Project Initiation

Systems projects are initiated by many different sources for a variety of reasons. Some of the projects suggested will survive various stages of evaluation to be worked on by you (or you and your team); others will not and should not get that far. Businesspeople suggest systems projects for two broad reasons: (1) because they experience problems that lend themselves to systems solutions, and (2) because they recognize opportunities for improvement through upgrading, altering, or installing new systems. Both situations can arise as an organization adapts to and copes with natural, evolutionary change.

Problems in an Organization

Managers do not like to conceive of their organization as having problems, let alone talk about those problems or share them with someone from outside. Good managers, however, realize that recognizing symptoms of problems or, at a later stage, diagnosing the problems themselves and then confronting them are imperative if the business is to keep functioning at its highest potential.

Problems surface in many different ways. In some instances, problems are uncovered because performance measures are not being met. Problems (or symptoms of problems) with processes that are visible in output and that could require the help of a systems analyst include excessive errors and work performed too slowly, incompletely, incorrectly, or not at all. Other symptoms of problems become evident when people do not meet baseline performance goals. Changes in employee behavior such as unusually high absenteeism, high job dissatisfaction, or high worker turnover should alert managers to potential problems. Any of these changes, alone or in combination, might be a sufficient reason to request the help of a systems analyst.

Although difficulties such as those just described occur in an organization, feedback on how well the organization is meeting intended goals may come from outside, in the form of complaints or suggestions from customers, vendors, or suppliers, as well as lost or unexpectedly low sales. This feedback from the external environment is extremely important and should not be ignored. A summary of symptoms of problems and approaches useful in problem detection is provided in Figure 3.1.

Defining the Problem

Whether using the classic systems development life cycle (SDLC) approach or an object-oriented approach, an analyst first defines the problems and objectives of the system. These form the foundation for determining what needs to be accomplished by the system.

A problem definition usually contains some sort of problem statement, summarized in a paragraph or two. This is followed by a series of issues or major independent pieces of the problem. The issues are followed by a series of objectives or goals that match the issues point by point. Issues are the current situation; objectives are the desired situation. The objectives may be very specific or worded as a general statement.

Needless to say, a systems analyst needs to understand how a business works.

Finally, the problem definition contains requirements—the things that must be accomplished—along with the possible solutions and the constraints that limit the development of the system. The requirements section may include security, usability, government requirements, and so on. Constraints often include the word not, indicating a limitation, and may contain budget restrictions or time limitations.

The problem definition is produced after completing interviews, observations, and document analysis with the users. The result of gathering this information is a wealth of facts and important opinions in need of summary. The first step in producing the problem definition is to find a number of points that may be included in one issue. Major points can be identified in the interview in a number of ways:

1. Users may identify an issue, a topic, or a theme that is repeated several times, sometimes by different people in several interviews.
2. Users may communicate using the same metaphors, such as saying the business is a journey, a war, a game, an organism, a machine, a family, and so on.
3. Users may tell a story to illustrate a problem that includes a beginning, a middle, and an ending; a hero; obstacles to overcome; and a successful (or hoped for) resolution.
4. Users may speak at length on a topic.
5. Users may tell you outright, “This is a major problem.”
6. Users may communicate importance using body language or may speak emphatically on an issue.
7. The problem may be the first thing the user mentions.

Once the issues have been identified, the objectives must be stated. An analyst may have to do a follow-up interview to obtain more precise information about the objectives.

After the objectives are stated, the relative importance of the issues or objectives must be determined. If there are not enough funds to develop the complete system, the critical objectives must be completed first. Users are the best people to identify critical objectives (with the support of analysts) because users are domain experts in their business area and know how they work best with technologies in the organization. In addition to looking through data and interviewing people, a systems analyst should try to witness the problem firsthand.

Identifying, Forecasting, and Comparing Costs and Benefits

Costs and benefits of a proposed computer system must always be considered together because they are interrelated and often interdependent. Although a systems analyst tries to propose a system that fulfills various information requirements, decisions to continue with the proposed system will be based on a cost-benefit analysis, not on information requirements. In many ways, benefits are measured by costs, which becomes apparent in the next section.

Forecasting

Systems analysts are required to predict certain key variables before submitting a proposal to the client. To some degree, a systems analyst will rely on a what-if analysis such as, “What if labor costs rise only 5 percent per year for the next three years, rather than 10 percent?” The systems analyst should realize, however, that he or she cannot rely on what-if analysis for everything if the proposal is to be credible, meaningful, and valuable.

A systems analyst has many forecasting models available. The main condition for choosing a model is the availability of historical data. If historical data are unavailable, the analyst must turn to one of the judgment methods: estimates from the sales force, surveys to estimate customer demand, Delphi studies (a consensus forecast developed independently by a group of experts through a series of iterations), creation of scenarios, or historical analogies.

If historical data are available, the next differentiation between classes of techniques involves whether the forecast is conditional or unconditional. Conditional implies that there is an association among variables in the model or that such a causal relationship exists. Common methods in this group include correlation, regression, leading indicators, econometrics, and input/output models.

Unconditional forecasting means the analyst isn't required to find or identify any causal relationships. Consequently, systems analysts find that these methods are low-cost, easy-to-implement alternatives. Included in this group are graphical judgment, moving averages, and analysis of time-series data. Because these methods are simple, reliable, and cost-effective, the remainder of the section focuses on them.

ESTIMATION OF TRENDS Trends can be estimated in a number of different ways. One way to estimate trends is to use a moving average. This method is useful because some seasonal, cyclical, or random patterns may be smoothed, leaving the trend pattern. The principle behind moving averages is to calculate the arithmetic mean of data from a fixed number of periods; a three-month moving average is simply the average of the past three months. For example, the average sales for January, February, and March are used to predict the sales for April. Then the average sales for February, March, and April are used to predict the sales for May, and so on. When the results are graphed, it is easily noticeable that the widely fluctuating data are smoothed. The moving average method is useful for its smoothing ability, but it also has many disadvantages. Moving averages are more strongly affected by extreme values than by using graphical judgment or estimating by using other methods, such as least squares. An analyst should learn forecasting well, as it often provides information that is valuable in justifying an entire project.

Identifying Benefits and Costs

Benefits and costs can be either tangible or intangible. Both tangible and intangible benefits and costs must be taken into account when systems are considered.

TANGIBLE BENEFITS Tangible benefits are advantages that are measurable in dollars that accrue to the organization through the use of the information system. Examples of tangible benefits are an increase in the speed of processing, access to otherwise inaccessible information, access to information on a more timely basis than was possible before, the advantage of the computer's superior calculating power, and a decrease in the amount of employee time needed to complete specific tasks. And there are other tangible benefits. Although measurement is not always easy, tangible benefits can be measured in terms of actual dollars, resources, or time saved.

INTANGIBLE BENEFITS Some benefits that accrue to an organization from the use of an information system are difficult to measure but are important nonetheless. They are known as intangible benefits.

Intangible benefits include improving the decision-making process, enhancing accuracy, becoming more competitive in customer service, maintaining a good business image, and increasing job satisfaction for employees by eliminating tedious tasks. As you can see from this list, intangible benefits are extremely important and can have far-reaching implications for a business as it relates to people both outside and within the organization.

Although intangible benefits of an information system are important factors that must be considered when deciding whether to proceed with a system, a system built solely for its intangible benefits will not be successful. You must discuss both tangible and intangible benefits in your proposal because presenting both will enable decision makers in the business to make a well-informed decision about the proposed system.

TANGIBLE COSTS The concepts of tangible and intangible costs present a conceptual parallel to the tangible and intangible benefits discussed already. Tangible costs are costs that a systems analyst and the business's accounting personnel can accurately project.

Included in tangible costs are the cost of equipment such as computers and printers, the cost of resources, the cost of systems analysts' time, the cost of computer programmers' time, and other employees' salaries. These costs are usually well established or can be discovered quite easily; they are costs that will require the business to make a cash outlay.

INTANGIBLE COSTS Intangible costs are difficult to estimate and may not be known. They include losing a competitive edge, losing the reputation for being first with an innovation or the leader in a field, declining company image due to increased customer dissatisfaction, and ineffective decision making due to untimely or inaccessible information. As you can imagine, it is nearly impossible to accurately project a dollar amount for intangible costs. To aid decision makers who want to weigh a proposed system and all its implications, you must include intangible costs even though they are not quantifiable.

Comparing Costs and Benefits

There are many well-known techniques for comparing the costs and benefits of a proposed system. Two are break-even analysis and payback.

BREAK-EVEN ANALYSIS By comparing costs alone, a systems analyst can use break-even analysis to determine the break-even capacity of a proposed information system. The point at which the total costs of the current system and the proposed system intersect represents the break-even point, the point where it becomes profitable for the business to get the new information system.

Total costs include the costs that recur during operation of a system plus the developmental costs that occur only once (one-time costs of installing a new system)—that is, the

tangible costs that were discussed previously. Figure 3.10 illustrates a break-even analysis on a small store that maintains inventory using a manual system. As volume rises, the costs

of the manual system rise at an increasing rate. A new computer system would cost a substantial sum up front, but the incremental costs for higher volume would be rather small.

The graph shows that the computer system would be cost-effective if the business sold about 600 units per week.

Payback is useful when a business is growing and volume is a key variable in costs. One disadvantage of break-even analysis is that benefits are assumed to remain the same, regardless of which system is in place. From our study of tangible and intangible benefits, we know that this is clearly not the case.

Payback analysis can determine how long it will take for the benefits of the system to pay back the costs of developing it. Figure 3.11 illustrates a system with a payback period of three and a half years.

Managing Time and Activities

The process of analysis and design can become unwieldy, especially when the system being developed is large. To keep the development activities as manageable as possible, systems analysts usually use some of the techniques of project management to help get organized.

The Work Breakdown Structure

Systems analysts are responsible for completing projects on time and within budget and for including the features promised. To accomplish all three of these goals, a project needs to be broken down

into smaller tasks or activities. These tasks together make up a work breakdown structure (WBS). When defined properly, the tasks that comprise a WBS have special properties:

1. Each task or activity contains one deliverable, or tangible outcome, from the activity.
2. Each task can be assigned to a single individual or a single group.
3. Each task has a responsible person monitoring and controlling performance.

Activities in a work breakdown structure do not need to take the same amount of time or involve the same number of team members. The activities defined must add up to 100 percent of the work, however.

The main method for developing a WBS is decomposition, or starting with large ideas and then breaking them down into manageable activities. This subdivision of ideas into smaller ideas and eventually into tasks stops when each task has only one deliverable.

There are different types of work breakdown structures. A WBS can be product oriented. In other words, building a website can be broken down into many parts, with each set of pages having a specific purpose. You could divide a website into its home page, product description pages, a FAQ page, a contact page, and an ecommerce page. Each of these pages could contain activities that you could use in your work breakdown structure.

1.0 Project Initiation

1.1 Assemble and welcome project team

1.2 Conduct historical research about the business

1.3 Discuss objectives with client

2.0 Early Planning Phase

2.1 Investigate feasibility

2.2 Consider a make vs. buy decision

2.3 Develop a work breakdown structure

2.4 Provide time estimates

2.5 Develop a project schedule

2.6 Calculate cost estimates

2.7 Prepare project proposal to client

2.8 Present proposal to client

3.0 Develop supporting plans

3.1 Develop a quality management plan

3.2 Identify risks and build a risk management plan

3.3 Describe a communications plan

3.4 Develop a procurement plan

4.0 Analysis

4.1 Conduct interviews of key personnel

4.2 Administer questionnaires

4.3 Read company reports

4.4 Analyze data flow

5.0 Design

5.1 Build prototype website

5.2 Obtain reactions from client

5.3 Modify prototype website

5.4 Seek final recommendations from client

5.5 Complete website

6.0 Launch

6.1 Create training manual

6.2 Document website features and logic

6.3 Present final website to client

Another way is to create a process-oriented work breakdown structure. An example of this is shown in Figure 3.12. This type of WBS is typical in systems analysis and design.

In this example, we show the development of a website, but rather than show the development of each page, this example emphasizes the importance of each phase in the systems development life cycle.

Project Scheduling

Planning includes all the activities required to select a systems analysis team, assign members of the team to appropriate projects, estimate the time required to complete each task, and schedule the project so that tasks are completed in a timely fashion. Control means using feedback to monitor the project, including comparing the plan for the project with its actual evolution. In addition, control means taking appropriate action to expedite or reschedule activities to finish on time while motivating team members to complete the job properly.

This section contains an example in which a systems analyst, acting as a project manager, begins with the basic activities of analysis, design, and implementation. Then the analyst uses decomposition to break apart the main activities into smaller subtasks, as shown in Figure 3.13.

Then the analysis phase is further broken down into data gathering, data flow and decision analysis, and proposal preparation. Design is broken down into data entry design, input and output design, and data organization. The implementation phase is divided into implementation and evaluation.

In subsequent steps, a systems analyst needs to consider each of these tasks and break them down further so that planning and scheduling can take place. Figure 3.14 shows how the analysis phase is described in more detail. For example, data gathering is broken down into five activities, from conducting interviews to observing reactions to the prototype. This particular project requires data flow analysis but not decision analysis, so the systems analyst has written in “analyze data flow” as the single step in the middle phase. Finally, proposal preparation is broken down into three steps: perform cost-benefit analysis; prepare proposal, and present proposal.

The systems analyst, of course, has the option to break down steps further. For instance, the analyst could specify each of the people to be interviewed. The amount of detail necessary depends on the project, but all critical steps need to appear in the plans.

Sometimes the most difficult part of project planning is the crucial step of estimating the time it takes to complete each task or activity. When quizzed about reasons for lateness on a particular project, project team members cited poor scheduling estimates that hampered the success of projects from the outset. There is no substitute for experience in estimating time requirements, and systems analysts who have had the opportunity of an apprenticeship are fortunate in this regard.

Planners have attempted to reduce the inherent uncertainty in determining time estimates by projecting most likely, pessimistic, and optimistic estimates and then using a weighted average formula to determine the expected time an activity will take. This approach offers little more in the way of confidence, however. Perhaps the best strategy for a systems analyst is to adhere to a structured approach in identifying activities and describing these activities in sufficient detail. In this manner, the systems analyst will at least be able to limit unpleasant surprises.

Using Gantt Charts for Project Scheduling

A Gantt chart is a tool that enables you to easily schedule tasks. It is a chart on which bars represent tasks or activities. The length of each bar represents the relative length of the task. Figure 3.15 is an example of a two-dimensional Gantt chart in which time is indicated on the horizontal dimension, and a description of activities makes up the vertical dimension. In this example, the Gantt chart shows the analysis or information-gathering phase of the project. Notice on the Gantt chart that conducting interviews will take three weeks, administering the questionnaire will take four weeks, and so on. These activities overlap part of the time. In the chart the special symbol ▲ signifies that right now, it is week 9. The bars with color shading represent projects or parts of projects that have been completed, telling us that the systems analyst

is behind in introducing prototypes but ahead in analyzing data flows. Action must be taken on introducing prototypes soon so that other activities or even the project itself will not be delayed as a result.

The main advantage of a Gantt chart is its simplicity. Not only is this technique easy to use, but it also lends itself to worthwhile communication with end users. Another advantage of using a Gantt chart is that the bars representing activities or tasks are drawn to scale; that is, the size of the bars indicates the relative length of time it will take to complete each task. You can use Microsoft Project to develop a Gantt chart.

Using PERT Diagrams

PERT is an acronym for Program Evaluation and Review Technique. A program (a synonym for a project) is represented by a network of nodes and arrows that are evaluated to determine the critical activities, improve the schedule if necessary, and review progress once the project is undertaken. PERT was developed in the late 1950s for use in the U.S. Navy's Polaris nuclear submarine project. It reportedly saved the U.S. Navy two years of development time. PERT charts are called network diagrams in Microsoft Project.

PERT is useful when activities can be done in parallel rather than in sequence. A systems analyst can benefit from using PERT by applying it to systems projects on a smaller scale, especially when some team members can be working on certain activities at the same time that other team members are working on other tasks.

Figure 3.16 compares a simple Gantt chart with a PERT diagram. Arrows in the PERT diagram represent the activities expressed as bars in the Gantt chart. The length of the arrows has no direct relationship with the activity durations. Circles on the PERT diagram are called events and can be identified by numbers, letters, or any other arbitrary form of designation. The circular nodes are present to (1) recognize that an activity is completed and (2) indicate which activities need to be completed before new activities may be undertaken (which is called precedence).

In the PERT example shown here, activity C may not be started until activity A is completed. Precedence is not indicated at all in the Gantt chart, so it is not possible to tell whether activity C is scheduled to start on day 4 on purpose or by coincidence.

A project has a beginning, a middle, and an end; in this example, the beginning is event 10, and the end is event 50. To find the length of the project, each path from beginning to end is identified, and the length of each path is calculated. In this example, path 10–20–40–50 has a length of 15 days, whereas path 10–30–40–50 has a length of 11 days. Even though one person may be working on path 10–20–40–50 and another on path 10–30–40–50, the project is not a race. The project requires that both sets of activities (or paths) be completed; consequently, the project takes 15 days to complete.

The longest path is referred to as the critical path. Although the critical path is determined by calculating the longest path, it is defined as the path that will cause the whole project to fall behind if even one day's delay is encountered on that path. Note that if you are delayed one day on path 10–20–40–50, the entire project will take longer, but if you are delayed one day on path 10–30–40–50, the entire project will not suffer. The leeway to fall behind somewhat on noncritical paths is called slack time.

Occasionally, PERT diagrams need pseudo-activities, referred to as dummy activities, to preserve the logic of or to clarify the diagram. Figure 3.17 shows two PERT diagrams with dummies. Project 1 and Project 2 are quite different, and the way the dummy is drawn makes the difference clear. In Project 1, activity C can be started only if both A and B are finished because all arrows coming into a node must be completed before leaving the node. In Project 2, however, activity C requires only activity B's completion and can therefore be under way while activity A is still taking place.

Project 1 takes 14 days to complete, whereas Project 2 takes only 9 days. The dummy in Project 1 is necessary, of course, because it indicates a crucial precedence relationship. The dummy in Project 2, on the other hand, is not required, and activity A could have been drawn from 10 to 40, eleminating event 20 completely.

Therefore, there are many reasons for using a PERT diagram over a Gantt chart. The PERT diagram allows:

1. Easy identification of the order of precedence
2. Easy identification of the critical path and thus critical activities
3. Easy determination of slack time