfies software requirements,[online], Available: http://www.csm.ornl.gov/~sheldon/cs330/PDF/SLIDES/c12.pdf

[4] Software architecture, [online], Available: http://azhar-paperpresentation.blogspot.com/2010/04/software-architecture.html

[5] P. Lago, P. Avgeriou, R. Hiliard: Software Architecture: Framing Stakeholders' Concerns, IEEE, IEEE Computer Society, vol. 27, no. 6, pp. 20-24, 2012