Project and Process Management

Chapter 5: Project Conceptualization: Defining Requirements

The Why, What and How of Project Conceptualization and Definition

OUTLINE

The First Stage
Using a Statement of Work
Defining Project Boundaries and Scope
Work Breakdown Structure
Why Getting this Right is so Important
The Use of Surveys and Interviews
Definition of exact Deliverables and Due Dates
Managing Stakeholder Expectations
What the Deliverables of this Stage are
Why this has to be done Expeditiously
Why Iterations Between This and the next Stage may be Necessary
How this Stage gets Done (The Process)
Using Goldratt's Thinking Process to Facilitate JRP Sessions
Defining Team roles and Determining the Organization
Assess Feasibility
Project Selection: Multi-attribute Utility Theory
Personal Project Selection
Infrastructure Project Selection
Making Decisions Amongst Multiple Projects with Interconnectedness
Summary
Exercises
Supplement: Best Practices for Stage 1

Learning Objectives: In this chapter, you will learn:

Why this conceptualization and definition stage is so important
How to identify the stakeholders
What the deliverables of this stage are
How this stage is performed (the process or methodology)
How to manage the stakeholders
How to conduct/facilitate Joint Requirements Definition sessions
How to assess feasibility
How to manage a portfolio of projects

Chapter

5

Starting off on the Wrong Feet

A system integrator and a Fortune 100 firm entered into an agreement in which the integrator was to convert some client/server applications into web-based Internet applications that used a browser front end. It was agreed that two project managers would be used, one on both sides of the contractual relationship. The two professional people selected to serve as project managers had more than 40 years of IT related industrial experience under their belts, but no project management experience between them. The two agreed to co-manage a $3 million project to convert the Fortune 100 firm's major applications for use on the Internet. Right away the two project managers started spending money. Many people (including the administrators themselves) were sent to this training seminar and to that one. By the end of the first year, over $350,000 had been spent on training in such loosely related disciplines as Visual Basic™ , Windows NT Server™, Oracle™, XML, MS Visual InterDev™, MS SQL Server™. Still, no one had any idea which major applications were going to get new Internet interfaces. Indeed, some of the firm's major applications were still old legacy mainframe systems with IBM 3270 user interfaces. These two PM individuals knew about screen scraping techniques and information technology but next to nothing about project management. They were given complete control of the budget as well as control over how they allocated their own time.

After spending the first year just training people on both sides of the contract, the two PM's established a steering committee of interested stakeholders. They paid for these stakeholders to see and assess other firms' software systems that were already Internet enabled.

The two project managers spent much of their first year going to conferences around the country and improving their skills in Internet development. They also taught classes and provided instruction to other personnel who were anticipated as serving on the project. And they organized and facilitated the annual steering committee meeting.

Two years later and half of the money spent, they began looking at some of the code associated with one of the legacy systems. It was written in RPG3, in the 1970's! There was no RPG expertise anywhere between the two firms. How was the team to make modification to this application? Another, application was written in COBOL, and COBOL expertise existed within the consulting side of the partnership but was totally committed to work on other applications. A third application was written in C. Expertise to make modification to that program was available but very expensive.

During this time, no formal deliverables of the project were ever defined; no formal project plan was ever developed. No applications were modified, as there was no agreement as to which applications should be modified. The interest of many steering committee members languished as the project lacked focus.

In the **conceptualization and definition** stage, the following steps are appropriate, as discussed in Chapter 4.

1. Verify that a detailed study is worth doing
2. Determine owners and stakeholders
3. Obtain user requirements
4. Define scope, size, and resource requirements
5. Ensure fit with business strategy and priorities
6. Assess technology consistency
7. Identify dependencies with other projects
8. Assess overall risk
9. Test alignment with/impact on strategies and plans
10. Test resource availability
11. *Make go/no-go decision*
12. Endorse/obtain funding
13. Review alternative approaches
14. Commit resources
15. Assign project management

Shown in Figure 5.1 below is a kind of block structure of the activities that make up this major stage in the lifecycle of a project. Note that Conceptualization and Definition is in a feedback loop with Planning and Budgeting. If, after the plan and budget are worked out, it may be discovered that the completion date is too far into the future. Therefore, it becomes necessary to go back to conceptualization to remove some of the functionality by adjusting scope. Alternatively, additional resources could be found which might enable the project to be completed quicker. Conversely, it may be discovered in the Planning and Budgeting stage that the project is too expensive. By returning to the Conceptualization and Definition stage, it is possible to remove expensive resources and replace them with cheaper ones, to reduce scope, or to slow the pace of the project.

In general, there are many tradeoffs between scope, time and cost. These tradeoffs get studied in a cyclical fashion in the Conceptualization and Definition stage as it interacts with the Planning and Budgeting stage.

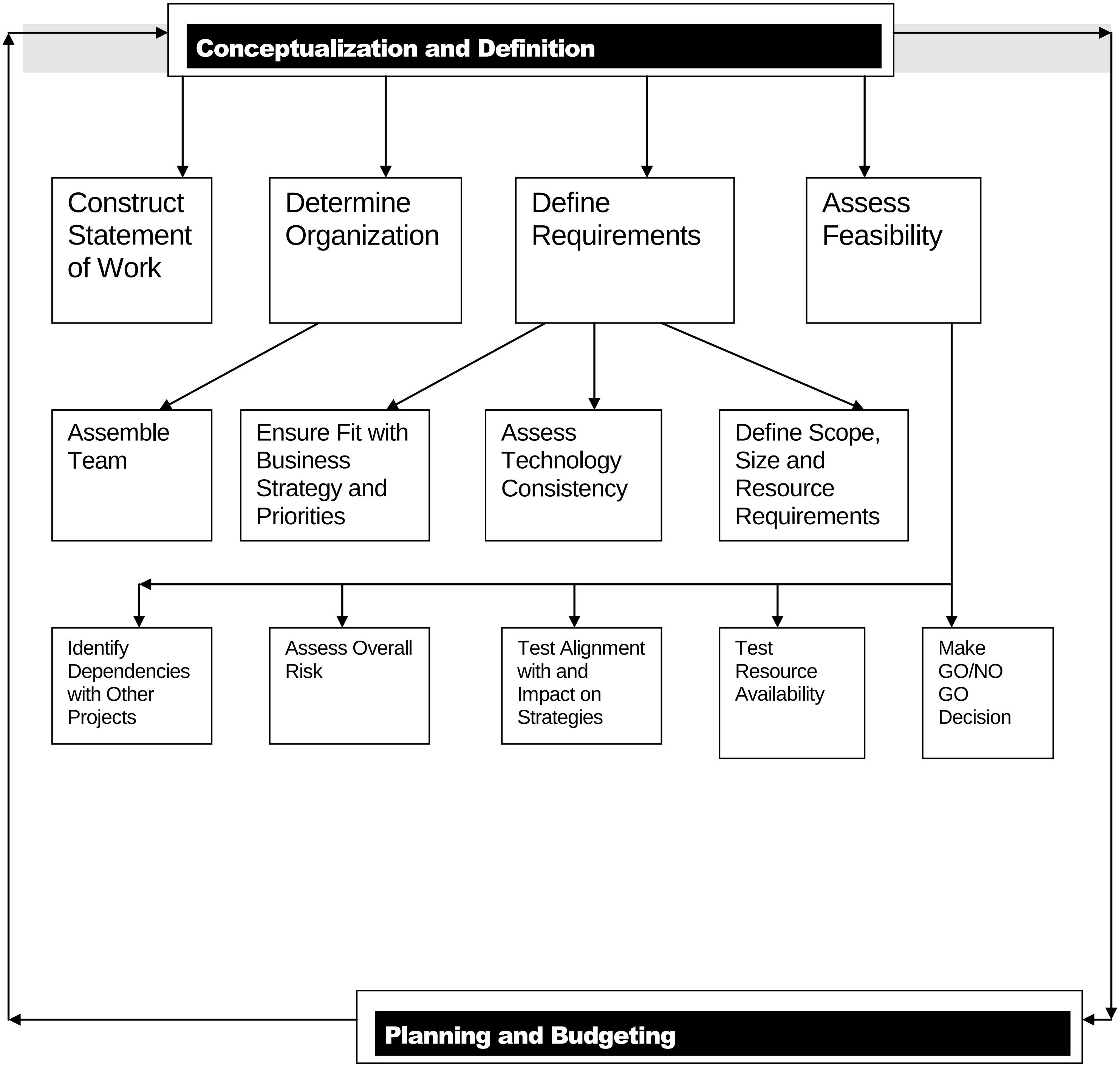

Conceptualization and Definition
Construct Statement of Work
Determine Organization
Define Requirements
Assess Feasibility
Assemble Team
Ensure Fit with Business Strategy and Priorities
Assess Technology Consistency
Define Scope, Size and Resource Requirements
Identify Dependencies with Other Projects
Assess Overall Risk
Test Alignment with and Impact on Strategies
Test Resource Availability
Make GO/NO GO Decision
Planning and Budgeting

Figure 5.1. The Definition and Conceptualization Stage

The First Stage

The first stage of a project has to launch the project, getting it going in the right direction. Like the first stage of any missile or rocket, a project has a mission and if it doesn't start out on its trajectory correctly, it may never hit its target or accomplish its mission. Sometimes, the launch is so bad, that the missile has to be destroyed before it gets to the second stage. At other times, significant redirections have to be applied in real-time during the first stage to get the missile moving in the right direction. Projects, in many ways, share similarities with missiles and rockets: they have a mission or target in mind, they are launched in stages and careful guidance and control must be applied throughout.

This stage begins with a formal definition of the project objectives. The objectives must be clearly defined in terms of the specifications or requirements for the final product. When external outsourcing is used, both customer and contractor must agree on the requirements. The objectives must be clear, attainable, specific, and measurable. As discussed in generality in Chapter 1, definition of the objectives requires consideration of scope, schedule, and cost. Examples of objective statements might be the following: "to transition from physical document management to Internet-based document management within twelve months at a cost of $700,000;" or "to convert all financial accounting applications to an ERP-based client-server or N-tier architecture within a period of fourteen months at a cost of $2,500,000." Still another might be "to convert all applications to utilize browser presentation management and to centralize the business logic component of these applications on a single application server at a cost of $5,000,000 within a period of twenty four months." These objective statements conform to the standards discussed for objective statements in Chapter 2. From the very beginning of the project, the objective of the project should be clear and concise.

Experience and lessons learned point to the benefits of hashing out the purpose, scope, and other factors in the project first, prior to developing the plan. If project managers wait until a plan is in place, there is so much detail that most managers will not question the plan, its detail or the concreteness of the concept behind it. Kliem, et al., articulate the need for project definition with clarity. According to them, projects with no direction are "more common than many project managers will admit." The consequences are disastrous; they are exemplars of a reactive style of project management. Projects that proceed without any direction have the pattern of a bag of marbles falling to the floor: every action resembles marbles scattering in different directions. Energy, consequently, is dissipated. People perform tasks in nonproductive ways. Duplicating effort and repeating tasks are common manifestations of a project plagued with unclear or no goals and objectives. Managers of such projects poorly employ resources…productivity declines as project costs increase. Projects without goals and objectives lead to poor morale…to dissension. People involved with the project…will likely start disagreeing among one another to the point that the project accomplishes nothing." [3]

Construct Statement of Work

Occasionally, the concept for the project is derived from a **Statement Of Work** (SOW) or a Project Overview Statement (POS). These one-page project initiators can be created by anyone in the organization who sees a need. In a project management culture, they are considered and graded by the Project Management Department. The format for the SOW is to include a brief objective statement, an explanation for why the need exists and what the benefits of the completed project will be. An example of such a document appears later in this chapter and in Chapter 11, where an explanation of it is also provided.

Project-oriented organizational cultures particularly encourage the use of a Statement Of Work. An outline for a SOW follows:

Proposed Project Name:
Date:
Name of Person Making the Submission:
Problems/Opportunities:
Goal(s):
Success Criteria:
Assumptions/Risks:
Impediments/Obstacles/Limitations/Exclusions:
Scope:
Deliverable(s):
Milestones:

Companies with a project orientation will encourage their employees to submit SOWs. These will get graded and discussed by a project review committee. A grading methodology is discussed later in Chapter 6. This committee will prioritize all of its pre-approved SOW's and then look at initial feasibility of each. If the SOW gets approved, the approval is for conceptualization and definition only. This is not to say that the actual project will get executed. But further studies relative to the SOW can take place; specifically, this stage, conceptualization and definition. If the perceived benefit appears to be significant, then the SOW gets approved for movement into the first stage of a project, conceptualization and definition. Information systems that are being proposed are considered if they directly contribute to the organizations' strategic plan, for example.

An SOW is just one way to launch a project. Projects are also authorized as a result of strategic planning that gives consideration to one or more of the following:

1. market demand, such as growth in demand for a product that necessitates a project to increase a firm's manufacturing capacity through enhancements to its scheduling software;
2. a business need, such as the ability for a firm's customers to see information relevant to them on the firm's website;
3. a customer request, such as a customer who wants its enterprise software to be upgraded to accommodate an N-tier architecture rather than the older client/server architecture;
4. a technological advance, such as the ability for sales people to close sales pitches in the field by uploading sales information via a cellular modem;
5. a legal requirement, such as a requirement by the government that specific firms report information to them on a monthly basis.

The deliverables of this stage of the project include a project requirements document exhibited later in this chapter and also showcased in Chapter 11. This deliverable articulates exactly what is needed, what interfaces with the existing information infrastructure are required, as well as generally when this product, configuration, system, structure or process is required.

Determine Organization

It is important that, once a project gets a preliminary go-ahead, that the project manager, leader and conceptualization team members be identified as soon as possible thereafter, if they have not been already. Stakeholders should be identified and they should be contacted. As mentioned in Chapter 1, **stakeholders** are professional people or organizations that have a vested interest in the project. Their interests may be positively or negatively affected as a result of project execution or successful project completion. Stakeholders include the customers who might be users of the system, human resource providers such as line managers, upper management who expects to profit from completion of the project, and the sponsor who provides the funds for the project.

A **Responsibility Assignment Matrix** is used to assign project players to specific responsibilities. More will be said about this later. Such a matrix will list the responsibilities in columns across the top and the names of the project players on the left. Then the matrix will show which player was assigned to which responsibility.

Define Requirements

A Joint Requirements Definition (JRD) Session is a good starting point for all projects. Stakeholders and potential team members should be invited to this session. Stakeholder groups should reflect as much diversity as possible to assimilate the gamut of constituent viewpoints. Out of these sessions should come cohesion and consensus. Conversation and dialogue are essential to discovery of the commonality. The person assigned to be project manager should be agreed to at this session. And, as its name implies, in a JRD Session, a rough-cut requirements document should be outlined.

With a common view of what requirements the ultimate deliverable should have, the project can go forward with a strongly held shared vision[1] of what the initiative is about. This is the purpose of the JRD Session. Now it becomes clearer what tasks will be required to complete it, and who should be on the ultimate project team.

After the JRD Session, the project manager should have an opportunity to develop a preliminary requirements document. Each stakeholder and team member should be involved in the further refinement of this document. Involvement builds commitment and ownership. Left to themselves, some stakeholders, particularly end-users, do not generally build good requirements documents. They don't know what is technically possible. And, they may not even perceive their own needs/requirements correctly.

After the project manager has sketched a rough-cut requirements document, he or she should transmit it to the stakeholders. Another session with the stakeholders should be convened to refine the requirements.

The VISION and MISSION

The requirements phase of any project establishes the vision and mission for the project. Without these, projects will languish and wonder out of scope. If planning is allowed to begin without formal requirements in place, the planning team will have difficulty deciding what exactly is to be done, what tasks should be included, and who should do them. If execution of the project is allowed to begin without concrete requirements and a formal project plan, then there will be no basis for determining whether the project achieved its objectives because it, in effect, had no concrete objectives. The project will know when to stop, however, when all of the money allocated to it is consumed. When experienced team members see the project lacking direction, they will leave, leading to a failed or canceled project.

[1] Recall the discussion of shared vision in the Senge material that was appendicized to Chapter 2.

Defining Project Boundaries and Scope

Once the stakeholders involved with the problem are determined, a meeting of the stakeholders should be convened. At this meeting, project boundaries and scope should be hammered out. Stakeholders should look at the big picture and make recourse to the organizational strategic plan (OSP). That plan should state long-term goals and when these will be completed. The plan should include an assessment of the organizations' strengths and weaknesses. In addition, organizational opportunities and threats should have been identified in the OSP. The term SWOT (for Strengths, Weaknesses, Opportunities and Threats) is often used in this context. Alignment of the project proposal with the OSP should be tested. All stakeholders should agree that the SOW delineates a project that rigorously supports the OSP.

Work Breakdown Structure

The Work Breakdown Structure (WBS) is a deliverable or product of the scope definition process. It is a vehicle for graphical delineation of project scope. It breaks the work of the project into its logical components. The WBS is used in scope management. Creation of a WBS employs the system concept of decomposition. A project is hierarchically broken down into ever more detailed work elements necessary for the project to be completed. If the WBS is done correctly, work outside the WBS is outside the scope of the project. The WBS defines not only the product to be created, but also provides a breakdown of tasks that can be used for schedule and cost planning, the next stage of the project lifecycle as discussed in Chapters 7, 8, and 9.

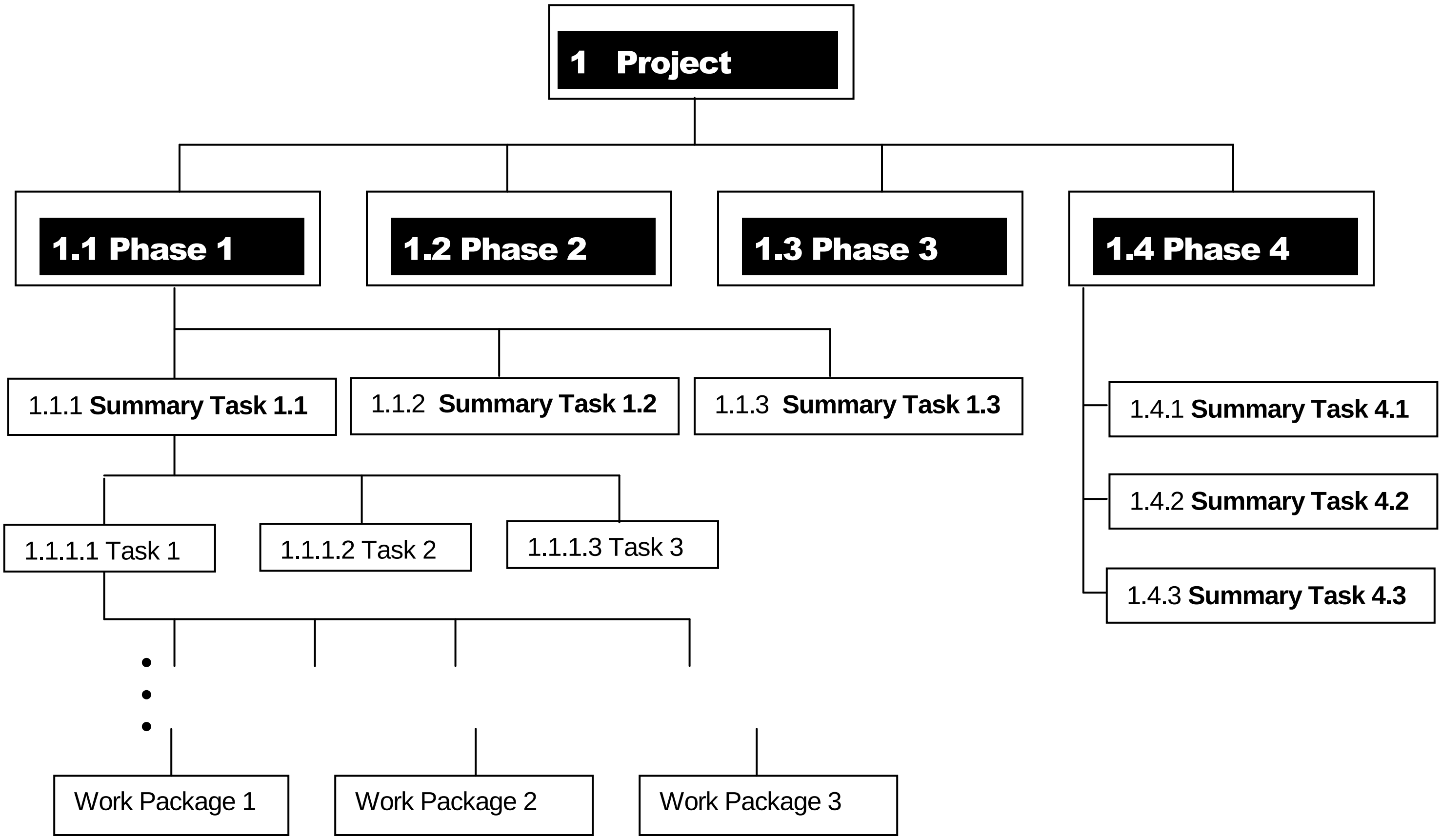


Figure 5.2 Work Breakdown Structure

Notice in Figure 5.2 above, that Work Breakdown Structure numbers in the left part of each box determine the level of the box and identify its parent box. For example, Task 1 has a WBS number of 1.1.1.1. From this number we know that Task 1 is on the fourth level of the WBS and that its parent is box 1.1.1 on the third level. Summary Task 4.3 has a WBS number of 1.4.3. It is on the third level of the WBS and its parent is box 1.4 (Phase 4) on level two. Clearly, boxes on the nth level will have n WBS numbers separated by periods. The bottom-most boxes of the WBS are called work packages. In an actual WBS, the work package boxes would have WBS numbers attached to them, but because it is not known in general at what level of the WBS the work packages will appear, WBS numbers have been left off of their boxes in the illustration above. The complete WBS is a structure that takes considerable work and inputs/assessments coming from many stakeholders to get it right. Complete delineation can be spread over the first two stages of the project life cycle—the first and second stages. In the first stage, the first three or four levels of the WBS might be delineated, depending on the size of the project. In the second stage, all subsequent levels down to work packages might be delineated. Delineation of the WBS should take place before Gantt charts and network diagrams are prepared and before such time management processes as activity sequencing, activity duration estimation, and schedule development are accomplished in the second stage of the life cycle.

The information in Figure 5.2 above is stored internally within a computer in a table that would appear as follows:

WBS Number	**NAME**
1	Project
1.1	Phase 1
1.1.1	Summary Task 1
1.1.1.1	Task 1
1.1.1.2	Task 2
1.1.1.3	Task 3
1.1.2	Summary Task 2
1.1.3	Summary Task 3
1.2	Phase 2
1.3	Phase 3
1.4	Phase 4
1.4.1	Summary Task 4.1
1.4.2	Summary Task 4.2
1.4.3	Summary Task 4.3
	Work Package 1
	Work Package 2
	Work Package 3

A WBS can be created from scratch or derived from use of a template. **Templates** are simply predefined work breakdown structures that were created for use by a previous project that has similarities to the current one. Frequently, projects are variants of previously defined, planned and executed projects. When such is the case, a good place in which to start is with an existing template. By reusing an existing template, the time required to formally decompose an existing project is eliminated or reduced significantly.

Creating work breakdown structures from scratch is a bit more work. The basic idea is to decompose each main activity down to work packages as described above. A **work package** is a well-defined piece of work that can be assigned to a single person. Usually one to three weeks in duration, work packages are the bottom-most boxes on the WBS. Often, a project manager will assign an appropriate person to do the work required by the work package. Work package duration can be estimated by the person assigned to do the work. A work package represents units of work at levels where work is performed (Department of the Navy, 1987). The work package is assign-able to an identifiable resource. The work package can be scheduled with specific start and completion dates. It's cost can be easily determined.

Work breakdown structures can be created by decomposition or by recourse to a predefined template. **Decomposition** involves sub-dividing project elements into smaller, and more manageable components in order to provide better management control.

Decomposition is what one way to create a work breakdown structure (WBS). The question is, how far should you break a project down? A WBS is begun by listing the major components of a project. This is level 2 of the WBS. Level 1 is just the title of the project, as illustrated in Figure 5.2 above. One way to start a project is to decide upon what major project type is being implemented and begin with the major phases defined in Chapter 4.

Why Getting this Right is so Important

The Conceptualization and Definition stage is one of the most critical in all of project management. According to William V. Leban, program manager at Keller Graduate School of Management, lack of proper project definition and scope is a main reason why projects fail.

Consider the article entitled 'starting off on the wrong feet' at the beginning of the chapter. Clearly, bad decisions at the beginning of or prior to the start of a project can lead to a project that fails.

Getting the problem defined right is 90% of the battle. The wrong solution to the right problem is no great dis-benefit, but the right solution to the wrong problem is a disaster. The requirements definition must address not only the functions and features of the system but also expected performance characteristics, the business and technical environment in which the systems will operate, and the acceptance test methods and procedures.

Managing Stakeholder Expectations

The stakeholders must be assembled into a work group and invited to the Joint Requirements Definition Session. The problem that often arises is that these persons have different expectations in terms of outcomes for the project. The sponsor expects the project to be completed within budget irregardless of whatever else happens. The customer expects the system to fulfill their functional expectations, irregardless of what it costs and so forth. One job of the project manager is to manage all of these differing stakeholder expectations. In so far as possible the project manager should seek to get consensus and alignment amongst all the stakeholders. Recall Chapter 3 and the discussion there about conflict resolution. When all efforts have failed to resolve the issue, the differences should be resolved in favor of the customer.

Noteworthy is the fact that there can be different groups within the customer base. A web-based software system to assist patients, insurance companies and doctors to track the medications prescribed by pharmacists would be of great value to all three of the pharmacists' constituencies. But all groups would have different views, values and expectations of the product. The patient would like to see what medications he has taken, what he is currently taking, what medications he is allergic to and so forth. Similarly, the doctor would like to see some of this but she is more interested in the medication history, the results of taking those medications, as well as what is currently being taken and what the interactions are. Further, the insurer would like to see what inoculations, drugs it has paid for so that it doesn't pay for them twice and so it doesn't pay for something that is not covered by the medical insurance.

Managing the different views of the various stakeholders is comparable to managing the diversity of values they all bring to the table. Generally, people agree on ends values. It is easy to get people to agree that a project should be done or even that it should fall within a certain budget amount and produce a product with a specified functionality. It is harder to get people to agree on the means by which this gets done. Some may want the entire project out-sourced; others may want it done in house; still other stakeholders might want to do it themselves and so forth. While there is a good possibility for consensus on ultimate goals; the processes, mechanisms by which these are achieved are always much more controversial.

Matrix delineation of stakeholders in terms of their commitment to the proposed project, their organizational affiliations, their roles, the decisions they can make regarding the project as well as suggestions for managing the relationship are commonplace in the project management literature. This is called **stakeholder analysis** and it is helpful in dealing with the diverse viewpoints, personalities and roles among the stakeholders. Especially important is an understanding of the possible veto or “project kill” power that certain stakeholders have. Certainly, the project manager will want to make sure these are “happy campers.” An example of stakeholder analysis follows:

	JILL	**JACK**	**JOHN**	**JIM**
ORGANIZATION	Customer	Project team member	Hardware vendor	Project sponsor
Personal traits	Not very computer literate, doesn’t know what she wants	Terrific developer; lots of experience on similar projects	Good sense of humor; a little lax on delivery due dates	Very diluted; not convinced project is needed; doesn’t like Jill
Relation to project	Works for the marketing dept.	Will do the database definitions of the product	Provides both client and server hardware components	Provides funds for project; can kill the project if he deems necessary
Level of interest	Moderate; is the end user of the deliverable	High	Moderate	Moderate
Level of influence	None	Moderate	None	High; can kill the project
Suggestions for managing the relationship	Involve her heavily in the user interface and database development	Don’t’ distract him; keep him happy	Stay after him, make certain he knows you absolutely have to have it on 3/3/3	Keep him informed; do as he says, now

It should be obvious that this kind of information is “eyes only” for the project manager and should not appear in any deliverable. The project manager manages the stable of stakeholders. But project managers can, to a certain extent, influence who the actual, participating stakeholders are. Persons with an obvious conflict of interest ought not to be allowed to be stakeholders. Project managers can talk to sponsors, upper management about the possible exclusion of persons who are openly hostile to the concept of the project.

Stakeholder Profiling: The Project Sponsor

Projects need proactive and involved sponsors, not just high-level people who allocate funds and then walk away from the project. Sponsors should serve as champions of the project and should maintain a position of advocacy relative to the projects' continued progress. If they do this, sponsors can contribute to the success of the project by protecting it from political adversaries. In any organization there are always those who would otherwise like to plunder the project because it conflicts with their personal agendas and takes resources away from those agendas. The sponsor is also an outsider in the sense that he is not involved in the details of the project nor does he do the work of creating any of the products of the project. He or she can be of extreme importance to the project by 1) providing the funds necessary to sustain it, and 2) protecting it from adversarial outsiders.

Other ways a project sponsor can contribute to the project include: 1) reviewing and approving the SOW, 2) contributing to the definition of the project, 3) singing off on the project requirements doc and the project charter and 4) providing quality assurance. Regular, informal interactions with the project manager can be very helpful in terms of the advice that is given.

Project sponsors should be advised of the consequences of making changes to project requirements. Such changes are costly in terms of time and budget. If a sponsor discovers some aspect of the requirements that are inadequate, he or she may have the power to kill the project at this stage or to force a change in the requirements without providing additional funds to cover the incremental cost. Either of these alternatives are unreasonable to the project manager and the team.

Stakeholder Profiling: The Project Customer

Customers are professional people with final authority over product requirements. In a marketing department, market research is performed to determine what the requirements are and these are then written into the product specifications.

What should happen when the number of customers is large? For example, a new water treatment software billing system that is being conceived will ultimately impact 300,000 customers. Do we need signature signoff from all of these? Fortunately, the city municipalities have processes for customer interaction that do not involve a signoff of all the customers. Typically, this will consist of a "town meeting" in which the plans and their implications are presented, followed by a time of customer questions and responses. Usually, the media does a good job of getting this word out to the myriad of customers who could not attend.

Selection and Adaptation of a Methodology

One major decision the project manager must make within this phase is selection and adaptation of a methodology. Starting with one or more of the methodologies presented in the previous chapter, the PM should make adaptations that will improve the fit with his or her particular project. Later an entire chapter, Chapter 13, is devoted to transitioning project processes (methodologies) toward ever more effective and efficient forms. Discussion of various methodologies appeared in Chapter 4.

What about rules?

The expectations, standards, processes by which the project will be undertaken from beginning to end should be prescribed. Usually, these take the form of rules. It's always best to have these written down and made explicit up front so every one is on the same "wavelength." What is needed here is an expert system for project management. Such a system could provide consultation, advice and perspective to the intellectually challenged in the area of project management.

Some examples of rules are: "don't spend longer that 25% of the total allotted time on conceptualization and definition of the project." Another might be, "if the deliverables for any given stage are incomplete, then the project should not be allowed to proceed into the next stage." If changes to the requirements are necessitated by discovery or by evolution of the environment or technology, then a change management process is activated. That process is one of the various "rules prescribed for the game."

Delineation of a Project Charter

The purpose of a project charter is to create visibility for a project, to announce the initiation of the project, and to set expectations for it. The project charter is more detailed and refined than the SOW and is used to communicate the project vision, scope, processes, and stakeholders. It is part of the overall communication plan.

Project charters will name the project, give an approximate start date, describe GO/NO-GO criteria, declare who the project manager, project sponsor are, and reassert the project objectives. In addition, a good charter will state assumptions.

The most important contribution of the project charter is its scope statement. This statement should provide some justification for the project in terms of business need, should state what products, deliverables will be forthcoming from the project, and what constitutes success for the project. The processes used to create the deliverables of the project should also be discussed.

Occasionally, project managers loose focus after beginning execution of the project, forgetting the original goals/objectives as they become immersed in the details. The result is myopia and straying from the path leading to achievement of the goals and objectives. A project charter can serve as a reminder to the project manager and team as to what the original objectives were and can help to maintain focus.

To create a project charter, a meeting of the stakeholders should be convened at which a meeting facilitator, who is usually the project manager, endeavors to get consensus on the scope, objectives, assumptions, constraints, rules and deliverables. If the stakeholders cannot come to an agreement before the project commences, they are even less likely to do so after project execution/control begins and money is being spent.

A project charter should delineate any specialized rules or processes that are anticipated. For example, the rules for change management could be delineated in the charter. Alternatively, the charter could describe when such rules will be made up and who will make them.

A project charter should describe project governance. For example, does the project manager have complete authority to spend funds as he or she thinks best? Does the project manager have the authority to remove a non-performing person from the project? Can the project manager procure the necessary materials needed, send people to training seminars and charge this to the project budget?

How will major project decisions faced by the project team get resolved? What project culture will the project aspire to—monastery, sweatshop, egalitarianism, personal growth, what?

A written project charter helps keep the stakeholders focused and the charter establishes a project manager's right to make decisions and lead the project. It gives the project manager referential authority. Referential authority simply means that certain stakeholders have empowered the project manager to act in their behalf—to engage resources, spend money, make decisions and report project status.

An outline for an example project charter follows:

NAME:
OBJECTIVES:
STAKEHOLDERS:
PROJECT MANAGER:
SCOPE:
DELIVERABLES:
ASSUMPTIONS:
RULES/PROCESSES:
GOVERNANCE:
COMMENTS:
SIGNATURES/SIGNOFFS:

Notice that the project charter contains some of the same content as the SOW. But the SOW is an informal document, not requiring stakeholder signoffs. The charter replaces the SOW and supercedes it. The project charter is like a contract between many (more than two) different parties. Contracts are legal documents between just two parties. It is necessary that the project manager gain consensus on most, if not all, of the terms/stipulations of the project charter. This may require several meetings of the stakeholders to do so.

Deliverables of this Stage

Project conceptualization and definition is the first stage of any project and among the most important. In this stage the deliverables of the entire project get defined. If these are not carefully delineated, then the whole project may wind up producing nothing of value. The deliverables do not have to be detailed in totality, but major characteristics should be declared. One deliverable of this stage is identifying the deliverables of the entire project.

Also, this stage must deliver a set of requirements, a requirements document. A detailed outline for a requirements document is provided in Chapter 11.

As just discussed, this stage should also deliver a project charter. The authorship of the project charter can vary or be distributed among the stakeholders. Certainly, the project sponsor, project manager, customer and management should have some involvement in creating it. Most appropriately, the project sponsor should make the announcement by attaching a cover letter with his or her signature on it.

Why Conceptualization and Definition has to be done Expeditiously

Requirements change over time. If the requirements aren't defined expeditiously and the whole project (all four stages) completed in a timely manner, the requirements for the project product (or project deliverable) may change mid-stream, causing the whole project to be canceled or to be greatly changed at considerable expense. In the late 1980's some 80 billion of IT projects were canceled per year in the U.S., primarily because the products of the projects could not be delivered fast enough to bring value to the customers. Time and timeliness is so important when it comes to projects that some managers will insist they would rather have an incomplete project than to have it late.

Leadership

More so than any of the other major components or stages of project management, leadership is needed to effect successful conceptualization and definition of projects. Strong leaders are able to articulate their vision for what a more desirable future would be, if the product being conceived were created and put in place. They are able to articulate with clarity the current reality and they are able to maintain that creative tension between current reality and that more desirable future. Goal erosion is not a possibility unless some project constraint clearly cannot be realized: cost, duration, or functionality.

There isn't agreement as to what exactly constitutes good leadership. Are the competencies associated with leadership acquired or inherited? Is leadership a science or an art? But all can agree that leadership has a tremendous impact on the outcome of a project in terms of cost, schedule, functionality, and quality. Charismatic, positive, leadership that is able to see the total "picture" while being immersed in details is contagious. Courageous, faithful professionals who do not break down under pressure but continue to exude a positive perspective while treating others with both firmness and kindness make attractive project leaders and managers.

Motivating people to perform effectively and efficiently entails the following: providing vision, communicating details to all, inspiring the stakeholders, maintaining direction, being supportive, encouraging the team, and building camaraderie, to name just a few.

Senge (1990, pages 339-360) alludes to the leader's work as involving: 1) design, 2) stewardship of the purpose story or corporate vision, 3) teaching and 4) maintaining or holding the creative tension associated with current reality and the corporate shared vision for how things could be. If we think of a project as a rocket ship taking people to Mars, who would the project manager be? Would he be the captain, the navigator, the entertainment director, the communications officer? Senge would say no to all of these—he would be the designer of the rocket ship. Why? Because it is the designer of the rocket ship more so than any other person that determines the dynamics of the rocket itself and whether the ship is able to accomplish its mission. As designer, the project manager must finalize the design of the processes that make up the project. Listed in the box below are processes requiring design by the project manager. Planning and budgeting in the next stage is, by any other word, designing. As steward, the project manager lives the corporate vision, which is also akin to his personal story and mission; that is, he "walks the talk." As teacher, others realize they have much they can learn from him; he is respected for what he is. As maintainer, he continually articulates the current reality taken in relation to the shared future vision everyone on the project has bought into.

Criteria for Good Project/Process Design

Generally, the criteria by which processes are measured include: speed or completion time, quality of the end-result, cost and flexibility.

1. Overall, the project/process adds value that the customer would be willing to pay for.
2. The project/process should result in a quality deliverable at the lowest possible cost and duration.
3. The project/process should not contain non-value-adding steps.
4. Handoffs among participants should be minimized; this is accomplished through the use of meetings or non-meetings over the Internet.
5. Operations should be concurrent rather than sequential to speed completion, improve quality and reduce costs.

Processes Requiring Design within each Stage

Stage 1: Conceptualization and Definition

JRD meeting facilitation
Scope definition
Feasibility assessment
Initiation
Resource assignment process
Change management

Stage 2: Planning and Budgeting

Activity definition
Activity sequencing
Activity duration estimating
Schedule development
Resource planning
Cost estimating
Cost budgeting
Quality planning
Organizational planning
Project plan development
Staff acquisition
Communications planning
Risk management planning
Risk identification
Qualitative risk analysis
Quantitative risk analysis
Risk response planning
Procurement planning
Solicitation planning

Stage 3: Execution and Control

Periodic status review meetings
Slip announcements
Project expediting
Quality assurance
Cost control
Quality control
Team development
Information distribution
Solicitation
Source selection
Performance reporting
Risk monitoring

Stage 4: Termination and Closeout

Contract close-out
Administrative close-out

Stakeholder Interviews

One possible strategy for definition of the project is to conduct stakeholder interviews. Two interviewers should approach just one interviewee (the person being interviewed.) One interviewer should ask questions from a predefined script of questions. The second person should take notes. The first interviewer should listen carefully to discern whether the question was answered adequately and should follow-up with clarifying questions if the scripted question was not adequately addressed.

Interviews can be very significant in defining requirements. First, the right people should be interviewed. Second, in all of the interviews, the right questions must be asked. It's always important to ask about the flow of information within the firm. Start with outputs—what decisions are made. Then proceed to inputs in terms of what information is needed to make those decisions at this level. What information is currently available, what additional information is needed. Recall the five W's of journalism—Who, What, Where, When, and Why. Occasionally, it is necessary to re-interview some individuals because only after all of the interviews are complete is it sometimes discovered that additional questions are necessary to fully ascertain how a process works. Interviews are most appropriate when information is needed in order to understand a process. To elicit requirements, build consensus and create concurrent designs, a JRD (Joint Requirements Definition) Session is much more effective.

Joint Requirements Definition

A JRD meeting will determine the requirements, and the products, deliverables that will meet those requirements. After the JRD session has been completed and feasibility studies are finished, there is a GO/NO GO decision. Occasionally, the decision is NO, as the expectation is that there is something else that will meet the requirements better. The JRD effort is focused at really nailing down objectives. What can be used--data models, object models, any reusable components? Who is going to participate? The critical people from IT will include a facilitator, a scribe, a DBA[2]. The business unit should send people with decision-making authority. The preparation of the agenda for the meeting is done beforehand and transmitted to all attendees one week prior. The purpose of the meeting is to get consensus on these agenda items.

At the end of the JRD session, there may be a lot of un-resolvable issues. Does that kill the project? Sometimes. Typically, three days are allocated to work through all of the issues. On the first day, all of the participants have their own agenda. The personal agendas are well defended. On the second day, the facilitator must convince that the purpose is to bring benefit to the whole organization. She must get these people working as a team. By the third day, the facilitator has used every "trick in the book" to get consensus across the group and to isolate those people who will not bend to the needs of the majority. A trained facilitator is enormously valuable. Once the workshop is done, all of the participating organizations should pull together and there should be buy-in throughout. What platforms will be used, what databases, what tools, have been resolved. What is left dangling will impact the GO/NO GO decision.

[2] DBA—Database Administrator

Requests for Proposal (RFP)

Yet another deliverable of this stage is a decision to outsource either part of or all of the work. This is typically done by submitting a request for proposal or RFP to qualified contractors. A RFP will contain much of the same detail as a requirements document.

Before a RFP is submitted to outside vendors, the project manager must be sure that preliminary approval to solicit proposals from the outside has been granted. Also, a determination as to how much of the total project is to be out-sourced should be made and the RFP should reflect only the work that is to be out-sourced. Usually, this is not known until after a project plan is determined in the next stage.

Proposals that get returned in response to a RFP can be graded much like SOWs can be graded. To do that the criteria by which the proposals will be judged should be known together with the weights. It is a good idea to have all of this determined before the RFP goes out. The evaluation criteria might even be discussed as part of the RFP, at the project manager's discretion. A methodology for grading everything from job offers to computers being contemplated for purchase is presented in the next chapter.

To create a proposal, the responder must engage in the activities of stage two of the project life cycle, Planning and Budgeting. Two very important parameters that must appear in any proposal are the duration and the cost. In order for these numbers to be accurate, a thorough plan and budget must be prepared. More will be said about proposals in the chapters on planning and budgeting (Chapters 8 and 9).

Using Goldratt's Thinking Process to Facilitate JRD Sessions

There are many JRD sessions in which there is no obvious solution, but there are many obvious "problems." In what follows, a methodology for finding a root cause and generating a solution from that is presented. This methodology is particularly germane when firms are resolved to address core problems rather than "symptoms." In some cases, the general strategy or plan has already been established and the JRD meeting is just about refinement of the generally-conceived solution. In such cases the methodology proposed herein would not be appropriate.

Requirements definition entails a good collective understanding of current reality, a strong vision of what could be down the road, and an agenda for evolving the current reality to a more desirable future reality. In the words of Eli Goldratt, the session should address three questions:

1) What to change?
2) What to change to?
3) How to cause the change?

The first question above should assess issues like success criteria, system or object under scrutiny, and the current reality for that system or object. The second question above should entail finding a breakthrough solution and verification that the breakthrough solution will achieve the desired end result. The third question should focus attention on the process by which the system or object is taken from its current undesirable state to a future more desirable condition or state.

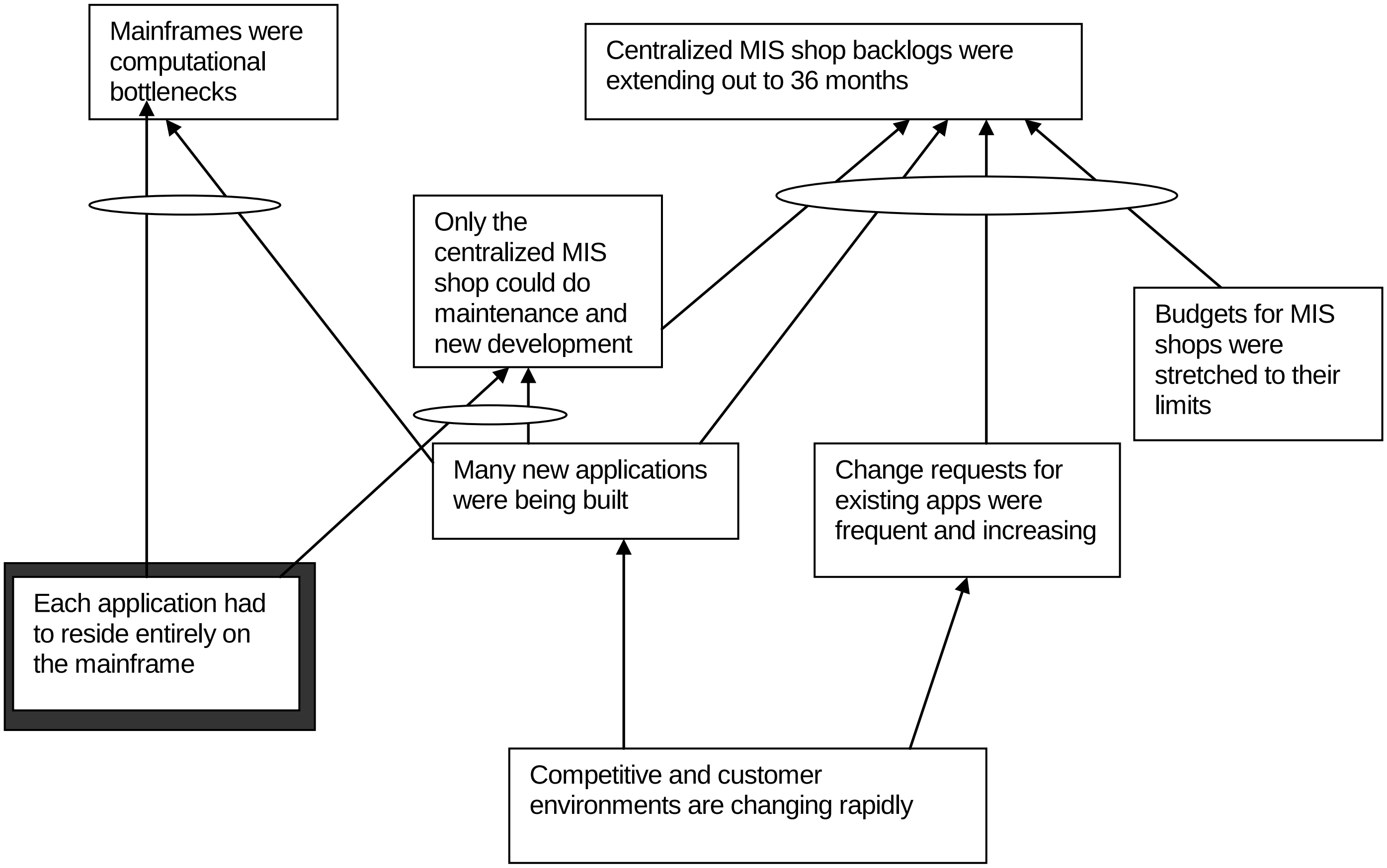
Typically, stakeholders have a good handle on "what's broke." But frequently stakeholder participants are focused on symptoms rather than root causes, events rather than patterns and behaviors. To address the first question above, participants need a methodology for mapping symptoms and superficial events down to root causes. For that Goldratt recommends a tree that causally traces undesirable effects down to root causes and core problems. Senge, on the other hand, recommends the use of causal loops to get a handle on the underlying dynamics. In both cases, the intent is not to respond to symptoms and events but to understand the root causes, the underlying dynamics.

To get started, participants are asked to list undesirable events, annoying effects that make up the current reality. These should be written on post-it notes and placed on a sticker board where they are visible to all of the participants.

The goal of this part of the methodology is to get a better understanding of root causes, and underlying dynamics. To do this two constructs are used, a tree and a diagram. The purpose of the tree is to map all superficial effects back to root causes, working backwards. The intent of the diagram is to understand cyclical systemic causal structures in which feedback loops of causality are created. Two constructs are used because trees cannot be trees and have causal loops embedded within them. The tree construct captures the underlying static detail causality, while the diagram captures the dynamic complexity.

Consider the following scenario that was true of corporate computing a few years ago when the mainframe architecture was prevalent. Centralized mainframes were computing bottlenecks. Centralized MIS shops had long lead times of 36 or more months. End users had very little control and ownership over their own applications. On the other hand, network PC's were idle more than 95% of the time. The legacy mainframe applications did not share data easily. The sales module could not see the financial/accounting data or the manufacturing data, for example. New mission critical applications had to run entirely on mainframes. There was very little opportunity for reuse of existing software.

Each software application consisted of three components: presentation management, business logic management, and data management. Development/creation of each application was essentially a re-write of each of these components; there was no re-use of existing components. Many new applications were being anticipated and changes to existing applications were rampant.


Mainframes were computational bottlenecks
Centralized MIS shop backlogs were extending out to 36 months
Only the centralized MIS shop could do maintenance and new development
Budgets for MIS shops were stretched to their limits
Many new applications were being built
Change requests for existing apps were frequent and increasing
Each application had to reside entirely on the mainframe
Competitive and customer environments are changing rapidly

This diagram reads as follows, starting in the lower left hand corner. “If _Each application had to reside entirely on the mainframe_ and _Many new applications were being built_, then _Maintrames were computational bottlenecks_. Also, if _Each application had to reside entirely on the mainframe_ and _Many new applications were being built_, then _Only the centralized MIS shop could do maintenance and new development_. Moreover, if _Only the centralized MIS shop could do maintenance and new development_ and _Many new applications were being built_ and _Change requests for existing apps were frequent and increasing_and_Budgets for MIS shops were stretched to their limits_, then _Centralized MIS shop backlogs were extending out to 36 months_. Notice that the use of the ellipse in conjunction with the causation arrows. It’s purpose is to designate that the supporting causal statements are to be ANDED rather than ORed. This is to say, in order for the consequent phenomenon to occur, all precedent causes connected by a causation arrow must happen when an ellipse appears. In an ORed situation, no ellipse appears and any of the precedent causal relations will force the consequent phenomenon to occur.

As we examine this diagram, we have to ask what the root cause is. Because it is impossible to change the environment, the committee doing this investigation is forced to examine the premise that _Each application had to reside entirely on the mainframe_.

Consider another problem that existed circa 1990 within corporate computing, as indicated below.

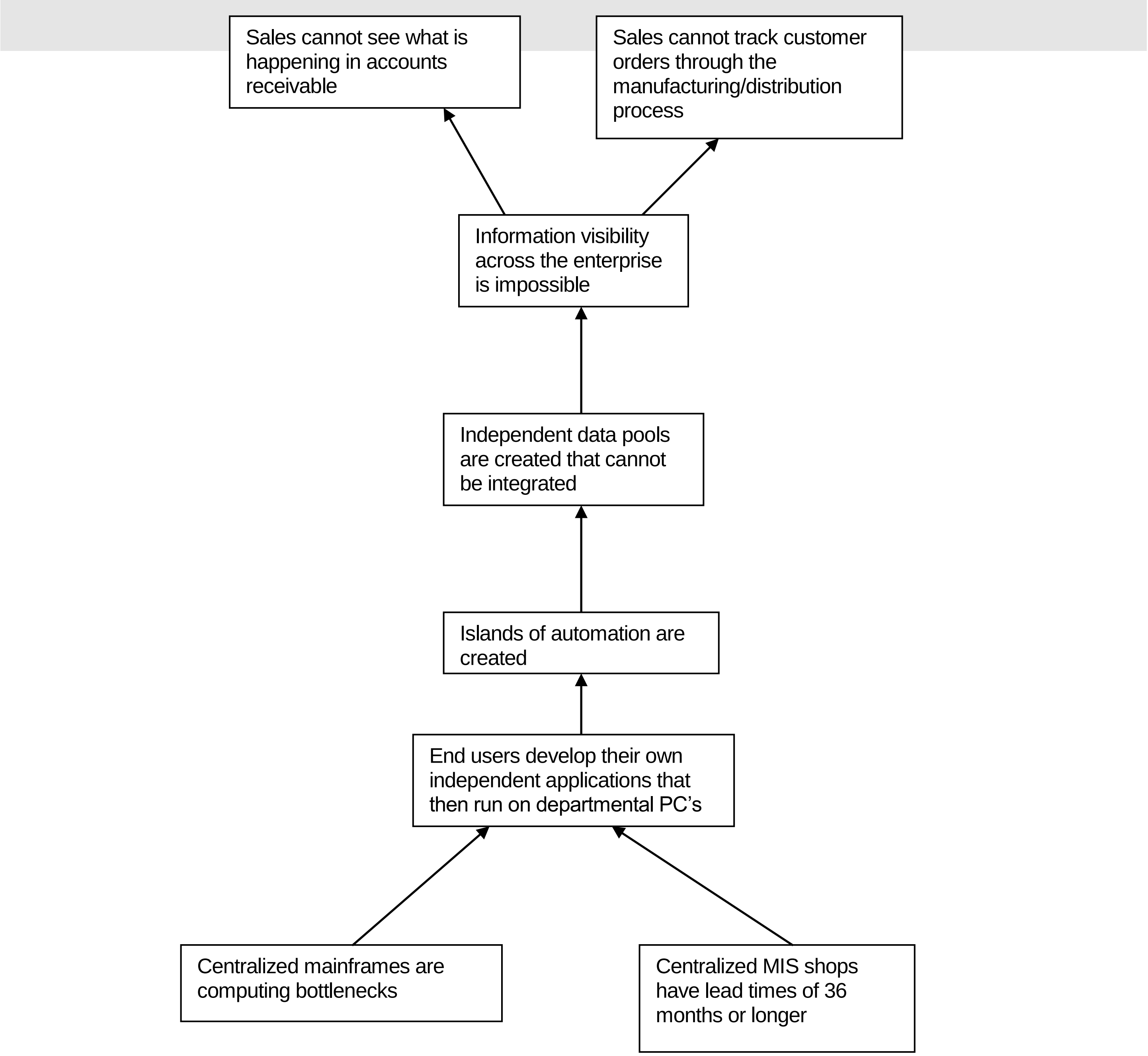

Sales cannot see what is happening in accounts receivable
Sales cannot track customer orders through the manufacturing/distribution process
Information visibility across the enterprise is impossible
Independent data pools are created that cannot be integrated
Islands of automation are created
End users develop their own independent applications that then run on departmental PC's
Centralized mainframes are computing bottlenecks
Centralized MIS shops have lead times of 36 months or longer

Figure 5.3. Current Reality Tree for Understanding the Root Cause of the Undesirable Effects of Mainframe Architectures

In the above, we observe yet more undesirable effects resulting from the computing bottlenecks caused by mainframe computers OR the long lead times in centralized MIS shops. From the previous tree, we observe that both of these phenomena, (_Centralized MIS shops have lead times of 36 months or longer_, _Centralized mainframes are computing bottlenecks_) are derived from _Each application had to reside entirely on a mainframe_. Now we have a better idea as to what the implications of this technology constraint were.

Still other problems were observed in conjunction with the old mainframe architecture. Because each application resided entirely on the mainframe and had its own data management component that managed its own pool of data, it was very difficult for applications to share data. Yet sharing of data was precisely what was needed to order to support such new developments as activity-based costing and business process integration. Each software program (application) had an architecture like that shown below. The cylinder at the bottom represents a magnetic disk drive where data were permanently stored.

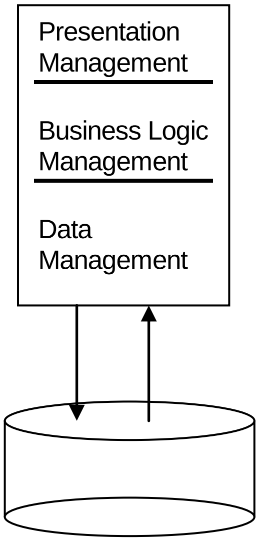


Figure 5.4. The Three Components Making up any Application

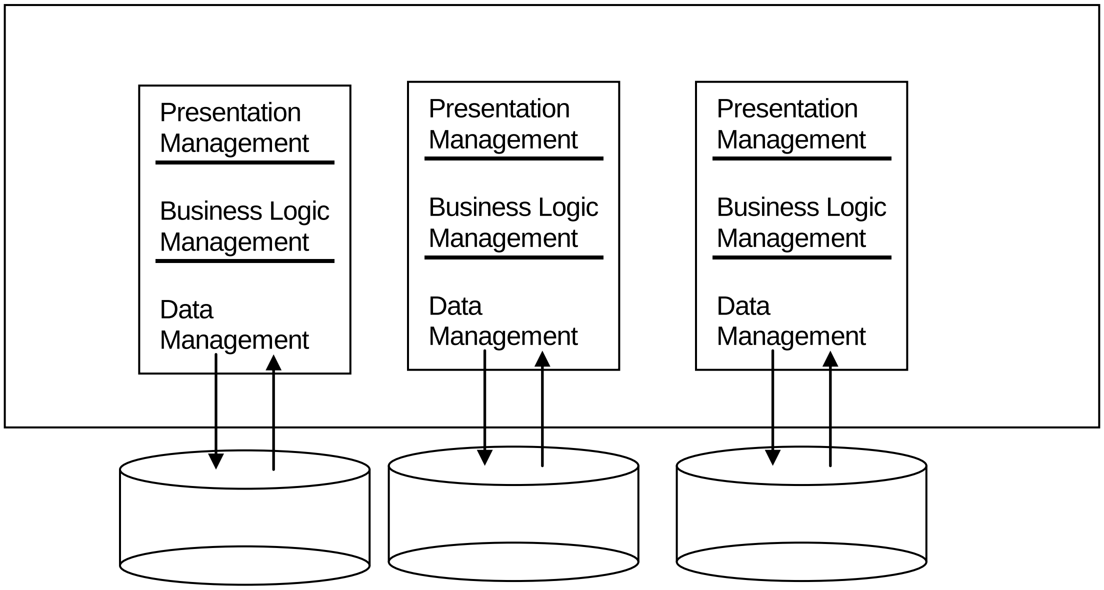


Figure 5.5. Mainframe Architecture showing the Independent Data Pools of each Application

The figure above depicts typical mainframe architecture in which several applications would run on the same mainframe concurrently. Notice that each application resides entirely on the mainframe. Notice how each of the data pools were completely independent and non-integrated. This absence of data integration prevented management from being able to combine the data in new and useful ways that would've resulted in better information and better decision making. One recent development within the accounting/finance area is the interest in using activity-based costing to calculate a unit cost on products produced. To do this the software needs two types of data, accounting/finance data and logistics/manufacturing data. Traditionally, these data were isolated within independent data pools. Today, all of the data resides behind a single database engine which is able to provide any one application, such as activity-based costing, access to all of the data.

The following tree of additional current reality becomes apparent.

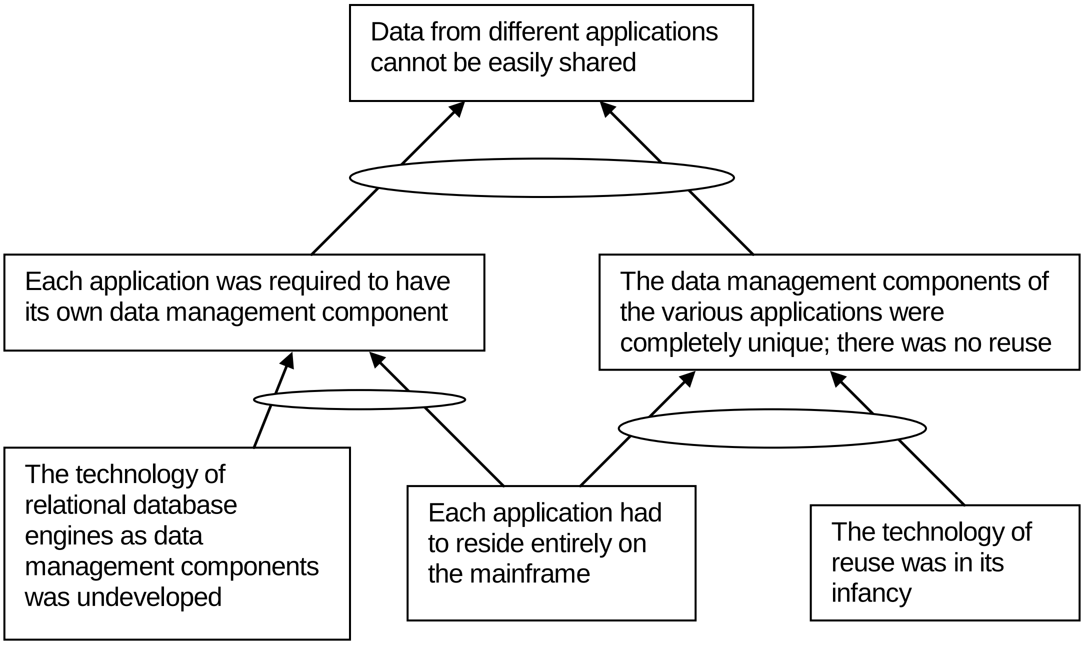

Data from different applications cannot be easily shared
Each application was required to have its own data management component
The data management components of the various applications were completely unique; there was no reuse
The technology of relational database engines as data management components was undeveloped
Each application had to reside entirely on the mainframe
The technology of reuse was in its infancy

Figure 5.6. Current Reality Tree showing Why Data from Different Applications Cannot be Shared

From the above it should be apparent that a "core problem" or "root cause" was the fact that _each application had to reside entirely on the mainframe_. This will be our "focal point" in terms of answering the question "what to change."

The next major question addressed by the methodology is "what to change to?" To do this, we begin by stating the core problem in its contra-positive format: "_applications may reside partially on the mainframe or server and partially elsewhere_" And we ask our selves, "is there any technological or other reason why this cannot happen?" Is there a conflict here? Apparently, there is no conflict; simply the early technology would not allow for this to happen.

This simple technique is only a fraction of the total Goldratt methodology and already, we have a good feeling about what the core problem actually is. Notice how distributing the application across the enterprise leads to solution of a number of symptoms or difficulties. It opens the door for data sharing. It gets some of the computing off of mainframes which were computing bottlenecks. It enables PC's to be better utilized, as portions of the application are loaded onto them. It decentralized corporate computing so there is less dependency on the centralized MIS shop where lead times are 36 months long.

Without the use of this technique, you could expect each of the symptoms to be addressed separately, with southern-engineered fixes that in total would probably be more expensive and less substantial in terms of symptom removal. For example, users would hire MIS professionals to build applications that ran on their local hardware to get around the 36 month backlogs in the centralized MIS shop. And, some mainframe legacy applications would be modified to reach into and fetch data out of other legacy application data sets to fix the problem is not being able to reach other applications data sets. For further information on the Goldratt Thinking Process, consult Dettmer, [9].

A causal loop diagram ala Forrester and Senge reveals other negative ramifications of having applications reside entirely on the mainframe. Because mainframe applications can only be maintained and created by use of a centralized MIS shop, that shop will see its backlog of unfilled change orders grow. As the lead times get longer, departments will attempt their own "work arounds," resulting in less integration of data and applications as islands of automation are created.

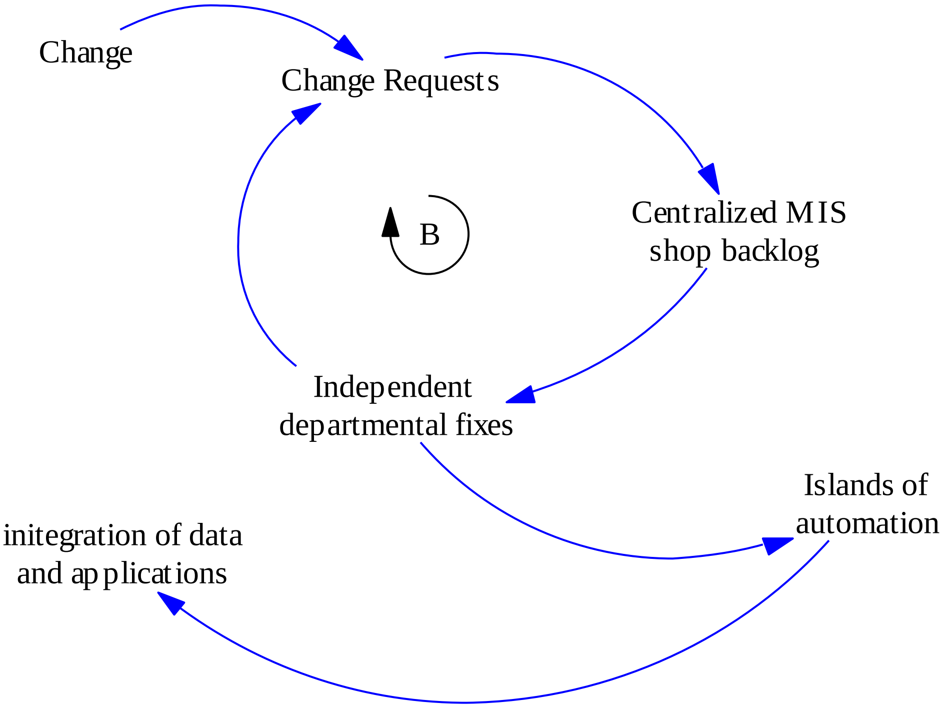

Change
Change Requests
B
Centralized MIS shop backlog
Independent departmental fixes
Islands of automation
initegration of data and applications

Figure 5.7. Causal Loop Diagram Showing how Change Creates Islands of Automation, leading to Integration of Data and Applications

This causal diagram is telling us that, as the backlog is growing, other undesirable things are happening. Namely, departments are creating their own little "work-arounds" that involve localized applications running on departmental computers with their isolated data pools. These actions would take some of the pressure off of centralized MIS shops but result in yet other undesirable outcomes.

For further information on the use of causal modeling consult Anderson and Johnson (1997).

What goes into a Requirements Document?

Your requirements document should consist of the following:

***Introduction*.** Identify the company, the vendors to whom the RD is targeted and state the problems that need to be fixed, the history, examples of the problem situation, motivation to fix it, etc. Describe the business case.

***Project Goal*.** A simple statement of why you are proposing the project. Major constraints of time or money can be mentioned.

Major functions . Simple statements about how the system will function, based on the project goals are to be included here. Each statement is a separate requirement which must be identified by a reference number, such as RD-MF-1, RD-MF-2, RD-MF-3, etc. For example, a search capability may be needed to identify a specific record. This might be stated as follows:
RD-MF-1. A search engine will be required that is able to conduct searches on a customer table so as to retrieve specific customers by name, by social security number and by policy number.

Outputs Simple description of information required from the system. Each major output set (such as a form, or web-page, or other integral output) should be identified by a reference number, such as RD-GO-1, RD-GO-2, RD-GO-3, etc.

Inputs Go through the list of output items above, and see what input data are necessary to produce the outputs. These should also be categorized in groups and identified by a reference number, such as RD-GI-1, RD-GI-2, RD-GI-3, etc. Since each set of inputs was determined by recourse to a General Output set determined above, each RD-GI should reference a specific RD-GO as the "driver."

Performance. How many transactions are to be processed in what period of time, how much data must be stored, how frequently must reports be produced. Each performance requirement should be identified by a reference number, such as RD-P-1, RD-P-2, RD-P-3, etc.

Growth. In this section you're trying to include enough capacity for growth over a specified period of time. Each growth requirement should be identified by a reference number, such as RD-G-1, RD-G-2, RD-G-3, etc.

Operation and Environment. Where the computers will reside, where the terminals are, who will use them. Each operation and environment requirement should be identified by a reference number, such as RD-OE-1, RD-OE-2, RD-OE-3, etc.

Compatibility, Interfaces Deals with data communication issues, distributed computing, and distributed databases, etc. Each performance requirement should be identified by a reference number, such as RD-CI-1, RD-CI-2, RD-CI-3, etc.

Reliability, Availability. Quotes MTBF and MTTF. Each performance requirement should be identified by a reference number, such as RD-RA-1, RD-RA-2, RD-RA-3, etc.

Human Interface Outline the computer experience required of the user, state how the system is to handle the brand new user. Each performance requirement should be identified by a reference number, such as RD-HI-1, RD-HI-2, RD-HI-3, etc.

***Organizational Impact*.** Which departments will be affected and how must their work be changed. Each performance requirement should be identified by a reference number, such as RD-OI-1, RD-OI-2, RD-OI-3, etc.

***Maintenance and Support*.** Warranties required: how long, to what extent, how it will be delivered. Each performance requirement should be identified by a reference number, such as RD-MS-1, RD-MS-2, RD-MS-3, etc.

Documentation and training List the general documents and/or courses that would be required. Each performance requirement should be identified by a reference number, such as RD-DT-1, RD-DT-2, RD-DT-3, etc.

Stakeholder Signatures and Signoff. Provide here places for signoff of all identified stakeholders.

As previously mentioned, requirements documents can be made into **requests for proposal. A request for proposal is a formal way for users and customers to communicate their requirements to contractors who would then competitively bid on doing the work.**

Defining team roles and Determining the Organization

A determination of team roles within this stage will help in the planning, scheduling and budgeting of the next stage. While determination of the roles seems appropriate for this stage, it doesn't make sense to assign resources or determine task durations. That can only happen after the Go/No Go decision has been made and the project is allowed to move into the next stage.

Defining roles at this point simply means acknowledging that specific area of expertise will be needed to complete certain aspects of the project. A project manager may know "on the surface" that end users from marketing and sales should be on the team, that a network specialist is required and that several Internet developers are needed. This is the kind of role identification required at this point.

Assess Feasibility

As indicated in Figure 5.1, feasibility assessment consists of several steps. These are listed below for convenience and discussed in what follows.

1. Identify dependencies with other projects
2. Assess overall risk
3. Test alignment with/impact on strategies and plans
4. Test resource availability
5. Submit deliverables for a quality gate inspection
6. ***Make go/no-go decision***
7. Obtain funding
8. Review alternative approaches
9. Obtain necessary signatures/signoffs
10. Move to next stage

In the literature on project management one can find suggestions of the use of quality gates through which the project must pass before it is allowed to proceed on to the next stage. At the end of every stage, there ought to be a quality gate in which the deliverables and other documents, are inspected and a judgement is made as to whether these are complete and the project should be allowed to proceed on to the next stage. If these are judged to be inadequate, this does not kill the project, but it does mean that the project cannot be allowed to move on to the next stage, until the "quality" problems are fixed.

As part of the feasibility assessment, a number of things ought to be considered as indicated in the list above. Consider first, "dependencies with other projects." If there are dependencies, then the importance of this project is partly derived from the importance of the other projects on which it depends. If a project in question provides a critical kernel for another very important project, then your project becomes very important. On the other hand, if your project provides a piece of functionality considered "sufficient and desirable," but not "necessary," then that piece may get scrapped or postponed in times of budget austerity and negative cash flow.

Risk assessment is always an important step in this stage of considerations/ deliberations and a thorough consideration of it appears in the next chapter, Chapter 6. Risk vs. reward ought to be an important parameter in the final GO/NO GO decision. It is sufficient to say here, that risk should take into consideration a number of risk factors, such as technology risk, end-user acceptability risk, financial risk, and miscellaneous risks. The latter would include such situations as improper working environment, failure of other party-supplied resources to materialize when needed, inappropriate technical team members, etc.

The next step listed above addresses the question, “suppose the project is pulled off with complete success and precision---so what?” Will the project make a solid contribution to the corporate bottom line? Can the firm expect a monetary payback in 12 to 18 months? Does the project align with the overall business strategy and plan of the firm?

At this point it should be possible to do a rough-cut determination of the number of persons that will be involved in the project. On the basis of that, the project manager can, along with other stakeholders such as line managers and upper managers asses whether there is enough idle human resources to do the project inside the firm, or whether an outside contractor is going to be needed. Having done that, an assessment of resource availability can be performed and a rough-cut strategy for acquisition of the needed resources can be determined.

Feasibility studies will assess risk, determine benefit/cost, and calculate overall comparative gain for each project. These are subjects about which much will be said in the next chapter.

The deliverables of the project at this point include the SOW, the project charter, the requirements document and a feasibility statement. These should all be given to the designated quality gate inspector who will judge whether the project has a good definitional grounding, whether there is strong consensus among the stakeholders, whether requirements are thorough and complete, etc. This assessment is not whether to do the project or not, but instead whether this stage has been thoroughly and conscientiously completed. If the answer is NO, then certain aspects of the stage may have to be re-completed. The project sponsor may be qualified to make such judgements in behalf of the project he or she is funding.

If the quality gate inspection surfaces a YES, response, then it is time to convene all of the stakeholders in a meeting that will determine whether the project is a go or not. In front of the stakeholders either prior to or at the time of the meeting should be all of the deliverables of this stage—SOW, charter, requirements document, and feasibility statement.

Assuming the result of the GO/NO GO session is a YES, then a commitment to fund should be forthcoming, and there should be a signature signoff on the requirements. The project manager should assess alternative methodologies, solutions and provide explanation as to why the chosen product and process are the best solution to this problem. All of this should take place within the GO/NO GO session. If all is well, the project should move to the next stage, planning and budgeting.

Project Selection: Multi-attribute Utility Theory

Once the cost of all the alternative proposals has been computed, it is possible to relate the cost to the proposal grade. This is done by means of a scatter plot as shown in Figure 5.8. Notice that the horizontal axis is the present-value, after-tax cost whereas the vertical axis is the grade on a scale of zero to one or ten, depending on how the attribute grades were assigned.

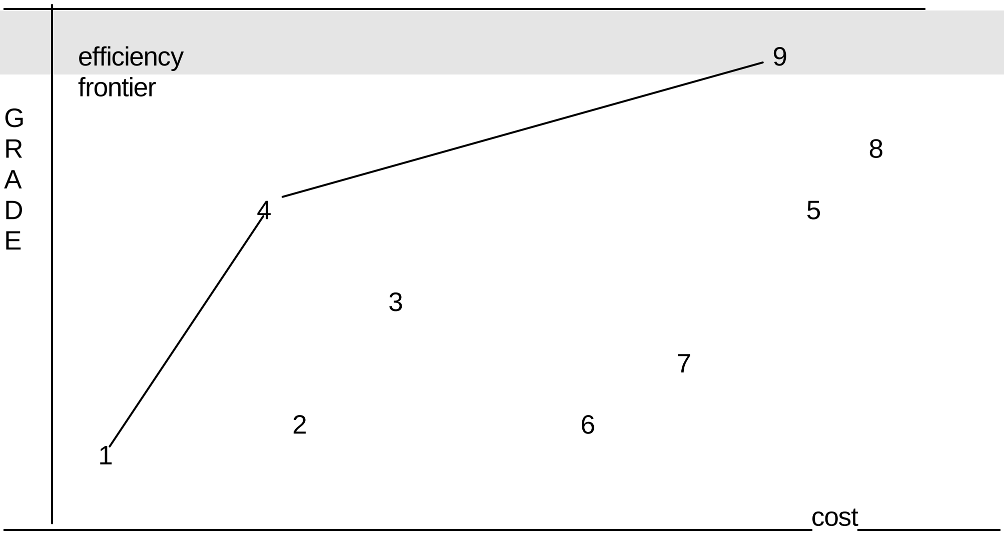
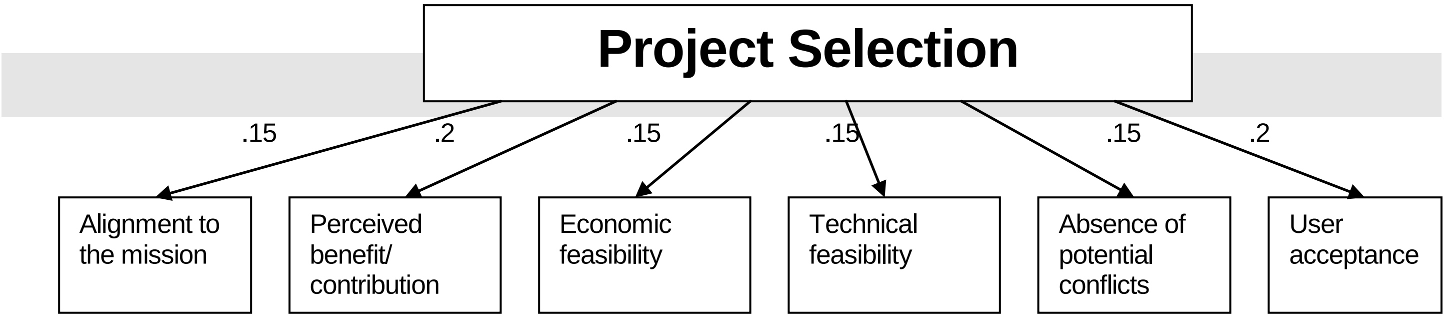


Figure 5.8. Plot for Relating Proposal Cost to Grade

Each number shown in Figure 5.8 above corresponds to a proposal. Each numbered proposal is positioned in the chart on the basis of its grade and its cost. Proposals worthy of further consideration are generally those that are on the efficiency frontier. It should be clear, for example, that proposal 4 would be preferred to proposal 5 because both have roughly the same grade or performance, but proposal 5 is much more costly.

Each proposed project must be assessed in terms of whether that project should be funded. To make such decisions, each project must be graded. Then based on the grade and the project's sequencing relative to other projects, a decision regarding funding must be made. The funding decision is the GO/NO GO decision. Grading is accomplished by use of a multi-attribute grading technique that we shall illustrate. Based on the grades, decisions regarding project funding can be accomplished by use of a capital investment model, involving the use of integer programming, if project administrators want to get sophisticated.

First, each project is graded. To do this managers must decide what dimensions to use for grading. The dimensions might include the following: alignment to the mission of the organization, perceived benefit/contribution, feasibility, potential conflicts (with other projects, say), user acceptance. The dimensions have to be mutually exclusive (no overlap between them in terms of concept). The dimensions should also be collectively exhaustive; that is, all bases should be covered. The evaluation should not leave out any consideration that might impact on the final selection. Feasibility might have several dimensions to it like economic feasibility, and technical feasibility. Economic feasibility has to do with whether the organization can afford the project, whereas technical feasibility has to do with whether the organization has the right technical resources to achieve the goal(s) of the project. Neither of these questions can be addressed very well until thorough planning and budgeting of the proposed project takes place.



Project 1	85	80	70	30	90	90
Project 2	80	78	90	75	80	75
Project 3	75	88	85	85	85	75

Figure 5.9. A Multiattribute Tree for Grading Project SOWs

An overall grade for each project can be calculated as the sum of the products of the individual dimension grades and the dimension weights, as illustrated below:

	.15		.2		.15		.15		.15		.2		
					Eco		Tech		No		Use		
	Alignment		Benefit		Feas		Feas		Confl		Accept		Total
Project1	85	12.8	80	16	70	10.5	30	4.5	90	13.5	90	18	75.3
Project2	80	12.0	78	15.6	90	13.5	75	11.3	80	12	75	15	79.4
Project3	75	11.3	88	17.6	85	12.8	85	12.8	85	12.8	75	15	82.1

Clearly, Project 3 is the winner here with the highest score, of 82.1.

Projects are not the only entities that can be graded using this methodology. Project proposals as generated in response to an RFP can be graded in this fashion as well. See the exercises for examples of this.

A somewhat more complex project selection problem is one in which overall grades for each project can be calculated as above, but in which it is possible to select several projects instead of just one. In addition, there are also a collection of pre-specified conditions, constraints on the selection of projects, such as project two cannot be selected unless project one is selected; or, the total number of projects selected cannot exceed five, say; or, the total budget for all projects cannot exceed $5 million, for example. These types of project selection problems are best handled using a discipline called "mathematical programming."

Personal Project Selection

The criteria by which individual project professionals make personal project management selection decisions is very different from the criteria used by organizations. Personally, you would look for projects that have a high probability of success, that won't bankrupt the company should they go wrong and that fit well with your interests and concerns. However, you should beware that individuals tend to pick "pet projects" that may not represent the best use of the firm's corporate resources. This is where alignment with the organization's objectives is so important. Without it, people will use corporate resources to achieve their own personal objectives. And the best interests of the firm will not be achieved.

If you're getting into project management for the absolute first time, you should pick a project that is small, short in duration, limited in scope and involves a half-dozen or fewer people. The project should have a good probability of completion *on-time* and *within-budget*. Beware of projects whose contractual arrangements are fixed in terms of price, functionality and duration. Before you say 'YES" to one of these you should be certain that you can pull the project off successfully, which means you have to diligently plan and budget the whole project. This is the reason for the discussion of the tradeoff triangle in Chapter 3.

Infrastructure Project Selection

Value-added assessments may be very difficult for some infrastructure projects; nevertheless, these may be necessary in order to support other projects that have very obvious and tangible benefits. For example, projects that accomplish data integration across functional areas don't, of themselves, appear to add very much value. However, such projects are necessary to support projects that require data integration in order to do such value-added computations as activity-based costing, process integration and the like.

Projects that address broad organizational concerns are much more likely to return quick payback benefits to the bottom line, even though the measurement of tangible benefit may be difficult. A project that improves data security, increases organizational communications and information flow, and enhances customer service makes contributions across many areas of the organization and thus is said to address broad organizational concerns.

Infrastructure projects may be necessary prerequisites to other projects that have obvious importance. Many companies have a budget for projects covering a period from 1 to five years. They are faced with making capital decisions about which projects to fund, which to pass on for now. In the following section, you will find material on how this gets done, when there are dependencies amongst the various proposed projects. This is called capital budgeting.

Making Decisions Amongst Multiple Projects with Interconnectedness

With the help of linear programming, it is possible to formulate and solve an optimization model that specifies which projects should be funded in order to maximize some measure of corporate benefit, such as ROI or IRR. Linear programming is simply a model that can be solved within Excel using a tool called Solver. The resulting solution is an optimal one for the parameters used in the model. For persons unfamiliar with linear programming, this section will be difficult to master and can be skipped. An example follows.

A CIO named Bud Capital has a six million budget from which to spend on IT enhancement within the various functional areas of the firm. So far just the basic ERP modules have been implemented—

accounting/finance, human resources, and manufacturing/materials management. The marketing side of the house is most interested in implementation of a customer relationship management system. However, before the CRM module can be installed, a data warehouse facility is required and before a data warehouse facility can be implemented much data must be "scrubbed" and normalized for inclusion in the data warehouse.

The manufacturing and purchasing departments have extreme interest in a supply chain management module that will streamline the procurement of raw materials from the firms' suppliers and assist with new product development. The supply chain management module requires a data warehouse. Fortunately, the same data warehouse that supports the CRM system marketing is interested in can be used. But again, data to be included in the warehouse require preparation. They are also looking at a finite-capacity scheduling (FCS) module, but this module cannot be implemented until the supply chain management module has become operational.

Engineering is interested in a new CAD/CAM system as well as programs that will do process flow analysis, manufacturing costing, process charting and value-stream mapping. At least two of these must be purchased together as a suite.

Finance has their sights set on a new collaborative product commerce (CPC) module that manages data throughout the lifecycle of the product as well as coordinates product and process redesign. However, the CPC system requires the supply chain management module as a prerequisite.

So as not to bias the project work toward one functional area, Bud Capital does not want to do more than three projects within any one functional area. However, because the data-warehousing project accommodates two functional areas, marketing and manufacturing, it will not be counted for either. The three functional areas are marketing, engineering, and manufacturing.

The table below lists the various projects Bud has to decide upon and provides some measure of benefit for each, as determined from a multi-attribute grading strategy previously discussed.

PROJECT NAME	ABBREVIATION	BENEFIT SCORE (0-100)	COST
Customer Relationship Management	CRM	85	1.5M
Data Warehouse Facility	DWF	95	1.3M
Scrubbing of Marketing Data	SMD	55	.5M
Supply Chain Management	SCM	80	1M
Supply chain data Prep	SCD	50	.4M
CAD/CAM System	CAD	75	.8M
Finite Capacity Scheduling System	FCS	30	.2M
Process Flow Analysis	PFA	35	.01M
Manufacturing Costing Analysis	MCA	30	.01M
Process Charting System	PCS	30	.01M
Value Stream Mapping	VSM	40	.01M
Collaborative Product Commerce	CPC	85	1M

For simplicity, we will use the abbreviations above as the decision variable names of the projects Bud wants to consider. Each of these variables is a binary integer variable in the sense that it will be allowed to take on a value of 0 or 1 only. A value of 0 indicates the project will not be funded; a value of one indicates the project has been selected for funding within the five million budget allotted Bud. The selection will be accomplished by the algorithm used within Solver. Mathematically formulated, the model is the following

MAX 85*CRM + 95*DWF + 55*SMD + 80*SCM + 50*SCD + 75*CAD + 30*FCS + 35*PFA + 30*PCS + 40*VSM + 85*CPC

s.t.

1.5*CRM + 1.3*DWF + .5*SMD + .1*SCM + .4*SCD + .8*CAD + .2FCS + .01*PFA + .01*PCS + .01*VSM + 1*CPC <= 6

CRM – DWF >= 0

DWF – SMD >= 0

2SCM – DWF – SCD >= 0

FCS – SCM >= 0

CAD + PFA + MCA + PCS + VSM >= 2

CPC – SCM >= 0

An integer programming solution to this problem suggests there is budget sufficient to cover all projects except the CAD/CAM project, resulting in 615 units of benefit at a cost of $5.94 million.

Summary and Conclusion

In this chapter we have endeavored to delineate what is involved in the first and most important of the project stages, Definition and Conceptualization. We have explained how the concept for a project gets started and how it will proceed from there.

The bottom line for the project manager in this stage of the project game is this. Hope to get assigned to the project early so as to be intimately involved in the selection of stakeholders, the definition and conceptualization of the deliverables and in building consensus among all stakeholders. The project manager should make sure that the stakeholders involved understand the project and agree on what constitutes success. This involves using expertise and charisma to lead the stakeholders even though they have more formal authority than the project manager does. They need to be led by someone more focused on the needs of just the project itself. They have assigned a surrogate, a project manager who is more technically consistent with the needs of the project and more driven by the project itself than they are.

In order to plan the project, well-defined goals, deliverables and scope for the project are required. For this reason, this stage stops with definition and conception, to give the project participants time to “catch their breath” and do a quality check. Did we adequately define requirements, charter and deliverables? Is there strong consensus among project participants? Is the project feasible; i.e., aligned with the strategic business plan, resourceable from either inside the firm or from without, and sufficiently beneficial.

In this chapter we presented a methodology for facilitation of JRD sessions. The basic motivation behind this methodology is to implement breakthrough solutions to problems, not just temporary work-arounds. Using Goldratt's thinking process, we proposed the use of his current reality tree to map symptoms, undesirable effects back to a root cause or core problem. The basic idea here is to find solutions to root causes, not just not just short-term fixes for symptoms. The current reality tree is the first of five constructs Goldratt gives us for problem-solving, critical thinking. The others are conflict resolution diagram (which we studied in Chapter 3), future reality tree, prerequisite tree and transition tree. We will study some of these in forthcoming chapters. For example, the transition tree helps us determine the methodological steps to achievement of a breakthrough solution determined by the thinking process

Finally, nowhere in this first stage was there ever a step in which quick and dirty estimates of duration and cost were determined. It is the author's opinion, backed up by the experience of many in project management that "ball park" estimates have gotten firms and project managers in trouble. This is why a cycle, involving this stage with the next, is proposed in which thorough planning and budgeting is performed before ever once pronouncing any estimates at all. As project manager, you should refuse to give any stakeholder any estimates of duration and cost until the next stage, discussed in Chapter 8 is rigorously performed. Toward that end, let's move on.

Chapter 5: Exercises

Define what is meant by:

Alignment
Decomposition
JRDS
Feasibility
Project charter
Project governance
Project rules
Project scope
Resource availability
Stakeholders
Work Breakdown Structure

Exercises

1. Name some deliverables of the conceptualization and definition stage of the project lifecycle.

2. Why is iterative/repetitive application of this phase with the next major stage, planning and budgeting sometimes necessary?

3. Describe why it is impossible to begin the next stage planning and budgeting if this stage is not done with thoroughness.

4. In addition to deliverables, what are some of the other "items" that require definition within the project charter?

5. Do you think it is a good idea to delineate at a high level the processes that will be used within the project in this stage? Why or why not?

6. How would you handle the problem of differing customer and stakeholder expectations?

7. Describe a simple technique for stakeholder analysis.

8. When in your opinion is the best time to bring on the project manager? How about the project team members? Why?

9. A firm has decided to move into the next stage, Planning and Budgeting, without a project charter, citing that everyone already knows about this project. However, the firm does not do many projects, hasn't done one for a while and the projects that have been done have been quite heterogeneous. What problems do you anticipate down the road?

10. In the Goldratt and Senge approaches, the first thing that must happen in the JRP session is what?

11. Construction of the Goldratt current reality tree begins with what items first?

12. In the Goldratt methodology, what construct is used to find an injection, a breakthrough solution? Where else has this diagram been used in our previous discussions?

13. Construct a current reality tree for the following situation. A corporate client/server architecture is now more than seven years old. IT has noticed that applications, because they reside mostly on clients are hard to maintain. The developers were decentralized, mixing with the end-users directly. However, the functional areas in which these MIS folk "lived" did not have strong career paths for them. So some of them left. Servers are now being used instead of mainframes and unlike, seven years ago, server MIPS are cheap, less than $10 per MIPS and readily available. Users both inside and outside of the firm want an Internet-like user interface. The firm is constantly changing and has a difficult time keeping its applications current with the rampant process change. Because the business logic portion of the application is hosted by the client, there is little standardization of processes, components. The various applications do not use a WEB browser as the "standardized" front-end. Construct the logic necessary to work backwards from the undesirable symptoms described here to a core problem. Determine what that core problem is.

14. Construct a current reality tree for the following situation. The current inventory system that is being used at Tom's Restaurant is out of date and very inefficient. The method of keeping track of live bands that will perform at Blue Light (also owned by Tom's) is also not very efficient. Tom is also needing to track the certification of the bartenders. Currently, Tom's uses an 800MHz Pentium II computer machine with 5 Gbytes of hard drive storage that is 90% full. Tom's will need to upgrade its current computer system(s) to handle this new, automated system.

15. Construct a current reality tree for this situation. A&T and Associates is a company in Denver, CO. Their main focus is to provide a web site for various antique dealers throughout the Denver area. Currently their service entails someone going out to the dealers stores. An outside picture is taken, an inside picture is taken, and various pieces the dealer chooses are photographed. The quantity of individual photos determines how much the service will cost. There is an up-front charge for this service and the contract is for A&T to support their web site with those images taken for one year. The following details will have to be implemented for more efficient production of web pages, to increase value to the dealers (thus more revenue for A&T), and for the possibility of going national. Up to this point, the web pages provided to the dealers have been static. When you visit the home page you have the option of searching for dealers by geographic location or by the dealers name. The geographical search isn't like a database. There are static listings of major areas of Denver that have a cluster of antique dealers. The search for dealers by name is also a static listing that shows the dealers in alphabetical order. Once a user selects a dealer that he or she is interested in, that dealer's web page comes up. The format for all the dealers' web pages is somewhat generic. Image backgrounds may change as well as page backgrounds, but the layout of the images and the links to other areas of that dealer's directories are the same. This is a small company with approximately 90 dealers who are the customers. In order to gain more customers, some formats and layouts are quite different at the dealer's request. This makes the task of creating a new non-generic web page more time consuming. The owner works 90+ hours a week going out to dealers and selling his service, taking new pictures, renewing customers, creating web pages for new dealers, and editing images. He is struggling to stay on top of the things he has to do and is looking to find a way to make his functions more efficient and at the same time have the value of his service rise or at least maintain.

16. Describe the difficulties associated with the opening scenario associated with this chapter. Delineate a current reality tree for the project. What would you say is the core problem? How could the money's have been better spent?

17. Create a work breakdown structure for the following list of tasks, using the WBS notation in the left-hand column.

No.	NAME
1	SOFTWARE DEVELOPMENT PROJECT
1.1	Definition
1.1.1	Interviews
1.1.2	Prep Req.Doc.
1.1.3	Prep Pre Proj Plan
1.1.4	Prep Proposal
1.2	Analysis
1.2.1	Interviews
1.2.1.1	Management
1.2.1.2	Supervisor
1.2.1.3	Technical
1.2.1.4	Clerical
1.2.2	Anal Existing Doc
1.2.3	Synthesis
1.2.4	Functional Spec
1.2.5	Re-estimate
1.2.6	Development Prop
1.2.7	Presentation
1.3	Design
1.3.1	System Design
1.3.2	Walk-through
1.3.3	File Design
1.3.4	Walk-through
1.3.5	Acc. Test Plan
1.3.6	Design Spec
1.4	Construction
1.4.1	Plan the integration
1.4.2	Module Design
1.4.3	Walkthrough
1.4.4	Plan module testing
1.4.5	Coding
1.4.6	Module Test
1.4.7	User documentatin
1.5	System Test
1.6	Acceptance Test
1.7	Operation

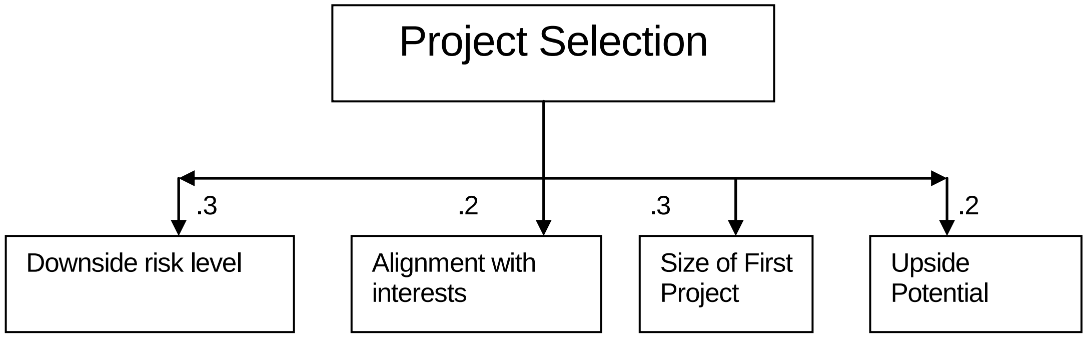
18. Create a work breakdown structure for the following list of tasks, using the WBS notation in the left-hand column. Do only the first three levels of the WBS.

No.	NAME
1	Infrastructure Deployment Template
1.1	Scope
1.1.1	Determine project scope
1.1.2	Secure project sponsorship
1.1.3	Define preliminary resources
1.1.4	Secure core resources
1.1.5	Scope complete
1.2	Analysis
1.2.1	Review Current Infrastructure
1.2.1.1	Review hardware environment

1.2.1.2 Review software environment
1.2.1.3 Review communications environment
1.2.1.4 Review connectivity LAN/WAN
1.2.1.5 Review support environment
1.2.1.6 Review geographic factors
1.2.1.7 Review current infrastructure complete
1.2.2 Review business goals/direction/vision
1.2.3 Identify Target Areas for Improvement
1.2.3.1 Identify hardware improvements
1.2.3.2 Identify software improvements
1.2.3.3 Identify communications considerations
1.2.3.4 Identify connectivity LAN/WAN improvements/issues
1.2.3.5 Identify support environment improvements/additions
1.2.3.6 Identify mitigation of geographic factors
1.2.3.7 Identify target areas for improvement
1.2.4 Define system requirements
1.2.5 Define target performance metrics
1.2.6 Review Current Market Solution Vendors
1.2.6.1 Hardware vendors
1.2.6.2 Software vendors
1.2.6.3 Communications vendors
1.2.6.4 Design partners
1.2.6.5 Implementation partners
1.2.6.6 Review current market solution vendors complete
1.2.7 Analysis complete
1.3 Design
1.3.1 Secure necessary architectural resources
1.3.2 Draft Preliminary Infrastructure Design Documents
1.3.2.1 Preliminary hardware design
1.3.2.2 Preliminary software design
1.3.2.3 Preliminary communications design
1.3.2.4 Preliminary connectivity LAN/WAN design
1.3.2.5 Preliminary support environment design
1.3.2.6 Draft preliminary infrastructure design document complete
1.3.3 Review preliminary design documents
1.3.4 Obtain feedback/input on design
1.3.5 Develop Detailed Infrastructure Design Documents
1.3.5.1 Develop detailed hardware design
1.3.5.2 Develop detailed software design
1.3.5.3 Develop detailed communications design
1.3.5.4 Develop detailed connectivity LAN/WAN design
1.3.5.5 Develop detailed support environment design
1.3.5.6 Develop detailed infrastructure design documents complete
1.3.6 Align long/short term infrastructure design with business goals
1.3.7 Design complete
1.4 Budget
1.4.1 Develop budget based on detailed infrastructure documents
1.4.2 Align budget request with business goals
1.4.3 Align budget request with infrastructure requirements timeline
1.4.4 Present long/short term budget request
1.4.5 Secure multi-phase/year funding for key initiatives
1.4.6 Budget complete
1.5 Finalization/Validation
1.5.1 Secure resources based upon approved initiatives
1.5.2 Finalize detailed design based upon budgetary considerations
1.5.3 Develop detailed implementation strategy

1.5.4 Validate implementation strategy in test environment
1.5.5 Review implementation strategy noting other initiatives
1.5.6 Secure approval to proceed
1.5.7 Finalization/validation complete
1.6 Deployment
1.6.1 Logistics
1.6.1.1 Secure deployment resources
1.6.1.2 Train deployment resources in deployment methodology
1.6.1.3 Logistics complete
1.6.2 Pilot
1.6.2.1 Select infrastructure component to deploy
1.6.2.2 Review deployment team tasks and timeline
1.6.2.3 Communicate impact to community
1.6.2.4 Deploy infrastructure component
1.6.2.5 Test infrastructure component
1.6.2.6 Release to production/operations environment
1.6.2.7 Obtain feedback
1.6.2.8 Evaluate pilot feedback
1.6.2.9 Pilot complete
1.6.3 Determine readiness to proceed
1.6.4 Final Deployment
1.6.4.1 Review deployment team tasks and timeline
1.6.4.2 Communicate impact to community
1.6.4.3 Deploy infrastructure components
1.6.4.4 Test infrastructure component
1.6.4.5 Release to production/operations environment
1.6.4.6 Obtain feedback
1.6.4.7 Final deployment complete
1.7 Post Implementation Review
1.7.1 Document lessons learned
1.7.2 Distribute to team members
1.7.3 Create software maintenance team
1.7.4 Post implementation review complete
1.8 Infrastructure deployment template complete

19. Suppose that Sharon, in the case discussed at the end of Chapter 5, has come up with the following attribute tree. Note the weights Sharon has attached to each of her dimensions. She has total control over choice of dimensions and the weights attached to them.



project 1	.65	.95	.55	.75
project 2	.30	.50	.75	.85
project 3	.50	.50	.50	.80

Notice that Sharon has given project 2 a .3 for downside risk level because the project absolutely has to be completed in three months and because no-one really knows how large and time-consuming this project really is. She has given project 1 the highest grade because she really enjoys WEB-based development. Determine which of the three projects Sharon should select, based on this analysis. Does this approach match with the selection you made earlier for Sharon? What is your assessment of this approach?

20. An undergraduate MIS major has three job offers to consider. The attributes for evaluation taken in relation to the weights are:

ATTRIBUTES	WEIGHT	OFFER 1	OFFER 2	OFFER 3
location	.1	85	75	50
opportunity to advance	.2	90	95	60
alignment with personal goals	.2	75	80	85
training	.1	85	95	85
chemistry match with "boss"	.1	80	75	65
salary	.2	80	90	95
fringe benefits	.1	90	80	70

Draw the multi-attribute tree. What Is the overall grade of each offer? Which offer should be selected, based on the overall grade?

21. An automated inspection technology firm (a startup) is looking to create a world-wide-web presence on the Internet. The firm is anxious to get this facility going as soon as possible. The firm will need hardware, software as well as application software to support an active server pages technology. An RFP has been transmitted and proposals have been received. Each proposal must be graded on a scale of 0 to 100 where 100 is perfection and 0 is no contribution. The major attributes are **Software, Hardware, Upgrading Potential, Time Considerations, and Vendor**. The relative weights for each of these are .3, 2., .2, .1, and .2 respectively. The software component is further decomposed into **Systems software and Applications software**, each with relative weights of .5. The Time Considerations component is further decomposed into an Equipment Delivery date and an Application on-line-by-date, each with relative weights of .5.

Draw the multi-attribute utility tree. Determine the absolute weights of all the lowest-level elements on the tree. Determine the grade of each of the two proposals below. Assume the first costs $250,000, the second $350,000. What is the benefit/cost of each proposal? Which would you choose?

		Grades	
ATTRIBUTE	**WEIGHT**	**Proposal 1**	**Proposal 2**
Software			
Systems Software		78	78
Applications Software		85	90
Hardware		88	90
Upgrading Potential		90	90
Time Considerations			
Equipment delivery date		75	90
Application on-line-by		88	90
Manufacturer/Vendor		67	80

Chapter 5: Brief Case

A young IT professional has been "on the job" for just six months when she is approached by her boss, John. "Sharon, I have good news for you. Right now we have three new projects that need project managers, which we are short of. We'd like you to pick one of them. Because of your excellence here these past few months, we're giving you the right to choose which of these you'd most want to be project manager for. The first project is a re-write of the user interface for the enterprise sales module. The task there is to build a WEB-based, browser-driven interface to replace the old GUI/client/server one."

Sharon responded, "Cool. I enjoy building WEB interfaces as you know. How many subordinates will I have?" "Three," was the bosses answer.

"And how much time is allocated?" "Six months."

"How many screens," Sharon asked. "If you're talking about forms, we think there could be as many as fifteen---maybe more. We're not sure."

"Any database revision?," Sharon inquires. "Yes. We want to move the entire sales automation system into the WEB-based ERP system.

"Do you have a budget," asked Sharon. "We think it should take less than $300,000."

"What about the other two projects," queries Sharon, feeling a little overwhelmed by the size of the sales project, yet modest budget for it. "The second one involves replacement of a broad-band LAN with a cheaper but faster VDSL one based on copper. This would be for just the three buildings that make up the north campus here in Los Colinas."

"John, you know I prefer development. Why would you even consider me for this job?"

"Because you have extraordinary training in LAN software and systems, with your Microsoft certification in NT. Sharon, we already have installed the copper wiring throughout the complex. But we need someone with the maturity and competencies to manage this project. We believe you can make the component selections decisions associated with this project as well—choice of hubs, routers—stuff like that. Then the testing of the system using NT server and client software will be your baby. We believe you can bring this system live. Our current system is really slow and we have to get this up and running in three months before we complete our transition to the B2B WEB interfaces so we'll have the bandwidth to accommodate the extra load. Our customers and suppliers are expecting sub-second response times.

"Again, how many subordinates and how much budget?"

"We're going to let you out-source two network engineers to help you complete the project. As you know we don't have a lot of network expertise in this shop. Everybody here is busy so contracting this help from the outside makes the most sense. We believe the hardware costs will run $150,000 and the whole project will cost $450,000 max. It just needs to happen stat; I mean ASAP." "And the third project?," Sharon queried.

"The third project is a maintenance job on the old engineering CAD system. The engineers custom built this software package five years ago. They're hooked on it and we spent 25 million developing it. There's nothing like it, but it doesn't interface with the new ERP system and it doesn't even run in a client/server environment. Ultimately, we want real-time CAD/CAM product development and production so we can be putting our ideas into the market while the competition is still sharpening their pencils. This will be a major re-write of this entire system to convert it over to operate with our DB2 database and to inter-operate with the marketing and manufacturing and CAM enterprise modules. This is a must if we are going to decrease our new product introduction lead times. You would have six people to work with you on this and a budget of at least $1.2M, plus a year to complete the project.

What additional questions, or requests would you make of Sharon's boss, John? Which project would you choose for Sharon and why?

References

1. Baker, Sunny and Kim Baker, **The Complete Idiot's Guide to Project Management**, New York: Alpha Books, 1998.

2. Chalfin, Natalie, "Four Reasons Why Projects Fail," PM Network, Project Management Institute, June 1998, p. 7.

3. Kliem, Ralph L., Irwin S. Ludin, and Ken L. Robertson, **Project Management Methodology**, New York: Marcel Dekker, Inc., 1997.

4. Lientz, Bennet P., and Kathryn P. Rea, **Breakthrough Technology Project Management,** San Diego: Academic Press, 1999.

5. Senge, Peter, **The Fifth Discipline: The Art & Practice of The Learning Organization**, New York: Doubleday/Currency, 1990.

6. Verzuh, Eric, **The Fast Forward MBA in Project Management**, New York: John Wiley, 1999.

7. Anderson, Virginia & Lauren Johnson, **Systems Thinking Basics: From Concepts to Causal Loops**, Cambdridge, MASS: Pegasus Communications, Inc., 1997.

8. Dettmer, H. William, **Goldratt's Theory of Constraints: A Systems Approach to Continuous Improvement**, Milwaukee, Wis.: ASQ Quality Press, 1997.

Measurement

Measurement provides the antidote to the common problems of poor estimates, poor scheduling, and poor progress visibility. Companies that have active measurement programs tend to dominate their industries. Measurement should have high-level management commitment and be enacted through a permanent group. Measurement has a high potential for reducing the duration of the project schedule, for improving progress visibility, and for reducing schedule risk.

Software products and projects can be measured in dozens of ways: size in lines of code or function points, defects per thousand lines of code, hours spent designing, coding and debugging, and developer satisfaction

Measurement provides status visibility. The only things worse that being late with a project is not knowing that you are late. Detailed measurement can help improve visibility.

Measurement focuses people's activities. As Goldratt would say, "Tell me how you will measure me and I will show you how I will respond." People get focused on visible measurements that are rewarded. They will reduce defects if that increases their chances for a better merit raise; they will increase lines of code produced per week if they get more salary as a result. What gets measured and rewarded gets optimized by the project personnel.

Measurement improves morale. If you measure developer stress, developer satisfaction with tasks assigned, availability of resources for training, and you actively do something about these concerns when there is a "problem," you will get improvements in developer morale.

Measurement can help set realistic expectations. If, as a developer, you are asked to produce something in less time than you think it should take, and if there is a history database of measured durations, you can use the history database to back up your claim for inadequate time to complete the task.

Measurement lays the groundwork for long-term process improvement. The most significant benefit of measurement cannot be realized in the short-term on a single project, but will produce results over several projects as processes and practices are improved and measures provide hard evidence of that improvement. A measurement program helps you avoid wasting time on practices that aren't paying off. It also helps you identify sliver-bullet technologies that aren't living up to their claims. It enables you to accumulate a base of experience that will support more accurate project estimation and more meaningful planning. Measurement is the cornerstone of any long-term improvement program.

Using Measurement

There is a temptation to measure everything just in case you need it. A better practice is to allow the measurements to be driven by goals, questions and metrics (Basili and Weiss 1984). Goals determine how you want to improve your projects and products. One good goal is to reduce the number of defects that make their way into the software initially, which then take so much time to find and remove later is the lifecycle. Secondly, determine what questions you need to ask in order to achieve your goals. A legitimate question might be "what types of defects are most expensive to find and remove?" Finally, set up measurements (or metrics) that will answer your questions. For the goal and question above, one would start collecting data on defect types, creation times, detection times, cost to detect, and cost to correct.

Measurement Group. Obviously, this group would have to have knowledge of the goals, questions of interest to the larger organization. Additionally, the group should have knowledge in the following areas (Jones 1991) software:

1) statistics and multivariate analysis
2) software engineering
3) software project management
4) software planning and estimating methods and tools

5) design of data-collection forms
6) survey design
7) quality control methods, including reviews
8) walk-throughs, inspections, and all standard forms of testing
9) pros and cons of specific software metrics
10) accounting principles

Most measurement groups at AT&T, DuPont, Hewlett-Packard, IBM, and ITT have this skill set.

What to Measure. Based on its goal priorities and questions, each organization must decide what to measure. At a minimum most organizations will want to keep historical data on project sizes, schedules, resource requirements, and quality characteristics.

Classic Mistake. One classic mistake is not collecting data at a fine enough granularity so that analysis relative to planning future projects can be performed. Accountants require less granularity than do analysts, generally.

Development speed. If you are concerned about development speed, one of the most significant things you can do is apply Pareto analysis—look for the 20 percent of activities that consume 80 percent of the time. Optimizing a software project for speed is akin to optimizing a software program for speed—measure where you spend the most time and then look for ways to make the most time-consuming areas more efficient.

How much data to collect. One of the lessons NASA's Software Engineering Laboratory has learned over the years is to spend more effort on analysis and less on data collection. Today, NASA's SE Laboratory spends about three times as much on analysis as they do on data collection. Organizations just starting out collect about a dozen measurements; organizations that have been collecting data for some time collect about two dozen measurements. (Brodman and Johnson 1995)

Feedback. Once the measurement program is in place, you can begin to provide feedback to the developers, managers and stakeholders as to the results of the measurements. Only then can perceptual improvement begin to take place as people see how they are doing and how that is changing over time.

Baseline report. One specialized kind of feedback that measurement organizations provide is an annual software-baseline report. The baseline report is similar to an annual financial report, but it describes the state of the organization's software-development capability. It includes summaries of the projects conducted that year; strengths and weaknesses in the areas of people, process, product, and technology; staffing levels; schedules; productivity levels; and quality levels (McConnell 1996).

Measurement accuracy. The fact that something is measured doesn't mean that the measurement is accurate. Measurements of the software process contain lots of error (McConnell 1996). Sources of errors include unpaid and unrecorded overtime, charging time to the wrong project, unrecorded user effort, unrecorded management effort, unrecorded specialist effort on projects, unreported defects, unrecorded effort spent on prior to activating the project-tracking system and inclusion of non-project tasks. Capers Jones (Jones 1991) reports that more corporate tracking systems tend to omit 30 to 70 percent of the real effort on a software project.

Managing the Risks of Measurement

In general measurement is a risk reduction practice. The more you measure, the fewer places there are for risks to hide. However, measurement carries risks of its own. If you measure lines of code produced you might get some developers to change their style and make the source code more verbose. Some will completely forget about code quality and focus only on quantity. If you measure only defects, you might get a significant drop in productivity measured in lines of code produced per day or week or whatever. Measurements must create a balanced commitment to productivity and quality. Comparing lines of code produced for comparable projects across high-and low level languages can create some difficulties. The low-language will give the appearance of being more productive because far more lines of code are required. Similar biases can be created for comparisons of quality in quality in terms of defects/per 1000 lines of code.

Side Effects of Measurement

The main side effect of measurement is the effect it has on human behavior. The measure will get optimized by human behavior. If that is the behavior you want, fine; if not, then the measure does not achieve the results that are wanted.

Measurement's Interactions with Other Practices

A measurement program provides the foundation for improvement in areas such as estimation, scheduling, and productivity. A well-designed measurement program should contribute to rapid development.

Key concepts for Measurement

Setup a measurement group. Use goals, questions, metrics to decide on what to measure. Create a balanced measurement strategy. Track time accounting data to a fine-level of granularity. Analyze the collected data and feed it back to the people whose work it describes.

Measurement guru Capers Jones reports that organizations that have established a full software metric program have improved quality by about 40 percent per year and productivity by about 15% a year for four or five years. (Jones 1991, 1994). He points out that only a handful of U.S. organizations currently have accurate measures of software defect rates/removal and that those organizations tend to dominate their industries (Jones 1991). The cost for this level of improvement is typically from 4 to 5 percent of the total software budget.