Risk Management: Introduction

Risk Management

- 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

- Risk Management is associated with a software project and solving the problem 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 Risks

- Project risks refer to scheduling problems, resources, person allotment, and some client-related difficulties.
- The software product cannot be seen 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, the slippage in scheduling is a major project risk.

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- In a few cases, there is a technical risk due to outdated software.

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.

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.

Risk Management Process

Risk Management Process

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.

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.
- 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.

Risk Planning

- In the Risk Planning process, for each identified risk, different strategies are developed.
- Each risk should be handled separately.
- Risk planning includes the following steps
  - Avoidance Strategies
  - Risk Reduction/Minimization strategies
  - Contingency plans
-

Risk Monitoring

- In this process, the assumptions, and the business risk description is verified.
- Any 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.

Risk Monitoring

- 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.

Software Reliability

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.

Software Reliability

- If the input to the system is selected such that it always gives the correct output, then the reliability is high.
- 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.

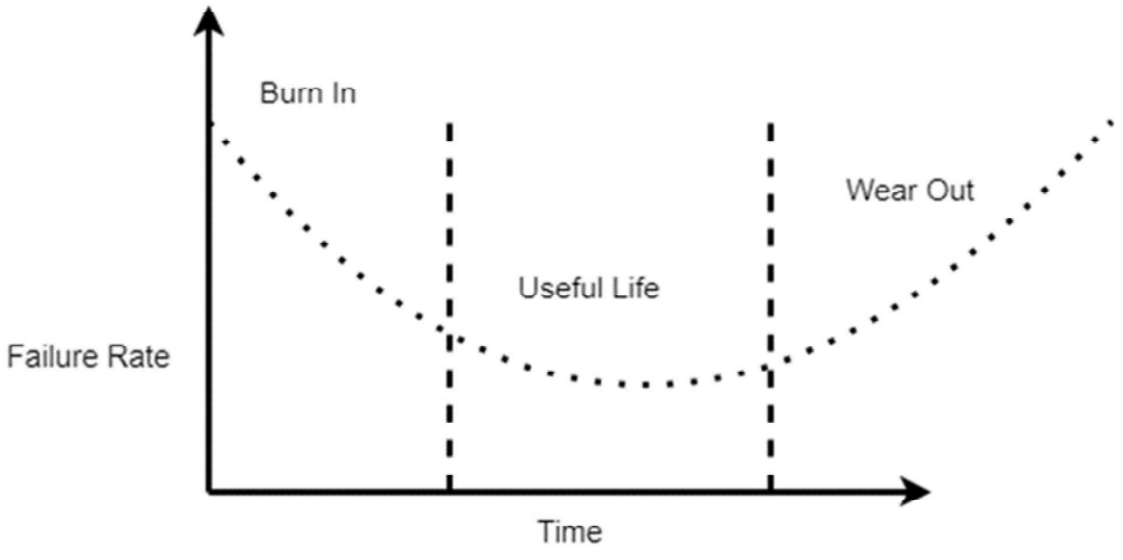
Software Reliability

- Measuring software reliability is difficult due to several reasons:
- More errors, poor is the reliability, and the errors are in the code.
- The reliability depends on how well the software product is observed and tested using suitable input.
- Based on fixing bugs, the reliability keeps changing.

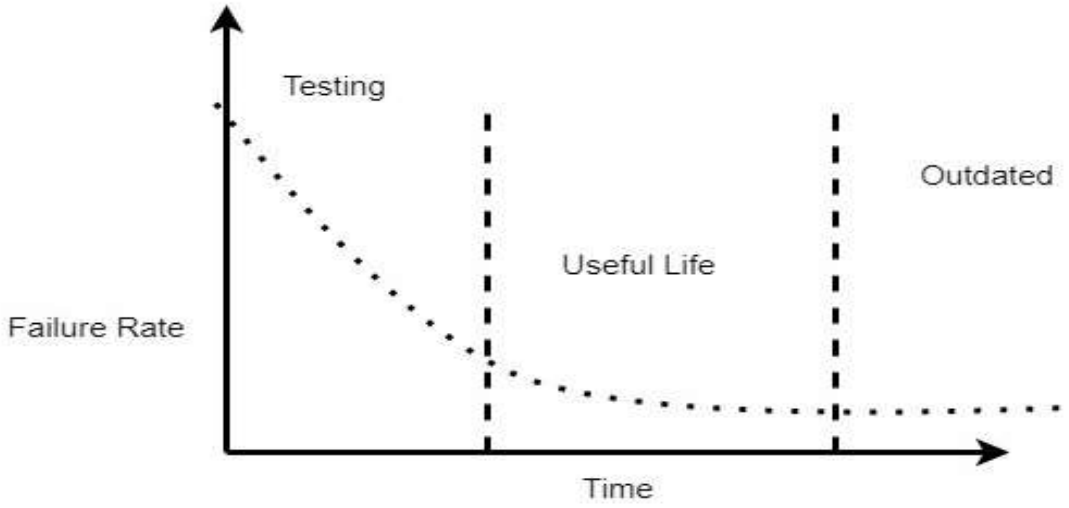
Hardware and Software Reliability

- Hardware reliability refers to the component's wear and tear over time, and there is hardware failure.
- The software fault will remain till the error is tracked or 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.
- Hardware reliability study is related to stability, and software reliability aims at reliability growth.

Hardware Reliability



Software Reliability



Unit 5 – Maintenance

Reliability Metrics

Reliability Metrics

- 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.

Reliability Metrics

- 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.

Reliability Metrics

- 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.

Reliability Metrics

- 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

Reliability Metrics

- 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.

Reliability Metrics

- 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.

Software Failures

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.

Software Failure

- 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.

Software Failure

- 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.

Unit 5 – Maintenance

Software Quality

Software Quality

- Software quality is not determined like 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.

Factors for Software Quality

- **Portability:** When a software product can be made to work on different operating system environments, different systems, etc.,
- **Usability:** When both expert seniors and new persons can use the software without any difficulty.
- **Reusability:** Different components can be reused several times to develop new software.

Factors for Software Quality

- **Correctness:** The software is correctly implemented according to the requirement specifications.
- **Maintainability:** The software product can be easily corrected, and new functions can be added without much overhead.

Software Quality Management

- A quality management system or quality system 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
2. Quality System Activities

Software Quality Management

- Managerial Structure and Individual Responsibilities:
- Quality is a responsibility of the whole organization with 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 produced products are of the expected quality standard.

Software Quality Management

- **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.

Unit 5 – Maintenance

ISO 9000

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.

ISO 9000 Certification

- ISO 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

1. ISO 9001
2. ISO 9002
3. ISO 9003

ISO 9001

- This standard applies to most software organizations.
- The design, development, production, and servicing of the products use this standard.

Types of ISO 9000

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.

Types of ISO 9000

- **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.**

Need for ISO Certification

- 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.

SEI CMM

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.

Software Engineering Institute Capability Maturity Model (SEI CMM)

- There are five maturity levels of software development industries using SEE CMM.
- Initial
- Repeatable
- Defined
- Managed
- Optimized

Software Engineering Institute Capability Maturity Model (SEI CMM)

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

Software Engineering Institute Capability Maturity Model (SEI CMM)

CMM Level	Focus	Key Process Areas
Managed	Product and process quality	Software metrics and quality management
Optimizing	Improvement of continuous process	Defect prevention, process change management, technology change management.

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.
- ISO 9000 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.

ISO 9000 certification vs. SEI/CMM

- The ISO 9000 is focused on the customer and supplier relationship, whereas SEI/CMM 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.

Contact

For any queries contact:

vaishali.kulkarni@geu.ac.in