Design for Electrical and Computer Engineers

Theory, Concepts, and Practice

Ralph M. Ford and Christopher S. Coulston

This document was prepared with LaTeX.

Contents

Part I - The Engineering Design Process

Chapter 1

Project Management

If you fail to plan, then you plan to fail.—Anonymous

The engineering community has led in the development of project management practices because building complex systems is a tremendous technical and managerial challenge. Currently, businesses tend to organize around projects that have significant value to the organization. Consequently, project management is consistently rated by employers as one of the most desirable skills sought in new college engineering hires [Par03]. The project management field includes topics such as initiating a project, team management, cost management, risk management, controlling, resource management, and performance management, to name a few. Many of these are addressed throughout this book from an engineering design viewpoint such as controlling (design process), initiating (project selection), performance management (requirements and testing), and team management.

The three important objectives of project management are to complete projects that are on-time, within budget, and meet the requirements of the user. Since user requirements were addressed in Chapters **??** and **??**, this chapter addresses the remaining two objectives of time and cost management. Time management introduces the work breakdown structure, which identifies the activities (combined tasks and deliverables) required to complete the project. Responsibility for completing the activities is then assigned to members of the team. Two graphical representations of the work breakdown structure, the network diagram and the Gantt chart, are introduced. These visual depictions show dependencies between tasks and allow for a quantitative analysis of the project plan. The chapter concludes with methods of cost estimation.

Learning Objectives

By the end of this chapter, the reader should:

- Be able to create a work breakdown structure.
- Be able to create network diagrams and Gantt charts.

- Be able to determine the critical path for completing a project and the float time for each activity in the plan.
- Be able to conduct break-even analysis and understand some basic methods of cost estimation.

1.1 The Work Breakdown Structure

The ***work breakdown structure*** (WBS) is a hierarchical breakdown of the tasks and deliverables that need to be completed in order to accomplish the project objectives. Creating the WBS is typically the first step in project planning. A WBS is an ordered set of activities required to complete the project. An ***activity*** is a combination of a task and its associated deliverables. ***Tasks*** are actions that accomplish a job, while ***deliverables*** are entities that are delivered to the project based upon completion of tasks. Examples of deliverables include a circuit design, a software module, the integration and test of modules, a report, a presentation, or obtaining an approval. Example tasks include conducting research or writing a program.

The concept of the WBS was formalized by the United States Military in the 1993 document <u>Work Breakdown Structure Handbook</u> [MIL-HDBK 881]. The WBS has gained wide acceptance in industry and is described in MIL-HDBK 881 as follows:

- *A product-oriented family tree composed of hardware, software, services, data, and facilities. The family tree results from systems engineering efforts.*
- *A WBS displays and defines the product, or products, to be developed and/or produced. It relates the elements of work to be accomplished to each other and to the end product.*
- *A WBS can be expressed down to any level of interest. However the top three levels are as far as any program or contract need go unless the items identified are high cost or high risk. Then, and only then, is it important to take the work breakdown structure to a lower level of definition.*

This description indicates that WBS results from systems engineering efforts and that the structure of the WBS follows the design hierarchy. The second bulleted item focuses on the activities for the project. Gray and Larson [Gra02] recommend identifying the following activity attributes for each activity:

- A definition of the work to be done or delivered.
- A timeframe for completion of the activity.
- Resources needed to complete the activity.
- Person(s) responsible for the activity.
- Predecessors (or dependencies) for the activity. Predecessors are other activities that must be completed before the work can start.
- Checkpoints for monitoring progress.

The collection of activities and their attributes are gathered in the WBS table. The rows of the WBS table represent the project activities. These activities are arranged in a hierarchical fashion; each major activity is followed by its constituent sub-activities. The columns of the

Table 1.1: Example work breakdown structure for the design of a temperature monitoring system.

ID	Activity	Description	Deliverables / Checkpoints	Duration (days)	People	Resources	Predecessors
1	**Interface Circuitry**						
1.1	Design Circuitry	Complete the detailed design and verify it in simulation.	• Circuit schematic • Simulation verification	14	Rob (1) Jana (1)	• PC • SPICE simulator	
1.2	Purchase Components		• Identify parts • Place order • Receive parts	10	Rob		1.1
1.3	**Construct & Test Circuits**	Build and test.					
1.3.1	Current Driver Circuitry	Test of circuit with sensing device.	• Test data • Measurement of linearity	2	Jana (1) Rob (2)	• Test bench • Thermometer	1.2
1.3.2	Level Offset & Gain Circuitry	Test of circuit with voltage inputs.	• Test data • Measurement of linearity	3	Rob (1) Jana (2)	• Test bench	1.2
1.3.3	Integrate Compo-	Integrate the current driver and offset	• Test data • verifying functionality and	5	Rob (1) Jana	• Test bench •	1.3.1 1.3.2

WBS table are the activity attributes proposed by Gray and Larson. Table 1.1 contains the WBS table for the temperature monitoring design examined in Chapter **??** (Section **??**).

Each activity is assigned an identification number (ID) and a name. The hierarchical nature of the activities is reflected in the ID numbering scheme. The three highest-level activities in the project plan are the Interface Circuitry (1), LED and Driver Circuitry (2), and System Integration and Test (3). The first two activities are further refined into sub-activities shown by the indented items. For example, the Interface Circuitry activity contains three sub-activities: Circuit Design (1.1), Purchase Components (1.2), and Construction and Test (1.3). The Construction and Test activity is further subdivided, producing a total of three levels in the hierarchy. In this example, the WBS follows the hierarchical breakdown of the design architecture itself.

Descriptions and deliverables for each activity are provided. The deliverables also serve as checkpoints for monitoring the activity. The identification of checkpoints becomes more important as the complexity and duration of an activity increase. The deliverable items identified for this example include circuit design schematics, simulation results, and test data. Deliverables are even defined for activities that can be somewhat nebulous, such as Research A/D Converters (2.1). Doing so ensures that activities do not become open-ended and have specific deliverables for monitoring their progress.

The fifth column in Table 1.1 is the estimated duration for each activity. The ability to estimate durations is heavily influenced by past experience. The general tendency is to underestimate the amount of time required. Keep in mind that credibility is lost if delivery of the final system is repeatedly delayed, so durations should be estimated as accurately as possible. Take into account time for unexpected problems, such as equipment failure, delayed delivery of parts and equipment, and illness. System integration and testing are tasks that are notoriously time-consuming. A method for estimating activity duration comes from the Project Evaluation Review Technique (PERT) developed by the US Navy in the 1950s. Empirical studies show that durations typically follow a Beta probability density. Based upon this model, the duration of an activity is estimated as

$$t_e = \frac{t_a + 4t_m + t_b}{6} \tag{1.1}$$

where t_a is the most optimistic time estimate, t_b is the most pessimistic time estimate, and t_m is the most realistic time estimate. The advantage of this is that it forces one to consider best and worst-case scenarios in the plan.

The person(s) responsible for each deliverable are identified in the WBS. Teams need to consider how to assign responsibility to ensure mutual accountability is achieved. Assigning one person to an activity provides a single person who is responsible for it. However, what if that person needs additional support or backup? Who is going to help them if they are unable to deliver? What if a team member becomes ill? It may make sense to assign multiple people to activities based upon complexity. In Table 1.1, when a number is placed adjacent to a person's name, it indicates that the person has primary (1) or secondary (2) responsibility for that activity.

The resources needed to complete the activity include material, equipment, and labor. Material includes items that are consumed in creating the deliverable, such as electronic components and printed circuit boards. Examples of equipment include test equipment, computers, and software.

Predecessors for an activity are the activities that must be completed before the given activity can begin. Identifying predecessors is necessary for determining the sequencing of the activities and the time to complete the project.

1.2 Network Diagrams

A ***network diagram*** is a directed graph representation of the activities and dependencies between them for a project. Activity B is dependent on activity A when A must be completed before B can be started. The network diagram allows a graphic visualization of the project that also allows for quantitative analysis. In the ***Activity on Node*** (AON) form, the activities are represented by nodes and the dependencies by arrows. Examples of the basic connections allowed in the AON representation are shown in Figure 1.1.

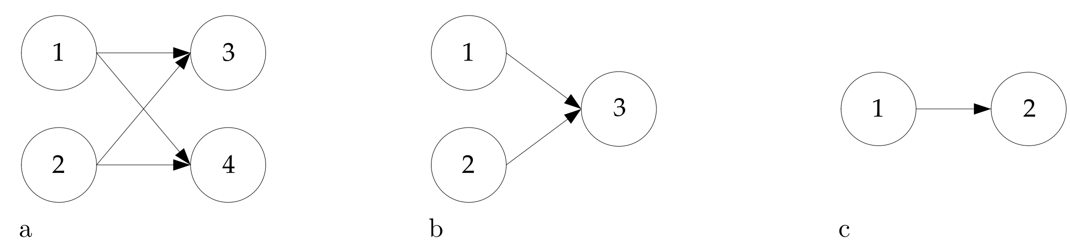


Figure 1.1: Activity on Node (AON) representations of activities. (a) Activity 1 must be completed before activity 2 can begin. (b) Activities 1 and 2 must be completed before activity 3 can begin. (c) Activity 1 and 2 must be completed before activities 3 and 4 can begin.

An example network diagram for a simple project is shown in Figure 1.2. Dummy activity nodes for the start and end of the project have been included. The ID number and estimated duration in days are indicated inside each node. A given network diagram will have multiple paths from the start to the end of the project. A path is any connected sequence of activities from the start node to the end node. In this example, there are four paths to completion: $P_1 = \{1,4\}, P_2 = \{1,5\}, P_3 = \{2,3,4\}, P_4 = \{2,3,5\}$. The completion of an individual path does not result in completion of the project—all paths (and consequently all activities) must be completed for the entire project to finish.

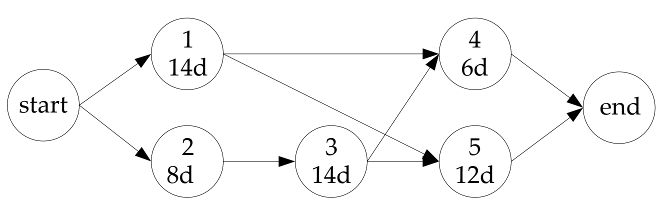


Figure 1.2: Example network diagram. Each activity contains the activity ID and the estimated duration in days.

The duration of each path is determined by summing the duration of all activities on the path, which in this case is 20, 26, 28, and 34 days for paths P_1 to P_4 respectively. The path with the longest duration is known as the ***critical path*** since it represents the minimum time required to complete the project. In this example the critical path is P_4. Activities on the critical path are of particular interest, since, if they fall behind schedule, or experience ***slippage***, the overall completion time of the project is delayed. The other paths can become critical paths if their activities experience sufficient slippage to form a new critical path. A quantity known as ***float*** quantifies this margin. Float is the amount of time an activity can slip without extending the overall completion time of the project. Thus, all activities on the critical path have zero float by definition. The following method is utilized to determine the float for activities:

1. Identify all paths on the network diagram and the duration for each path.

2. The path with the longest duration is the critical path. Label the critical path duration as t_{cp} All activities on the critical path have zero float.

3. To determine the float for an activity that is not on the critical path, find all paths that the activity lies on and identify the one with the longest duration. Label this as t_{lp} The float for the activity is calculated as

$$Float = t_{cp} - t_{lp} \tag{1.2}$$

Examples 1.1 and 1.2 examine float time computation.

Example 1.1 Float time calculation.

Problem: Calculate the float for activities in the network diagram shown in Figure 1.2.

Solution: The first two steps of the process have already been completed, which are the identification of the paths, their duration, and the critical path. To summarize, the paths are $P_1 = \{1, 4\}, P_2 = \{1, 5\}, P_3 = \{2, 3, 4\}, and P_4 = \{2, 3, 5\}$ with durations of 20, 26, 28, and 34 days. The critical path isP_4 and $t_{cp} = 34$ days, so activities 2, 3, and 5 have zero float. Thus

$$Float_{2,3,5} = 0 days$$

The only two activities that are not on the critical path are 1 and 4. Let's examine activity 1 first. It lies on paths P_1 and P_2 durations 20 and 26 days, thus the longest path to completion t_{lp1}, for activity 1 is 26 days. The float is calculated from equation 1.2 as $Float_1 = t_{cp} - t_{lp1} = 34 - 26 days = 8 days$.

Activity 4 lies on P_1 and P_3, so $t_{lp4} = 28$ days and the float is

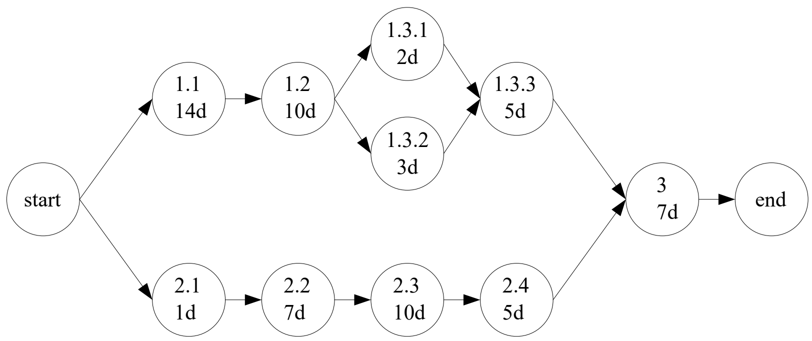
$$Float_4 = t_{cp} - t_{lp4} = 34 - 28 days = 6 days$$

This means that activity 1 can slip by 8 days and activity 4 can slip by 6 days without impacting the time to complete the project.

Example 1.2 Network diagram construction and float time calculation for the temperature display.

Problem: For the example WBS in Table 1.1, (a) create a network diagram, (b) determine the critical path and project completion time, and (c) determine the float for all activities not on the critical path.

Solution: (a) The network diagram is constructed from the dependencies identified in Table 1.1 and is shown below.



(b) The three paths from start to end are: $P_1 = \{1.1, 1.2, 1.3.1, 1.3.3, 3\}, P_2 = \{1.1, 1.2, 1.3.2, 1.3.3, 3\}, and P_3 = \{2.1, 2.2, 2.3, 2.4, 3\}$ which have durations of 38, 39, and 30 days respectively. Thus the critical path is P_2 and its duration is

$$t_{cp} = 39days$$

(c) All activities on path P_1 are also part of the critical path, with the exception of activity 1.3.2, which has

$$Float_{1.3.2} = 39 - 38\text{days} = 1\text{day}$$

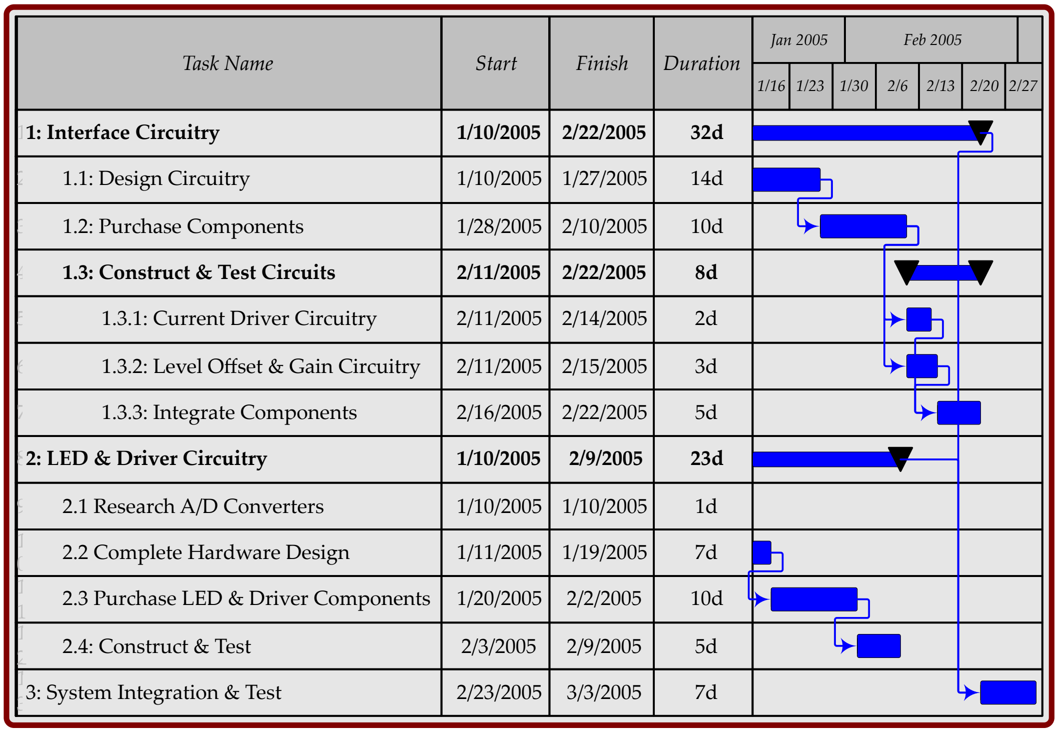
Activities 2.1-2.4 have the collective float

$$Float_{2.1-2.4} = 39 - 30\text{days} = 9\text{days}$$

The strength of the network diagram is that it provides an intuitive graphical representation of activities and their dependencies on one another. This is particularly valuable for complex projects where the paths to completion may not be obvious. It also allows identification of the critical path and float times for activities. A disadvantage of the network diagram is that it may be difficult to encapsulate the amount of information required for an in-depth project on a single page in an easy to read format.

1.3 Gantt Charts

Gantt charts, developed by a mechanical engineer named Henry Gantt (1861-1919), are a bar graph representation of activities on a timeline. An example Gantt chart is shown in Figure 1.3 for the temperature display design. The Gantt chart effectively shows the WBS and the timeline for completion. A traditional weakness of the Gantt chart has been the inability to show the dependencies between activities. However, as seen in this example, this has been remedied by modern project management software where the dependencies are indicated by the connecting arrows between tasks.



Task Name	Start	Finish	Duration
1: Interface Circuitry	**1/10/2005**	**2/22/2005**	**32d**
1.1: Design Circuitry	1/10/2005	1/27/2005	14d
1.2: Purchase Components	1/28/2005	2/10/2005	10d
1.3: Construct & Test Circuits	**2/11/2005**	**2/22/2005**	**8d**
1.3.1: Current Driver Circuitry	2/11/2005	2/14/2005	2d
1.3.2: Level Offset & Gain Circuitry	2/11/2005	2/15/2005	3d
1.3.3: Integrate Components	2/16/2005	2/22/2005	5d
2: LED & Driver Circuitry	**1/10/2005**	**2/9/2005**	**23d**
2.1 Research A/D Converters	1/10/2005	1/10/2005	1d
2.2 Complete Hardware Design	1/11/2005	1/19/2005	7d
2.3 Purchase LED & Driver Components	1/20/2005	2/2/2005	10d
2.4: Construct & Test	2/3/2005	2/9/2005	5d
3: System Integration & Test	2/23/2005	3/3/2005	7d

Figure 1.3: Gantt chart for the temperature display project created using Microsoft Visio™

1.4 Cost Estimation

The second main objective of this chapter is to address how to complete projects within budget. In order to do this, the costs associated with the design, development, and manufacture of the system need to be estimated. This section describes break-even cost analysis and economic considerations followed by methods of cost estimation.

Break-Even Analysis

A break-even analysis aims to determine the number of units that must be sold for costs and revenues to be equal—in other words, for there to be no profit or loss. The two types of costs that factor into this analysis are fixed and variable costs. ***Fixed costs*** are those that are constant regardless of the number of units produced and cannot be directly charged to a process or activity. Examples are rent, overhead, insurance, property taxes, design and development costs, capital expenditures, market research, and sometimes labor costs depending upon the situation. Capital expenditures are costs incurred for long-term assets such as equipment or buildings. ***Variable costs*** vary depending upon the process or items being produced, and fluctuate directly with the number of units produced. Examples are raw materials, inventory, energy costs, and labor costs.

The ***break-even point*** is the point where the number of units sold is such that there is no profit or loss. It is determined from the total costs and revenue. The total cost required to produce a product is the sum of fixed and variable costs. Assuming n units are sold, the total cost is

$$\text{Total cost} = \text{fixed cost} + nx\frac{\text{variable cost}}{\text{unit}} \tag{1.3}$$

The total revenue generated by the sale of the n units is directly related to the sale price

$$\text{Reveneue} = nx\frac{\text{sales price}}{\text{unit}} \tag{1.4}$$

The break-even point is where the revenue and total costs are equal

$$nx\frac{\text{sales price cost}}{\text{unit}} = \text{fixed cost} + nx\frac{\text{variable cost}}{\text{unit}} \tag{1.5}$$

The break-even analysis is shown graphically in Figure 1.4 with the costs and revenue plotted as a function of units sold. Equation 1.5 allows different scenarios to be examined. For example, based upon a certain cost structure and sales price, the volume of sales necessary to break-even can be computed. Or, based upon a cost structure and projected number of units sold, a target sales price can be selected. Example 1.3 demonstrates the application of break-even analysis for the development of the Hewlett-Packard (HP) DeskJet printer.

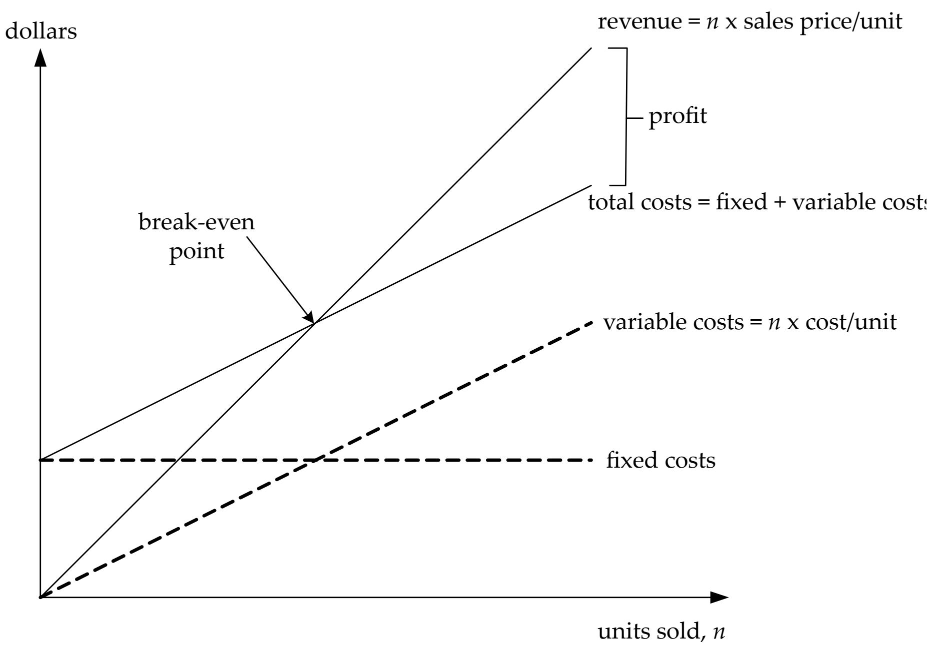


Figure 1.4: Graphical representation of the break-even analysis.

Example 1.3 Break-even analysis for the HP DeskJet printer.

Problem: The following data has been publicly reported for the development and sale of the HP DeskJet [Ulr03]: sales price = \$300, development cost = \$50 million, production investment = \$25 million, annual production (sales) volume = 4 million units per year, and the sales lifetime is 2 years. Assuming a fictitious variable production cost of \$225/unit, determine: (a) the number of units that must be sold to break even, and (b) the profit expected over an estimated sales lifetime of 2 years.

Solution:

(a) The objective is to determine the sales volume, *n,* necessary to break even. The fixed costs are the sum of the development costs and production investment. So

$$\text{Fixed costs} = \$(50 + 25)\text{million} = \$75\text{million}$$

.

This leads to a total cost for n units sold of

$$\text{Total cost} = \$75\text{million} + nx\$225$$

.

The revenue is

$$\text{Revenue} = nx300$$

.

Setting the revenue and total cost each equal at the break-even point produces

$$nx\$400 = \$75\text{million} + nx\$225$$

Solving for the final number of units gives

$$N = 1\text{million}$$

(b) Profit is the differential between the total revenue and the total cost and is expressed as

$$\text{Profit} = \text{total revenue - total costs}$$

For an expected volume of 4 million units per year over 2 years

$$\text{Profit} = nx\$300 - (\$75\text{million} + nx\$225$$
$$= 8\text{million}x\$300 - (\$75\text{million} + 8\text{million}x\$225$$
$$\text{Profit} = \$525\text{million}$$

Cost Models

The costs must be accurately estimated in order to realize the expected profit. The goal here is not to address the complete subject of cost estimation, which is beyond the scope of this book, but instead present some basic concepts and techniques for estimating development costs. Many projects go over budget during development, and this is a particularly important consideration from a design viewpoint. The WBS is a valuable tool for cost estimation because it divides the project into manageable pieces whose individual costs can be more readily estimated.

As identified in the WBS, the development costs associated with a project typically include labor, equipment, and materials. Equipment costs can be determined in a fairly straightforward manner, because many of the equipment needs are known *a priori*. Labor costs are tied directly to the length of the project and are often the largest expense.

Estimates of labor costs are usually based upon past experience and expert opinion. This means asking others to estimate the cost and use it as a guide. The estimation formula in (1) for activity duration can be applied for costs as

$$Cost\frac{cost_a + 4cost_m + cost_b}{6} \tag{1.6}$$

where $cost_a$ is the most optimistic cost estimate, $cost_m$ is the most likely cost estimate, and $cost_b$ is the most pessimistic cost estimate.

A more formal approach for estimating labor costs is to use empirical models that represent a quantification of past experience. The models estimate an output based upon quantifiable inputs related to the design or technology. Example inputs are the number of subsystems, the estimated complexity of a circuit design, or an estimate of the lines of code necessary for a software project. This changes the problem from opinion-based estimation to estimation of a quantity that is presumably easier to find and a better indicator of the cost. The output of the model is the cost or another quantity that is directly related to it, such as the estimated number of person-hours.

The simplest example is a linear model $y = mx + b$ for estimating an output y (cost or person-hours) based upon an input x. For example, IBM modeled software development project costs using the number of lines of code as the input [Jal97] as

$$\text{Effort} = a * KLOC + b \tag{1.7}$$

The output is an estimate of the effort in person-months and the input is the projected number of lines of code, KLOC. KLOC, pronounced "kayloc", equals thousands (kilo) of lines of code. The linear model was found to work well for relatively small development projects with between 4 and 10 KLOC. As the complexity increased, an exponential model was found to be more realistic where

$$\text{Effort} = a(KLOC)^b \tag{1.8}$$

By observation of 60 software development projects at IBM, the exponential model was fit to observed data and the parameters were estimated to be $a = 5.2$ and $b = 0.91$. The KLOC model is applied in Example 1.4.

Example 1.4 Effort estimation using KLOC.

Problem: Consider a software development project that has a team of 10 software development engineers. The team has proposed a design and estimates that it will require 50,000 lines of code to complete the project. The average cost to the company for an engineer is $100,000 per year, including salary, benefits, and overhead. Estimate (a) the time required to the complete the project and (b) the labor costs.

Solution:

(a) Based upon the projected value KLOC, the exponential model in equation 1.8 is most appropriate

$$\text{Effort} = 5.2(50)^{0.91} = 183\text{worker months}$$

Since there are 10 developers on the project, the estimated time is determined by dividing the effort by 10, to produce an estimated time of <u>18.3 months</u>.

(b) The labor costs for development are determined from the number of person-months and the average monthly salary of a development engineer as

$$\text{Labor cost} = 183 \text{ worker-months} x \frac{\$100,000}{year} x \frac{1 \text{ year}}{12 \text{ months}}$$

$$\text{Labor cost} = \$1.53\text{million}$$

The models in equations 1.7 and equ:softwareEffortExp are simplistic in that there is a single input to the estimator. For example, what if there were 100 engineers assigned to the project in Example 10.4? The model indicates that the project would be completed in 1.83 months at the same cost. Common sense dictates that this is unrealistic; this problem is commonly referred to as the "mythical man-month" (person-month). The mythical person-month refers to the fact that just adding more people to the team will not linearly reduce development time. The reality is that many factors impact the costs, which leads to effort estimation based on many inputs. An example of this is the Constructive Cost Model (COCOMO) that is also used in software development. There are different levels of COCOMO that allow increasingly complex model inputs, such as the type of technology employed, the maturity of the technology, the size of the team, the experience of the engineers, and the timeframe required to complete the project. Not only do the models estimate costs, but they can also be used to estimate how many engineers are needed to complete projects of a given complexity. Although the examples cited here are from the software field, the concepts are general and applicable to the development of many technologies.

Empirical cost models can also be developed for the materials necessary for the manufacture of items. Material estimates are often based upon the expected size and types of technologies used in the manufactured part. For example, a cost model for the manufacture of a printed circuit board may have as inputs the size of the board, the number of layers, and the type of technology used (such as through-hole vs. surface mount).

1.5 The Project Manager

Many engineering teams have a project manager responsible for planning and organizing the project. The project manager may take primary responsibility for developing the WBS, the network diagram and Gantt chart, the cost estimates, and the budget. Although the project manager may have primary responsibility for the project plan, all team members should have input and contribute to development of the plan. The project manager should monitor the checkpoints and deliverables against the plan and develop strategies for reacting to slippage in any activities. The plan should be updated as necessary and the changes communicated to all involved. The project manager may also have primary responsibility for the purchasing of materials and controlling spending. Project managers are not necessarily the boss in the traditional sense and should be viewed as member of the team. As such, it is important for the project manager to also be responsible for completing project deliverables in addition to the project management tasks.

1.6 Guidance

The following is guidance to consider when creating the project management plan, and is particularly relevant for those working on capstone design projects:

- *Build the plan after the design architecture is complete.* The project plan can be created at any point in the design process. Our experience shows that a good time to develop it is after the system design architecture is complete. The design serves as a good guide for developing the WBS.

- *Take the initial time estimates for activities and double them!* Most people tend to significantly underestimate the amount of time it takes to complete an activity. That is because people often have a conceptual idea of what it will take to complete the task and can envision the steps to completion. The desire to please superiors also influences people to underestimate the time to completion. Although it may not be necessary to double the time estimates, you can incorporate the most optimistic and pessimistic estimates and apply Equation 1.1 Or, if a proven mathematical model exists, such as the KLOC estimator in Equations 1.7 and 1.8, it can be used to estimate times.

- *Assign a lot of time for testing and integration.* During integration many people must work together to integrate components that may have been developed in isolation. Problems with a single component can bring the integration to a halt. Delays may be compounded by necessary re-design to correct the problem.

- *Factor in lead times for part ordering.* Even with the Internet and overnight delivery, you may find that needed parts and equipment are out-of-stock. Lead times for seemingly commonplace items can sometimes be quite lengthy.

- *Assign a project manager(s).* Consider assigning one individual who has primary responsibility for organizing and monitoring the plan. Again, the project manager must also be responsible for some of the deliverables for completion of the project.

- *Do not assign all team members to all tasks.* Experience shows that when this is the case nobody is responsible for anything and the work doesn't get done. There needs to be individual accountability for all team members. However, it may be a good idea to have more than one person responsible for activities for backup support as shown in the WBS in Table 1.1.

- *Track the progress versus the plan.* There is a tendency to create the plan and then ignore it. The plan is only valuable if it is monitored and progress is tracked.

- *Don't become a slave to the plan.* Circumstances usually dictate change. Be prepared to shift resources as needed. Monitor the plan to see if there are changes to the critical path or if a new critical path emerges.

- *Experience counts.* Get started now in developing this experience by creating a plan for your project.

1.7 Project Application: The Project Plan

A project plan should contain the following:

- *Work Breakdown Structure.* Identify the activities, deliverables, responsibilities, duration, resources, and dependencies as demonstrated in

  Table 1.1. Be sure to provide sufficient detail in the structure and identify clear deliverables.

- *Gantt Chart and/or Network Diagram.* Provide a graphical representation of the project plan. Network diagrams have the advantage of showing the dependencies, while Gantt charts show the timeframe. Modern software tools allow both to be integrated into the same graph and are a good compromise as demonstrated in Figure 1.3. The critical path should be identified and the float for non-critical path activities understood.

- *Costs.* Develop a tabulated list of costs and for the equipment, materials, and labor necessary to carry out the project. It may not be necessary to develop labor costs for a capstone project, yet it is good practice to estimate person-hours and compare the estimate to the actual at the end of the project.

1.8 Summary and Further Reading

Three main objectives of project management are to complete projects that are on-time, within budget, and meet the needs of the user. This chapter addressed the time and budget aspects. The key element in developing a project plan is the WBS, which is a hierarchical identification of the activities needed to complete the project. Both network diagrams and Gantt charts can be created from the WBS. A network diagram is a graphical representation of activities and their dependencies that provides for quantitative analysis of the project plan. This analysis includes computation of the critical path and float times. The Gantt chart is related to the network diagram, but provides a time scale representation of the activities. In terms of the budget issues, a simple profit and loss model was presented with a break-even analysis. Model-based techniques can be used for estimating labor costs, where the models are built from the analysis of similar projects. The models can be linear or nonlinear with single inputs. More complex models can be developed with multiple inputs in order to arrive at a more precise estimate.

Project management is a well developed field and there are many good textbooks and online resources available for delving deeper into the subject. Project Management by Gray and Larson [Gra02] is a comprehensive text on the subject that includes risk management, resource scheduling, leadership, and performance measurement.
Planning, Performing, and Controlling Projects by Angus et al. [Ang00] is written for an engineering and scientific audience, and integrates phases of the design process. MindTools (www.mindtools.com) is an online resource that addresses many career skills including project management.

1.9 Problems

1. In your own words, describe what is meant by the work breakdown structure.

2. Consider the set of activities, duration (in days), and predecessors for a project given below.

Activity	A	B	C	D	E	F	G	H	I
Duration	3	9	6	6	6	3	2	6	7
Predecessors	-	-	-	A,B	D,B	C	F,E	G	F

   a) Develop a network diagram representation for the project.
   b) Determine the critical path.
   c) Determine the float time for all activities that are not on the critical path.

3. Consider the set of activities, duration (in days), and predecessors for a project given below.

Activity	A	B	C	D	E	F	G	H	I	J	K
Duration	9	12	3	4	5	9	8	3	6	9	1
Predecessors	-	A	A	B,C	C	B	D	F,D	G	H,I	E

   a) Develop a network diagram representation for the project.
   b) Determine the critical path.
   c) Determine the float time for all activities that are not on the critical path.

4. Explain why a network diagram cannot contain cycles. A cycle is a sequence of activities where you can travel back to an activity already visited.

5. Describe the advantages and disadvantages of the network diagram and Gantt chart representations for a project.

6. Assume that the following data has been determined for the development and sale of a new digital thermometer for home use: development cost = \$250,000, production investment = \$500,000, annual production volume = 20,000 units per year, and the sales lifetime is 7 years. Assuming a variable production cost of \$5 per unit, determine: (a) the sales price necessary to break even within 2 years, and (b) the profit expected over the estimated sales lifetime.

7. Describe the difference between the cost estimation models in Equations 1.7 and 1.8, and the COCOMO cost estimation model.

8. Consider a software development project that has a team of 50 software development engineers. The team has proposed a design and estimates that it will require 500,000 lines of code to complete the project. The average cost to the company for an engineer is \$90,000 per year, including salary, benefits, and overhead. Estimate (a) the time required to complete the project, and (b) the labor costs.

9. Consider a software development project where the team has proposed a design and estimates that it will require 200,000 lines of code to complete. The average cost to the company for an engineer is $110,000 per year, including salary, benefits, and overhead. Estimate (a) the number of engineers needed to complete the project within 18 months, and (b) the labor costs.

10. **Project Application.** Develop a project plan for your project. A format and guideline for developing the plan is contained in Section 1.7.