PROJECT SCHEDULING AND TRACKING

PROJECT SCHEDULING AND TRACKING

- *Software project scheduling* is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks.
- During early stages of project planning, a *macroscopic schedule* is developed.
- This type of schedule identifies all major software engineering activities and the product functions to which they are applied.
- As the project gets under way, each entry on the macroscopic schedule is refined into a *detailed schedule.*

Basic Concept

Why software is delivered late?

- An unrealistic deadline established
- Changing customer requirements that are not reflected in schedule changes.
- 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.
- 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.

What should we do when management demands that make a deadline that is 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 that will deliver critical functionality by the imposed deadline. Document the plan.
- Meet with the customer and (using the detailed estimate), explain why the imposed deadline is unrealistic.
- Offer the incremental development strategy as an alternative

Basic principle

- **Compartmentalization:** The project must be compartmentalized into a number of manageable activities and tasks.
- To accomplish compartmentalization, both the product and the process are decomposed.
- **Interdependency.** The interdependency of each compartmentalized activity or task must be determined.
- Some tasks must occur in sequence while others can occur in parallel.
- Some activities cannot commence until the work product produced by another is available.

Contd.

- **Time allocation.** Each task to be scheduled must be allocated some number of work units (e.g., person-days of effort).
- In addition, each task must be assigned a start date and a completion date that are a function of the interdependencies
- **Effort validation.** As time allocation occurs, the project manager must ensure that no more than the allocated number of people have been scheduled at any given time.
- **Defined responsibilities**. Every task that is scheduled should be assigned to a specific team member.
- **Defined outcomes.** Every task that is scheduled should have a defined outcome. For software projects, the outcome is normally a work product or deliverable.
- **Defined milestones.** A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

Planning and Scheduling

People and Effort

“If we fall behind schedule we can always add more programmers and catch to late in the project”

Has a disruptive effect on the project

Schedules slip even further

Planning and Scheduling

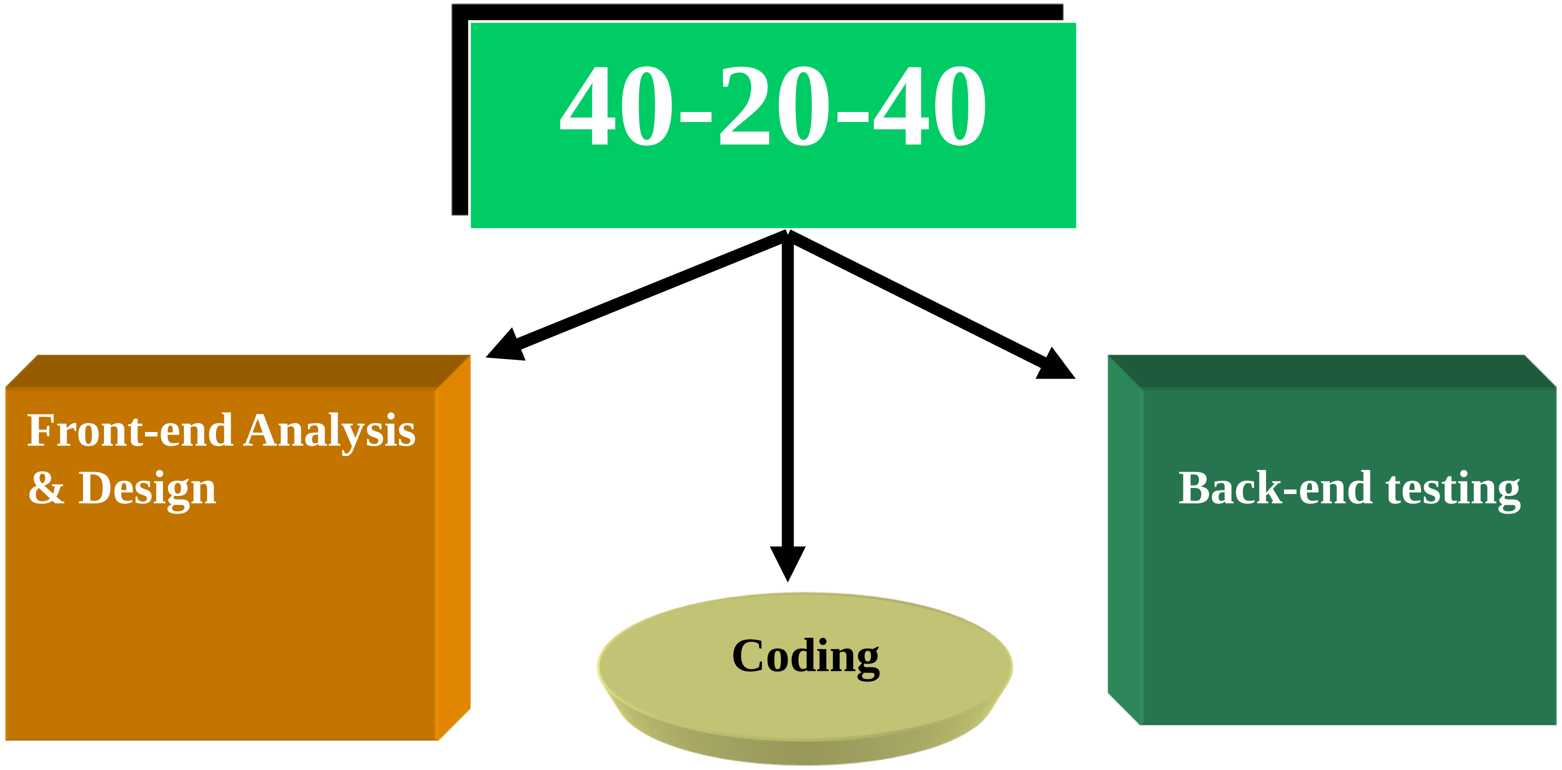
People and Effort

The relationship between the number of people working in software project and overall productivity is

not linear

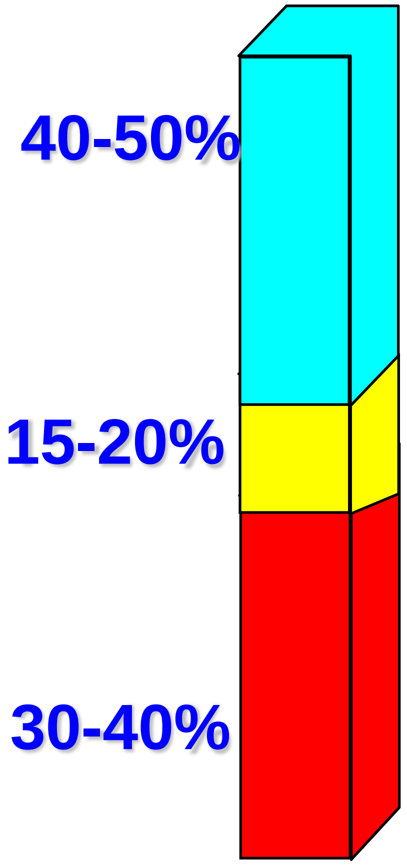
Fewer people and ***longer time period*** is a better option for software development

Effort Distribution



Planning and Scheduling

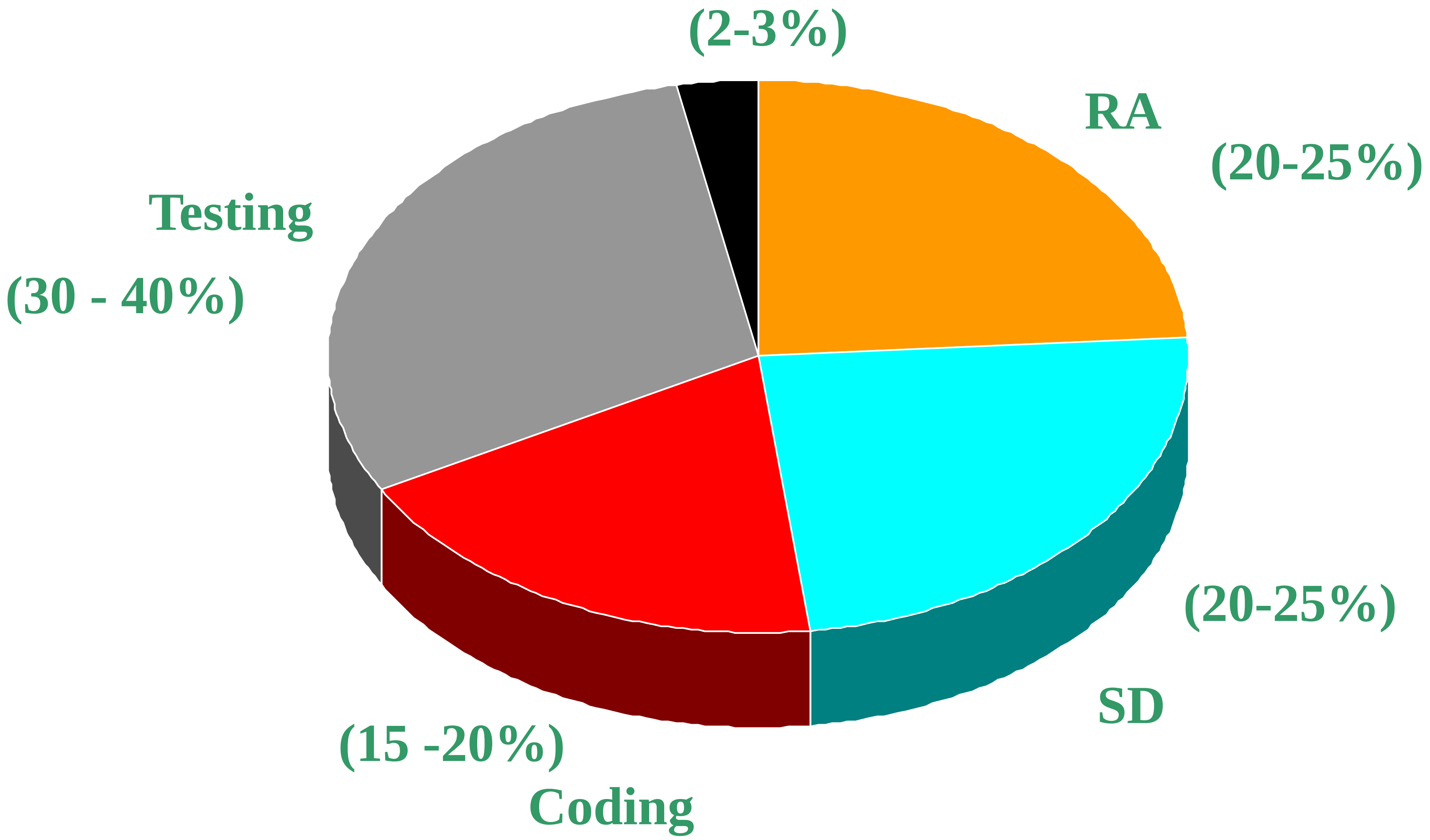
Effort Allocation



- “front end” activities
  - customer communication
  - analysis
  - design
  - review and modification
- construction activities
  - coding or code generation
- testing and installation
  - unit, integration
  - white-box, black box
  - regression

Planning and Scheduling

Effort Distribution



DEFINING A TASK SET

- A *task set* is a collection of software engineering work tasks, milestone, and deliverables that must be accomplished to complete a particular project.
- The task set must provide enough discipline to achieve high software quality. But, at the same time, it must not load the project team with unnecessary work.
- To develop a project schedule, a task set must be distributed on the project timeline. Task set will vary depending upon the project type and the degree of rigor.
- Task sets are designed to accommodate different types of projects and different degrees of rigor.

Contd.

- To define a Task set
  - Determine Project Type
  - Identify adaptation criteria
  - Assess the degree of rigor required
  - Select appropriate software engineering tasks.

- Determine the project type
  - *Concept development projects* that are initiated to explore some new business concept or application of some new technology.
  - *New application development projects* that are undertaken as a consequence of a specific customer request.
  - *Application enhancement projects* that occur when existing software undergoes major modifications to function, performance, or interfaces that are observable by the end-user.
  - *Application maintenance projects* that correct, adapt, or extend existing software in ways that may not be immediately obvious to the end-user.
  - *Reengineering projects* that are undertaken with the intent of rebuilding an existing (legacy) system in whole or in part.

Identify adaptation criteria

- Size of project
- Number of potential users.
- Mission criticality
- Application longevity
- Stability of requirements
- Ease of customer/developer communication
- Maturity of applicable technology
- Performance constraints,
- Project staff
- Reengineering.

- Each of the adaptation criteria is assigned a grade that ranges between 1 and 5, where 1 represents a project in which a small subset of process tasks are required and 5 represents a project in which a complete set of process tasks should be applied

Degree of rigor

- *Adaptation criteria* are used to determine the recommended degree of rigor with which the software process should be applied on a project.
- The degree of rigor is a function of many project characteristics. As an example, small, non-business-critical projects can generally be addressed with somewhat less rigor than large, complex business-critical applications.

Finally, apply software engineering task.

A task set example

- In order to develop a project schedule, a task set must be distributed on the project time line.
- Major software engineering tasks are applicable to all process model flows.
- As an example, we consider the software engineering tasks for a concept development project.

Major Task Set for concept development project are:

- **Concept scoping** determines the overall scope of the project.
- **Preliminary concept planning** establishes the organization’s ability to undertake the work implied by the project scope.
- **Technology risk assessment** evaluates the risk associated with the technology to be implemented as part of project scope.
- **Proof of concept** demonstrates the feasibility of a new technology in the software context.
- **Concept implementation** implements the concept representation in a manner that can be reviewed by a customer and is used for “marketing” purposes when a concept must be sold to other customers or management.
- **Customer reaction to the concept** ask for feedback on a new technology concept and targets specific customer applications.

- The software team must understand what must be done (scoping);
- Then the team (or manager) must determine whether anyone is available to do it (planning),
- Consider the risks associated with the work (risk assessment).
- Prove the technology in some way (proof of concept)
- Implement it in a prototypical manner so that the customer can evaluate it (concept implementation and customer evaluation).
- Finally, if the concept is viable, a production version (translation) must be produced.

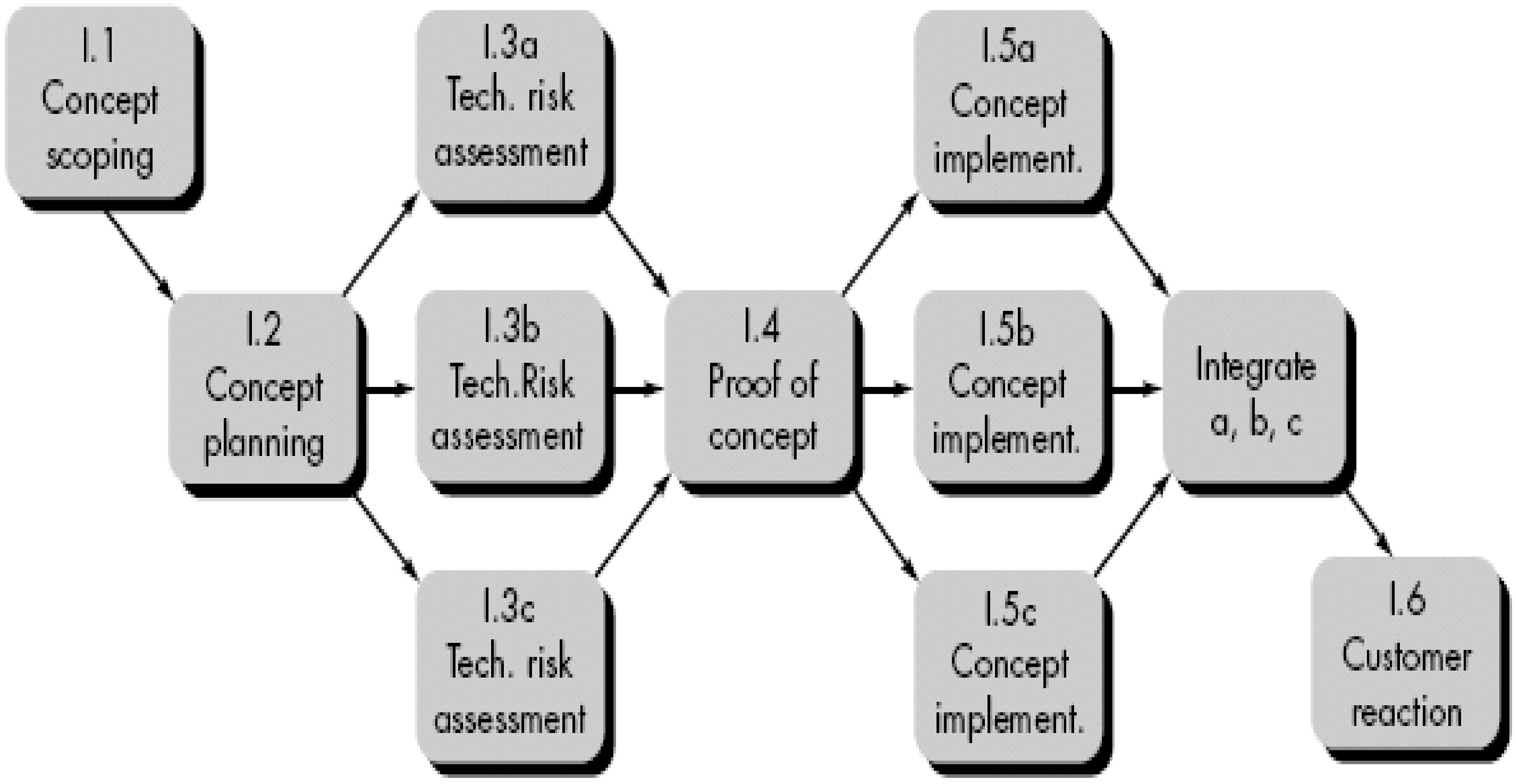
Refinement of Major Task

- Refinement begins by taking each major task and decomposing it into a set of subtasks (with related work products and milestones)
- As an example of task decomposition, consider *concept scoping* for a development Project
- Task refinement can be accomplished using an outline format a process design language approach is used to illustrate the flow of the concept scoping activity

```
Task definition: Task 1.1 Concept Scoping
1.1.1   Identify need, benefits and potential customers;
1.1.2   Define desired output/control and input events that drive the application;
        Begin Task 1.1.2
            1.1.2.1   FTR: Review written description of need[7]
            1.1.2.2   Derive a list of customer visible outputs/inputs
            1.1.2.3   FTR:  Review outputs/inputs with customer and revise as required;
        endtask Task 1.1.2
1.1.3   Define the functionality/behavior for each major function;
        Begin Task 1.1.3
            1.1.3.1   FTR:  Review output and input data objects derived in task 1.1.2;
            1.1.3.2   Derive a model of functions/behaviors;
            1.1.3.3   FTR:  Review functions/behaviors with customer and revise as required;
        endtask Task 1.1.3
1.1.4   Isolate those elements of the technology to be implemented in software;
1.1.5   Research availability of existing software;
1.1.6   Define technical feasibility;
1.1.7   Make quick estimate of size;
1.1.8   Create a Scope Definition;
endTask definition: Task 1.1
```

Defining a Task Network.

- A *task network,* also called an *activity network,* is a graphic representation of the task flow for a project.
- It is sometimes used as the mechanism through which task sequence and dependencies are input to an automated project scheduling tool.
- Therefore, the task network depicts major software engineering tasks.
- Fig. shows a schematic task network for a concept development project.
- The concurrent nature of software engineering activities leads to a number of important scheduling requirements.



Three I.5 tasks are applied in parallel to 3 different concept functions

- Because parallel tasks occur asynchronously, the planner must determine inter-task dependencies to ensure continuous progress toward completion.
- The project manager should be aware of those tasks that lie on the critical path. That is, tasks that must be completed on schedule if the project as a whole is to be completed on schedule.
- In a detailed task network each activity shown in fig.(refinement of Tasks 1.1) would be expanded.

SCHEDULING

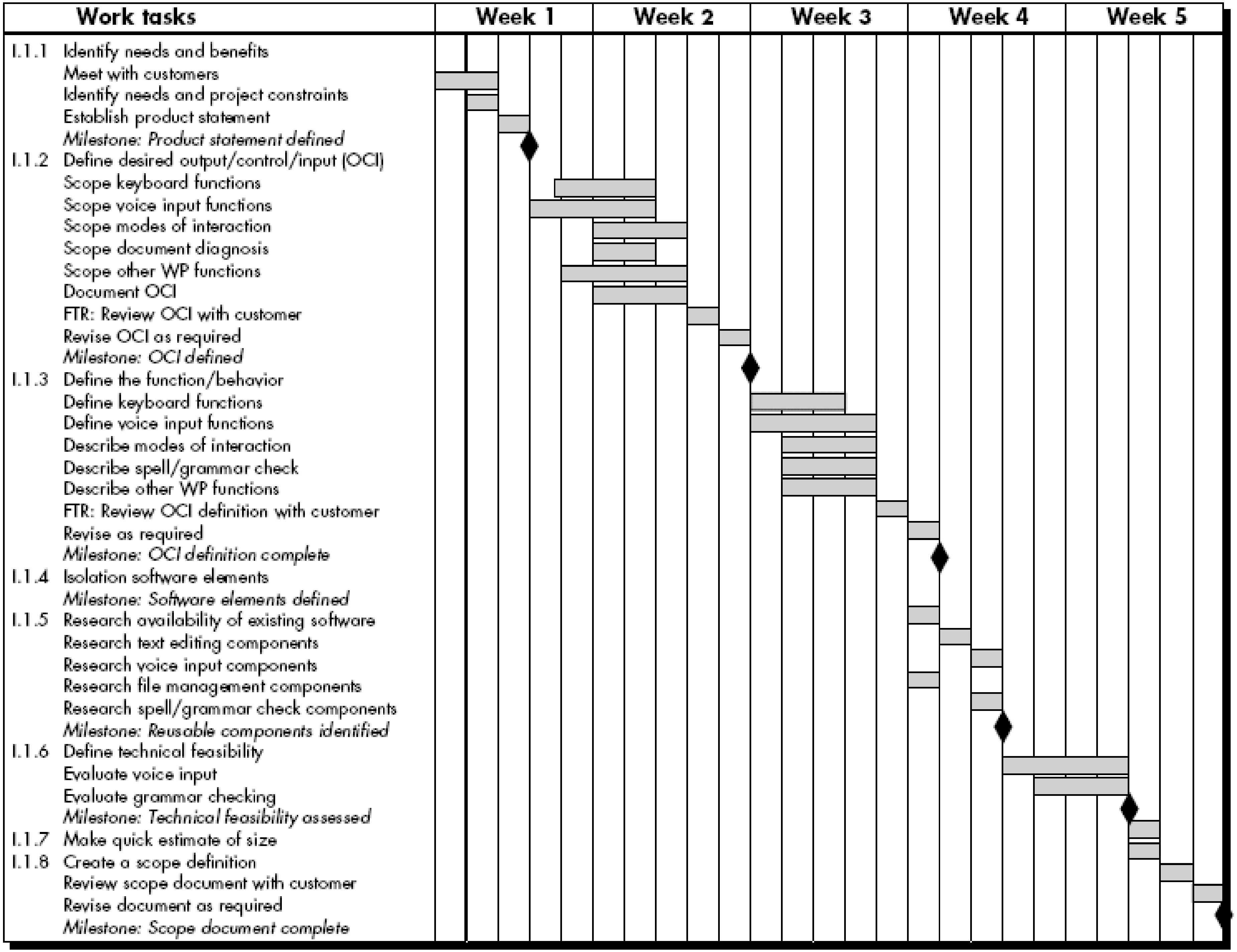
- Two project scheduling methods
  - *Program evaluation and review technique* (PERT)
  - *critical path method* (CPM)
- Both techniques are driven by information already developed in earlier project planning activities:
  - Estimates of effort
  - A decomposition of the product function
  - The selection of the appropriate process model and task set
  - Decomposition of tasks
- Interdependencies among tasks may be defined using a task network.
- Tasks, sometimes called the project *work breakdown structure* (WBS), are defined for the product as a whole or for individual functions.

- PERT and CPM provide quantitative tools that allow the software planner to
  - Determine the *critical path*—the chain of tasks that determines the duration of the project
  - Establish “most likely” time estimates for individual tasks by applying statistical models;
  - Calculate “Boundary times” that define a time "window" for a particular task.
- Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer

  E.g. Primevera software

Timeline Charts

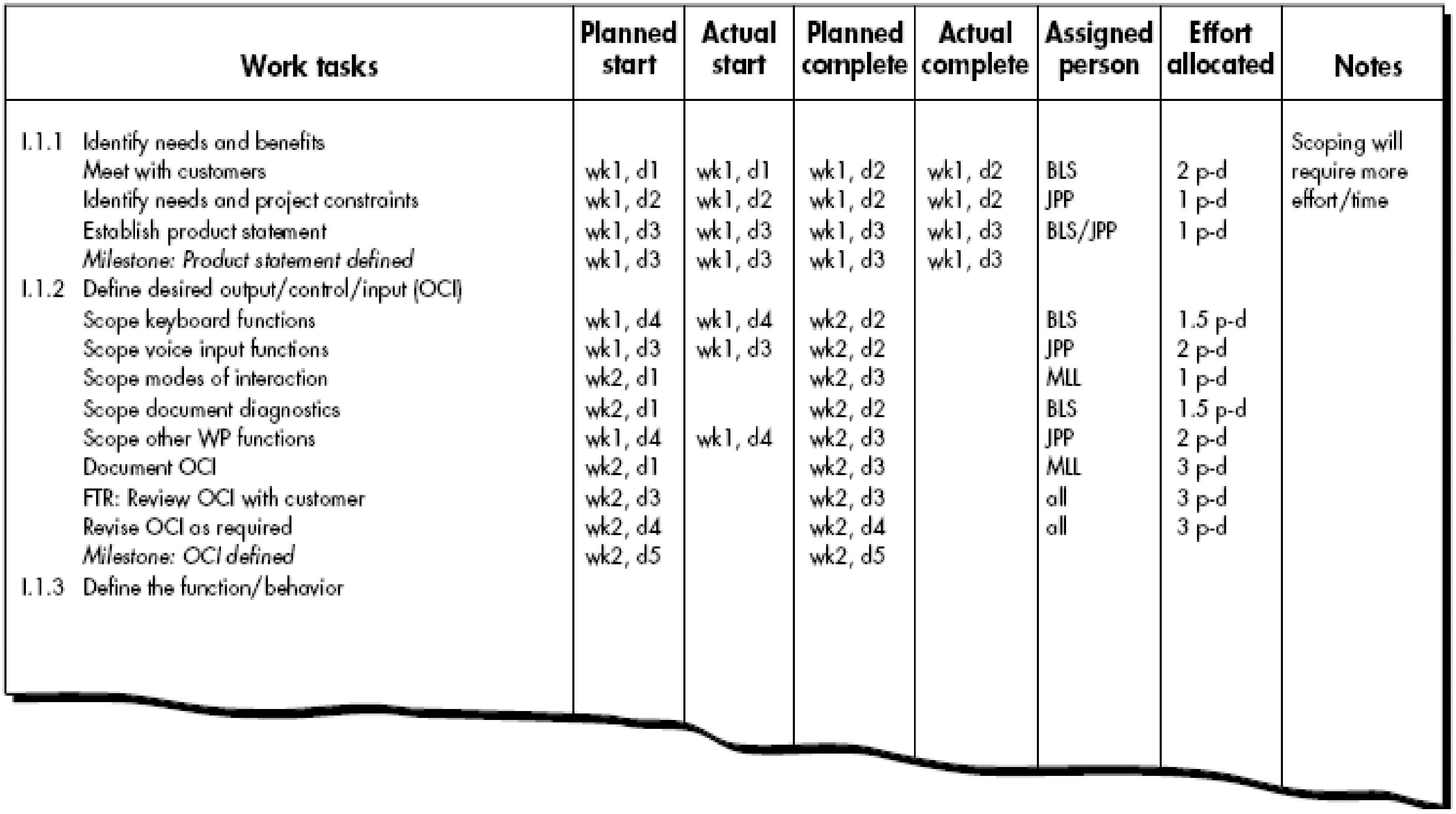
- The planner always begins with a set of tasks (the work breakdown structure).
- If automated tools are used, the work breakdown is input as a task network or task outline.
- Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals.
- A timeline chart enables you to determine what tasks will be conducted at a given point in time.
- 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.


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

Contd.

- All project tasks are listed in the left-hand column.
- The horizontal bars indicate the duration of each task.
- When multiple bars occur at the same time on the calendar, task concurrency is implied.
- The diamonds indicate milestones.
- Once the information necessary for the generation of a timeline chart has been input, the majority of software project scheduling tools produce *project tables*
- It is a tabular listing of all project tasks, their planned and actual start- and end-dates, and a variety of related information.
- Project tables enable the project manager to track progress.

Project Table



Work tasks	Planned start	Actual start	Planned complete	Actual complete	Assigned person	Effort allocated	Notes
I.1.1 Identify needs and benefits							Scoping will
Meet with customers	wk1, d1	wk1, d1	wk1, d2	wk1, d2	BLS	2 p-d	require more
Identify needs and project constraints	wk1, d2	wk1, d2	wk1, d2	wk1, d2	JPP	1 p-d	effort/time
Establish product statement	wk1, d3	wk1, d3	wk1, d3	wk1, d3	BLS/JPP	1 p-d	
Milestone: Product statement defined	wk1, d3	wk1, d3	wk1, d3	wk1, d3			
I.1.2 Define desired output/control/input (OCI)							
Scope keyboard functions	wk1, d4	wk1, d4	wk2, d2		BLS	1.5 p-d	
Scope voice input functions	wk1, d3	wk1, d3	wk2, d2		JPP	2 p-d	
Scope modes of interaction	wk2, d1		wk2, d3		MLL	1 p-d	
Scope document diagnostics	wk2, d1		wk2, d2		BLS	1.5 p-d	
Scope other WP functions	wk1, d4	wk1, d4	wk2, d3		JPP	2 p-d	
Document OCI	wk2, d1		wk2, d3		MLL	3 p-d	
FTR: Review OCI with customer	wk2, d3		wk2, d3		all	3 p-d	
Revise OCI as required	wk2, d4		wk2, d4		all	3 p-d	
Milestone: OCI defined	wk2, d5		wk2, d5				
I.1.3 Define the function/behavior							

Tracking the Schedule

- Project schedule defines the tasks and milestones that must be tracked and controlled as the project proceeds.
- Tracking can be accomplished in a number of different ways:
  - Conducting periodic project status meetings in which each team member reports progress and problems.
  - Evaluating the results of all reviews conducted
  - Determining whether formal project milestones (i.e. diamonds) have been accomplished by the scheduled date.
  - Comparing actual start-date to planned start-date for each project task listed in the resource table.
  - Using earned value analysis to evaluate progress quantitatively.

EARNED VALUE ANALYSIS

- The earned value system provides a common value scale for every task, regardless of the type of work being performed.
- Simply stated, earned value is a measure of progress.
- It enables us to assess the “percent of completeness” of a project using quantitative analysis
- The total hours to do the whole project are estimated, and every task is given an earned value based on its estimated percentage of the total.

To determine the earned value

- The *budgeted cost of work scheduled (BCWS) is determined for each work task* represented in the schedule.
  - BCWS*i is the effort planned for work task i.*
  - To determine progress at a given point along the project schedule, the value of BCWS is the sum of the $BCWS_i$ values for all work tasks for that point in time.
- 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.
  - BCWS - budget of the activities that were planned to be completed
  - BCWP - budget of the activities that actually were completed.”
- Given values for BCWS, BAC, and BCWP, important progress indicators can be computed:
  - Schedule performance index, SPI = BCWP/BCWS
  - Schedule variance, SV = BCWP – BCWS

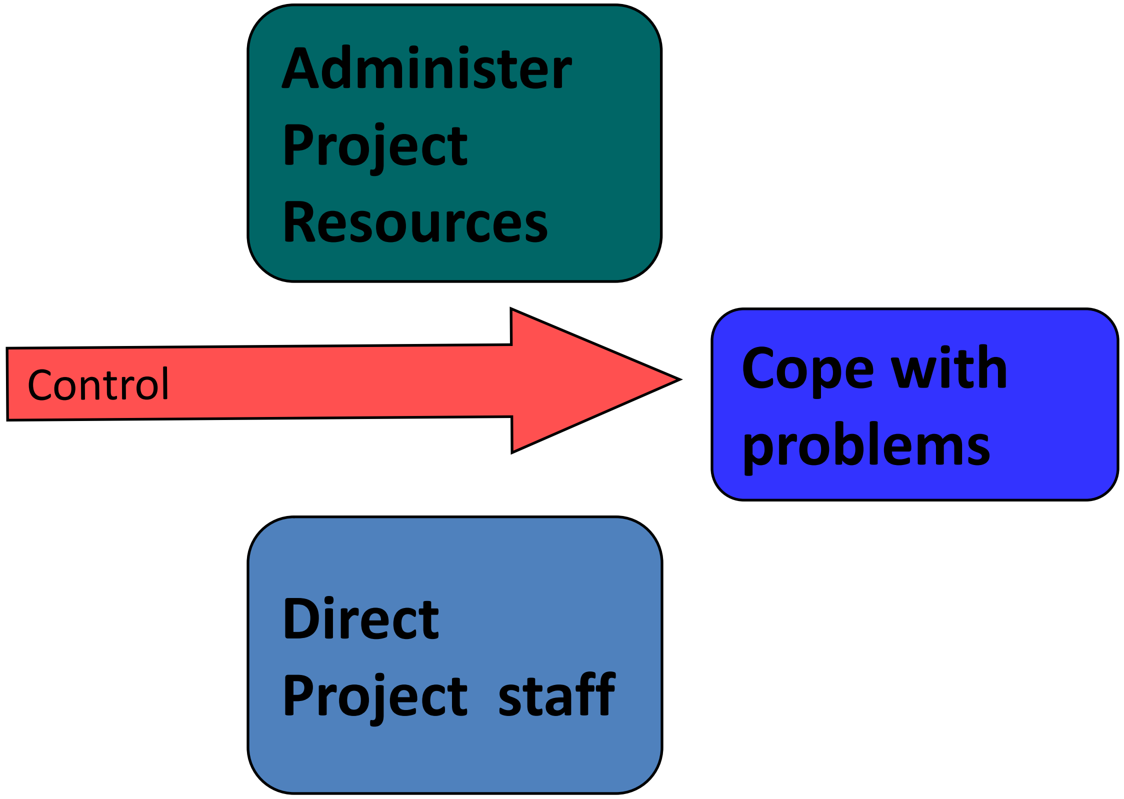
  SPI is an indication of the efficiency with which the project is utilizing scheduled resources.
- SPI value close to 1.0 indicates efficient execution of the project schedule.

- Percent scheduled for completion = BCWS/BAC
  - Indication of the percentage of work that should have been completed by time *t.*
- Percent complete = BCWP/BAC
  - provides a quantitative indication of the percent of completeness of the project at a given point in time, t.
- Actual cost of work performed, ACWP, is the sum of the effort actually expended on work tasks that have been completed by a point in time on the project schedule. It is then possible to compute
  - Cost performance index, CPI = BCWP/ACWP
  - Cost variance, CV = BCWP – ACWP
- CPI value close to 1.0 provides a strong indication that the project is within its defined budget

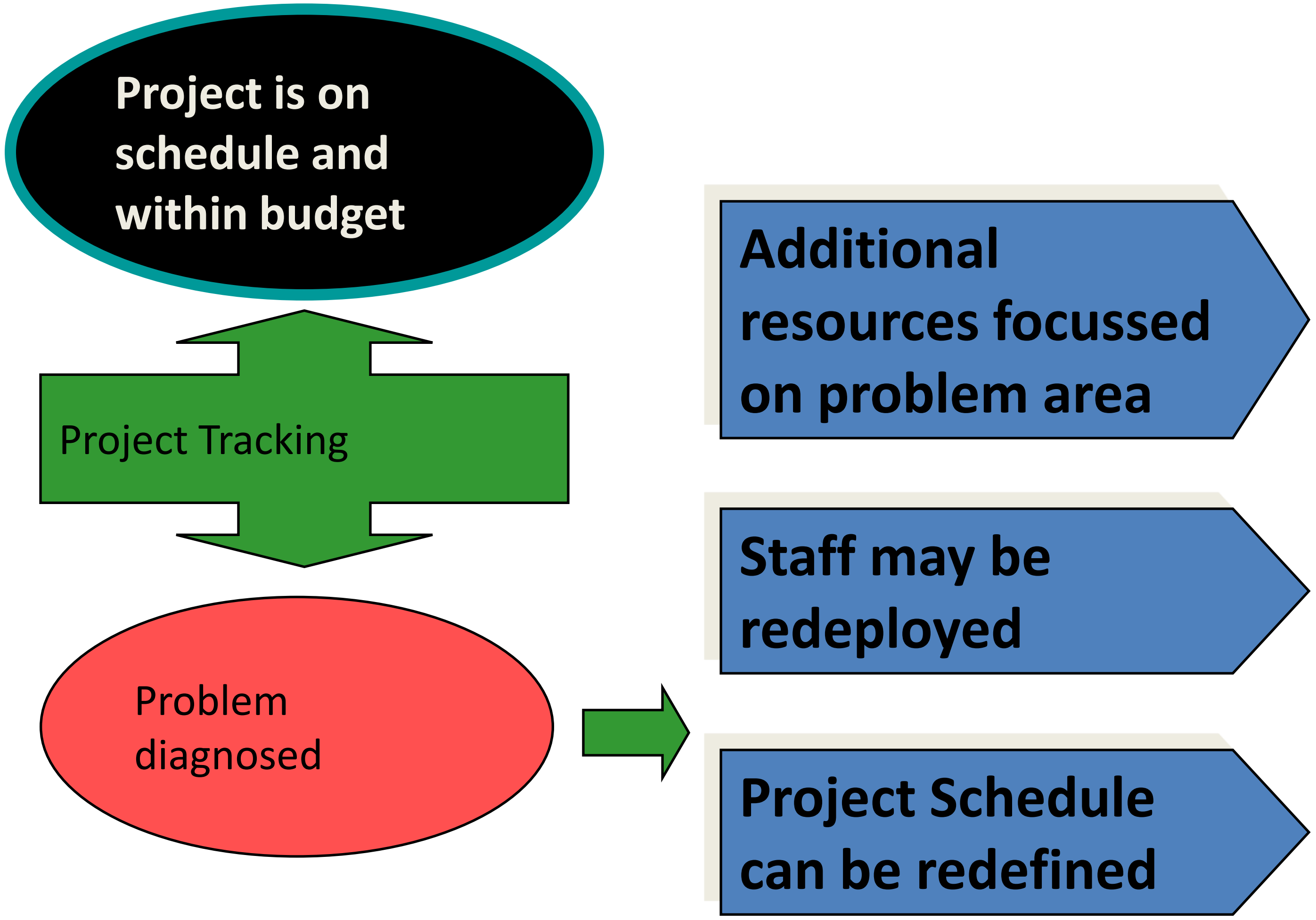
Example :

- Assume you are a software project manager and that you’ve been asked to computer earned value statistics for a small software project. The project has 56 planned work tasks that are estimated to require 582 person-days to complete. At the time that you’ve been asked to do the earned value analysis, 12 tasks have been completed. However, the project schedule indicates that 15 tasks should have been completed. The following scheduling data (in person-days) are available:

Tracking the Project Schedule



Tracking the Project Schedule


Project is on schedule and within budget
Project Tracking
Problem diagnosed
Additional resources focussed on problem area
Staff may be redeployed
Project Schedule can be redefined

Software Project Plan

I.. Introduction

- A. Scope and purpose of document
- B. Project objectives
  - 1. Objectives
  - 2. Major functions
  - 3. Performance issues
  - 4. Management and technical constraints

II. Project estimates

- A. Historical data used for estimates
- B. Estimation techniques
- C. Estimates

III. Project risks

- A. Risk analysis
  - 1. Identification
- B. Risk management
  - 1. Risk aversion options
  - 2. Risk monitoring procedures

IV. Project resources

- A. People
- B. Hardware and Software