software development The lifecycle starts here.

Decoding Business Needs

December 1999 – vol 7. no. 12.

Facilitated workshops reduce the risk of scope creep from 80% to 10%, accelerate the delivery of early lifecycle phases by 30% to 40%, and provide a 5% to 15% overall savings in time and effort throughout the entire lifecycle.

by Ellen Gottesdiener

Corporate software is designed to serve and support business needs, but how well it accomplishes its many missions often leaves much to be desired. Whether judged by internal measures (defects, process, or plan vs. delivery assessments) or external ones (surveys, metrics), it's clear that the industry falls far short of its ambitions.

In *Patterns of Software Systems Failure and Success* (Thomsen Computer Press, 1996), Capers Jones reports a high degree of risk in information technology projects, namely scope creep, schedule pressure and quality problems. The Standish Group's CHAOS report (www.standishgroup.com/chaos.html, 1995) shows that more than a quarter of software projects fail and that nearly half of all IT projects are "challenged" (completed, but over-budget, behind schedule, and delivering fewer features/functions than planned). Internal views of software production indicate high defect rates (see the Software Engineering Institute's Capability Maturity Model data at www.sei.cmu.edu/cmm/cmm.html) and immature software development processes.

The Standish Group suggests that the top four success factors for an IT project are, in ascending order: customer involvement, executive management support, clear statement of requirements and proper planning.

Active, visible business customer involvement is often elusive. Customers sponsor the project, specify requirements, test the product and use it after implementation. Yet developers are notoriously poor at establishing,maintaining and managing good customer relationships.

Producing successful software depends on five important factors:

- Business staff must have formal roles in IT projects.
- Business staff must participate from the beginning.
- Developers should not proceed with a project without business sponsorship.

- If business involvement is a risk factor, it should be part of a risk-management strategy including strong communications and an organizational change management plan.
- Internal project processes and techniques must maximize customer collaboration.

There are three practical ways to involve business customers in the software development process: establishing a project charter, using facilitated workshops and addressing organizational change.

The Commission

A charter begins with the end in mind, defining the who, what, when, where, why and how of the project. The charter documents:

- roles and responsibilities within the project organization (who),
- business functions in and out of scope, organizational scope, temporal scope, financial scope, and the priority of constraints such as time, cost, features/functions, quality, process and software metrics (what),
- the preliminary plan for the first phase of the project (when),
- where the work will get done and where the software will be deployed (where),
- goals and objectives (why), and
- risks and a risk-mitigation plan, methodologies, tools, assumptions, controls, a quality plan and knowledge transfer (how).

As Winston Churchill said, "The plan is nothing; the planning is everything." Indeed, the primary value of a project charter is not the resulting document but rather the process of creating, validating and closing the charter.

One of the roles that should be explicitly identified in the charter is the sponsor; that is, the person with the financial and logistical authority to make the project happen. The project manager must work with the sponsor to define what behaviors—actions and words—will be expected from the sponsor.

Depending on the organizational scope of the project, a more complex sponsorship team may be required. For example, I once participated in a global project that had an executive sponsor, a steering committee and a day-to-day project sponsor. In any case, the roles and responsibilities should be made clear to everyone connected with the project.

Displaying active, visible sponsorship entails making decisions, finding resources, promoting the project to peers and upper management and continually rewarding project members. Just as the act of specifying project risks tends to minimize them through team awareness, the very act of discussing, documenting and reviewing the expectations for the sponsor has the benefit of getting active sponsor participation. This simple activity is also a meaningful and important method of involving customers. Some sponsorship behaviors might include:

- defining and/or validating business rules and policies which are being implemented in the software,
- making decisions—and deciding how to make decisions—when multiple choices are possible,
- getting people resources to work on the project when they are needed,
- selling and marketing the value of the project,
- paying for the project or getting the money to pay for the project,
- making high-stake decisions quickly,
- ensuring that business objectives are being satisfied through the use of technology,
- ensuring that business objectives are being satisfied through the use of existing or new processes, and
- making the time to do all of the above.

In some projects, a written sponsorship contract can be useful as a concrete way for the team to ask for the behaviors it wants from the sponsor. If the sponsor is unable or unwilling to sign a sponsorship contract, it's better to know early. The project should not commence until sponsorship is secured.

How do you manage the chartering process? One way is to timebox the delivery of the charter in the kick-off phase initiated with a workshop for the project sponsor and the team. Prior to the workshop, portions of the charter should be drafted; decisions and definitions of missing or controversial pieces of the charter are made at the workshop.

In charter workshops that I have facilitated, the participants delivered:

- a list of functions that are in and out of scope,
- documented corrections or additions to the project roles and responsibilities,
- a visual scope model,
- a decision on the project sponsor,
- a list of risks,
- a ranking of risks and a risk- mitigation plan,
- a deliverables-based project plan,
- a statement of project strategy (project approach or method),
- a list of potential business analyses for the next phase,
- a project communications plan, and
- a decision on project constraints.

During the chartering stage, a sponsor must make decisions on roles/responsibilities, constraints and scope. But how does a team decide how it will decide?

Projects often get stuck when decisions are not made in a clear, timely manner. The sponsor is responsible for high-stakes decisions and must begin to exercise this authority early in the process. Using a decision rule and agreed-upon decision-making process is imperative.

At a recent business rules chartering workshop, I had to help the workshop sponsor define the deliverables of the encounter. One product was an intangible—a decision on the scope of business rules for the first phase of the project. As facilitator, I showed the sponsor a simple diagram of the possible decision rules she could exercise (leader decides after discussion, coin flip, delegation, majority rule, consensus, etc.). The sponsor is the leader, and thus is responsible for deciding how to decide a matter of project scope. (In other situations, the project manager or some other team role may inhabit this role.) Using a decision rule process, she could simultaneously check the participants' degree of agreement and model appropriate behavior.

To close out the chartering phase, conduct a walk-through workshop. This should result in a formal sign-off of the charter. If minor changes are still needed, the participants must agree on a sign-off date and responsibilities for those changes.

Facilitated Workshops

A facilitated workshop is a planned collaborative event in which participants, led by a neutral guide, deliver products in a concentrated period of time. Prior to the workshop, the participants agree upon what will be delivered and the ground rules for the interaction (including decision rules). The workshop process exploits the power of diverse groups of people joined together for a common goal. Participants act as a sophisticated team, working in a manner to achieve what psychologists call "consensual validation." A successful and productive workshop should be fun and energizing.

Facilitated workshops are most effective in the early stages of the software development lifecycle for chartering, planning, requirements, analysis and design. Not only can a well-run workshop provide the project artifacts (models, decisions, etc.) in a fast and high-quality manner, but it also has the benefit of building a team and establishing a spirit of real collaboration among all team members. Therefore, facilitated mid-point (or periodic) and debrief workshops are essential to team and project process improvement. Since business customers are participants in these workshops, and they provide about 19 percent of the total effort on IT projects, planning, requirements and debriefing workshops set the stage for and maintain active customer involvement.

Workshops for IT projects have their roots in Joint Application Design (JAD), a workshop technique developed and trademarked by IBM in the late 1970s. Since then, the process has evolved and been tailored to a variety of project types and technologies. The data around the increased quality and reduced costs of using JAD-like workshops are impressive.

According to Capers Jones, while 60% of software defects originate in the requirements and design phases, early facilitated workshops reduce those defects by 20% to 60%. Facilitated workshops reduce the risk of scope creep from 80% to 10%, accelerate the delivery of early lifecycle phases by 30% to 40%, and provide a 5% to 15% overall savings in time and effort throughout the entire lifecycle, writes Jones in *Assessment and Control of Software Risk* (Prentice Hall, 1994). Workshops are powerful devices to

deliver artifacts of project chartering, planning and requirements/analysis phases (see Figures 2 and 3 for examples of some of the artifacts that can delivered in workshops.) After going through team-development stages such as "forming, storming, norming and performing," a group can become very productive, very fast. The following are real examples:

- In a business-rule workshop, participants delivered 35 business rules at the end of the third workshop day (averaging nearly 12 per day), then were able to add an additional 20 by the end of day four and an additional 35 business rules by the end of day five.
- In a use-case modeling workshop, business participants were able to test their use case model and structural business rules using 55 scenarios in less than 90 minutes.
- In two hours, workshop participants generated 119 business events, classified them and then removed those not in scope.
- In three hours, a team validated a medium-complexity data model and made decisions about the scope of the data about which business performance metrics would be based.
- In 3.5 hours, a complete set of relationship maps were built by workshop participants which enabled them to identify, in detail, current business process and problem areas.
- In 75 minutes, a chartering workshop group delivered and categorized nine risk areas and created 13 risk-mitigation strategies for the two high-probability/high-impact risks.

How is this achieved? Planning is all. A facilitated workshop must be well-designed and planned in order to be successful, and it requires a process. The Facilitated Essentials method employs the Total Quality Management cycle of "plan, do, check, act" (PDCA), ensuring that the process is continually working. For example, a workshop contract, sometimes in the form of an agenda, will delineate decision rules and products. In other cases, an orientation is conducted to ensure agreement on the workshop process and understanding of the products and pre-work.

A framework for the Facilitation Essentials method (see Figure 1), helps manage the quality of the process. Like John Zachman's framework for information systems architecture (*IBM Systems Journal*, vol. 26, no. 3, 1987), these columns represent all the interrogatives necessary to plan a successful workshop. Attention to all these dimensions is paramount. Customers are involved from the start of the workshop-planning process, beginning with identifying the workshop goals. This promotes a shared sense of purpose for the workshop and the project.

Figure 1. The Facilitation Essentials Method for Managing Workshops

Purpose	**Participants**	**Principles**
Why do we do things?	*Who* is involved?	*How* do we function?

• Goals • Needs • Motivation • Justification	• People • Roles • Players • Contributors	• Guidelines • Ground rules • Process rules
Figure 1, Continued		
Products	**Place**	**Process**
What do we do?	*Where* is it located?	*When* do things happen?
• Deliverables • Models • Decisions • Plans	• Venue • Space • Time	• Activities • Concurrency • Sequence • Order

Organizational change management

Software modifies behavior: users navigate new work processes and procedures, performance expectations change—yet scant attention is paid to these post-implementation issues. Developers are in a unique position to improve the probability of success by paying attention to these aspects of change.

Organizational change management (OCM) is, in fact, what software development efforts are designed to do. The business goals and objectives which drive development are based on the expectation of change. So it's not surprising that when software projects fail or disappoint people, it often has nothing to do with the technology and everything to do with people factors—environment, rewards, feedback, procedures, measures, work aids, communications, etc. Here are some ways of mollifying customers undergoing a metamorphosis:

- Establish an OCM plan, or build it into the project plan. This includes a timeline for the change, metrics and who will act as sponsors (perhaps the project sponsor or other strategic individuals); change agents (for example, business analysts who participate in requirements workshops); and change targets (end users) who will participate in various life-cycle stages such as requirements, testing, prototyping, training and documentation.
- Identify the "end state" of the business environment during the chartering phase: How work will change, what procedures and methods in the business process will be altered, who will be affected, how the work flow might look, what metrics will be applied to the value of software, etc.
- Establish a communication plan for the project; include ongoing feedback mechanisms in this plan.
- Identify changes that can be expected during and after implementation, and identify the agents, targets, and sponsors (these should map to roles in the charter).

- Encourage your business partners to assess their readiness for change; this might mean stopping a project, revisiting project goals and objectives or taking other business actions to prepare for change.
- Establish a sponsorship contract for the change in which the specific behaviors which change sponsors must express (say), model (do) and reinforce (reward) are delineated; these contracts are established down the chain of influence in the organization to reach the ultimate targets of the change, e.g. the end users.

An orchestrated event

These three activities—creating a project charter, using facilitated workshops and addressing organization change—are highly interrelated. Software development is part of an orchestrated business change event. Delivering the wrong software correctly or delivering the right software incorrectly can make or break not only the project, but also the business. Satisfied customers are the product of continual business involvement in the life cycle.

Figure 2. Products of the Chartering/Planning Process

- Business goals and objectives
- Problem/opportunity fishbone diagram
- Relationship map
- Context diagram, scope diagram
- Event list
- High-level business process map
- Prioritized constraints (cost, time, features/functions, quality)
- Project critical success factors/assumptions
- Customer types and needs (QFD)
- Future changes/barriers
- Action plan, pert or chart
- Gantt chart
- Risks/risk mitigation strategy
- Sponsor
- Project roles, responsibilities and organization
- Business policies

Figure 3. Products of Requirements Analysis

- Business process thread
- Process dependency diagram
- Business and temporal events
- Role/event matrix
- Event partitioned dataflow
- Activity value analysis
- CRCs, class models
- Cross-functional process map
- Scenarios
- Swim-lane/process map
- Use cases
- Window hierarchy diagram
- Object states
- Interface prototype
- Entity-relationship model
- Non-functional requirements list
- Entity life history diagram
- Requirements validation action plan
- Statechart diagram
- Business rule list

- Functional decomposition diagram