Chapter 2: Project Selection and Management

Answer to Your Turn 2-1: To Select or Not to Select

Student answers will vary based on knowledge and experience. The response will more than likely describe a project that, though it would seem to be a perfect choice, did not fit into the organizational portfolio of projects. This is an appropriate time to discuss the topic of budgets which organizations must live within.

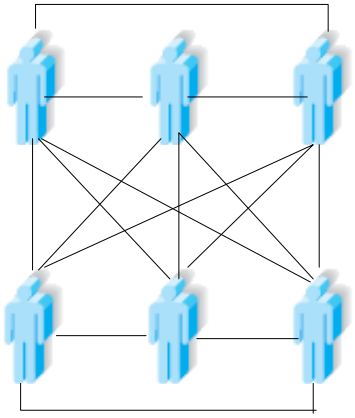
Answer to Your Turn 2-2: Selecting a Methodology

Throwaway prototyping would be a good choice for this scenario for a number of reasons. First, this is a brand new idea, so there may be some ambiguity or confusion as to the functionality of the system. Second, there are technical issues associated with integrating existing hardware and software due to the diversity at different locations around the world. Third, the time frame to delivery is one year.

The time frame would allow for an in-depth analysis to gather information and develop ideas for the system before the design phase. Once the initial requirements were documented, a series of design prototypes can be created, distributed and tested to determine whether issues dealing with functionality or technical problems have been addressed. Once the issues have been resolved, the project can move into design and implementation. Direct students to FIGURE 2-9 as it allows students to compare and contrast the methodologies with the selection criteria discussed in the chapter.

Answer to Your Turn 2-3: Communication Complexity

1. For a six member team, there are 15 communication channels.



For an eight member team, there are 28 channels. A good formula for figuring out how many channels there are is: [N * (N-1)] / 2, where N is the number of team members.

2. Answers will vary, but the larger the team the larger number of communication channels and the more complex communications will be.

Answer to Your Turn 2-4: CASE Tool Analysis

Student responses will vary based on the CASE tool selected. As a rule, CASE tools used during the analysis phase for creating integrated diagrams of the system and to store information regarding system components are referred to as **upper CASE**, whereas CASE tools used during the design phase to create diagrams and associated code for system functionality and databases are referred to as **lower CASE**.

Answer to Concepts in Action 2-A: Project Portfolio Management

Just a note that it an excellent idea to discuss the sheer size of the number of proposals and funded projects that a large organization fields on an annual basis. This stresses the importance of quality work with the System Request Form and that analysts must be adequately prepared in such a competitive area.

Answer to Concepts in Action 2-B, 2-C: Interview with CIO

Having your university or college CIO be a guest in class to answer questions from the students would be a great idea. Putting a name and a face on this very important position is an excellent thing to do.

Answer to Concepts in Action 2-D: Agile Development at Travelers

1. The projects certainly could be done with JAD sessions or with having the users review the project on a weekly basis. However, the success of an agile development depends upon the cohesiveness of the team, typically accomplished by having users meet and devote all their time to the project so that they are able to communicate ideas, comments, suggestions, and providing feedback immediately.

2. In order to work on an agile development project, an analyst needs to be highly motivated, dedicated to the project, able to communicate effectively, and work well in a high-paced team environment.

Answer to Concepts in Action 2-E: Where Agile Works and Doesn't Work

Chances are the agile methodology discussion is growing rapidly on your university or college. You may already have curriculum dedicated to Agile. Changing corporate culture of system development is difficult. This would be an opportune time to discuss how these changes will affect the development teams.

Answer to Concepts in Action 2-F: Trade-Offs

1. In considering the trade-offs among the factors of functionality, time and money, it's important to realize that a change in one of these factors affects the other two. Increasing functionality will most certainly increase the time and money needed to complete the project. Decreasing time or money will most certainly affect either the level or quality of the functionality in the finished project.

2. The project was considered successful from a 'product' standpoint, in that it met the business needs for which it was meant. However, once the time frame had been established, the project manager might have gone back to the clients and reworked the requirements, or the scope of the project. This would have provided the development team with realistic goals and perhaps would have produced a product that could have been used more than once.

Answer to Concepts in Action 2- G: Poor Naming Standards

Student answers will vary depending upon their programming experience, syntax language, and code comments. This is a great time to reinforce the concept of standards.

Solutions to End of Chapter Questions

1. *Describe how projects are selected in organizations.*

   An organization will base the selection of a project on an evaluation not only of the project's costs and expected benefits, but also the technical and organizational risks that have been identified for the project. Both the systems request and feasibility analysis can provide that type of information.

2. *Describe how project portfolio management is used by IT departments.*

   Portfolio management is the process of determining the mix of project types that will maximize benefit to the organization, while minimizing risks and costs to the organization. Included in portfolio management is the process by which an organization assesses how potential projects fit into the existing projects currently underway. Ideally, the organizational project portfolio consists of both high and low risk projects, large and small projects, etc.

3. *Describe the major elements and issues with waterfall development.*

   Waterfall development follows the phases of the life cycle in sequence (planning, analysis, design, and implementation). Each phase is thoroughly documented and approval is required before proceeding to the subsequent phase. It is difficult, though not impossible, to go backwards in the SDLC under waterfall development.

Waterfall development requires that the system requirements be precisely specified prior to implementation, and also often "freezes" those requirements during development. The high degree of effort devoted to specifying user requirements is a strength of waterfall development but specifying those requirements on paper is laborious and may lead to errors and omissions. "Freezing" the requirements during development helps assure that the system is developed according to specifications, but in a dynamic business environment, the system that is ultimately developed may bear little resemblance to what is actually needed at the time the project is completed. Therefore, extensive maintenance may be needed after implementation to revise the system to meet current conditions.

4. *Describe the major elements and issues with parallel development.*

Parallel development modifies the SDLC by altering the design and implementation phases. In parallel development, general design of the entire system is performed; then, the project is divided into sub-projects, each of which is designed in detail and implemented. Work on the sub-projects occurs simultaneously in an effort to reduce the time between analysis and delivery of the system. After all sub-projects are complete, the pieces are integrated into the final delivered system.

The total time to deliver the system can be reduced using parallel development as compared to waterfall development. However, there can be significant challenges in integrating the sub-projects since design decisions made in one sub-project may affect other sub-projects if they are not completely independent.

5. *Describe the major elements and issues with the V-model.*

The V-Model of development follows a waterfall approach in that each phase is completed before the next is begun. The V-Model differs in that during each phase of development not only are requirements specified and components designed, but the testing of those components is also defined. This allows for a well-defined testing plan which can improve the overall quality of the system.

As with waterfall development, the V-Model of development is rigid, and may not be a good choice for a project that is dynamic in nature.

6. *Describe the major elements and issues with iterative development.*

Iterative development is characterized by multiple versions of the overall project. The initial version contains the most important and fundamental requirements. This version can be accomplished with a mini-waterfall process. Once the version is complete, feedback is solicited as to the appropriateness of the system. The project then goes into the next version of the project, incorporating feedback that was received. The benefit to this is that an abbreviated version is quickly available for review.

The disadvantage to iterative development is that the early versions are abbreviated. Customers must understand that each version will have additional functionality until the final version is delivered.

7. *Describe the major elements and issues with prototyping.*

   Prototyping collapses analysis, design, and implementation into one phase, and this phase is cycled through several times until the system meets user needs. A "rough" version of the system is developed early and rapidly, and user feedback is solicited. Based on user comments, the system is modified and refined repeatedly. Ultimately, the system will be refined to the point where the users accept it and it can be put into production.

   Prototyping does provide rapid delivery of a system to the users, and user involvement in determining refinements helps assure a good fit with business needs. The haste to get "something" to the users quickly, however, may lead to poor initial specification of the prototype. Subsequently, substantial revisions may be needed to correct for inadequate initial analysis.

8. *Describe the major elements and issues with throwaway prototyping.*

   While similar to prototyping, throwaway prototyping differs in that the analysis phase is not collapsed, but is developed in full. Once complete, any ambiguous features or challenging design issues may be addressed using a *design prototype.* A design prototype is created to enable users to understand the issues under consideration. Many design prototypes may be built to minimize the risk associated with the system by confirming that important issues are understood before the actual system is built.

   Throwaway prototyping can result in a longer time frame to delivery due to the fact that the prototypes are not necessarily used for the final design. However, this methodology creates an opportunity to address and resolve key issues that arise before getting to the development stage.

9. *Describe the major elements and issues with agile development.*

   Agile development is a methodology in which the SDLC is streamlined. Much of the modeling and documentation is eliminated and the emphasis is placed on simple, iterative development in which each iteration is a small project on its own, complete with planning, requirements analysis, design, coding, and testing.

   Agile development is dependent upon a team that is highly motivated, cohesive, stable, and experienced. This type of development works best with small projects. If the project is not small, or the teams do not work well, then the likelihood of success is reduced.

10. Compare and contrast structured design methodologies in general to rapid application design (RAD) methodologies in general.

Structured design methodologies are usually fairly formal, step-by-step approaches to systems development. The project moves through the phases in a systematic way. The emphasis in most of these methodologies is development of paper-based specifications for the new system prior to implementation. RAD methodologies, on the other hand, tend to emphasize quick creation of a limited-capability version of the system or a model of the system. These methodologies focus on refining this preliminary system or model rather than trying to fully describe it on paper prior to implementation.

11. Compare and contrast extreme programming and throwaway prototyping.

Extreme programming uses continuous testing, simple coding performed by pairs of developers, and close interactions with end users. System functionality grows over time iteratively after a very superficial planning phase. Code is thoroughly tested as it is added to the system. By practicing discipline and keeping the development team small and focused, extreme programming can create good results very quickly. Throwaway prototyping focuses more on experimentation during the design phase. The prototypes are intended to clarify technical issues before the 'real' system is built. The prototypes will be discarded and do not become the final system.

12. What are the key factors in selecting a methodology?

How well are user requirements understood?
How familiar are the systems developers with the technology being employed?
How complex is the system?
How essential is system reliability?
How demanding is the time frame/project schedule?
How visible is the project schedule? (see FIGURE 2-9)

13. Why do many projects end up having unreasonable deadlines? How should a project manager react to unreasonable demands?

Unreasonable deadlines are often the consequence of trying to complete the project to accomplish some business goal rather than being based on a realistic assessment of how long the project will actually take to complete. For example, in the CD Selections case, the project sponsor wants the Internet Order System to be operational in time to sell CDs for holiday shopping. Too often such external factors are used to create target dates for project completion. The project manager must develop accurate and realistic time estimates for the project, and use these to convince the sponsor that his/her timelines can’t be achieved. The project manager is setting the project team up to fail if he/she goes along with a time frame that is known to be unachievable. If the time deadline is immovable, then the project manager should

employ timeboxing to negotiate a narrowed project scope that will be achievable in the time allotted.

14. Name two ways to identify the tasks that need to be accomplished over the course of a project.

One way to identify tasks is to follow a top-down approach, defining high level tasks first, and then breaking those tasks down into subtasks. A second way is to follow a methodology, which will provide a standard list of tasks for the SDLC. This standard task list can be modified as needed for the specific project.

15. What is the difference between a methodology and a work plan? How are the two terms related?

A methodology provides a standard, formalized list of steps and deliverables for the SDLC. Work plans include the specific steps, tasks, and deliverables that will be needed for a specific project. Methodologies form the basis for creating work plans; the project manager can select the parts of the methodology that apply to the specific project, and use that in defining the appropriate steps and tasks.

16. Some companies hire consulting firms to develop the initial project plans and manage the project, but use their own analysts and programmers to develop the system. Why do you think some companies do this?

It is possible that a company does not have skilled project managers available on its in-house staff. Project management requires a unique set of skills, and consulting firms may develop project management capabilities because of the many projects undertaken. It may also be useful to have an 'outsider' serving as project manager, since he or she may be better able to assess the project objectively and resist scope creep more effectively than an in-house employee.

17. Describe the differences between a technical lead and a functional lead? How are they similar?

The technical lead is typically a project team member who supervises the programmers and more technically-oriented project staff. The functional lead is a team member who oversees the systems and business analysts on the team. Both positions report to the project manager, and are responsible for managing, controlling, and coordinating the work of their assigned team members.

18. Describe three technical skills and three interpersonal skills that would be very important to have on any project.

Desirable technical skills might include programming experience in the chosen programming language, experience in configuring the hardware and communications technology platform correctly, and experience in utilizing the file/database

environment effectively. Desirable interpersonal skills might include interviewing skills, negotiation skills, and conflict resolution skills.

19. What are the best ways to motivate a team? What are the worst ways?

Research has shown that technically-oriented people are motivated by recognition, achievement, the work itself, responsibility, advancement, and the chance to learn new skills. The worst ways to motivate technical staff include setting unrealistic deadlines, failing to recognize good effort, accepting low quality output, rewarding all team members monetarily regardless of work quality, failing to include team members in important project decisions, and providing poor working conditions.

20. List three techniques to reduce conflict.

Clearly define the roles on the project, hold team members accountable for their assigned tasks, develop detailed operating procedures and make sure the team members understand them, have each team member commit to the project charter.

21. What is the difference between upper CASE and lower CASE?

Upper CASE refers to diagramming and other tools that are commonly used during the Analysis phase of the SDLC. Lower CASE refers to diagramming and prototyping tools and code generators that are used primarily to support the Design phase of the SDLC. These two components of CASE products are targeted at different segments of the life cycle.

22. Describe three types of standards, and provide examples of each.

Coding standards define the content and structures that are to be used in programs. An example would be that all programs are to be written following structured programming guidelines. Procedural standards define processes that are to be followed by all team members. An example would be required attendance at a weekly team progress meeting, and required honest progress reporting at that meeting. User interface design standards create a common understanding of the appearance and functioning of the screens the end users see. An example would be to create a standard group of icons that are used consistently on all screens.

23. What belongs in the electronic project binder? How is the electronic project binder organized?

All project deliverables, all internal communication, and all project documentation should be placed in the electronic project binder (binder). The sections of the binder should follow the phases of the life cycle, and each deliverable produced during the project should be placed in its appropriate place.

24. What are the trade-offs that project managers must manage?

The project manager must trade off system size, development time, and project cost. A larger project will require more time and money; while a short time frame may require more money or reduced project size. Since most projects have time and/or money constraints, the project manager must strike a balance between size, time, and cost in order to define an achievable project.

25. What is scope creep, and how can it be managed?

Scope creep refers to the addition of new requirements to the project after the initial project scope was defined and "frozen." Scope creep can be managed by doing the best possible job in determining the project requirements at the outset. Intensive meetings with users and prototyping are very beneficial in clarifying requirements early in the project. If a requirement is identified after the project begins, the consequences of adding it to the project should be carefully analyzed and presented to the users so that the impact of adding the requirements is understood (i.e., no free lunch). It may be possible to defer a requirement as a future enhancement to the system so that the current project scope is not increased.

26. What is timeboxing and why is it used?

Timeboxing is a technique that is used to organize a project when time is a critical issue. With timeboxing, a fixed deadline is established, and the project team prioritizes the functionality of the system so that the essential features are delivered within the set deadline. If some features must be omitted given that time frame, they are postponed to a later version of the system. With this technique, the users are assured of getting a system with essential functionality by the project deadline, and other, less essential features and refinements are added in later system versions.

27. Create a list of potential risks that could affect the outcome of a project.

Weak personnel, scope creep, poor design decisions, overly optimistic project estimates.

28. Describe the factors a project manager must evaluate when a project falls behind schedule.

Typically a project falls behind schedule when one of the risk factors listed above are realized. At that point the manger should determine which of those risk factors, or which combination of those risk factors, is responsible for the project falling behind. Once identified, the issue can be addressed.

Appendix 2A : Questions

1. What is a function point and how is it used?

A function point is a measure of program size that is based on the number and complexity of inputs, outputs, queries, files, and program interfaces. Each element of the system (input, output, query, etc.) will contribute some number of function points to the project based on the type of element and its perceived complexity. More complex elements add more function points than do more simple elements. Determining the number of function points provides a basis for determining the amount of programming code that must be written to complete the system. Conversion factors from function points to lines of programming code have been established for all major development environments.

2. *Describe the three steps of the function point approach.*

First, the size of the project is estimated in terms of the number of lines of code that must be written. This is accomplished using the concept of function points (a measure of program size based on the number and complexity of inputs, outputs, queries, files, and program interfaces). The number of function points are adjusted for perceived project complexity, and then can be used to determine the number of lines of code that must be written based on the language used for implementation. Second, the amount of effort needed to write the estimated number of lines of code is determined using an estimating algorithm (e.g., COCOMO model). This estimate will be expressed in the number of person-months of effort needed to produce the system code. Finally, the effort estimate is converted to a schedule time in months using another estimating algorithm.

3. *What is the formula for calculating the effort for a project?*

To calculate effort (in person-months), multiply the number of lines of code (in thousands) by a factor of 1.4. For example, a project requiring 25,000 lines of code would require 35 person-months of effort (25 * 1.4).

Appendix 2A : Your Turn Responses

2A-1. Calculate System Size

The following may be used as an example of student responses. Inputs, Outputs and Queries may be determined by the narrative provided. Files and Interfaces requirements will vary depending upon students' interpretation.

1. Inputs: Company Info, Interview/Office Visit Schedule, Offers Received
Outputs: Company Contact List, Interview/Office Visit Schedule, Thank You Letters
Queries: Number of Interviews by City, Average Offer Amount
Files: Company Information, Interview/Office Visit Schedule, Offers
Interfaces: Main menu which directs user to Input, Reports, Queries

2. **System Components**

Description	Complexity				
	Total Number	Low	Medium	High	Total
Inputs	3	1 x 3	1 x 4	1 x 6	13
Outputs	4	0 x 4	2 x 5	2 x 7	24
Queries	2	1 x 3	1 x 4	0 x 6	7
Files	3	0 x 7	3 x 10	0 x 15	30
Interfaces	1	0 x 5	1 x 7	0 x 10	7
Total Unadjusted Function Points (TUFP)					81

3. Total Unadjusted Function Points (TUFP) = 81

4. The Total Adjusted Function Point (TAFP) score will be calculated by multiplying the TUFP by the APC scores provided based on perceived Visual Basic skills.
.65 (Simple system to create) * 81 = 52.65
1 (Average system to create) * 81 = 81
1.35 (Complex system to create) * 81 = 109.35

5. Approximate number of lines of code per Function Point for Visual Basic = 30. To calculate approximate lines of code for this system, multiply the TAFP by the lines of code.
(Simple system to create) = 52.65 * 30 = 1579.50 lines of code
(Average system to create) = 81 * 30 = 2430 lines of code
(Complex system to create) = 109.35 * 30 = 3280.50 lines of code

2A-2. Calculate Effort and Schedule Time

The effort of the project in person-months is calculated by multiplying the projected lines of code (in thousands) from "Your Turn: 2A-1" by 1.4.

1. (Simple system to create) = 1579.50 lines of code = 1.6 * 1.4 = 2.24
(Average system to create) = 2430 lines of code = 2.4 * 1.4 = 3.36
(Complex system to create) = 3280.50 lines of code = 3.3 * 1.4 = 4.62

2. The schedule time in months for the project is calculated by the formula:
3.0 * (person-months$^{1/3}$)

(Simple system to create) = $3.0 * 2.24^{1/3} = 3.93$
(Average system to create) = $3.0 * 3.36^{1/3} = 4.49$
(Complex system to create) = $3.0 * 4.62^{1/3} = 4.99$

3. (Simple system to create) = 4 months
(Average system to create) = 4.5 months
(Complex system to create) = 5 months

Appendix 2B : Questions

1. *Compare and contrast the Gantt chart and the PERT chart.*

   A Gantt chart is a horizontal bar chart that shows the project tasks in rows, and project time across the chart horizontally. The starting point in time and duration of a task is shown as a horizontal bar under the appropriate time interval. Generally, the horizontal bar is open when a task has not been started, and is shaded as the task is completed.

   The PERT chart lays the project tasks out as nodes in a flowchart. Sequences and dependencies among tasks are easy to identify. The longest sequence of tasks is referred to as the critical path. If completion of the tasks on the critical path exceeds time estimates, then the project is going to be late.

2. *Of what value is the Gantt chart to the project manager? The PERT chart?*

   The Gantt chart is most useful in giving the project manager a snapshot in time of the progress through a project's tasks. Task completion is easily perceived. The PERT chart is best used to monitor progress on the critical path of tasks and to identify areas needing corrective action.

Solutions to End of Chapter Exercises

A. *Suppose you are a project manager using the waterfall development methodology on a large and complex project. Your manager has just read the latest article in Computerworld that advocates replacing the waterfall methodology with scrum and comes to your office requesting you to switch. What do you say?*

   Assuming I have the ability to stand up to my boss, I would try to encourage him to see that a methodology switch may not be appropriate during a large and complex project development. The waterfall methodology is quite valuable in complex projects, and that is probably why it is being employed. Scrum, on the other hand, is not suitable for complex projects because much of the modeling and documentation overhead is eliminated and this may prove to be inappropriate. I would try to dissuade the boss from mandating this change in the project, especially during mid-stream development.

B. *Suppose you were an analyst developing a new information system to automate the sales transactions and manage inventory for each retail store in a large chain. The system would be installed at each store and exchange data with a mainframe computer at the company's head office. What methodology would you use? Why?*

Throwaway prototyping should work well in this situation. The careful analysis at the beginning of the project should be valuable in determining the requirements. The biggest question is likely to be the unfamiliarity of getting the new system at the stores to communicate correctly with the existing mainframe system. Throwaway prototypes can be used to clarify the technical concerns of this situation so that the 'real' system works correctly when implemented.

C. *Suppose you were an analyst developing a new executive information system (EIS) intended to provide key strategic information from existing corporate databases to senior executives to help in their decision-making. What methodology would you use? Why?*

In this situation, requirements are likely to be unclear at the outset. The technology being used is probably familiar, and the system is probably not too complex since we are just reporting from existing corporate databases. Prototyping is an ideal approach to use in this situation. The biggest concern would be the time commitment on the part of the executives, since in prototyping their involvement and feedback is critical.

D. *Suppose you were an analyst working for a small company to develop an accounting system. What methodology would you use? Why?*

This situation has a well-understood problem domain that is not too complex but must be reliable. A structured methodology would be suitable since the business environment will not be too dynamic during the development process. Parallel development would work well since the parts of the system could be relatively independent of each other (accounts receivable, accounts payable, etc.) and could be successfully addressed as subprojects. As long as the final integration of subprojects is considered carefully, parallel development should work fine.

E. *Visit a project management Web site, such as the Project Management Institute (www.pmi.org). Most have links to project management software products, white papers, and research. Examine some of the links for project management to better understand a variety of Internet sites that contain information related to this chapter.*

Students' answers will vary.

F. *Select a specific project management topic like computer-aided software engineering (CASE), project management software, or timeboxing and search for information on that topic using the Web. The URL listed in exercise E or any search engine (e.g., Yahoo!, Google) can provide a starting point for your efforts.*

Students' answers will vary.

G. *Pretend that the Career Services office at your university wants to develop a system that collects student resumes and makes them available to students and recruiters over the Web. Students should be able to input their resume information into a*

standard resume template. The information then is presented in a resume format, and it also is placed in a database that can be queried using an online search form. You have been placed in charge of that project. Develop a plan for estimating the project. How long do you think it would take for you and three other students to complete the project? Provide support for the schedule that you propose.

Search for other schools that are using this type of system in the Career Services department and contact them to find out how long it took to develop such a system. Finding out some actual experiences with such a project would be one way to estimate this project. Another option is to use the function point estimation technique.

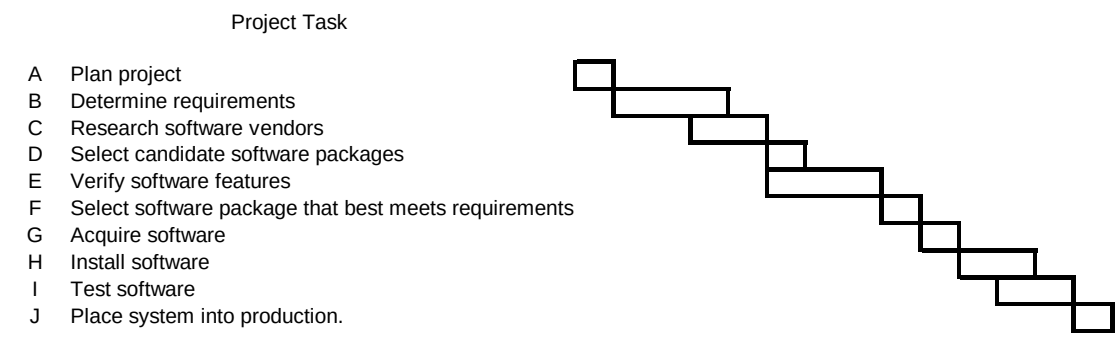
Assume that in the process of contacting other universities, it was found that projects like this typically required a team of three people about twelve weeks to complete. Applying standard percentages, this suggests that planning would take 2 weeks, analysis would take 2.5 weeks, design would take 4 weeks, and implementation would take 3.5 weeks. Since we have four people on our team, we might be able to shorten the project time to 9 weeks. Our schedule would be: planning - 1.3 weeks, analysis - 1.8 weeks, design - 3.2 weeks, and implementation - 2.7 weeks.

H. Refer to the situation in exercise G. You have been told that recruiting season begins a month from today and that the new system must be used. How would you approach this situation? Describe what you can do as the project manager to make sure that your team does not burn out from unreasonable deadlines and commitments.

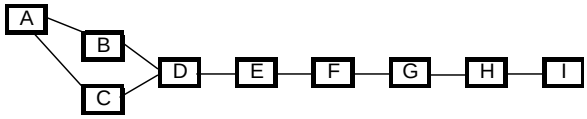
With a short deadline like this, the team will be under great pressure. One strategy I would investigate is the purchase of a software package that would meet the requirements of this project so that we would not have to code and test the system ourselves. If this is not feasible, I would employ timeboxing so that the team focuses on developing the bare-bones fundamentals of the system in the month we have available. Other 'nice-to-have' aspects of the system will be deferred to a later version of the system. We do not want to spend time working on frills, just the basics in order to meet this deadline.

I. Consider the system described in exerecise G. Create a work plan that lists the tasks that will need to be completed to meet the project's objectives. Create a Gantt chart and a PERT chart in a project management tool (e.g., Microsoft Project) or use a spreadsheet package to graphically show the high level tasks of the project.

Gantt Chart



PERT Chart



J. Suppose that you are in charge of the project that is described in exercise G, and the project will be staffed by members of your class. Do your classmates have all the right skills to implement such a project? If not, how will you go about making sure that the proper skills are available to get the job done?

Students may not have experience with the web-based technologies that would be used in the Career Services resume system. I would try to identify class members who do have the needed experience for my team if possible. If no one has the needed experience, I would try to arrange for training so that the team members can learn what they need for the project. This would require additional time in the schedule, however.

K. Consider the application that is used at your school to register for classes. Complete a function point worksheet to determine the size of such an application. You will need to make some assumptions about the application's interfaces and the various factors that affect its complexity.

Students' answers will vary.

L. Read "Your Turn 2A-1" in Appendix 2A of this chapter. Create a risk assessment that lists the potential risks associated with performing the project, along with ways to address the risks.

One risk is that the student may be overly generous in estimating his/her level of VB programming skills. This risk can be managed by being honest in this assessment, possibly even getting a VB instructor's assessment of the student's skills. This might help ensure that adequate time is devoted to the project.

Another risk is scope creep. The student may continue to generate good ideas on features that would be nice in the system. Unfortunately, it might never get completed. The student must finalize the requirements and stick to that list of requirements. Changes in scope should only be added after the basic system is complete.

A third risk would be inadequate design. The student may rush to coding rather than take the time to carefully outline the system requirements and design how the system will work. To manage this risk, the student must use discipline to ensure that the system has been well designed before the coding process begins.

M. Pretend that your instructor has asked you and two friends to create a Web page to describe the course to potential students and provide current class information (e.g., syllabus, assignments, readings) to current students. You have been assigned the role of leader, so you will need to coordinate your activities and those of your classmates until the project is completed. Describe how you would apply the project management techniques that you have learned in this chapter in this situation. Include descriptions of how you would create a work plan, staff the project, and coordinate all activities - yours and those of your classmates.

First, discuss with the instructor the requirements of the project. Then, create a work plan with project tasks clearly specified. Brainstorming with classmates might help here. Next, find two classmates who have prior experience in web page development to participate in the project. Develop a team charter that specifies the standards the team will use in terms of communication, documentation, and meeting times. Assign tasks to team members. Monitor performance. Reassign tasks as needed during project duration.

N. Select two project management software packages and research them using the Web or trade magazines. Describe the features of the two packages. If you were a project manager, which one would you use to help support your job? Why?

Students' answers will vary.

O. Select two estimation software packages and research them using the Web or trade magazines. Describe the features of the two packages. If you were a project manager, which one would you use to help support your job? Why?

Students' answers will vary.

P. A health insurance company had a computer problem that caused the company to overestimate revenues and underestimate medical costs. Problems were caused by the migration of its claims processing system from an older operating system to a UNIX-based system that uses Oracle database software and hardware. As a result, the company's stock price plummeted, and fixing the system became the number-one

priority for the company. Pretend that you have been placed in charge of managing the repair of the claims processing system. Obviously, the project team will not be in good spirits. How will you motivate team members to meet the project's objectives?

Since the team's morale may be very low, it is important to help them gain some early successes. First, make sure the team has a good working environment and that everyone understands the team's performance standards. Identify small, achievable goals at the start so that the team gets its confidence back. Make sure that deadlines are realistic; a team with low morale will be completely demoralized if they feel that deadlines are unachievable. Recognize and reward good performance at every opportunity, even if it's with small things such as gift certificates or tickets to a ballgame. Build team cohesiveness with little extras, like pizza for lunch on Fridays. Establish an environment of open, honest communication so that problems are identified and dealt with, rather than hidden until they become insurmountable.

Answers to Textbook Minicases

1. There are a number of things that may be helpful. Key among them are:
   - develop a project charter that clarifies the rules and group norms that Emily expects to follow
   - establish clear responsibilities for each team member
   - emphasize accountability for assigned tasks; don't let team members deflect responsibility to others
   - establish clear lines of communication, and emphasize shared responsibility for project success
   - get top management to commit to some type of recognition for the team to reward success

2. Student answers will vary depending upon which they feel is more important; organizational communication skills, or existing knowledge about the proposed system.

   If Marcus hires Barry, there will be no one on the project with an understanding of expert systems. Can Barry acquire that knowledge in a reasonable amount of time? Is training available? Does Marcus feel that a shallow level of knowledge about the system is enough for Barry to make a contribution?

   If Marcus hires Kim, there will be no one on the project with organizational communication skills. Again, can Kim acquire those skills in a reasonable amount of time? Or will her lack of organizational communication skills have a detrimental affect the outcome of the project?

3. **System Components**

Description	Complexity

	Total Number	Low	Medium	High	Total
Inputs	4	1 x 3	2 x 4	1 x 6	17
Outputs	4	0 x 4	4 x 5	0 x 7	20
Queries	7	1 x 3	4 x 4	2 x 6	31
Files	3	0 x 7	3 x 10	0 x 15	30
Interfaces	2	0 x 5	2 x 7	0 x 10	14
Total Unadjusted Function Points (TUFP)					112

Total Unadjusted Function Points (TUFP) = 112

The Total Adjusted Function Point (TAFP) score will be calculated by multiplying the TUFP by the APC scores provided based on perceived Visual Basic skills.

1.2 (Average system to create) * 112 = 134.4

Supplemental Minicases

1. Cal Strong is a project manager for a systems development project involving a new database application for his organization's human resources department. Despite Cal's best efforts at estimating the time requirements for the project, the team has missed the deadline for completing the Analysis Phase and producing the System Proposal. Cal had estimated during his 6-week Planning Phase that Analysis would require 2 months, but the team finished the System Proposal 2 weeks behind their planned target date. Based on his discussions with the team members, Cal believes that the system is more complex than he originally thought. Should Cal adjust his timeline, or can he try to make up the lost two weeks? Cal needs your advice.

   Answer: Based on standard percentages, Cal planned for 1.5 months in planning (15%), 2 months in Analysis (20%), 3.5 months in Design (35%), and 3 months in Implementation (30%), for a total project length of 10 months. The team missed their target in the Analysis phase by 25% (2 weeks/8 weeks). Since the remaining work is more complex than Cal originally thought, he cannot hope for the team to make up that time. It would be best for Cal to acknowledge his underestimate of project complexity right now, and extend the remaining phases by an additional 25%. This will push the completion date back 6.5 weeks. If this is unacceptable to the project sponsor, Cal should negotiate a reduction in project scope in order to give the team a reasonable chance to complete the project on time.

2. Mark Foster has recently been hired by a mid-sized machine shop that supplies various machine parts on contract to a large manufacturer. Mark has several years of systems development experience, including programming, analysis, and design. He has been hired with the express purpose of developing the EDI system that has been mandated by the firm's large business partner. Mark has not been in the position of

project manager before, and he is not sure what to do or where to begin. Based on this chapter's materials, what are three initial tasks that Mark should undertake?
Answer:

- *Identify project size*
- *Develop a project workplan*
- *Arrange to fulfill staff requirements*
- *Establish ground rules for project team and coordinate activities.*