1.1 Software project management

Software project managers are frequently called upon to manage and deliver many different types of projects. While the specifics of these projects may vary, one constant remains that is the manner in which a project is first communicated and addressed to its stakeholders will greatly influence the likelihood of its success.

Any software project has two main activity streams: engineering and project management. The engineering activity is concerned with building the system (or artefact) and focuses on those software issues as how to design, code, and test. The project management activity is largely concerned with risk, cost, and quality. To this end successful project management depends not on the type of technical project, but on consistently applying the right project and people methodologies. Methodologies provide the cornerstone by which we can plan, execute, measure, monitor, control, and reduce project risk. According to the Standish Group: formal methodology provides a realistic picture of the project and resources committed to it, and it results in steps and procedures the team can reproduce and reuse. It also enables the team to maximize consistency and it incorporates lessons learned into active projects. Methodologies encourage go or no-go decision checkpoints. They also help the project team proceed with a higher level of confidence, or halt or alter steps to fit changing requirements. Figure 1.1 provides one view of an integrated stakeholder and project management framework.

In the last decade project management theory has to some extent concentrated on the questions of structure, control, and monitoring. For a project manager, the structure of a given project is
often imposed and often stands at odds with the environment into which this structure is to be transplanted. It is the craft of the project manager to improvise within this imposed structure according to the demands of the environment and the stakeholders who are inherent within it.

What is distinctive for projects is their parameters. Parameters are not static entities, but more of an existential condition of knowing that the defining characteristic for one’s actions is their connection to a predetermined goal, be this goal of time, cost, or quality. But, even though a project by definition is usually a planned, scripted activity, it is the definitional characteristic of uniqueness that exerts the most influence over the actual network of activities that becomes the project. One can plan a project forever, but the project execution will still be at the mercy of the uniqueness that, in a way, is the project.

The management of software projects is a less straightforward activity than is usually assumed in the literature. This management is less a following of a project plan, and more the handling of continuous action, some ordered, some not. In the case of complex projects, such as the design and build of an entirely new software system, innovation must be at the forefront of a project manager’s thinking. A recurring theme in project management is that performance and success are closely related to innovative (or lateral) thinking and project management capability.

One manager and educator Edward de Bono has written extensively about the process of lateral thinking. De Bono identifies four critical factors associated with lateral thinking: (1) recognize dominant ideas that polarize perception of a problem,
(2) searching for different ways of looking at things, (3) relaxation of rigid control of thinking, and (4) use of chance to encourage other ideas. This last factor has to do with the fact that lateral thinking involves low-probability ideas, which are unlikely to occur in the normal course of events. I return to this subject in Section 3.6.1.


Project failure at times is associated with poor decision-making and lack of stakeholder involvement in searching for solutions to project management problems. One thing I learned from reading De Bono’s works is that an attitude of continuing to search for a solution, even when it seems quite impossible, often does lead to a solution, you must include your key stakeholders, and inclusion is a must.

Research undertaken by McManus and Wood-Harper (2003) for their book Information Systems Project Management noted that relationship and stakeholder management were the key factors mitigating failure and reducing project risk. Key points noted in relation to stakeholder management and project failure include:

- Failure to manage stakeholder expectations
- Failure to share knowledge with stakeholders
- Failure to share bad news with stakeholders
- Lack of project leadership (too much reinventing the wheel)
- Lack of shared information, that is:
  - What information is needed?
  - What level of urgency?
  - Who needs it?
  - When or how frequently is it to be produced?
  - In what form or way will it be sent?
  - Where is it to be kept?
- Lack of team motivation to undertake work
- Lack of communication and co-ordination, that is lack of management strategy to deliver the project.

In larger companies only 9% of the projects come in on time and on budget. Failed projects exhibit all or many of the above symptoms, but arguably the most important flaw in failed projects is the failure to manage stakeholder expectations.

(Source: The Standish Group, as reported by Solutions Integrator, June 30, 1999.)
1.1.1 The right project methodology

With the exclusion of PRINCE2 (which I will come back to in Chapter 2) most of the current thinking about how to deliver software engineering projects has come from Specialist Interest Groups (or SIGs) or individual software professionals. The ideas they developed have proven to be quite effective when used by small to medium organizations to create custom methodologies. Even though the best ideas could also be applied to projects undertaken by larger system integrators to date this has not tended to happen. Why? Structure and the imposition of non-valued management would seem to be major obstacles together with poor judgement and lack of control where it really matters at the coalface.

Advances in methodologies such as the Dynamic Systems Development Method (DSDM), Rational Unified Process (RUP), and lightweight approaches such as eXtreme Programming (XP), Crystal and Scrum place significant currency on user participation, and stakeholder involvement. Taken together, the current methodologies offer a wide range of choices to meet the particular needs of many software projects and the skill level of any organization. The most important reason for undertaking software projects is to improve the way a business process is performed. Software projects should follow a disciplined methodology. Common to many of these methods are the following attributes:

- Business vision
- Business models
- Configuration management
- Communication
- Contingency planning
- Data entry standards
- Development models
- Documentation
- Metrics
- Project plans
- Performance criteria
- Quality assurance
- Risk planning
- Software standards
- Software process
- Software testing
- Support tools
- Stakeholders
- Team structures.
Stakeholders and context of software project management


For completeness I would like to discuss each of these methods in brief. Starting with DSDM.

1.1.1.1 DSDM overview

The DSDM method is based on nine principles, these principles include:

1. Active user involvement
2. Teams must be empowered to make decisions
3. Focus is on frequent delivery of software products
4. Fitness of business purpose is the key criterion
5. Iterative and incremental development is necessary
6. All changes are reversible
7. Requirements are baseline at a high level
8. Testing is integrated throughout the life cycle
9. Collaborative and co-operative approach between stakeholders is essential.

1.1.1.2 RUP overview

The RUP method addresses the issues on how object-oriented development works. The principles include:

1. The process is iterative and incremental development process
2. The method allows increasing understanding of the problem through successive refinement
3. The method gives an effective solution and multiple iterations
4. The method consists of four phases: inception, elaboration, construction, and transition:
   - *Inception*: establishes the business rational for the project
   - *Elaboration*: collects detailed requirements to establish an architecture baseline and create a plan for construction
   - *Construction*: consists of many iterations, in which each iteration analyses designs, builds, tests, and integrates a subset of requirements of a project
   - *Transition*: includes beta testing, packaging, performance tuning, and training
5. The method relies on iterative user involvement and collective risk assessment methods.
1.1.1.3 XP overview

XP is a lightweight discipline of software development based on principles of simplicity, communication, feedback, and courage. XP is designed for use with small teams who need to develop software quickly in an environment of rapidly changing requirements. The method subscribes to the values of simplicity, communication, feedback, and courage. XP currently recommends 12 practices to support the four values:

1. **Planning**: the XP planning process allows the XP “customer” to define the business value of desired features, and uses cost estimates provided by the programmers, to choose what needs to be done and what needs to be deferred. The effect of XP’s planning process is that it is easy to steer the project to success.

2. **Small releases**: XP teams put a simple system into production early, and update it frequently on a very short cycle. Cycles lengths of a couple of months are recommended. This allows the business value of the product to be evaluated in the real world.

3. **Metaphor**: XP teams use a common system of names and a common system description that guide development and communication.

4. **Simple design**: a program should be the simplest one that meets the current requirements. There is not much building “for the future”. The focus is on providing business value. Of course, it is necessary to ensure that you have a good design. In XP this is achieved through relentless refactoring.

5. **Testing**: XP teams focus on validation of the software at all times. Programmers develop software by writing tests first, then software that fulfils the requirements reflected in the tests. Customers provide acceptance tests in advance of development that enable them to be certain that the features they need are provided.

6. **Refactoring**: XP teams improve the design of the system throughout the entire development cycle. Keeping the software clean does this: without duplication, with high communication, simple, yet complete. All code should be refactored as often as possible. Refactoring is a process of improving a program’s structure without changing its functionality.

7. **Pair programming**: XP programmers write all production code in pairs, two programmers working together at one machine. Pair members rotate regularly. Less experienced members are constantly mentored, and the risk of less experienced code being added to the application is minimized.
Pair programming provides constant code reviews. No more dreary code review meetings – put two sets of eyes on the code as it is written. Experiments have shown that pair programming produces better software at similar or lower cost than programmers working alone.

8. **Collective ownership:** all the code belongs to all the programmers. This means anyone is authorized to improve any part of it at any time. This lets the team go at full speed, because when something needs changing, it can be changed without delay. Collective ownership of the code by all members of the team helps ensure even more eyes will see the code, increasing the amount of code review performed.

9. **Continuous integration:** XP teams integrate and build the software system multiple times per day. This keeps all the programmers on the same page, and enables very rapid progress. Integrating frequently tends to eliminate integration problems that plague teams who integrate less often.

10. **Forty-hour week:** tired programmers make more mistakes. XP teams do not work excessive overtime, keeping them fresh, healthy, and effective.

11. **On-site customer:** an XP project is steered by a dedicated individual who is empowered to determine requirements, set priorities, and answer questions as the programmers have them. The effect of being there is that communication improves, with less hard-copy documentation – often one of the most expensive parts of a software project. Ideally, the customer sits at a desk next to the programmers for the length of the project.

12. **Coding standard:** for a team to work effectively in pairs, and to share ownership of all the code, all the programmers need to write the code in the same way, with rules that make sure the code communicates clearly.

### 1.1.1.4 Crystal overview

Crystal collects together self-adapting family of “shrink-to-fit”, human-powered software development methodologies based on these understandings:

1. Every project needs a slightly different set of policies and conventions, or methodology.
2. The workings of the project are sensitive to people issues, and improve as the people issues improve, individuals get better, and their teamwork gets better.
3. Better communications and frequent deliveries communication reduce the need for intermediate work products.
Managing Stakeholders in Software Development Projects

Crystal is a family of human-powered and adaptive, ultra light, “shrink-to-fit” software development methodologies. “Human-powered” means that the focus is on achieving project success through enhancing the work of the people involved (other methodologies might be process-centric, or architecture-centric, or tool-centric, but Crystal is people-centric). “Ultra light” means that for whatever the project size and priorities, a Crystal-family methodology for the project will work to reduce the paperwork, overhead and bureaucracy to the least that is practical for the parameters of that project. “Shrink-to-fit” means that you start with something small enough, and work to make it smaller and better fitting. Crystal is non-jealous, meaning that a Crystal methodology permits substitution of similar elements from other methodologies. Key points are:

- two base techniques: methodology tuning technique, and reflection workshop technique retrospective
- incremental development of 4 months or less
- clear ownership model of work products
- have regression testing.

Crystal is evolving in tandem with the understanding of the principles of lightweight software development processes and people-centric project management. It aligns itself with the manifesto for software development (refer to Agile Software Development by Alistair Cockburn, 2001).

1.1.1.5 Scrum overview

Scrum as applied to software development was first referred to in “The New Product Development Game” (Harvard Business Review, 86116, 137-146, 1986) and later elaborated in “The Knowledge Creating Company” both by Ikujiro Nonaka and Hirotaka Takeuchi (Oxford University Press, 1995).

Scrum is an iterative, incremental process for developing any product or managing any work. It produces a set of functionality at the end of every iteration. Its key attributes are as follows. Scrum is:

- an agile process to manage and control development work
- a wrapper for existing engineering practices
- a team-based approach to iteratively, incrementally develop systems and products when requirements are rapidly changing
Stakeholders and context of software project management

- a process that controls the chaos of conflicting interests and needs
- a way to improve communications and maximize co-operation
- a way to detect and cause the removal of anything that gets in the way of developing and delivering products
- a way to maximize productivity
- scalable from single projects to entire organizations and has controlled and organized development and implementation for multiple interrelated products and projects with over a thousand developers and implementers
- a way for everyone to feel good about their job, their contributions, and that they have done the very best they possibly could.

Scrum’s goal is to deliver as much quality software as possible within a series (3–8), of short time-boxes (fixed time intervals) called Sprints that typically last about a month. Each stage in the development cycle (requirements, analysis, design, evolution, and delivery) is now mapped to a Sprint or series of Sprints. The traditional software development stages are retained for convenience primarily for tracking milestones. So, for example, the requirements stage may use one Sprint, including the delivery of a prototype. The analysis and design stages may take one Sprint each. While the evolution stage may take anywhere from three to five Sprints.

As opposed to a repeatable and defined process approach, in Scrum there is no predefined process within a Sprint. Instead, Scrum Meetings drive the completion of the allocated activities. Each Sprint operates on a number of work items called a Backlog. As a rule, no more items are externally added into the Backlog within a Sprint. Internal items resulting from the original pre-allocated Backlog can be added to it. The goal of a Sprint is to complete as much quality software as possible.

The Scrum process is fully described in the book *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle (Prentice Hall, 2001), from which this paragraph was summarized.

### 1.1.2 The right project team

In software engineering projects considerable effort is expended in the analysis and design phases. These phases require creative and talented people, I have always adhered to the maxim that
talented people can make just about anything happen (provided they have stakeholder support) and although this is not always popular with managers who allocate resources I believe it holds up when delivering complex projects.

Identifying people with the right technical and soft skills for the project is a key challenge for many project managers. Selecting people to fill key roles in projects is not always easy especially when multiple projects are on the go or when the organization is contracting. Key positions to fill will most certainly include:

- Deputy project manager
- Quality assurance manager
- Technical design authority
- Technical designers
- Software developers
- Business analysts and modelers
- Configuration manager
- Business user
- Domain expert
- Sponsors representatives.

While no single selection process is guaranteed to deliver the people you are looking for, the project manager should try to minimize the risk by having a multiple selection process. Most software projects require people to work together in small groups for intensive periods of time. To do this effectively it is useful to understand something about how individuals may behave towards each other and within the project in different situations. Belbin as devised a questionnaire to ascertain the mix of personality types (and soft skills) that make up a balanced team. Belbin identifies nine roles (Figure 1.2). The Belbin role types include: plant, resource investigator, co-ordinator, shaper, monitor/evaluator, team worker, implementer, completer, and specialist. Generally project teams work best when they have a good balance of the nine roles.

1.1.3 The right project manager

From an organization, client and stakeholder perspective, selecting the right project manager is critical. When senior management selects a project manager several factors should be considered. For example, project managers working within the software industry must possess a high level of technical competence. Required skills and knowledge include: an understanding of software risk, and of the technology of software engineering,
<table>
<thead>
<tr>
<th>Role</th>
<th>Description of role traits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant</td>
<td><strong>Positives</strong> Creative, imaginative, and unorthodox. Can solve difficult problems. <strong>Negatives</strong> Ignores details and can be too pre-occupied to communicate effectively.</td>
</tr>
<tr>
<td>Resource investigator</td>
<td><strong>Positives</strong> An extrovert, enthusiastic, and communicative. Develops contacts. <strong>Negatives</strong> Can be overly optimistic, and may lose interest once initial enthusiasm has passed.</td>
</tr>
<tr>
<td>Co-ordinator</td>
<td><strong>Positives</strong> Mature, confident, and a good chairperson. Clarifies goals, promotes decision-making and delegates well. <strong>Negatives</strong> Can be seen as manipulative; delegates personal work.</td>
</tr>
<tr>
<td>Shaper</td>
<td><strong>Positives</strong> Challenging, dynamic, and thrives on pressure. Has the drive and courage to overcome obstacles. <strong>Negatives</strong> Can provoke others and hurt their feelings.</td>
</tr>
<tr>
<td>Monitor/evaluator</td>
<td><strong>Positives</strong> Sober, strategic, and discerning, sees all options and judges accurately. <strong>Negatives</strong> May lack drive and ability to inspire others and be overly critical.</td>
</tr>
<tr>
<td>Team worker</td>
<td><strong>Positives</strong> Co-operative, mild, perceptive, and diplomatic. Listens, builds, averts friction, calms the waters. <strong>Negatives</strong> May be indecisive in crunch times, and can be easily influenced.</td>
</tr>
<tr>
<td>Implementer</td>
<td><strong>Positives</strong> Disciplined, reliable, conservative, and efficient. Turns ideas into practical actions. <strong>Negatives</strong> Somewhat inflexible; slow to respond to new possibilities.</td>
</tr>
<tr>
<td>Completer</td>
<td><strong>Positives</strong> Painstaking, conscientious, and anxious; searches out errors and omissions. Delivers on time. <strong>Negatives</strong> Inclined to worry too much. Reluctant to delegate. Can be a nitpicker.</td>
</tr>
<tr>
<td>Specialist</td>
<td><strong>Positives</strong> Single-minded, self-starting, and dedicated. Provides knowledge and skills in rare supply. <strong>Negatives</strong> Contributes on only a narrow front. Dwells on technicalities. Overlooks the &quot;big picture&quot;.</td>
</tr>
</tbody>
</table>

**Figure 1.2** Belbin roles.

*(Figure 1.2 is adapted from Belbin, R.M. (1993) Team Roles at Work, Butterworth Heinemann, 1993.)*
Managing Stakeholders in Software Development Projects

awareness of programming tools and techniques used in software development, knowledge of current analysis methods, and knowledge of project management techniques. Project managers need not necessarily be technical experts in all aspects of a project. Of course, the project manager should also have above average diplomacy and managerial abilities.

While all this seems obvious, according to research many project managers are perceived to be average or below average in the areas of leadership skills, interpersonal relationships, and administration.

Because there is a high correlation between the quality level of the attributes needed of project managers and the use of soft skills, careful analysis of the project managers’ managerial experience and capability of working with different stakeholder groups should be confirmed prior to selection or appointment.

One characteristic a project manager must possess is the ability to see the “big picture”. Successful project managers recognize that many factors affect the outcome of a project. They must consider not only the technical aspects but also the economic, social, and legal aspects. Their perspective must be broad, seeing the whole picture, and requires taking a holistic approach. It is argued by McManus and Wood-Harper that many project managers lack this characteristic. They often fall into the trap of emphasizing the technical side while neglecting other important areas. As a result, relations with stakeholders often deteriorate. This may lead to legal complications rising due to lack of compliance and trust.


1.2 The nature of stakeholder theory

The best way to begin to develop a stakeholder approach line of thinking is to start at the beginning with some theory. Edward Freeman, a major protagonist in the stakeholder literature has developed a stakeholder theory of a modern corporation which includes owners, suppliers, employees, customers, local communities, and managers that are considered to be the primary stakeholders to which an organization as a moral responsibility (Freeman, 1984; 1997). A wider definition of stakeholders would include any group or individual who can affect or is affected by the achievement of the organization’s objectives (Freeman, 1984, p. 46). In organizations that use traditional project management
methodologies such as PRINCE2, stakeholders are normally defined as internal (direct) and external (indirect). Those directly affected by a proposed project are clearly among the key stakeholders. They are the ones who stand to benefit or lose from the organization's operations or who warrant redress from any negative effects of such operations. It is these directly affected stakeholders, who are the most significant and occasionally the most difficult to identify and involve in participatory efforts.

(The term PRINCE2 stands for Projects in a Controlled Environment and is the UK’s defacto project management methodology.)

1.2.1 Identifying stakeholders

In many software project stakeholders are numerous and sometimes difficult to identify. One method for identifying stakeholders is the use of a “contrast” or “maximum” variation sampling procedure. This can be used to define local groupings around project issues. Each potential stakeholder is interviewed and asked to identify another who will have the most different perceptions on the issue than his/her own. The process of interviewing and identifying new respondents with contrasting views is repeated until several main issues or themes emerge. These themes each represent a stakeholder group. This approach enables the identification of stakeholder groups with conflicting or different values without asking direct questions that may be socially unacceptable to answer. See also Section 2.1.1.

Another approach to stakeholder identification is to produce a stakeholder map. Figure 1.3 identifies key stakeholder groups and organizations that will be included in an National Health

![Figure 1.3](image)
Simple example of key stakeholder map. (GPs, general practitioners; PCTs, patient-controlled trials.)
Managing Stakeholders in Software Development Projects

Service (NHS) project. All of these parties have a stake in the successful implementation of a project, so it is important to identify which stakeholders will have an impact on the work of the project team. For example, at the final implementation phase of a project, the potential for misuse and malfunction of a proposed software system must be analysed in terms of its impact on all of the projects present and future stakeholders.

As pointed out in Section 1.1.2, a typical software project team consists of a project manager, analysts, designers, developers, testers, and quality assurance personnel. The team may also include users or their representatives. Unfortunately, the literature on software project management often fails to clearly classify types of stakeholders and to describe strategies for their management. For example, Mitchell et al. (1997), give a list of 27 definitions of the term stakeholder used from 1993 to 1995 showing the intention of researchers to answer the fundamental question of which entities can be defined as stakeholders that deserve the managers’ attention. A stakeholders’ significance will depend upon the situation and the issues encountered during the project life cycle. Of all the possible stakeholders (see example in Figure 1.4), the ones who will be relevant to the organization’s management team will depend on the particular problem. The bigger, the more complex the problem, the more it is likely to involve a wider array of stakeholders. As a result, the more assumptions needed. It is characteristic and fundamental feature of complex problems that not everything of basic importance can be known prior to working on the problem. Rather, such a statement often only emerges with difficulty over time and only as a direct result of our working on it. In such circumstances both the stakeholders’ motivation and opportunity to act are particularly sensitive to specific issues (Freeman, 1984). The motivation of stakeholders for their contribution can vary considerably; for example, the customer would like to introduce change with maximum benefit, the project manager wants to successfully complete the project with the appropriate resources, and the an analyst would like to specify the requirements on time and within budget. Stakeholders have various possibilities to influence the outcome of the project life cycle, for example during the requirements analysis (and evaluation) stage stakeholders can and do manipulate the requirements specification to articulate their own interests. Davis (1982, p. 10), names some cognitive aspects to this bias. Stakeholders tend to rate current information higher than less recent information, they base their judgement on information available, are particularly influenced
by recent events and are not good at as intuitive statisticians so that they draw unwarranted conclusions from small samples or a small number of occurrences. Another major influence results from personal interests within the issues touched by the object or situation to be evaluated. Blumberg and Gerwin (1981) note: Managers (and other stakeholders) do not have the expertise to second-guess the judgements of technical experts when evaluating whether or not to purchase equipment (IT solutions). Consequently, they tend to employ mainly financial criteria in judging requirements.

### 1.2.2 Supportive stakeholder relationships

There are many reasons to believe that adoption of a stakeholder approach to software project management and management in general will contribute to the long-term survival and success of a project organization. Positive and mutually supportive stakeholder relationships encourage trust, and stimulate

<table>
<thead>
<tr>
<th>Software project life cycle stage</th>
<th>Primary, secondary, external, and extended stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. System requirements analysis</td>
<td>- Project board member's</td>
</tr>
<tr>
<td>2. System architectural design</td>
<td>- Non-executive directors</td>
</tr>
<tr>
<td>3. Software requirements analysis</td>
<td>- Manager/director</td>
</tr>
<tr>
<td>4. Software architectural design</td>
<td>- Department manager</td>
</tr>
<tr>
<td>5. Software detailed design</td>
<td>- Account managers</td>
</tr>
<tr>
<td>6. Software coding and testing</td>
<td>- Project sponsor</td>
</tr>
<tr>
<td>7. Software integration</td>
<td>- Champion user</td>
</tr>
<tr>
<td>8. Software qualification testing</td>
<td>- Human resources manager</td>
</tr>
<tr>
<td>9. System integration</td>
<td>- Department users</td>
</tr>
<tr>
<td>10. System testing</td>
<td>- Programme manager</td>
</tr>
<tr>
<td>11. Software installation</td>
<td>- Project manager</td>
</tr>
<tr>
<td>12. Acceptance support</td>
<td>- Business architect</td>
</tr>
<tr>
<td></td>
<td>- Business analysts</td>
</tr>
<tr>
<td></td>
<td>- Technical architect</td>
</tr>
<tr>
<td></td>
<td>- Database architect</td>
</tr>
<tr>
<td></td>
<td>- Database analyst</td>
</tr>
<tr>
<td></td>
<td>- Developers</td>
</tr>
<tr>
<td></td>
<td>- Integration manager</td>
</tr>
<tr>
<td></td>
<td>- Configuration manager</td>
</tr>
<tr>
<td></td>
<td>- Change control manager</td>
</tr>
<tr>
<td></td>
<td>- QA manager</td>
</tr>
<tr>
<td></td>
<td>- Test manager</td>
</tr>
<tr>
<td></td>
<td>- Testers</td>
</tr>
<tr>
<td></td>
<td>- Infrastructure support teams</td>
</tr>
<tr>
<td></td>
<td>- Software support team</td>
</tr>
</tbody>
</table>

**Figure 1.4** Example of stakeholders involved in NHS-related software projects. (MPs, Members of Parliament; MEP, Members of European Parliament; GMC, General Medical Council; QA, quality assurance.)
Managing Stakeholders in Software Development Projects

<table>
<thead>
<tr>
<th>Principle</th>
<th>Condition</th>
<th>Narration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Monitor</td>
<td>Managers have moral obligation to monitor the concerns of legitimate stakeholders and to take their interests into account</td>
</tr>
<tr>
<td>2</td>
<td>Communicate</td>
<td>Managers should keep an open mind when making decisions and activity involve stakeholders in the decision-making process</td>
</tr>
<tr>
<td>3</td>
<td>Behaviour</td>
<td>Managers should support processes and modes of behaviour that are sympathetic, sensitive, and respectful to stakeholders and their needs</td>
</tr>
<tr>
<td>4</td>
<td>Risk</td>
<td>Managers should recognize the individual stakeholder sacrifices and take positive steps to ensure that an just risk reward strategy is implemented</td>
</tr>
<tr>
<td>5</td>
<td>Co-operation</td>
<td>Managers should work to harmonize events that may destabilize stakeholder involvement</td>
</tr>
<tr>
<td>6</td>
<td>Rights</td>
<td>Managers should avoid activities that jeopardize human rights or give rise to risks that are unacceptable to stakeholders</td>
</tr>
<tr>
<td>7</td>
<td>Conflict</td>
<td>Managers have moral and legal responsibilities to stakeholders and should address conflicts through open communication</td>
</tr>
</tbody>
</table>

**Figure 1.5** Clarkson’s principles of stakeholder management.

*(Figure 1.5 is adapted from Clarkson, M. (1995) A stakeholder framework for analyzing and evaluating corporate social performance. Academy of Management Review, 20, 92–117.)*

collaborative efforts that lead to relational wealth, that is, organizational assets arising from familiarity and teamwork. By contrast, conflict and suspicion stimulate formal bargaining and limit efforts and rewards to teams, which result in time delays and increased costs. In addition, more and more managers are recognizing that a reputation for “ethical and socially responsible behaviour” can be the basis for a competitive edge in both market and public policy relationships. Finally, in spite of the specification and measurement difficulties involved, research studies have found evidence of positive associations (few have found negative associations) between various socially and ethically responsible practices and conventional economic and financial indicators of performance (profitability, growth, etc.). Thus, there is no reason to think that the conscientious and continuing practice of stakeholder management will conflict with conventional financial performance goals see Figure 1.5.
In the world of software projects, project managers tend to hold high office and may be regarded as Captains of the project organization. Jones (see Hill and Jones, 1992) has advanced one form of instrumental stakeholder theory. Jones makes a theoretical case for the general proposition that if a project organization’s contract (through their managers) with their stakeholders on the basis of mutual trust and co-operation, they will have advantage over organizations that do not. Put another way they will deliver and win future business. No assumption is made that managers will try to develop trusting and co-operative relationships with stakeholders, but an argument is made that if they do, competitive advantage will result.

As intimated in Section 1.1.2 many organizations have limited resources and as such stakeholders compete for these resources. Often as not stakeholder values and needs differ widely and there is usually a highly skewed distribution of resources among stakeholder groups. Typically on projects stakeholders have different priorities and different objectives. Unequal influence and distribution of resources exacerbate conflict of interests. For example, senior managers are normally part of the elite that have responsibility for and make the key decisions for the business direction of the project organization. Their role is to generally ensure that technology and information technology is integrated to the business processes of the organization, whereas employees the non-elite may be users of systems without any real influence. They are however, important stakeholders in the exploitation process because, in order to plan and implement software systems that can be efficient and integrated, their specific needs and requirements must be taken into account. However, they tend to fall into the underprivileged stakeholder category.

### 1.2.3 Gaining stakeholder commitment and trust

The importance of stakeholders during the software life cycle process is based on the premise that their activities in the development phases largely determine the quality of the finished product itself and as such stakeholders yield power and influence over the project manager (Schulmeyer and McManus, 1999). Research by McManus and Wood-Harper (2003) emphasises the need to gain stakeholder trust and commitment.

Project managers sometimes use their reputations to create trust with stakeholders. Reputation involves an estimation of ones
character, skills and reliability, and other attributes important to the exchanges and is important under exchange conditions of uncertainty. As uncertainty within a project increases, exchanges between stakeholders become more focused with information about their own and others repudiation (Kollock, 1994). Reputation can reduce behavioural uncertainty by providing information about the reliability and good will of others. Reputations do have limitations in their use and so project managers must be able to legitimize their actions in the eyes of those stakeholders who are affected or who can affect the project's outcomes. In essence they need to establish credibility and engender trust.

It is relatively easy to tell when a project manager is behaving in ways that will reduce credibility and cause damage to the team or project. Trust-busting behaviours (or language) often kill hope of consensus or negotiation. This is not generally the fault of the stakeholders; but instead is strictly the failing of those who use this approach in the mistaken belief that it is effective or will somehow avoid having to deal with issues and questions.

McManus and Wood-Harper offer some strategic advice to the project manager looking to engender trust – their advice includes:

- establishing good personal relationships, expertise alone does not inspire trust and credibility
- illustrating that actions are being driven by the needs of the stakeholders, and that their needs and requirements are being considered seriously
- using the recommendations of stakeholders or established formal methodologies to support the project
- involving stakeholders as project champions to lend the project authority.

Mintzberg (1994) argues for caution when developing stakeholder strategies. He comments: “Intended strategies have no value in and of themselves; they take on value only as committed people infuse them with energy ... that is why every problem of implementation is also one of formulation – not only for the actual strategies conceived but also for the process by which conceptualisation occurs”.

In essence Mintzberg view is that managers should encourage active consultation and participation of as many stakeholders as possible, by engaging them in the ongoing dialogue, and involving them in the strategic process, to generate a feeling of consensus and ownership of the process and the outcomes throughout the organization. Stakeholders who have been consulted and have participated in the strategic process will better
Stakeholders and context of software project management

understand the trade-offs between project benefits and disadvantages and have greater trust. In this respect, stakeholder consultation can minimize the risk of unexpected negative reactions and protect and ensure a continued “licence to operate” and indirectly may lead to financial savings by avoiding social unrest, political or legal disputes, or negative publicity that can delay projects and be very costly. These aspects of software projects should not be forgotten. The project manager must be aware of the sensitive nature and issues of power and personal interests within the project or organization. The more high level and strategic a project is, the more such aspects have a vital importance on the success of the project. There is no clear recipe of what to do except to engage and consult stakeholders.

Boundaries of stakeholder consultation can be limited. For example, in Government or Health sector projects restrictions on resources or opportunities may act to curtail the involvement of the general public, leading to a reliance on key individuals or groups to represent stakeholder interests. However, organizations hoping to undertake a project involvement must realize that stakeholder consultation is just one component. Stakeholder processes are not public processes. If a company bases its actions on stakeholder consultation alone there is a possibility that the tax paying public will want to know why they were not consulted before a decision was reached and may even turn to the courts or officials to block decisions.

1.3 The principles of stakeholder management

I would like to discuss in more detail the seven principles of stakeholder management referred to in Figure 1.5 starting with principle number one.

Principle No. 1: Project managers have obligation to monitor the concerns of legitimate stakeholders and to take their interests into account.

In answering the question of what is a legitimate stakeholder? The first requirement of stakeholder management is an awareness of the existence of multiple and diverse stakeholders, and an understanding of their involvement and interest in the project. Many stakeholders are readily identified because of their express or implied contractual relationship to the organization. Others may identify themselves because of the impact, positive or negative, of the firm’s activities on their own well-being. And, of course, some third parties may claim a stake in the firm...
when no such relationship, in fact, exists. Managers are not obligated to respond favourably to every request or criticism; they are, however, obligated to examine all such claims carefully before passing judgment on their validity.

The salience of specific stakeholder concerns varies among different areas of managerial decision-making, and according to the time horizon involved. Current working conditions are of greatest concern to employees; the cost and quality of products are of greatest concern to customers. Long-term survival and growth may be of greatest concern to investors and to the communities within which the firm operates. In taking particular decisions and actions, project managers should give primary consideration to the interests of those stakeholders who are most intimately and critically involved.

Principle No. 2: Project managers should keep an open mind when making decisions and activity involve stakeholders in the decision-making process.

Communication, both internal and external, is a critical function of project management, and effective communication involves receiving, as well as sending, messages. Hence, to understand stakeholder interests and to integrate various stakeholder groups into an effective wealth-producing team, managers must engage in dialogue. A commitment to engage in dialogue, however, does not constitute a commitment to collective decision-making; there are obvious limits as to the amount and content of information (particularly information about strategic options under consideration) that can be appropriately shared with particular stakeholder groups. Nevertheless, the more open managers can be about critical decisions and their consequences, and the more clearly managers understand and appreciate the perspectives and concerns of affected parties, the more likely it is that problematic situations can be satisfactorily resolved. Open communication and dialogue are, in themselves, stakeholder benefits, quite apart from their content or the conclusions reached.

Principle No. 3: Project managers should support processes and modes of behaviour that are sympathetic, sensitive, and respectful to stakeholders and their needs.

Stakeholder groups differ not only in their primary interests and concerns, but also in their size, complexity, and level of involvement with the corporation. Some groups are dealt with through formal, and even legally prescribed, mechanisms, such as
collective bargaining agreements and shareholder meetings. Others are reached through advertising, public relations, or press releases; still others (e.g., government officials) are reached largely through official proceedings and personal contacts. Both the mode of contact and the type of information presented, or the opportunity for dialogue, can appropriately vary among different stakeholder groups, although the descriptions of situations and explanations of actions offered by managers should be consistent among all stakeholders. Extreme caution is required when project managers deal with stakeholder groups that have limited capacity to assimilate and evaluate complex situations and options.

Principle No. 4: Project managers should recognize the individual stakeholder sacrifices and take positive steps to ensure that any just risk reward strategy is implemented.

A firm’s a purposive organization in which all voluntary stakeholders collaborate for mutual benefit. Involuntary or consequential stakeholders (e.g., communities or third parties) may also be affected by the operation of the enterprise. And both voluntary and involuntary stakeholders are vulnerable, and differently vulnerable, to the effects of uncertainty and change over time. Successful project managers will see that all stakeholders receive sufficient benefits to assure their continued collaboration in the enterprise, and that their burdens and risks are no greater than they are willing to bear. Again, the openness and demonstrable fairness of the distribution of benefits and burdens among stakeholders are, in themselves, stakeholder benefits. Project managers may need to make special efforts to demonstrate stakeholder interdependence and the collaborative nature of the enterprise to non-contractual and involuntary stakeholders.

Principle No. 5: Project managers should work to harmonize events that may destabilize stakeholder involvement.

Wealth creation necessarily gives rise to consequences that may not be fully mediated through the marketplace. Some of these may be beneficial and welcome; others may be harmful. Monitoring and ameliorating undesirable consequences (i.e., negative externalities) often requires co-operation with other firms, private sector organizations, public agencies, and units of government. Project managers should be proactive in developing contacts with relevant groups and in forging coalitions aimed at reducing harmful impacts and compensating affected parties.
Managing Stakeholders in Software Development Projects

The often-true observation that one firm cannot solve this problem alone should be a stimulus to multi-party co-operation, not an excuse for neglect and inaction.

*Principle No. 6: Project managers should avoid activities that jeopardize human rights or give rise to risks that are unacceptable to stakeholders.*

The ultimate consequences of most human endeavours (particularly endeavours involving large expenditures, diverse interests, and long-time periods) can never be fully anticipated in advance. Hence, managerial decisions and corporate operations necessarily give rise to multiple and diverse risks. Project managers should communicate openly with stakeholders concerning the risks involved with their specific roles in the corporate enterprise, and should negotiate appropriate risk-sharing (and benefit-sharing) contracts wherever possible. When stakeholders knowingly agree to accept a particular combination of risks and rewards, then the arrangement is usually considered satisfactory. However, some projects may have consequences for which no conceivable compensation would be adequate, or risks that cannot be fully understood or appreciated by critical stakeholders. In these circumstances, managers have a responsibility to restructure projects to eliminate the possibility of unacceptable consequences, or to abandon them entirely if necessary.

*Principle No. 7: Managers have moral and legal responsibilities to stakeholders and should address conflicts through open communication.*

Up to this point, we have spoken of project managers as if they were disinterested co-ordinators of stakeholder interactions. However, project managers also form a distinct stakeholder group, with privileged access to information and unique influence on corporate decisions. As stakeholders, project managers are naturally interested in the security of their jobs, the level of their rewards, and the scope of their discretion in the use of corporate resources. Other stakeholder groups (shareowners and boards of directors, in particular) have devised a variety of arrangements intended to align the interests of managers with those of the corporation as a whole, and to prevent opportunistic abuse of managerial positions. However, the tension between the interests of project managers as stakeholders, on one hand, and those of other stakeholder groups and of the corporation itself as an ongoing entity, on the other, is unavoidable. Responsible project managers will recognize this, and will therefore accept and encourage organizational practices intended to
control this source of intra-organizational conflict. Project managers gain credibility when they establish procedures to monitor their own performance and, when appropriate, to facilitate third party review. Credibility matters when project managers ask other stakeholders to align their interests with those of the corporation, and to act responsibly rather than opportunistically. Without mutual credibility, stakeholder trust diminishes and the collaborative character of the organization may be jeopardized.

(Principles 1–7 are based on the work of Max Clarkson (Published by the Clarkson Centre for Business Ethics, Toronto, ON, 1999. ISBN 0-7727-8609-7 (paper). [59 pp.]) And are reproduced with permission.)

1.4 Key questions in stakeholder management

A number of the stakeholder issues have already been covered in Sections 1.1, 1.2, and 1.3. There are a number of key stakeholder questions that I would like to explore within this chapter. These questions relate to the what? Namely:

- What are our stakeholder’s stakes?
- What challenges do the stakes and stakeholders present?
- What responsibilities does the organization have to stakeholders?
- What strategies should management adopt to manage its stakeholders?

1.4.1 What is a stake?

To appreciate the concept of stakeholders, it also helps to understand the idea of a stake. A stake can be described as an interest or share in a project undertaking to achieve business, technical or social goals. If we accept this broad definition, it must be asked, what are the stakeholder’s interest, concerns, and perceptions of rights, expectations, or even ownership? As examples, consider two sets of stakeholders, the Organization and Government. The organizations interests are in maximizing profits, protecting intellectual property rights, balancing resource and demand, keeping the customer happy and other interests. The government’s interest in an organization can be as regulators, tax collectors, customer of defence products, trade
Managing Stakeholders in Software Development Projects

Figure 1.6
Defined components of a stake.

balance manager, and many others related to the public good. In each of these cases, it is important to determine the power, criticality and rationality of each group with regard to each interest, Figure 1.6.

1.4.2 What challenges do the stakes and stakeholders present?

Challenges normally take the form of demands of or responses by the organizations. Challenges also arise when crises occur that appear to be the responsibility of business. For example, organizations engaged in the production of software systems sometimes face litigation for poor quality or late delivery. Another example is anti-trust cases, such as those recently publicized involving Microsoft. Where an organization accused of an anti-trust violation can face new challenges from government, customers, competitors, the media, the financial community, as well as others. A final example is software piracy where multiple stakeholders are affected.

The high rate of software piracy leads to more than just a loss of profit for companies. It also leads to a loss of jobs, wages, and tax revenue. In addition, companies have to charge higher prices for their software to account for their losses due to piracy. While it is easy to justify saving a few Euros by getting a piece of software for free, the impact on society as well as those around us must be considered. Each interest, of course, is of different degrees of importance.
1.4.3 What responsibilities does the organization have to stakeholders?

Mutual and joint responsibilities for stakeholders separate into two general categories or types of situation:

1. Between the organization and its stakeholders
2. Among stakeholders themselves

Organizations that make “demands” on stakeholders bear some responsibility of assurance that their “demands” do not generate unintended negative consequences for the stakeholder. Organizations and their project managers have both a moral and ethical duty to their stakeholders. If organizations and their project managers do not practice mutual respect they are presumably forced to turn to non-moral ways of dealing with moral conflict. In projects they are driven to count on procedural agreements, political deals, and threats – all of which obviously stand in the way of moral care. The underlying assumption is that organizations should value reaching conclusions through reason rather than force, and more specifically through moral reasoning rather than through self-interested bargaining. Nevertheless people holding such divergent moral values are still equal participants in a democratically structured decision process. Professor David Wong (Duke University, Durham, USA) has articulated principles for achieving this. Act on one’s moral position in a way that minimizes potential damage to one’s broader relationship to others who have opposed positions. Other things being equal, select issues that minimize opportunity for serious disagreement. In this case, we should decide policies on a fairly specific, rather than general level. A project-by-project framework is dictated by the nature of the diversity and complexity of the consequences of different proposed interventions. Organizations must adopt a willingness to bridge any differences. This permits domains of agreement to be ascertained by removing the “us vs. them” attitude. This is emphasized in organizations that promote internal value systems among stakeholders themselves.

1.4.4 Values among stakeholders

Based on practical experience I suggest that a stakeholder consensus would identify the following values as a minimum requirement for software engineering projects to be considered ethical:

- The project manager should promote quality. Quality includes technical variables such as the skills and knowledge of the
professional but also that ethical issues are handled well and that compassionate and respectful attitudes are promoted and rewarded.

- The project manager should promote equality. All potential stakeholders should have an equal opportunity to access information.
- The project manager should promote the independence of the project professional. This is essential if the project manager is to perform the role of advocate. It is a prerequisite for the maintenance of trust.
- The project manager should not contain perverse incentives. Such incentives include those, which might encourage project managers to over service or to under service, and those, which might encourage stakeholders to make unreasonable demands on, project managers.
- The project manager should promote professional accountability. There should be no rights without responsibility, and conversely, there should be no responsibility without the right of professional autonomy.
- The project manager should be able to reconcile conflicts between values. The demand of stakeholder confidentiality and privacy, access to information to make informed choice, autonomous decision-making, and the values of stakeholders and those of the project manager are not always mutually compatible. Organizations should allow for the discretionary exercise of judgement.
- The project manager should not make unreasonable demands on any of the stakeholders. The requirement for equity of access to the service has been noted. However, software service providers should not be expected to provide services for which resources have not been provided.
- The project manager should promote co-operation rather than competition between service providers.
- The project manager should promote continuity of service and the exchange of values and beliefs over time.

1.4.5 What strategies should management adopt to manage its stakeholders?

After management has determined the responsibilities and stakeholders involved, a highly important question is, “What strategies should management adopt?”

In large projects where numerous stakeholders are involved it is perhaps true to acknowledge that it is highly unlikely that all
Stakeholders and context of software project management

stakeholders’ expectations will be met. Therefore, the project manager must somehow ascertain which stakeholders should be satisfied. Since stakeholders have the ability to positively or negatively influence the project, integrating and satisfying the right people is essential. Specific organizational and project strategies used to integrate stakeholders will differ, depending on the issue and the groups potential to co-operate or threaten the organizations performance. In developing strategy, the project manager needs to consider that each stakeholder has the ability to both threaten and co-operate, the objective of the game is to reduce the threatening element and increase the co-operative behaviour of the stakeholder.

It is important to realize that the stakeholders potential to act and their willingness to act are not directly related. Therefore, when looking at strategies, it is important to examine not only strategies addressing stakeholders who are positively disposed towards a project but those who are negatively disposed towards a project as well. Some strategies may only be appropriate for a stakeholder with a specific disposition towards the project, that is, positive or negative. In other cases a given strategy may be appropriate for either type of stakeholder, that is both. Strategies should not be mutually exclusive; some are appropriate for more than one type of stakeholder.

This topic will be discussed in more detail in Chapter 5.

1.5 Chapter summary - 10 key points

The most important points to take away from this chapter are as follows. Remember:

1. Software projects need to follow a disciplined methodology
2. Selection of the right project manager with the right stuff is critical
3. Successful project managers recognize that many factors affect the outcome of a project
4. One characteristic a project manager must possess is the ability to see the big picture
5. Project managers should recognize dominant ideas that polarize perception of a problem
6. Creative and talented people matter
7. A stakeholder approach to project management will contribute to the long-term survival and success of a project
8. The bigger, the more complex the project, the more it is likely to involve a wider array of stakeholders
9. Project managers should support processes and modes of behaviour that are sympathetic, sensitive, and respectful to stakeholders and their needs
10. Credibility matters when project managers ask other stakeholders to align their interests with those of the project, and to act responsibly rather than opportunistically.

Chapter reading