Software project management

TAMEEM ABDULBASET ABDULWAHID ABDO HEZAM, *master degree*

Abstract—The ability to deliver projects on schedule, budget, and alignment with business objectives is critical in determining the limit on highly competitive global business conditions. Project managers play a complex role, including organizational skills, analytical thinking, and interpersonal skills. Also, Software is a non-physical product. Software development is a new business distribution, and has little experience in creating software products. Most software products are designed to fit the needs of the customers. Most importantly, basic technologies change and re-emerge more and more quickly so that the knowledge of one product can be applied to another. Such types of business and environmental issues increase the risk of software development, so it is important to properly manage software projects.
The organization should deliver the highest quality product, keep costs below the customer budget and deliver the project as planned. Therefore, respectively, software project management is required to include user needs and budget and time constraints.

CONTENTS

• *TAMEEM ABDULBASET ABDULWAHID ABDO HEZAM is with computer science and engneering school, nanjing university of science and technology , nanjing, 210094, jiangsu, China, Tel.: +8613270717933, Email: tamem20142016@gmail.com*

1 NOMENCLATURE

2 INTRODUCTION

Until the 1980s software was used only by professional people. The whole world was curious about computers, software systems but even so, the software had not yet entered people's lives. Today as we see, the software has become an integral part of human life. Almost everything we can look around is owned, controlled by software be it our internet, smartphone, mobile network, electronics, television, travel system, food chains, retail and learning centers, health care centers, weather, agriculture, and so on. . Almost every aspect of our lives is associated with one or more other software. And going forward, it will be a skill and a very important part of our lives.

Software project management although this, in some ways, such changes have always resulted in 'good work ethic' performing the so-called 'software project management. Building software or creating software is a process and a well-defined process for making software is called software project management. in this report we will

3 PROJACT MANGEMENT

3.1 what is the project ?

AA general definition of project is: "It is an temporary endeavor with set of well-defined activities that leads achievement of a specific goal(s)" [7] . A Project has following characteristics:

- Project has specific goal(s)
- It has a definite start date and end date
- It is not group of routine tasks or daily activities rather involves planned activities Unlike routine activities.
- project comes to end when its goal(s) is achieved.
- Every project requires enough resources in terms of time, skilled workforce, budget, material and other support

3.2 How do you define a project?

Before we get into project management, we need to define what a "project" really is.

Project Management Institute defines "project" [2] as "a concerted effort to create a unique product, service or outcome."

There are a few important things you can see in this description:

The term "temporary" means that projects must have a defined beginning and end. This means that every project must include a timeline, scope, and resources. The fact that it is temporary with a beginning and an end also mean that it is not part of the ongoing process. This brings us to the second point ...

The aim of the project should be to "create a unique product, service, or outcome." This means that a project will be built to achieve a common goal outside the day-to-day business environment. This means that the project team can include people who are not used to working together, and who need resources outside of daily work.

However, domain.com describes the project in more detail: "big or big work, especially one that involves a lot of money, labor, and equipment."

No matter, the whole project should have the following:

- Purpose: What are you trying to achieve?
- Time: When are you trying to achieve it?.
- Budget: How much will it cost to achieve?
- Stakeholders: Who are the main players interested in this project?.
- Project Manager: Who will ensure that everything that needs to be done is eliminated?

The project is not a standard one. Daily work or care is not considered a project because it has no definite beginning and end.

3.3 What is project management?

Project management is the practice of applying knowledge, skills, tools and techniques to complete a project according to specific needs. It depends on the problem, the problem-solving process, and the process until the problem is solved. That may sound simple, but there is more to it than meets the eye [3] .

The roots of project management can be traced as far back as the construction of the Pyramids in Giza and China's Great Wall. However, the modern development of project management began in the 19th century when railway companies purchased tons of equipment and employed thousands of people to operate the railways across continents.

In the early 20th century, Frederick Taylor applied the principles of day-to-day project management, developing strategies to work smarter and improve efficiency, rather than requiring workers to work longer hours. Henry Gantt, a collaborator with Taylor, took those ideas and applied the bars and charts to the graph when certain tasks or series of tasks were completed, creating a new way to visualize project management.

During World War II, military and industrial leaders used more elaborate management strategies, which eventually led to more intensified processes as a more sensitive one. [9]

These practices grew in popularity throughout the industry, and in 1965 and 1969, the International Project Management Association and the Project Management Institute were established, respectively. In 2001, Agile project management approaches were integrated into the implementation of the Agile Manifesto.

The project management field continues to evolve as a growing competitive environment, the need for rapid change, and new technologies (automation, AI, etc ...) enter the market.

3.4 What are the stages of project management?

The project management process teams are:

- Getting started: The purpose of this section is to describe a project.
- Planning: This phase includes building a road for everyone to follow.
- Performance & Monitoring: At this stage, the project team is formed and deliveries are being built. Project managers will monitor and evaluate project performance to ensure it stays on track.
- Closing: The project is completed, a corpse is made, and the project is transferred to another team to keep it.

3.5 Why is project management important?

Project managers will help your organization:

- have a more predictable project planning and execution process
- Adhere to project budgets, schedules, and scope guidelines.
- Resolve project roadblocks and escalate issues quicker and easier.
- Identify and terminate projects that do not have relevant business value.
- Become more efficient.
- Improve collaboration across and within teams.
- adentify and plan for risks.

3.6 What do project managers do?

In short, project managers are responsible for planning, executing, monitoring, managing and completing projects. However, that is just the tip of the iceberg of project management. Here are some of the key responsibilities of a project manager:

- Create a plan: Project managers are responsible for planning the actual course of the project. The plan should include the scope of the project, the timeline, and the budget. This can include identifying the right tools for the job.
- Join the team: Finding the right team is essential to the success of the project. All project teams will vary depending on the scope of the action and the tasks required to complete the project. Finding experts and experts on the topics for each job required is good..
- Assign tasks: Project managers must provide their team with a clear description of specific tasks and a timeline for all parts of the project. While each team member will be committed to his or her responsibilities, many tasks will require the cooperation of internal and external team members.
  Leading the team: Now that the team is organized and their tasks are assigned, the project manager must keep the machine properly installed. This will include looking at specific people to get status updates, identifying and removing roadblocks, negotiating disagreements, maintaining team ethics, and providing training and counseling.
  Budget management: Most projects will require some expense, which means understanding how to integrate project budget and cost management is critical to success. This will include comparing real-life costs with estimates, and adjusting the project plan if necessary.
  Managing time lines: As a budget, project managers are tasked with keeping everything up to date so the team meets the expected deadline. This will require setting realistic deadlines throughout the project life plan, consistently communicating with their team to receive status updates, and maintaining a detailed plan.
- Involvement of stakeholders: Stakeholders play a major role in your project. They are usually influential people involved in the project. Project managers need to maintain good relationships with an open line of communication with stakeholders who not only help clear roadblocks and empower your team but also create unnecessary issues and disrupt the project if they are unhappy with the direction.

- Handover project: Just because the project objectives have been delivered does not mean that the project manager's job is over. The project manager should now submit the project to a team that will manage, maintain, and move forward. At this point, the project manager will no longer be the "go" person and will be assigned a new project.
- Write a process: Identifying and documenting "lessons learned" is not only a good process for developing a personal project manager, but also for passing that knowledge on to other groups around the organization for future use. This will help others to avoid making similar mistakes, or to use the shortcuts available. [15]

4 THE TOP PROJECT MANAGEMENT METHODOLOGIES

"You mean there's more than one project management methodology?" There are quite a lot of them, actually, and some even combine to form new hybrid approaches. But what are they exactly? How do they help project teams work better? And what makes one methodology better than another?

Project management methodologies are essentially different ways to approach a project. Each one has a unique process and workflow. [18]

4.1 A. The traditional, sequential methodologies

4.1.1 Waterfall project management methodology

The most common way to plan out a project is to sequence the tasks that lead to a final deliverable and work on them in order. This process is also known as the waterfall methodology — the traditional method for managing projects and the one that is simplest to understand. You have to complete one task before the next one begins in a connected sequence of items that add up to the overall deliverable. It's an ideal method for projects that result in physical objects (buildings, computers), and you can easily replicate project plans for future use.

4.1.2 Critical path method (CPM)

The critical path method was developed in the 1950s, based on the idea that there are some tasks you can't start until you finish the previous one. When you string these dependent tasks together from start to finish, you plot out your critical path.

Identifying and focusing on this critical path allows project managers to prioritize and allocate resources to get the most important work done and reschedule any lower priority tasks that may be clogging up your team's bandwidth. This way, if you need to make changes to the project schedule, you can optimize your team's work process without delaying the results.

4.1.3 Critical chain project management (CCPM)

Critical chain project management takes the critical path method one step further. CCPM is a methodology that focuses on the resources needed to complete the project's tasks by adding resource availability to the critical path. It also builds buffers of time around these tasks in the project's schedule, ensuring the project meets its deadlines.

4.2 The Agile family

Agile project management methodologies are growing in popularity, thanks to a highly competitive business environment and increased innovation. In general, Agile methodologies prioritize shorter, iterative cycles and flexibility.

Let's take a look at some of the most popular Agile frameworks.

4.2.1 Agile project management methodology

The core of the Agile methodology was developed in 2001 with four central values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The Agile Manifesto of Software Development put forth a groundbreaking mindset on delivering value and collaborating with customers. Today, Agile can refer to these values as well as the frameworks for implementing them, including Scrum, Kanban, extreme programming, and adaptive project framework.What do these various Agile frameworks have in common?Project objectives are made clear by the customer (internal or external), while the final deliverable can change as the project progresses. The project team works in iterative cycles, always evaluating results at the end. Depending on the results of these evaluations, the final deliverable may be modified to better answer the customer's needs. Continuous collaboration is key, both within the project team and with project stakeholders.

4.2.2 Scrum

Scrum is the most popular Agile development framework because it is relatively simple to implement. It also solves many problems that software developers struggled with in the past, such as convoluted development cycles, inflexible project plans, and shifting production schedules.In Scrum, a small team is led by a Scrum master whose main job is to clear away all obstacles to working efficiently. The team works in short cycles of two weeks called "sprints," though the team members meet daily to discuss their work and any roadblocks that need clearing. This methodology allows for rapid development and testing, especially within small teams.

4.2.3 Kanban

Kanban is another framework for implementing Agile based on a team's capacity. It originated in Toyota's factories during the 1940s. The departments used a visual system of cards ("Kanban") to signal that their team was ready for more raw materials and had more capacity to produce.

Today, this visual approach to managing a project is well-suited to work that requires steady output. Project teams create visual representations of their tasks, often using sticky notes and whiteboards (or online Kanban boards), moving the notes or tasks through predetermined stages to see progress as it happens and identify where roadblocks could occur.

4.2.4 Extreme Programming (XP)

Extreme programming (XP) is another offshoot of Agile. XP is a methodology designed to enhance software quality (and simplicity) and a development team's ability to adapt to customers' needs. Much like the original Agile formula, XP features short work sprints, frequent iterations, and constant collaboration with stakeholders. Change can happen within a sprint. If work hasn't started on a specific feature, it can be swapped out and replaced by a similar task.

4.2.5 Adaptive Project Framework (APF)

Adaptive project framework grew from the difficulty in managing most IT projects using traditional project management methods due to uncertain and changing requirements.

APF begins with a requirements breakdown structure (RBS) to define strategic project goals based on product requirements, functions, sub-functions, and features. The project proceeds in iterative stages, and at the end of each step, teams evaluate previous results to improve performance and practices. Stakeholders can also change the project's scope at the start of each stage so the team can produce the most business value.

4.3 The change management methodologies

Some methodologies deal with managing projects, but with an extra focus on change management — especially planning for risks and taking control of change when it happens. Notable methods include:

4.3.1 Event chain methodology (ECM)

The underlying idea behind event chain methodology is that potential risks often lie outside the project's scope. It's essential to prepare for these risks and plan your response since unexpected events will impact your project's schedule, deliverables, and potentially its success.

4.3.2 Extreme Project Management (XPM)

Extreme project management (XPM) is the opposite of waterfall. It offers you a way to manage massive change and still move forward to project completion. In XPM, you can alter the project plan, budget, and even the final deliverable to fit changing needs, no matter how far along the project is. It's a good option when managing projects with a short timeline of anywhere from a few weeks to mere days.

4.4 D. The process-based methodologies

Next, we have the project management methods that practically veer into business process management (BPM), where each approach focuses on work as a collection of processes. While project management purists may argue that these methods belong on a different list, we think these are still good ways to plan and execute a project.

4.4.1 Lean

Lean is a methodology focused on streamlining and cutting out waste. The first step is to create a work process breakdown to identify and eliminate bottlenecks and delays. The goal is to do more with less — to deliver value to the customer using less manpower, less money, and less time.

4.4.2 Six sigma

Six sigma is a statistics-based methodology seeking to improve the quality of a process by measuring the defects or bugs present and eliminating as many as possible. A process can attain a six sigma rating if 99.99966% of the final product — your project deliverable — is defect-free.

4.4.3 Lean six sigma

Combining the minimalist approach of lean ("no waste!") And the quality improvement of six sigmas ("zero defects!"), lean six sigmas focuses on eliminating waste so that projects are more efficient, cost-effective, and truly answer customers' needs.

4.4.4 Process-based project management

Process-based project management is a methodology aligning all project objectives with a company's larger mission and corporate values. All project goals and tasks remain strategic and must roll up to the larger corporate objectives. The steps involved include defining the process, establishing metrics, measuring methods, adjusting goals when these prove unstable, planning improvements, and implementing them.

4.5 E. Other methodologies

4.5.1 PRINCE2

PRINCE2 stands for Projects In Controlled Environments. It's a method for managing projects used by the UK government and characterized by a product-based planning approach. In PRINCE2, a structured project board is in charge of high-level activities such as setting the business justification and resource allocation. A project manager takes care of the lower level, day-to-day activities like scheduling. This methodology gives teams greater control of resources and the ability to mitigate risk effectively.

4.5.2 PRiSM

PRiSM stands for Projects Integrating Sustainable Methods and aims at managing change while incorporating environmental sustainability into its processes. The goal with PRiSM is to complete tasks while reducing a company's negative environmental and social impact. It is, quite literally, green project management.

4.5.3 Benefits realization

From conception to execution to delivery and beyond, the benefits realization methodology focuses on whether your deliverables satisfy the benefits the customer expects, and not just whether you delivered it on time or within budget. This methodology ensures that you provide real value to customers and stakeholders.

5 Software project management

A Software Project can be considered as the basis for a General Project. It involves managing the software life cycle from the collection of software requirements, design, testing and maintenance, carried out according to specific project management methods, promptly to achieve the intended product.

5.1 software project mangement process

Until the 1980s software was used only by professional people. The whole world was curious about computers, software programs but even so, software had not yet entered people's lives. Today as we see, software has become an integral part of human life. Almost everything we can look around is controlled, controlled by software be it our internet, smartphone, mobile network, electronics, television, travel system, food chain chains, educational and educational institutions, health care centers, climate, agriculture and more. Almost every aspect of our lives is associated with one or more other software. And going forward, it will be a skill and a very important part of our lives.

Software project management although this is, in a sense, such a change is due to the 'good work ethic' performing the so-called 'software project management'. Building software or creating software is a process and a well-defined process for making software is called software project management .

5.2 What is software project management?

Software project management is the art and discipline of planning and managing software projects. It is a small software project management regulation in which software projects are planned, implemented, monitored and controlled.
It is a process of managing, allocating and time resources to make computer software that meets the requirements.
In Software Management, the client and developers need to know the length, duration and cost of the project.

5.3 Why Software Project Management Is Required?

Unlike machines or structures, the software does not have a physical form or a portable product. Today organizations use software to drive business processes. One can imagine the difficulties involved in mapping the software business process. Also, the process of one business may not be the same as others; means that the need for software in one organization will be different. Given the rapid changes in the technology platform and globally, integrated economies pose a threat to the software component that is already developed or under construction. Therefore, to reduce risk and ensure that project delivery will meet the expectations of stakeholders, there is a need to follow a systematic, process-based approach, which is nothing but software project management.

5.4 Prerequisite of software project management?

There are three software project management requirements. These are:

- Quality
- Cost
- Time

It is an important part of a software organization to deliver a high quality product, save costs within the client budget and deliver the project according to each schedule. There are various factors, both external and internal, that can affect this triple object. Any of the three can negatively affect the other two.

Time Cost of Quality (TQC) Threefold issues are also known in general project management. But because of technological changes that are faster in the case of software, the significance of the triple barrier to TQC is far greater than in these projects.

TQC represents the project timeline, project quality and project cost issues that impact on the project stage and vice versa. Any changes in any of these three factors affect other factors e.g. The change in the timeline states that reducing project time will change the quality of the project, increase and decrease the cost of the project, just as the change in project scale can have an impact on everything. Time line, cost and quality.

One can therefore easily understand that project management is very much-needed in a software project to simplify, evaluate, adapt user needs, change the scope of the project by carefully assessing its impact on the project budget, project timeline and project quality.

5.5 IT Project Manager / Software Project Manager

Once it has been accepted that the delivery of a software project is not a simple or harmless process and requires a proper process, then there is a need for experienced and competent professionals who can explain the project process and interpret team effort, stakeholder communication and other aspects of project delivery. We are talking about an IT project manager or a software project manager.

The software project manager is responsible for project planning, project implementation, monitoring and closure. The PM should be familiar with all aspects of project management, also called Software Development LifeCycle (SDLC). The project manager prepares a software development plan, evaluation system, identifies and provides the necessary resources, tracks progress, and manages project communication between team members and other stakeholders. All objectives are to deliver the project within the timeline, budget, resources and quality issues and ultimately deliver software that meets / exceeds customer expectations. Therefore, an IT PM may not be doing a software program or software testing, but should be aware of the challenges, technologies, and expert issues involved in these areas.

5.6 PM's Key Responsibilities While Managing the Project

- Act as a key coordinator with the anchor in the project...
- ommunicate with stakeholders and keep them informed/involved when needed.
- Define and manage the scope of the project.
- Create and manage a project plan.
- Track project progress and track timeline, financial performance.
- Track and manage issues/events.
- Analyze risks and devise a response strategy.
- Manage project resources and track resource time.

5.7 How Does Software Project Management Reach?

When business entities or critical software applications are introduced, software project management is of paramount importance. In order to make a product or service reliable, appropriate, the IT software delivery project team must establish and follow sound project engineering process to improve product quality, reduce costs and adhere to the delivery schedule.

5.8 Software management Activities

Like standard project management, IT project management involves multiple tasks planning, tracking, monitoring and delivering a project. Software project management includes specific tasks in software development and maintenance. Software project management may include the following functions:

- IT Project Resource Management.
- IT Project Budget, Cost Management and Paymentsg.
- T Project Bug / Issues Tracking.
- IT project participant management.
- IT Project Application Management.
- IT Project Document Management.
- IT adjustment management.
- IT Proprement Management.
- IT Project Risk Management.
- IT Project Communication Management.
- IT Integration Management.
- IT Project Planning and Tracking.
- For a complete view and better control over project managers.
- Choose to use project management software in IT teams [39], [41] .

6 Software Development Life Cycle

Software Development Life Cycle, SDLC in a nutshell, a well-defined, structured sequence of software engineering categories to develop a targeted software product.

6.1 SDLC stages

SDLC provides a series of steps to follow in designing and developing a successful software product. The framework of the SDLC includes the following stages: [8], [11]

6.1.1 Communication

This is the first step in which the user starts the application for the software product you want. You are in contact with your service provider and are trying to negotiate terms. He submits his application to a service that provides the organization in writing.

6.1.2 Requirement Gathering

This is a step forward for the software development team working on running the project. The team holds discussions with various stakeholders from the troubled domain and tries to provide as much detail as possible about their needs. Requirements are considered and categorized by user requirements, system requirements and operational requirements. Requirements are collected using a number of practices as provided -

- to study an existing or obsolete program and software.
- conducting user interviews with developers.
- efers to a database .
- collect answers from the questionnaire.

6.1.3 Feasibility Study

After collecting the requirements, the team comes up with a bad software process plan. In this step, the team analyzes whether the software can be customized to meet all user needs and whether the software is out of date. It turns out that, if a project is funded, it works both physically and technically which the organization can take over. There are many algorithms available, which help developers to control the performance of a software project.

6.1.4 System Analysis

In this step, the developers decide on a map of their strategy and try to come up with a software model that is most relevant to the project. System analysis includes Understanding software product limitations, learning program-related issues or changes to existing programs, identifying and resolving project impact on the organization and staff etc. The project team analyzes project size and organizes the schedule and resources accordingly.

6.1.5 Software Design

The next step is to streamline all the information needs and desk analysis and design the software product. Input from users and information collected in the required collection section is for input for this step. The release of this step comes in the form of two designs; logical structure and body design. Engineers produce dictionaries for meta-data and data, logical diagrams, data flow diagrams and in some cases fake codes. [25], [26]

6.1.6 Coding

This step is also known as the planning phase. The implementation of software design begins with the process of writing program code in the appropriate language for planning and developing error-free programs.

6.1.7 Testing

The rating means that 50% of the entire software development process should be evaluated. Errors can damage software from a critical level to its removal. Software testing is performed while coding by developers and complete testing is performed by testing specialists at various levels of code such as module testing, system testing, product testing, internal testing and product testing at the end of the user. Early detection of errors and solutions is the key to reliable software.

6.1.8 Integration

IntegrationSoftware may need to be integrated with libraries, databases and other programs. This section of the SDLC is involved in software integration with external international organizations.

6.1.9 Implementation

This means installing software on user equipment. In some cases, the software requires a rear end installation at the user's end. The software has been tested to carry and adapt and the issues related to integration are resolved during implementation.

6.1.10 Operation and Maintenance

This means installing software on user activity. In some cases, the software requires a backup installation at the user's end. The software is tested to manage and respond to and integration-related issues are resolved during implementation.

6.1.11 Disposition

Over time, the software may slow down before operating. It may be completely out of date or may need some major upgrades. There is therefore an urgent need to eliminate a large part of the system. This category includes data storage and required software components, shutdown, system configuration and termination of the system at the appropriate end of the program.

6.2 Software Development Paradigm

Software processes are a consistent set of tasks for defining, designing, operating and testing software programs. A software process model is an ambiguous representation of a process that expresses the meaning of a particular process. There are many software processes, but all include:

- Clarification — describes what the system should do;

  Design and implementation : defines system planning and implementation;
- Verification — checking whether it complies with customer requirements;.
- Evolution — transforming the system in response to changing customer needs.

6.3 Types of Software Process Model

Software processes, processes and frameworks range from specific steps that can be used directly by the organization in day-to-day work, to the flexible frameworks used by the organization to produce a set of steps tailored to the needs of a particular project or group. In some cases, a "sponsoring" or "maintenance" organization distributes an official set of documents outlining the procedure.

Software Process and Software Development Lifecycle Model

One of the basic concepts of the software development process is the SDLC models that represent the Software Development Life Cycle models. There are many advanced health care systems designed to achieve the various goals needed. Models specify the various stages of the process and the order in which they are performed. The most widely used, popular and important SDLC models are provided below:

6.3.1 Waterfall Model

The waterfall model is the simplest example of how to improve software. It states that all sections of the SDLC will operate in a straightforward manner. Only when the first phase is over will only the second phase begin again.

RAD Model (Rapid Application Development)
RAD is a sequential software development process model that emphasizes a short cycle of development using a building-based construction approach. If the requirements are well understood and defined, and the size of the project is a barrier, the RAD process enables the development team to develop a fully functional plan in the short term.

6.3.2 RAD Model

RAD is a sequential software development process model that emphasizes a short cycle of development using a building-based construction approach. If the requirements are well understood and defined, and the size of the project is a barrier, the RAD process enables the development team to develop a fully functional plan in the short term.

(Rapid Application Development) concept is that products can be built quickly and of high quality through:

- Collect requirements using workshops or focus groups.
- Prototyping and initial user testing of designs.
- Reuse of software components.
- A solid schedule that directs design development in the next product version.
- To do less in review and team communication.

6.3.3 Spiral Model

The job pattern of an IT company engaged in software development can be seen split into two parts:

Software Creation
Software Project Management
A project is a well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). A Project can be characterized as:

Every project may have a unique and distinct goal.
Project is not a routine activity or day-to-day operations.
The project comes with a start time and end time.
Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

6.3.4 V-Model

In this type of SDLC model testing and development, the step is planned accordingly. Therefore, there are verification stages on one side and a confirmation phase on the other side. The IV-Model joins the Coding category.

6.3.5 Incremental Model

The ascension model is not a different model. It is actually a series of waterfall cycles. Requirements are divided into groups at the beginning of the project. In each group, an SDLC model is followed to develop software. The SDLC process is repeated, with each release it can add more functionality until all the requirements are met. In this way,

each cycle acts as a maintenance phase for the previous software release. Modification to a growing model allows for developmental cycles to be dispersed. After that next cycle you can start before the previous cycle ends.

7 Requirement Engineering

Engineering requirements (RE) refer to the process of defining, documenting, and maintaining requirements in the engineering construction process. Demand engineering provides a comprehensive way to understand customer needs, analyze needs, and evaluate feasibility, negotiate a viable solution, articulate a clear solution, ensure clarity and control of needs as they are transformed into an effective system. Therefore, engineering requirements require the directed use of proven principles, methods, tools, and notation to define the intended behavior of the proposed system and its associated implications.

7.1 Requirement Engineering Process

It is a four-step process, which includes :-

- Performance Research.
- The Need for Accuracy and Analysis.
- Software Requirement Specification.
- Software Requirement Verification.
- Software Requirement Management.

8 Software Project Planning

A Software Project is the complete methodology of programming advancement from requirement gathering to testing and support, completed by the execution procedures, in a specified period to achieve intended software product.

8.1 The Need to Software Project Management

Software development is a form of all new streams in the global business, and is not on the side of being involved in the development of system components. Many planning materials are designed to meet the needs of customers. Most importantly, basic technology is changing and evolving normally and quickly so that the knowledge of one object may not be linked to another. All of those business and environmental factors pose a threat to software development; therefore, it is important to properly manage software projects.

9 SOFTWARE RISK MANGEMENT

"Tomorrow's problems are today's danger." Therefore, a clear definition of "accident" is a problem that can cause some loss or threaten the continuation of a project, but which has not yet happened.

These potential problems could damage the cost, schedule or technical success of the project and the quality of our software device or the project team's behaviour.

Risk Management is a process of identification to fix and eliminate these problems before they damage the project.

We need to classify risks, such as potential problems, into current project problems.

Different approaches are needed to address these two types of problems.

For example, because we have not been able to select people with the right technical skills, staff retention is a current problem, but the threat of our highly qualified technical staff is dangerous.

9.1 risk Management in software projects

A software project can deal with a wide variety of risks. To systematically identify major risks that may affect a software project, it is important to classify risks into different categories. The project manager can determine which risks from each class are appropriate for the project.

Three main risk categories can affect a software project:

- Project risk
- Technical risks
- Business risks

9.2 Principle of Risk Management

- Global Vision: In this regard, we are reviewing the overall description of the program, structure, and implementation. We look at the potential and impact of the risk.
- Take a look ahead: Think of a potential threat in the future, and plan for future events.
- Open Communication: This is done to allow for free movement of communication between the client and the team members to ensure the risk.
- Integrated management: In this way, risk management is made an integral part of project management.
- Ongoing process: At this stage, the risk is continuously followed throughout the risk management process.

10 Software Requirement Specifications

Production section of the software development process requirements (SRS) (also called the requirements document). This report lays the foundation for software engineering activities and builds where all requirements are requested and analyzed. SRS is an official report, which serves as a software presentation that enables customers to review whether (SRS) complies with their needs. Also, it contains system user requirements and system specifications.

SRS is a description of a specific software product, program, or set of applications that performs specific functions. It serves several purposes depending on who writes it. First, SRS can be written by a program client. Second, SRS can be written by a program engineer. These two methods create completely different situations and create completely different purposes for the text. The first case, SRS, is used to describe the needs and expectations of users. The second case, SRS, is written for various purposes and serves as a text of the agreement between the customer and the developer.

11 Software Configuration Management

When we make software, the product (software) undergoes many changes in the storage phase; we need to deal effectively with them.

Several people (programs) work together to achieve these common goals. This person generates the product (SC objects), e.g., intermediate type of modules or test data used during debugging, components of the final product.

Items that contain all the information generated as part of a software process are called software configuration.

As software development progresses, the number of Software Configuration (SCI's) items grows rapidly.

These are managed and controlled by SCM. This is where we need software configuration management.

Product optimization refers not only to produce content but also to a specific type of item.

Therefore, SCM is a discipline.

Point to the change
Monitoring and controlling change
Ensure proper implementation of changes made to the item.

Audit and reporting on changes made.
Configuration Management (CM) is a technology for identifying, editing and managing software conversion developed by a program team.

The goal is to increase productivity by minimizing errors (errors).

CM is important for asset management, library management, and project management refinement management.

11.1 Why do we need Configuration Management?

Multiple people are working on software which is consistently updating. It may be a method where multiple version, branches, authors are involved in a software project, and the team is geographically distributed and works concurrently. It changes in user requirements, and policy, budget, schedules need to be accommodated.

11.2 Importance of Software Configuration Management

- It effectively controls and controls access to different SCIs, e.g., by preventing two team members from testing the same object modification simultaneously.
- Provides a tool to ensure that changes are made correctly.
- It has the ability to define and retain various software components.
- SCM is used to keep the system consistent by automatically reproducing the version obtained when converting the same object.

12 Software Quality

A quality software product is defined in terms of the strength of its purpose. That is, a quality product does exactly what users want it to do. For software products, user compatibility is generally defined in terms of satisfaction with the requirements set out in the SRS document. While "firmness of purpose" is a satisfactory description of the quality of many devices such as car, table lover, milling machine, etc., for software products, "purpose fit" is not a completely satisfactory definition of quality.

Example: Think of a software product that works well. That is, do all the work as specified in the SRS document. However, it has an almost unusable user interface. Although it may work well, we cannot take it as a quality product.

Today's concept of quality associated with the software product has several quality methods, such as the following:

Portability: A software device is portable if it can be freely deployed in a variety of application environments, multiple devices, and other software products, etc.

Usability: A software product has better usability if different categories of users can easily request product functions.

Recycling: A software product has good reuse if different product modules can be reused quickly to make new products.

Fixing: The software product is fine if the various requirements described in the SRS document are properly executed.

Maintenance: Software product is secure if bugs are not easily repaired, and when they do appear, new functions can be easily added to the product, and product performance can be easily modified, etc.

12.1 Software Quality Management System

The quality management system is the main method used by organizations to provide that the products they develop are of the quality you want.

The quality system lives as follows:

- Management Structure and Individual Responsibilities: A quality system is the responsibility of the organization as a whole. However, every organization has high-level departments to perform a variety of quality functions. The quality system arrangement should support senior management. In addition to the company's high-quality program support, other employees will take the quality plan seriously.
- Quality System Functions: Quality system functions include the following:

1) Project research.
2) Quality system review.
3) Development of standards, methods, and guidelines, etc.
4) Production of senior management documents that summarize the performance of a quality system in an organization.

12.2 Evolution of Quality Management System

Quality programs have emerged more and more in the last fifty years. Prior to World War II, the general task of producing high-quality products was to test finished products to remove defective equipment. Since then, organizational quality programs have undergone four evolutionary steps, as shown in figs. The first product test function provided a quality control (QC) method.

The quality control policy detects faulty devices and removes them, and determines the causes of the malfunction. Therefore, quality control aims to address the causes of bedbugs and not just to reject products. The next development of qualitative methods was the development of quality assurance methods.

The basic premise of modern quality assurance is that if the organizational processes are right and strictly followed, the products are bound to be of the right quality. New quality-based activities include visual guidance, interpretation, analysis, and improvement of the production process.

Comprehensive quality management (TQM) encourages the organisation's process should continue to be improved through process standards. TQM goes beyond quality assurance and aims to improve the process over and over again. TQM goes beyond recording the steps to use by reorganizing. The name associated with TQM is Business Process Reengineering (BPR).

The BPR aims to revitalize the way business is conducted in the organization. From the above discussion, it can be said that the quality paradigm has changed from product validation to process validation over the years, as shown in figs [17], [22] .

13 Software Design

Software design is a way of transforming the user's needs into an appropriate form, which helps the developer write and use it. It works on behalf of the client's needs, as described in the SRS text (Requirement Specification Software), in the form, that is, it is easy to use using programming language.

The software design phase is the first step in the SDLC (Software Design Life Cycle), which shifts the problem domain's focus into the solution domain. In software design, we consider the program to be a set of elements or modules with clear behaviours and limitations.

13.1 Objectives of Software Design

the purposes of Software designas the Following :-

1) Correctness:Software design should be correct as per requirement.
2) Completeness:The design should have all components like data structures, modules, and external interfaces, etc.
3) Efficiency:Resources should be used efficiently by the program.
4) Flexibility:Able to modify on changing needs.
5) Consistency:There should not be any inconsistency in the design.
6) Maintainability: The design should be so simple so that it can be easily maintainable by other designers.

14 Coding

Coding is the process of converting program structure into a computer language format. This software development coding section deals with translating software-specific design specifications into source code. It is necessary to write source code and internal documents so that the code's consistency in its meaning is easily verified.

Coding is done with coders or programs that are more independent than the developer. The goal is not to reduce the coding phase's effort and expense but to reduce future costs. The cost of testing and care can be significantly reduced.

15 Software Reliability

Software Reliability Means Reliability. It is defined as the system or ability of a component to perform its required functions under certain fixed conditions.

Software reliability is also defined as a software system's ability to perform its assigned function in a given area of a predefined number of installation cases, assuming that the hardware and installation are faulty.

Software reliability is an important link to software quality, built-in functionality, functionality performance, performance, power, installation, storage, and documentation. Software reliability is difficult to achieve because the complexity of the software is high. While any software with a high level of complexity, containing software, will be difficult to achieve a certain level of reliability, program developers are prone to complicating the software layer's complexity with rapid program size and ease of doing so through software development.

For example, larger next-generation airlines will have more than a million lines of software resources on board; next-generation aviation control systems will contain between one and two million lines; the upcoming International Space Station will have more than two million lines on board and more than 10 million lines of ground support software; critical security systems will have more than 5 million online software resources. While complex software is associated with software reliability, it is directly related to other important aspects of software quality, especially performance, [5] power, etc.

16 Software Maintenance

Software maintenance is part of the System Development Life Cycle. Its main goal is to modify and update the software application after delivery to fix bugs and improve performance. Software is a real-world model. When the real world changes, software needs to be transformed where possible.

Software Repair is an integrated function that includes bug fixes, skills development, removal of inefficient skills, and optimization.

16.1 Need for Maintenance

Software Maintenance is needed for:-

1) Correct the mistakes
2) Switch to user needs over time
3) Changing hardware/software requirements
4) Improving system efficiency
5) Enable code too quickly
6) Converting parts
7) Minimize any unwanted side effects.

Therefore, maintenance is required to ensure that the system continues to satisfy the user's needs.

16.2 Types of Software Maintenance

- Corrective Correction:-The debugging aims to fix any remaining errors regardless of where they can create data, format, coding, testing, documentation, etc.

- Consensitive Nutrition:-It contains software modification to match the changes in the ever-changing environment.
- Protective maintenance:-That's how we prevent our system from working. It involves the concept of renewal and new engineering in which the old system and old technology are redesigned using new technology. This backup prevents the system from running out.
- Complete Nutrition :-It defines performance enhancement or performance or limiting software to improve flexibility. This may include existing system performance improvements, improvements in computer performance, etc.

17 Software Testing

Software Testing is a software analysis comparing needs collected from users and system specifications. Tests are performed at the phase level in the life cycle of the software development or the module level in the program code. Software testing includes Verification and Verification.

17.1 What the software testing

Testing is a group of strategies for determining an application's accuracy under a predefined text, but testing may not detect all application errors. The test's main purpose is to detect application failures so that failures can be detected and corrected. It does not show that the product works well under all conditions, but it does not work in certain conditions.

The test provides a comparison that compares performance with the state of the software against processes because the operating system can detect the problem. This method may include previous versions of the same set product, comparable products, and a combination of expected objectives, standards, or alternatives but not limited to this.

Testing includes code testing and coding in different areas, environments, and in all aspects of code testing. In the current software development case, the testing team may differ from the development team so that the test-based information can be used to correct the software development process.

The software's success depends on the acceptance of the target audience, the interface of a simple click, a powerful performance test, and so on. For example, the bank audience is completely different from the video game audience. Therefore, when an organization develops a software product, it can evaluate whether it will benefit its customers and other audiences.

18 Conclusion

The ability to deliver projects on schedule, on budget, and aligned with business objectives is essential in determining the limit on highly competitive global business conditions. Project managers have an incredibly complex role, which includes organizational skills, analytical thinking, and interpersonal skills. Also, Software is a non-physical product. Software development is a new stream of business and has very little experience in building software products. Most software products are customized to fit the needs of the customers. Most importantly, basic technologies change and evolve more often and quickly so that the knowledge of one product can be applied to another. Such types of business and environmental issues increase the risk to software development which is why it is important to manage software projects well. It is necessary for the organization to deliver the highest quality product, keep costs below the customer budget and deliver the project as planned. Therefore, respectively, software project management is required to include user needs as well as budget and time constraints.

Acknowledgment

I would like to express my gratitude to my primary supervisor chao xia, and professor :-Yu Tou Ge software mangement subject teacher at NJUST who guided me throughout this project. I would also like to thank my friends and family who supported me and offered deep insight into the study.

References

[1] R. Pressman, *Software Engineering: A Practitioner's Approach*. New: McGraw-Hill.
[2] *Sandia Software Guidelines*, vol. 4, pp. 85–2347.
[3] A. Std, pp. 1042–1987, 1993.
[4]
[5] E. H. Bersoff, "Elements of Software Configuration Management," *IEEE Transactions*.
[6] Ansi/Ieee.
[7] T. HEZAM, "software project devalopment and mangement," in *software project devalopment and mangement*. sanaa -yemen: ALMAJED, 03 2020, pp. 260–255.
[8] S. Development, . Approach, and H.-P. Halvorsen, pp. 978–82, 2018. [Online]. Available: https://halvorsen.blog
[9] ——, pp. 978–82, 2018. [Online]. Available: https://halvorsen.blog
[10] [Online]. Available: http://en.wikipedia.org/wiki/Software_development_process
[11] H. P. Halvorsen, "So You Think You Can MATLAB," *References Part*, vol. 5, 2017.
[12] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Pair_programming
[13] Agile, 2017. [Online]. Available: http://agilemanifesto.org
[14] H. P. Halvorsen, 2017. [Online]. Available: https://www.halvorsen.blog
[15] ——, 2017. [Online]. Available: https://www.halvorsen.blog
[16] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/VModel_(software_development
[17] H. P. Halvorsen, 2017. [Online]. Available: https://www.halvorsen.blog
[18] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Agile_software_development
[19] [Online]. Available: http://en.wikipedia.org/wiki/Software_Requirements_Specification
[20] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Waterfall_model
[21] [Online]. Available: http://www.noop.nl/2008/07/the-definitive-list-of-software-development-methodologies.html
[22] A. Std, 1983.
[23] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Mockup
[24] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," in *Proc. Future of SE Track*, 2000, pp. 35–46.
[25] I. Richardson, V. Casey, F. McCaffery, J. Burton, and S. Beecham, "A Process Framework for Global Software Engineering Teams," *Information and Software Technology*, vol. 54, no. 11, pp. 1175–1191, 2012. [Online]. Available: 10.1016/j.infsof.2012.05.002;https://dx.doi.org/10.1016/j.infsof.2012.05.002
[26] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Scrum_(development

[27] . Apple, O. X. U. Mac, and Guidelines, 2073. [Online]. Available: https://developer.apple.com/library/mac/-documentation/userexperience/conceptual/applehiguidelines/UEGuidelines/UEGuidelines.html
[28] Wikipedia, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Software_process
[29] H. P. Halvorsen, 2017. [Online]. Available: https://www.halvorsen.blog
[30] E. J. Braude and M. E. Bernstein, 2011.
[31] . K. Wikipedia, J. Schwaber, and Sutherland, "Unified Process," *The Scrum Guide. Available: scrum.org*, 2011.
[32] I. Sommerville, 2016.
[33] H. P. Halvorsen, 2017. [Online]. Available: https://www.halvorsen.blog
[34] Microsoft, 2017. [Online]. Available: http://msdn.microsoft.com/enUS/library/windows/apps/hh465424
[35] H. P. Halvorsen, 2017. [Online]. Available: https://www.halvorsen.blog
[36] Ansi/Ieee.
[37] E. Blankenship, M. Woodward, G. Holliday, and B. Keller, *Professional Team Foundation Server*, 2012.
[38] Wikipedia, 2017.
[39] T. Bucher, M. Klesse, S. Kurpjuweit, and R. Winter, "Situational Method Engineering," *Situational method engineering: fundamentals and experiences*, pp. 33–48, 2007.
[40] L. Malka, 2013. [Online]. Available: http://www.lior.ca/publications/api_design.pdf
[41] A. I. Wasserman, "Information system design methodology," *Journal of the American Society for Information Science*, vol. 31, no. 1, pp. 1–24, 1980. [Online]. Available: 10.1002/asi.4630310102;https://dx.doi.org/10.1002/asi.4630310102
[42] D. E. Damian and D. Zowghi, "RE challenges in multi-site software development organisations," *Requirements Engineering*, vol. 8, no. 3, pp. 149–160, 2003. [Online]. Available: 10.1007/s00766-003-0173-1;https://dx.doi.org/10.1007/s00766-003-0173-1
[43] G. N. Aranda, A. Vizcaíno, A. Cechich, and M. Piattini, "A Methodology for Reducing Geographical Dispersion Problems during Global Requirements Elicitation," in *Proc. 11th Workshop on Requirements Engineering (WER2008)*, 2004, pp. 117–127.
[44] F. Tsui, O. Karam, and B. Bernal, 2014.
[45] [Online]. Available: http://www.itinfo.am/eng/software-development-methodologies/

TAMEEM ABDULBASET ABDULWAHID ABDO HEZAM Author of the project management book, programmer and web developer, member of the Arab Coders Club