Unit 5 – Chapter 14

Software Project Management (Part 2)

Prerequisites:

Software development life cycle, software testing. Software Metrics

Unit Outcomes:

The objective of this chapter is to introduce the concept of risk management and software reliability and quality. At the end of this chapter, students will be able to:

- Identify various risks affecting software projects.
- Discuss risk identification, risk assessment, and risk abatement/containment
- Determine the reliability metrics and software failures.
- Discuss quality management.

14.1 Overview of Risks

Risk indicates the problem that may cause some damage or loss or predict that the project may not be successful. Risk is a threat that is not yet happened. Risk is that input or event or situation that affects or becomes an obstacle to the project completion.

Risk Management can be used to identify these problems, address them, and remove them before they damage the project. Risk management helps in analyzing the impacts of risks so that the quality of the software is improved. It is like an investment or an insurance policy. Project managers find out what might go wrong and plan how to handle the problem using risk management strategies.

The main purpose of risk management is to identify and manage the risks associated with a software project and solve the problem.

Types of risks: There are different categories of risks that can damage the software project. The risks are classified as given below.

1. **Project Risks:** Project risks refer to scheduling problems, resources, person allotment, and some client-related difficulties. The software product cannot be seen

TRANSFORMING DREAMS INTO REALITY
GRAPHIC ERA

or touched like any mechanical, electrical, or other engineering-based product. For example, in house construction, the different stages of construction such as bricks level, slab or carpentry work, etc. can be monitored very easily. But in software, this is not the case. As the software is invisible, the slippage in scheduling is a major project risk.

2. **Technical Risks:** The problems in the main design, programming, interfacing, testing, and maintenance problems can lead to technical risks. The technical risks exist because the development team has less knowledge about the domain, or the software used. If the software requirement specifications are confusing, incomplete, or frequently changing then there is a technical risk. In a few cases, there is a technical risk due to outdated software.
3. **Business Risks:** This type of risk is related to the budget available for the product. The software product may be very good, but it may include business risks due to its unaffordable price.
4. **Other Risks:** The other risks include **people risks** which indicate the risk with the people associated with the product development. Some people may quit the ongoing project, and this creates a disturbance. If the organizational environment is not feasible for the employee, then there is a threat **of organizational risk**. **Tool risk** is like technology risk that is derived from the supporting software and other tools.

14.2 Risk Management Process

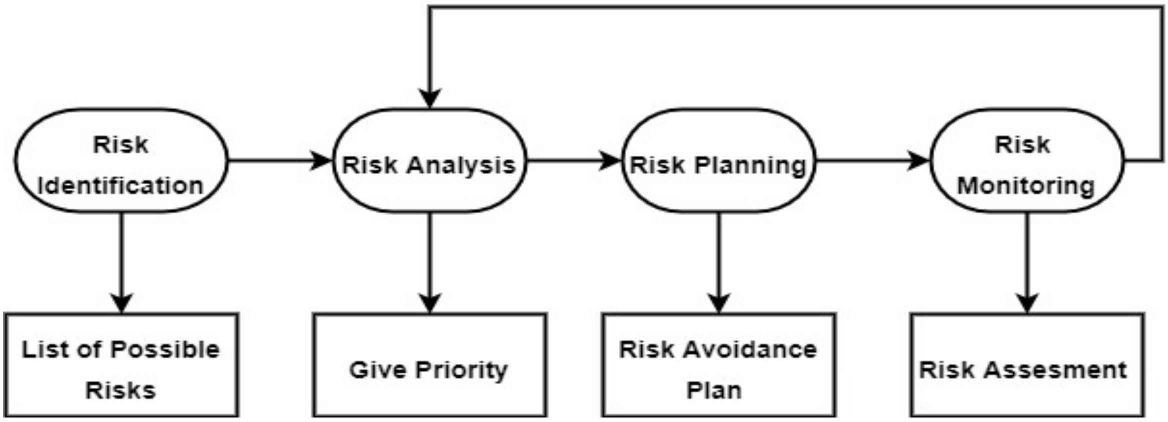

Figure 14.1 Stages in Risk Management

Different stages of the risk management process are depicted in Figure 14.1.

1. **Risk Identification:** This is the starting stage of the risk management process. This stage includes brainstorming where a team of members gather and identify the type of risks. Generally, this process is led by the project manager. The manageable list of risks is summarized at the end of this stage.
2. **Risk Assessment/Analysis:** In the risk assessment, the rank of the risks is identified which describes the potential for damage to the software**. The rank of the risk** is calculated as follows.
3. Let r be the possibility of a risk coming true and the consequences or effect of the problems associated with these risks is indicated by the term s, then the priority of each risk is calculated as:

$$p = r * s$$

With the priority of the risks, the most damaging risk is handled first.

4. **Risk Planning:** In this process, for each identified risk, different strategies are developed. Each risk should be handled separately. Risk planning includes the following steps
    a. **Avoidance Strategies**: In this step, the focus is on reducing risks. Replacing defective components in the product, buying a high-performance database, discussing changed requirements with the client, etc. are some of the risk avoidance strategies.
    b. **Risk Reduction/Minimization strategies:** In this stage, the focus is on minimizing the impact of the risk. For example, giving a hike in salary or incentive to employees so that they do not leave their job.
    c. **Contingency plans:** This plan represents the strategy for handling the worst-case damage. Preparing a document representing how the project is making a very good contribution to the business is an example of a contingency plan.
5. **Risk Monitoring**: In this process, the assumptions, and the business risk description is verified. The change in the identified risk is monitored in this step. The risks can be avoided or transferred or reduced. For selecting the specific strategy for handling risk, the project manager estimates the cost using risk leverage. The risk leverage is nothing, but the cost, and it is calculated as:

$$R\_L = (R\_B - R\_A)/C$$

Where R_L is the risk leverage, R_B is the risk exposure before reduction, R_A is the risk exposure after reduction and C is the cost of risk reduction.

TRANSFORMING DREAMS INTO REALITY
GRAPHIC ERA

14.3 Software Reliability

Software reliability is the probability of the software working correctly. Software reliability depicts the trustworthiness or dependability of the system. Software with many defects is known as unreliable. Reliability depends on the location of errors and how that product is used. If the input to the system is selected such that it always gives correct output, then the reliability is high, whereas if the input always invokes errors in the software, then the reliability is low. The reliability can be improved by reducing the number of defects. Measuring software reliability is difficult due to several reasons:

- More errors, poor is the reliability.
- The reliability depends on how well the software product is observed and tested using suitable input.
- Based on fixing bugs, the reliability keeps changing.

Hardware and Software Reliability:

Hardware reliability is different than software reliability as most of the time, the components wear and tear over time, and there is hardware failure. The software fault will remain till the error is tracked and either the design or program is changed. After the hardware repair, the reliability is maintained, whereas, in the case of software reliability, it may increase or decrease. Thus, the hardware reliability study is related to stability, and software reliability aims at reliability growth. The difference between hardware and software reliability is shown in Figure 14,2

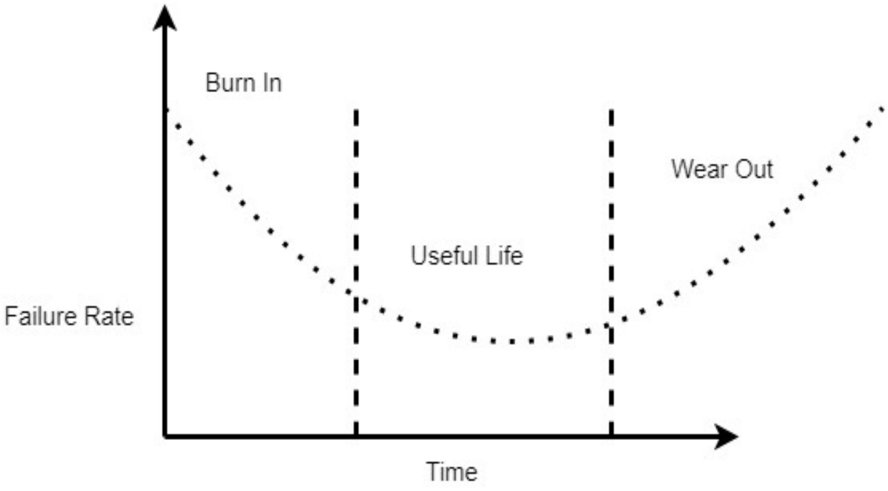

Figure 14.2 Hardware Product

As shown in the graph, the hardware and software reliability depend on how much hardware component is deteriorated over the usage or after home many days the software is outdated. The software product growth is shown in Figure 14.3.

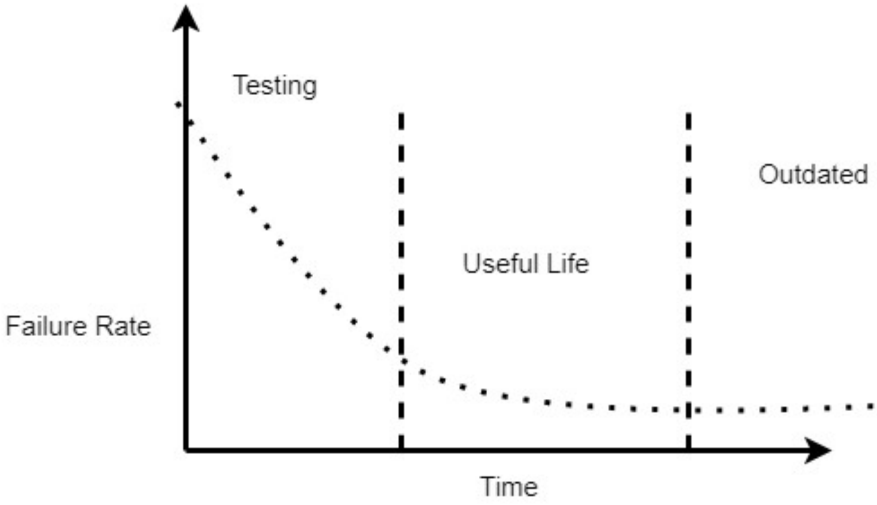

Figure 14.3 Software Product

To measure the reliability of the product, different software metrics are used as follows:

1) **Rate of occurrence of failure (ROCOF):** this metric is used to measure the occurrence of software failure. The behavior of the software is observed over a period and the total number of failures is measured.
2) **Mean Time to Failure (MTTF):** This metric is used to measure the average time between two successive failures. For this metric, only run time is used and not the booting time, system shutdown, etc.
3) **Mean Time to Repair (MTTR):** This metric is the measurement of time to fix the error. Duration for tracking an error and fixing is considered in this metric.
4) Mean Time Between Failure (MTBR): This metric combines MTTF and MTTR by considering the real-time and not the execution time.
   MTBF = MTTF + MTTR
5) **Probability of Failure on Demand (POFOD):** This metric is used to measure the likelihood of the system failure when a request is made, and it does not include time measurement. For example, a POFOD of 0.002 would mean that 1 out of every 2000 service requests would fail.

TRANSFORMING DREAMS INTO REALITY GRAPHIC ERA

6) **Availability:** This metric is used to determine the failure and repair time of the system. Availability indicates how likely the system will be available to clients during a certain time.

14.4 Software Failure

The reliability is based on the software failure. The software failures are classified into five different types as follows.

1) **Transient:** This failure is only due to certain input values given to the system
2) **Permanent:** Software failures for all input values while invoking a function of the system are called permanent failures.
3) **Recoverable:** When a system can recover with or without any manual help from the failure, then it is called recoverable failure.
4) **Unrecoverable:** A software failure where the system needs to be restarted.
5) **Cosmetic:** The system failure due to cosmetic lead to minor irritable software failures, e.g., not being able to click on an option for selecting a menu.

14.5 Software Quality

Software quality is determined unlike a traditional product such as the working of a fan or any machine. Software is not considered a quality product if it only satisfies the functionality. If the software is working as per requirement specification but it takes a long time and memory, then it is not quality software.

The software which is defect free, delivered within the allocated budget and time, meets the client's requirements, and can be easily maintained is known as quality software. The software quality reflects both functional as well as structural quality.

As per the modern view of quality, several factors determine software quality.

1) **Portability:** When a software product can be made to work on different operating system environments, different systems, etc., then it is known as portable software.
2) **Usability:** When both expert senior and novice or new persons can use the software without any difficulty, then it is said to have usability quality attributes.
3) **Reusability:** This factor determines the reusability of the software, where different components can be reused several times to develop new software.
4) **Correctness:** This factor determines whether the software is correctly implemented according to the requirement specifications.

TRANSFORMING DREAMS INTO REALITY
GRAPHIC ERA

5) **Maintainability:** This factor determines whether the software product can be easily corrected, and new functions can be added without much overhead.

14.5.1 Software Quality Management System

A quality management system is also referred to as a quality system. It is a fundamental methodology used by organizations to give assurance that the products have desirable quality. A quality management system must contain the following.

1. **Managerial Structure and Individual Responsibilities:**
   Quality is a responsibility of the whole organization with many departments. There is a quality department in every organization. Quality management is used to establish a framework for the organization's standards. The standard should be applied to software-related documentation such as system requirements, design, and code. At the project level, every process is checked whether it is a quality process producing the products that are of expected quality standard.
2. **Quality System Activities:**
   The quality system activities are related to auditing the projects, reviewing the quality, developing guidelines, methods, etc. for organizations, generating reports for the top management, etc. The quality goals are defined in the quality plan to define what processes are to be used.

The most used terms are **quality assurance or quality control** in the manufacturing industry.

Quality management is evolving over the past several years. In the earlier days before World War II, the quality was checked only to eliminate the defects. Since then, the quality system has undergone different stages of evolution as shown in Figure 14.4.

TRANSFORMING DREAMS INTO REALITY
GRAPHIC ERA

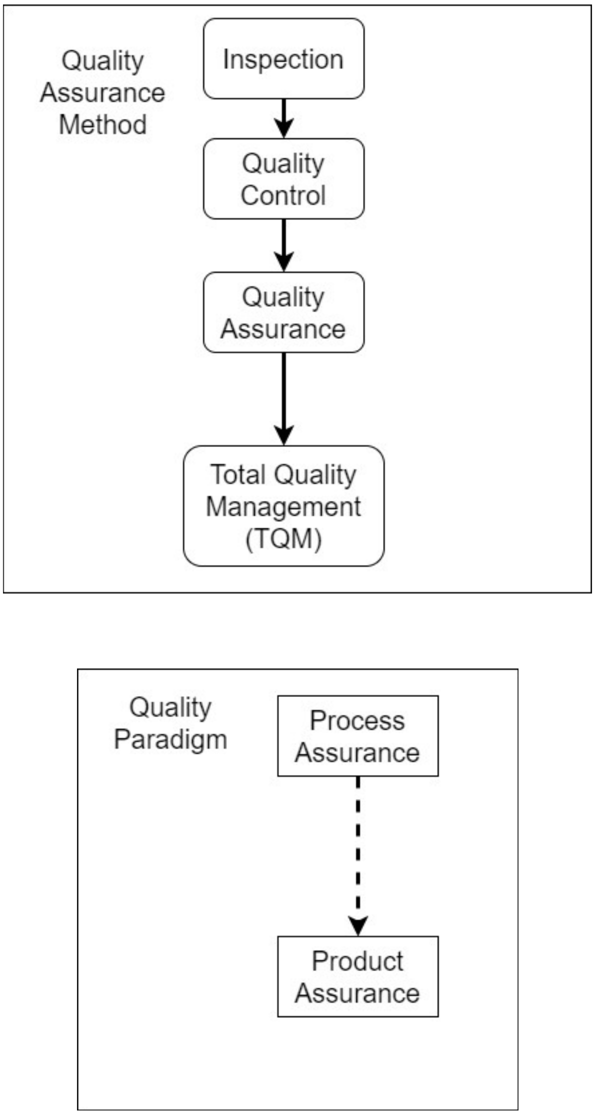

Figure 14.4 Evolution of Quality Management

Initially, the quality system was only to determine the defects but in recent days, the reasons for the cause of defects are also considered for correction. Thus, quality control is not just rejecting the products but also correcting them.

14.6 ISO 9000 certification

ISO (International Standards Organization) is an international standard development organization. It is composed of representatives from the national standards organizations of member countries. ISO published its 9000 series of standards in 1987. It provides guidelines for maintaining quality ISO certification and serves as a reference for contracts between independent parties. The operations and responsibilities, and report aspects are addressed by the ISO for producing high-quality development. The ISO 9000 standard gives guidelines for producing the product and it is not concerned with the product itself.

Types of ISO 9000:

- **ISO 9001:** This standard applies to most software organizations. The design, development, production, and servicing of the products use this standard.
- **ISO 9002:** This standard applies to those organizations which are not involved in the design of the product but are concerned with the production. The industries such as car or steel manufacturing companies buy their design from external parties and only focus on manufacturing. Therefore, this standard does not apply to software products.
- **ISO 9003:** This standard is applied only to organizations that are involved only in the installation and testing of the products.

Need for ISO Certification:

With the use of ISO certification, the customer gets confidence in the product. ISO 9000 uses a well-structured and documented process in place, and this assures that the developed software is of high quality. The development process is much more efficient and cost-effective with the ISO 9000. If there are any weak points, then they are pointed out by the ISO 9000 and the remedies are given. Thus, the use of ISO 9000 sets the basic framework for an efficient process and results in total quality management (TQM).

However, the ISO 9000 sets the steps for the software production process but **does not guarantee the high quality of the process.** It is not a full-proof agency, and it may downplay the domain experts. It requires a heavy emphasis on documentation and takes a lot of time and effort.

14.7 Software Engineering Institute Capability Maturity Model (SEI CMM)

The quality of the software can be improved using the SEI CMM. With the help of the SEI CMM model, there will be business benefits. SEI CMM can be used for capability evaluation and software process assessment.

- With the help of capability evaluation, we can understand a way to assess the software process capability of an organization.
- To improve the process capability, the software process assessment is used by an organization.

TRANSFORMING DREAMS INTO REALITY
GRAPHIC ERA

There are five maturity levels of software development industries using SEE CMM.

- Initial
- Repeatable
- Defined
- Managed
- Optimized

There are several key areas provided by SEI CMM for the organizations. The key process areas (KPA) are as given in Table 14.1.

The quality is improved gradually from one level to the next over several stages.

Each stage is carefully designed so that one stage enhances the already existing capability.

CMM Level	Focus	Key Process Areas
Initial	Competent people	
Repeatable	Project Management	Software project planning and configuration management
Defined	Defining of processes	Process definition, training programs, and peer reviews
Managed	Product and process quality	Software metrics and quality management
Optimizing	Improvement of continuous process	Defect prevention, process change management, technology change management.

Table 14.1 KPA s and CMM Levels

14.7.1 ISO 9000 certification vs. SEI/CMM

- ISO 9000 includes a set of international standards on quality management, and it is awarded by the international standards body. It helps companies to get efficient documentation for quality whereas the SEI CMM was developed specifically for the software industry and therefore addresses many issues which are specific to the software industry alone.

- The ISO 9000 is focused on the customer and supplier relationship, whereas SEICMM is only on the improvement of intermediate processes to achieve a high-quality product.
- ISO 9000 is accepted by most countries, but the SEI CMM is used in USA and less in other countries.
- SEI CMM model provides a list of KPAs in an organization for gradual quality improvement from one level to the next.

14.8 Multiple Choice Questions (MCQs)

1. CMM model is used for
   - a) Maintain software reliability
   - b) Inspect the software
   - c) Improve the process
   - d) Design the software model
2. The number of expected failures per one billion hours of operation for a device is given by:
   - a) MTTF
   - b) MTBF
   - c) MTSF
   - d) MTTR
3. Quality is mainly focused on:
   - a) Software
   - b) Programmers
   - c) Internet
   - d) None of the above
4. The external failure cost for software belongs to:
   - a) Help line support
   - b) Testing
   - c) Warranty work
   - d) Complaint resolution

5. Software reliability is the same as software safety

   a) True

   b) False

14.9 Review Questions

1. What are the major classes of risks affecting a software project?
2. With a suitable example, discuss how to assess project risk.
3. Define reliability and list the reliability metrics in brief.
4. Compare software and hardware reliability.
5. What are the different software failures? Explain with a suitable example.
6. Explain the term risk leverage and scheduling slippage, Identify the suitable case studies.
7. Identify the different levels of the SEI Capability Maturity Model.
8. Explain the key process areas of a software organization provided by the SEI Capability Maturity Model.
9. Differentiate between ISO 9000 certification and SEI CMM.
10. How the reliability of the product can be enhanced? Discuss in brief.

14.11 MCQ solution keys

1: c	2: b	3: a	4: b	5: b