PROJECT SCHEDULING

❑What is PROJECT SCHEDULING?

❑Why it is important?

❑What are the steps?

❑Basic Concepts.

❑What should we do when management demands that we make a deadline that is impossible?

❑Basic Principles.

❑What are time line chart and its example.

❑How to track OO project?

❑Earned value analysis.

What is PROJECT SCHEDULING?

✓ You’ve selected an appropriate process model.

✓ You’ve identified the software engineering tasks that have to be performed.

✓ You estimated the amount of work and the number of people, you know the deadline, you’ve even considered the risks.

✓ Now it’s time to connect the dots. That is, you have to create a network of software engineering tasks that will enable you to get the job done on time.

✓ Once the network is created, you have to assign responsibility for each task, make sure it gets done, and adapt the network as risks become reality.

Why it’s Important?

- ✓ In order to build a complex system, many software engineering tasks occur in parallel.
- ✓ The result of work performed during one task may have a profound effect on work to be conducted in another task.
- ✓ These interdependencies are very difficult to understand without a schedule.
- ✓ It’s also virtually impossible to assess progress on a moderate or large software project without a detailed schedule.

▪What are the steps?

- ✓ The software engineering tasks dictated by the software process model are refined for the functionality to be built.
- ✓ Effort and duration are allocated to each task and a task network (also called an “activity network”) is created in a manner that enables the software team to meet the delivery deadline established.

Basic Concept of Project Scheduling

✓ An unrealistic deadline established by someone outside the software development group and forced on managers and practitioner's within the group.

✓ Changing customer requirements that are not reflected in schedule changes.

✓ An honest underestimate of the amount of effort and/or the number of resources that will be required to do the job.

✓ Predictable and/or unpredictable risks that were not considered when the project commenced.

✓ Technical difficulties that could not have been foreseen in advance.

Continue....

- ✓ Human difficulties that could not have been foreseen in advance.
- ✓ Miscommunication among project staff that results in delays.
- ✓ A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem.

Why Should we do when the management demands that we make a dead line impossible?

- ✓ Perform a detailed estimate using historical data from past projects.
- ✓ Determine the estimated effort and duration for the project.
- ✓ Using an incremental process model, develop a software engineering strategy that will deliver critical functionality by the imposed deadline, but delay other functionality until later. Document the plan.

Continue….

✓ Meet with the customer and (using the detailed estimate), explain why the imposed deadline is unrealistic.

✓ Be certain to note that all estimates are based on performance on past projects.

✓ Also be certain to indicate the percent improvement that would be required to achieve the deadline as it currently exists.

Basic Principles of Project Scheduling.

1. **Compartmentalization:** The project must be compartmentalized into a number of manageable activities and tasks.
2. **Interdependency:** The interdependency of each compartmentalized activity or task must be determined.
3. **Time allocation:** Each task to be scheduled must be allocated some number of work units (e.g., person-days of effort).
4. **Effort validation:** the project manager must ensure that no more than the allocated number of people have been scheduled at any given time.
5. **Defined responsibilities:** Every task that is scheduled should be assigned to a specific team member.

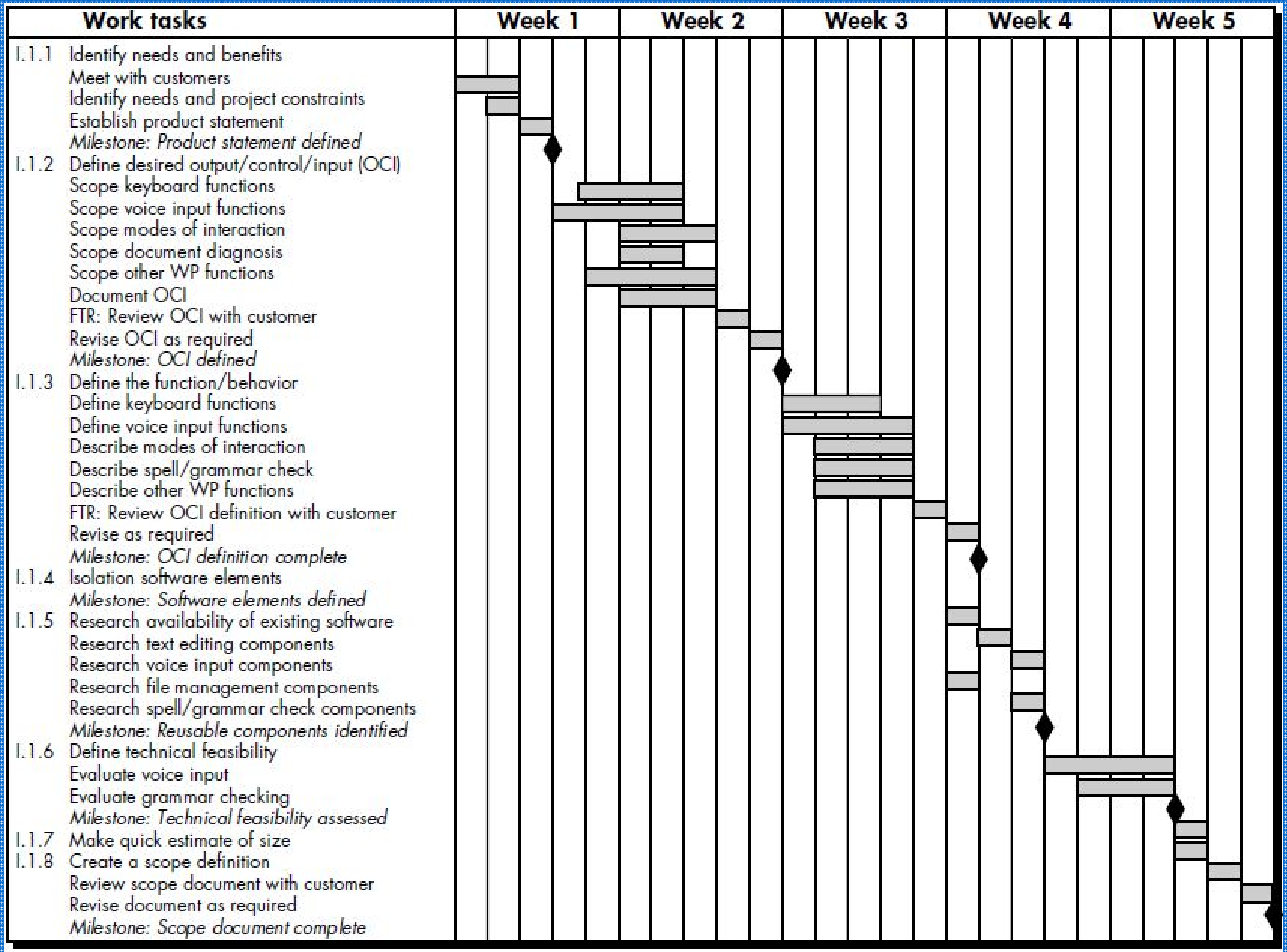
Continue

6. **Defined outcomes:** Every task that is scheduled should have a defined outcome.
7. **Defined milestones:** Every task or group of tasks should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

▪Timeline charts

- ✓ When creating a software project schedule, the planner begins with a set of tasks.
- ✓ If automated tools are used, the work breakdown is input as a task network or task outline.
- ✓ Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals.
- ✓ As a consequence of this input, a ***timeline chart*** is generated also called ***gantt chart***.
- ✓ A timeline chart can be developed for the entire project. Alternatively, separate charts can be developed for each project function or for each individual working on the project.

▪Example for Timeline charts



Work tasks

Week 1 Week 2 Week 3 Week 4 Week 5

I.1.1 Identify needs and benefits
Meet with customers
Identify needs and project constraints
Establish product statement
Milestone: Product statement defined

I.1.2 Define desired output/control/input (OCI)
Scope keyboard functions
Scope voice input functions
Scope modes of interaction
Scope document diagnosis
Scope other WP functions
Document OCI
FTR: Review OCI with customer
Revise OCI as required
Milestone: OCI defined

I.1.3 Define the function/behavior
Define keyboard functions
Define voice input functions
Describe modes of interaction
Describe spell/grammar check
Describe other WP functions
FTR: Review OCI definition with customer
Revise as required
Milestone: OCI definition complete

I.1.4 Isolation software elements
Milestone: Software elements defined

I.1.5 Research availability of existing software
Research text editing components
Research voice input components
Research file management components
Research spell/grammar check components
Milestone: Reusable components identified

I.1.6 Define technical feasibility
Evaluate voice input
Evaluate grammar checking
Milestone: Technical feasibility assessed

I.1.7 Make quick estimate of size

I.1.8 Create a scope definition
Review scope document with customer
Revise document as required
Milestone: Scope document complete

▪How to track OO project?

Technical Milestone: OO Analysis completed

✓ All classes and the class hierarchy have been defined and reviewed.

✓ Class attributes and operations associated with a class have been defined and reviewed.

✓ Class relationships have been establish and reviewed.

✓ A behavioral model has been created and reviewed.

✓ Reusable class has been noted.

▪Continue…

Technical Milestone: OO Design completed

- ✓ The set of subsystems has been defined and reviewed.
- ✓ Classes are allocated to subsystems and reviewed.
- ✓ Task allocation has been establish and reviewed.
- ✓ Responsibilities and collaborations has been identified.
- ✓ Design classes have been created and reviewed.
- ✓ The communication model has been created and reviewed.

- Continue…

Technical Milestone: OO Programming completed

- ✓ Each new class has been implemented in code from the design model.
- ✓ Extracted classes have been implemented.
- ✓ Prototype are increment has been built.

Continue…

Technical Milestone: OO testing

- ✓ The correctness and completeness of OO analysis and design models has been reviewed.
- ✓ A class-responsibility-collabaration has been developed and reviewed.
- ✓ Test cases are designed, and cluster testing is completed and the classes are integrated.
- ✓ System level tests have been completed.

▪Earned value analysis

✓ The *budgeted cost of work scheduled* (BCWS) is determined for each work task represented in the schedule.

✓ The BCWS values for all work tasks are summed to derive the budget at completion, BAC. Hence,

BAC = (BCWS*k*) for all tasks *k*

✓ Next, the value for *budgeted cost of work performed*(BCWP) is computed.

✓ The value for BCWP is the sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.