

8.7 PROJECT PLANNING

Project planning is part of project management which relates to the use of schedules such as gantt charts to plan and subsequently report progress within the project environment.

Initially, project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined

using an activity network diagram that enables identification of the critical path. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the plan becomes what is known as the baseline. Progress will be measured against the baseline throughout the life of the project. Analyzing progress compared to the baseline is known as earned value management.

Typically project planning involves the following activities:

- (a) Methodology
- (b) Risks
- (c) Quality Plan
- (d) Configuration management plan
- (e) Project Schedule
- (f) Resource Plan

(a) Methodology ✓

A Methodology is nothing but a series of steps which need to be carried out to meet an end objective. In software development there are a number of methodologies which can be followed for development of a software product. These methodologies have different variations and each has thus own pros and cons. The popular methodologies used for software developments are:

- (I) Waterfall model
- (II) Prototyping
- (III) Rapid application development

A description of the above methodologies is given as:

(I) Waterfall Model : The waterfall model derives its name due to the cascading effect from one phase to the other. In this model each phase well defined starting and ending point, with identifiable deliveries to the next phase.

(II) Prototyping : The Prototyping Model is a systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users.

(III) Rapid application development : RAD is a linear sequential software development process model that emphasis an extremely short development cycle using a component based construction approach. If the requirements are well understood and defines, and the project scope is constraint, the RAD process enables a development team to create a fully functional system with in very short time period.

(b) Risks

Risks are events or conditions that may occur, and whose occurrence, if it does take place, has a harmful or negative effect". Risk analysis is a set of techniques used to investigate problems created by uncertainty and to assess their effects.

(c) Quality Plan

A quality plan describes the various checks that will be done to improve the quality of deliverables. Typically this will contain the following:

- (i) Process for reviewing deliverables
- (ii) Walk through
- (iii) Process Audits
- (iv) Testing process

(d) Configuration Management Plans

In software engineering, software configuration management (SCM) is the task of tracking and controlling changes in the software. Configuration management practices include revision control and the establishment of baselines.

The goals of SCM are generally:

- Configuration identification – What code are we working with?
- Configuration control – Controlling the release of a product and its changes.
- Status accounting – Recording and reporting the status of components.
- Review – Ensuring completeness and consistency among components.
- Build Management – Managing the process and tools used for builds.
- Process management – Ensuring adherence to the organization's development process.
- Environment management – Managing the software and hardware that host our system.
- Teamwork – Facilitate team interactions related to the process.
- Defect tracking – Making sure every defect has traceability back to the source

(e) Project Schedule

Before a project schedule can be created, a project manager should typically have a work breakdown structure (WBS), an effort estimate for each task, and a resource list with availability for each resource. If these are not yet available, it may be possible to create something that looks like a schedule, but it will essentially be a work of fiction. They can be created using a consensus-driven estimation method like Wideband Delphi. The reason for this is that a schedule itself is an estimate: each date in the schedule is estimated, and if those dates do not have the buy-in of the people who are going to do the work, the schedule will be inaccurate.

In many industries, such as engineering and construction; the development and maintenance of the project schedule is the responsibility of a full time scheduler or team of schedulers, depending on the size

8.8 SOFTWARE MAINTENANCE

In software engineering "software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment." (ISO/IEC 14764). This international standard describes the 6 software maintenance processes as :

- The implementation process contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.
- The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.
- The process considering the implementation of the modification itself.
- The process acceptance of the modification, by checking it with the individual who submitted the request in order to make sure the solution provided a solution.

- The migration process (platform migration, for example) is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.

- Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

✓ The software products required maintenance on account of the following main reasons)

- ✓ (1) **Corrective Maintenance** : Reactive modification of a software product performed after delivery to correct discovered problems
- ✓ (2) **Adaptive Maintenance** : Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
- ✓ (3) **Perfective Maintenance** : Modification of a software product after delivery to improve performance or maintainability.
- ✓ (4) **Preventive Maintenance** : Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.