UNIT-10

Software Management

⊛Software Project Management:

Software project management is an art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled. It is an essential part of software engineering. Good project management cannot guarantee project success however bad management usually results in project failure.

Software Project Manager:

A software project manager is a person who undertakes the responsibility of planning, executing and controlling the software project. Project managers are responsible for planning and scheduling project development. Software project management is differing from other type of project management due to following reasons:

i) The product is intangible in nature and it can be realized only it cannot be seen or touched.

ii) There are no standard software processes

iii) Large software projects are often One-off projects.

⊛. Project Management Activities:

1) **Project Planning:** Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project. Project planning involves breaking down the work into parts, assign these to project team members, allocate resources, and prepare solutions to problems.

Project planning sections:

i) Introduction: It describes objectives of the projects and sets out the constraints like time, budget etc.

ii) Project organization: It describes the people involved and their role in team.

iii) Risk analysis: It describes possible project risks.

iv) Hardware and software resource requirement: It specifies the hardware and software required to carry out the development.

v) Work breakdown: The project is breakdown into activities and identifies milestones and deliverables associated with each activity.

vi) Project schedule: It shows the dependencies between activities.

vii) Monitoring and reporting mechanism: This defines the management report that should be produced.

Types of project plan:

i) Quality plan: It describes the quality procedures and standards that will be used in a project.

ii) Validation plan: It describes the approach, resources and schedule used for system validation.

iii) Configuration management plan: It describes the configuration management procedures and structures to be used.

iv) Maintenance plan: It predicts the maintenance requirements of the system, maintenance costs and effort required.

v) Staff development plan: It describes how the skill and experience of the project team members will be developed.

2) Project Scheduling:

Project scheduling is the process of separating the total work involved in a project into separate activities and judging the time required to complete these activities. It is a detailed plan showing dates when each activity should start and finish, also it includes when and how much resource will be required. Creating a project schedule includes four stages:

- → Constructing an ideal activity plan.
- → Risk analysis
- → Resource allocation
- → Schedule production.

Project schedules are usually represented as a set of charts showing the work breakdown, activities dependencies and staff allocations. Bar charts (for example Gantt chart) and activity networks are graphical notations that are used to illustrate the project schedule. Bar charts show who is responsible for each activity and when the activity is scheduled to begin and end. Activity networks show the dependencies between the different activities.

3) Risk Management:

Risk management is process of identifying risks that might affect the project schedule or quality of project being developed and taking action to avoid these risks. Risk management involves identifying and assessing major project risks to establish the probability that they will occur and the consequences for the project if that risk does arise.

There are mainly three categories of risks which are likely to occur in software projects:

1) **Project Risk:** Project risks influences the schedule of software project and that if occur, delay the development process.
Example: staff turnover, hardware unavailability etc.

ii) Technical risks: Technical risks involves implementation, testing, and maintenance issue. Most technical risks appear due to the development teams insufficient knowledge about the project.

iii) Business risk: It affects the organization business that is developing the software. Business risks could be quite dangerous for the long-term sustainability of the business. Example: Technology change, Product completion etc.

Risk Management Process:

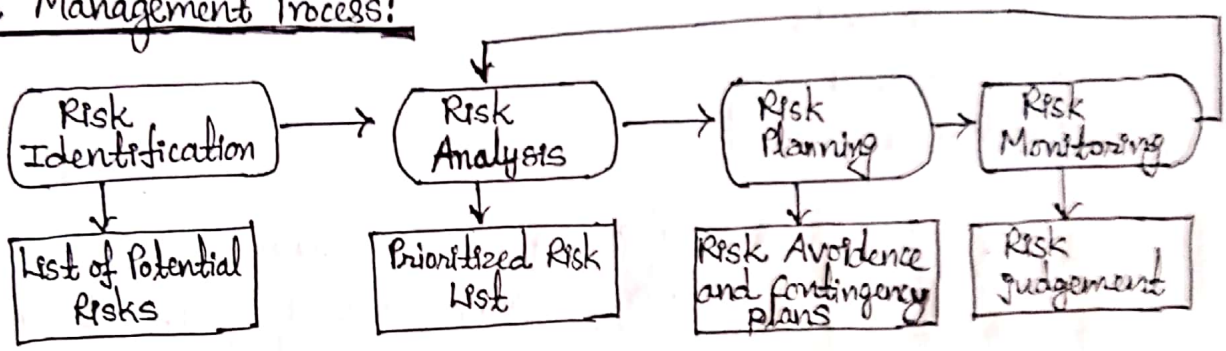

Fig: Risk management process.

The risk management process is an iterative process which continues throughout the project. Once an initial set of plans are drawn up, the situation is monitored.

i) Risk identification: It is the first stage of risk management. It is concerned with discovering possible risk to the project. It also involves prepration of risk list.

ii) Risk Analysis: In this phase all identified risks are considered and made a judgement about problems causing risk in the project, identify probability of occurrence of risk and the impact of the risk. The probability of risks may be judged as very low (< 10%), low (10-25%), moderate (26-50%), high (51-75%), and very high (>75%). Risks are tabulated in order according to the seriousness of the risk.

iii) Risk planning: It is the process of consideration of the key risks that have been identified and formulation of strategies to manage the risk. The strategies to manage risk are:

→ Avoidance Strategies (that tells reduction of arising risks).
→ Minimization Strategies (that tells impact of risk will be reduced).
→ Contingency Plans (Prepared for worst and have strategy to deal with it).

iv) Risk monitoring: Risk monitoring means regularly judging each of the identified risk to decide whether or not the risk is becoming more or less portable and whether the effect of risk have changed.

4) People Management:

The process of planning, organizing, directing, and controlling human resources is called people management. It is also called human resource management. People Management ensures that organizational, individual, and societal needs are satisfied. Human resource management includes all activities used to attract and retain employees.

5) Proposal Writing: Project manager have to write the project proposal to win a contract from the customer. Proposal writing is a skill that acquire through practice and experience. Proposal should include the objective of the project and how it will be carried out. It also includes cost and schedule estimates.

⊛. Milestones and Deliverable:

Project milestones are the predictable outcome of an activity where some formal report of progress are addressed. Milestones are recognizable end points of software process activity, they may be short report of what has been completed and presented to the management. Milestones should represent the end of a distinct, logical stage in the project.

A deliverable is a project report (result) that is delivered to the customer at the end of major project phase. Deliverables are usually milestones but milestones need not to be deliverables. A deliverable could be a report, a document, a server upgrade of an overall project. A deliverable may be composed of multiple smaller deliverables.

⊛ Software Pricing:

Software pricing must take into account broader organizational, economic, political and business consideration. The major factors that affect the software pricing are as follows:

→ Development organization quotes a low price if want to move into a new segment of software market. Accepting low profit in one project may give the opportunity of more profit later.

→ If an organization has no idea about cost estimation then it may increase its price by some contingency

→ If the customer is not clear about their requirement then there is chance of changing requirement at that time organization may lower its price.

→ If the financial health of the organization is not good then they may lower their price to gain contract and establish themselves in business.

⊛. Estimation Techniques:

The software estimation process includes estimating the size of software product, effort required, and overall cost of the project. The approaches to cost estimation can be tackled using either top-down or a bottom up approach.

1) **Top-down Approach:** Top-down estimating method is also called Marco Model. Using this method, an overall cost estimation for the project is derived from the global properties of the software project.

and then the project is partitioned into various low-level components. This method is more applicable to early cost estimation when only global properties are known. This approach starts at system level.

Advantages:

↳ It will not miss the cost of system-level functions.

↳ It requires minimal project detail, and it is usually faster, easier to implement.

Disadvantages:

↳ It often does not identify difficult low-level problems.

↳ It provides no detailed bases for justifying estimates.

2) Bottom-up Approach:

Using this method, the cost of each software components is estimated and then combines the results to arrive at an estimated cost of overall project. It aims at constructing the estimate of a system from the knowledge gained about small software components and their interactions. The leading method using this approach is COCOMO's detailed model.

Advantages:

↳ It allows software group to handle an estimate in an almost traditional fashion.

↳ It is more stable because the estimation errors in the various components have a chance to balance out.

Disadvantages:

↳ It tends to be more time-consuming

↳ It may not be accurate because the necessary information may not be available in the early phase.

⊛ COCOMO Model:

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model. This model predicts or estimates the effort required for the project, total project cost and scheduled time for the project. This model depends on the number of lines of code for software product development.

In COCOMO, projects are categorized into three types:

i) **Organic:** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past. Example of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

ii) **Semi-detached:** In this type the development consists of a mixture of experienced and inexperienced staff. Example of Semi-detached system includes developing a new operating system, a database management system, and complex inventory management system.

iii) **Embedded:** A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. For example: ATM, Air Traffic control.

Formulas for estimating the effort based on the code size:

Organic: Effort = 2.4 (KLOC) 1.05 PM

Semi-detached: Effort = 3.0 (KLOC) 1.12 PM

Embedded: Effort = 3.6 (KLOC) 1.20 PM

Formulas for estimating the development time based on the effort:

Organic: Tdev = 2.5 (Effort) 0.38 Months

Semi-detached: Tdev = 2.5 (Effort) 0.35 Months

Embedded: Tdev = 2.5 (Effort) 0.32 Months

(LOC means lines of code) (K means thousand)

Example: A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

Solution:

The semi-detached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence,

$$\text{Effort} = 3.0(200)^{1.12} = 1133.12 \text{ PM}$$ (indicates person months)

$$T_{dev} = 2.5(1133.12)^{0.35} = 29.3 \text{ Months}$$

$$\text{Average staff size} = \frac{\text{Effort}}{T_{dev}} = \frac{1133.12}{29.3} = 38.67 \text{ Mans}$$

$$\text{Productivity of software} = \frac{KLOC}{\text{Effort}} = \frac{200}{1133.12} = 0.1765 \text{ KLOC/PM}$$

$$= 176 \text{ LOC/PM}$$

⊛. COCOMO model types:

There are three types of COCOMO model:

1. Basic COCOMO: It is one type of static model to estimate software development effort quickly and roughly. It mainly deals with the number of lines of code and the level of estimation accuracy is less as we don't consider the all parameter belongs to the project. The estimated effort and scheduled time for the project are given by the relation:

$$\text{Effort}(E) = a*(KLOC)^b \text{ PM}$$

$$\text{Scheduled Time } (D) = c*(E)^d \text{ Months (M)}$$

where,

E = Total effort required for the project in Person-Months (PM)

D = Total time required for project development in Months (M).

KLOC = the size of the code for the project in Kilo lines of code.

a, b, c, d = The constant parameters for a software project.

ii) Intermediate COCOMO:

The intermediate model estimates software development effort in terms of size of the program and other related cost driver parameters (product parameter, hardware parameter, resource parameter) of the project. The estimated effort and scheduled time are given by the relationship:

$$\text{Effort } (E) = a^* (KLOC)^b * EAF \text{ PM}$$

$$\text{Scheduled Time } (D) = c^* (E)^d \text{ Months}$$

where, EAF = Effort Adjustment Factor, which is calculated by multiplying the parameter values of different cost driver parameters. For ideal, the value is 1.

iii) **Detailed COCOMO:** It is the advanced model that estimates the software development effort like Intermediate COCOMO in each stage of software development life cycle.

⊛. Introduction to Quality Management and Configuration Management:

Software Quality Management ensures that the required level of quality is achieved by submitting improvements to the product development process. Software quality assurance aims to develop culture within the team and it is seen as everyone's responsibility. Software quality management following activities:

i) **Quality Assurance:** QA aims at developing organizational procedures and standards for quality at organizational level.

ii) **Quality Planning:** Select applicable procedures and standards for a particular project and modify as required to develop a quality plan.

iii) **Quality control:** Ensure that best practices and standards are followed by the software development team to produce quality products.

Configuration Management is concerned with the policies, processes and tools for managing changing software systems. It is essential because it is easy to lose track of what changes and component versions have been incorporated into each system version. Configuration management activities are as follows:

→ Change management
→ Version management
→ System building
→ Release management.

Included at last. It is part of Unit-2

Q. What are the drawbacks of software reuse? Explain. [Model Set]

Ans:- Software elements of a product take time to create and reuse of them saves development time. But still it got some drawbacks:

i) Maintenance cost increase: Sometimes the old created components may fail doing their functions resulting high maintenance cost.

ii) Software tools require longer support: As the software tools that were used previously may not be in use today so we have to maintain those tools.

iii) Software tools may become obsolete: Software tools may get outdated after sometime, so we need to keep them up-to-date.

iv) Long Selection Time: It takes time to select which component can be used for long term.

v) Component Failure: One component may not work, then interconnected components will also not work, resulting failure of system.

Q. Differences between alpha testing and beta testing. [May be important, we can find solution everywhere read this in addition].

If my notes really helped you,then you can support me on esewa for my hardwork.

Esewa ID: 9806470952