

Chapter IX

Project Execution and Control

If you can't measure it, you can't manage it.

(Peter Drucker)

In fairy tales and traditional romance movies, the story ended when the prince found his soul mate, married her, and rode off with her into the sunset. The ending caption said: "They lived happily ever after." Well, we know that real life is not quite that simple; after the marriage comes the most difficult (and, one hopes, interesting) part. Similarly, a great project contract and plan is of little consequence without constant monitoring and control. Once the project is planned and underway, the project manager cannot simply ride away and assume that everything will go according to plan.

To insure success, many project matters need to be monitored; if a matter deviates from the plan, then some form of control must be exerted to bring the situation back in line with the plan. In this chapter I discuss the many matters that need to be monitored for IT projects, how best to monitor each matter, and what type of control actions may be appropriate for each.

The Control Process

The basic control process used in project management is the same process used in most engineering and business systems. It is based on the definition and establishment of key measures, and those measurements are then compared to some desired values or standards to formulate algebraic formulas, usually called metrics. If the difference

between the measurement and the desired value exceeds some threshold, then corrective action (feedback) of some type is invoked, and the degree of corrective action may be a function of the size of the difference (and/or the integration [accumulation] or differentiation [rate] thereof). The measurements may be of process outputs or of the process itself, and the measurement level may be process-related (generally, how things are being done) or product-related (generally, what things are being built). This is illustrated in Figure 9.1.

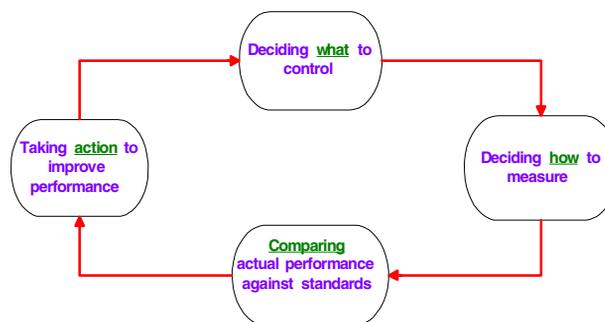
The project control processes go on during the execution of the project. The execution of the project is carried out primarily by the project team members, and the control of the project is carried out primarily by the project manager. PMI defines several processes that support the overall process of project execution (PMI, 2000), that is, activities that should be taking place while the team does it work:

- Information dissemination (i.e., reporting)
- Team development
- Scope verification
- Quality assurance
- Procurement activities (solicitation, source selection, contract administration)

The Software Engineering Institute's (SEI; www.sei.cmu.edu/cmm) CMM also implies necessary practices (Level 2) for project tracking and oversight:

- Are the project's actual results compared with estimates in the plans?
- Is corrective action taken when actual results differ significantly from the plan?
- Are changes agreed on by all affected parties?
- Does the project follow a written policy for tracking and control of activities?
- Is someone assigned specific responsibilities for tracking work products and activities?

Figure 9.1. The control process



- Are measurements used to determine the status of the tracking activities?
- Are activities for tracking and oversight reviewed with upper management?

What to Control

Once the decision is made to measure and control project matters, the next question is, What to control? PMI (2000) defined several processes that support the overall process of project control:

- Scope change control
- Schedule control
- Cost control
- Quality control
- Risk response control

For most projects, the consumption of resources (time and cost) is always controlled, and the previous list of supporting processes adds the control of scope, risk, and quality. Other matters need to be considered, however, for effective management of IT projects. The Standish Group Reports (called “CHAOS”), which are based on large surveys in years from 1994 to 2004 (Standish, 2004), provide some guidance. The reports list the leading causes of IT project failures as:

- Lack of executive management support
- Insufficient user involvement
- Inexperienced project manager
- Business objectives unclear
- Minimization and compromise of scope
- Lack of standard software architecture/infrastructure
- Lack of clear statement of requirements
- Lack of formal methodology
- Poor estimates
- Lack of proper planning
- Unrealistic expectations
- Scope minimized
- Lack of project “ownership” by team
- Team not hardworking and focused

- Vision and objectives unclear
- Incompetent staff
- Improper setting of milestones

The order of the Standish list represents the ranking of the problems in its 2000 report. The list in other years ordered the issues differently; for example, in 2004, user involvement came first and executive support was second. This list suggests that other matters that need to be tracked include management support, requirements verification and validation, user involvement and attitudes, project management practices, business objectives and vision, technology issues like architecture and infrastructure, methodology, and project team attitudes.

Another qualified source for matters that should be controlled in IT projects is the “Seven Core Metrics in the rational unified process (RUP), which was discussed in Chapter V (Royce, 1998):

- Work and progress
- Budgeted cost and expenditures
- Staffing and team dynamics
- Change traffic and stability
- Breakage and modularity
- Rework and adaptability
- Mean time between failure and maturity

All these project matters that need to be controlled (as suggested in the three previous lists) are already included in the general model for IT project critical success factors. This is a general model, and therefore some of these factors may not be relevant or important for any particular project or some other peculiar factors may need to be included. Figure 9.2 shows a spreadsheet that organizes these items, specifies key measurements for each, and indicates the book chapter(s) that provides detail coverage thereof. Each of these areas is briefly defined in Figure 9.2 and discussed in more detail with specific techniques in other chapters of this book.

Measurement of Completion Factors

Once it has been decided which project matters need to be controlled, the measurements that need to be made must be decided upon. Figure 9.2 indicates the measurements for our critical success factors of IT projects, and we will first examine the completion factors.

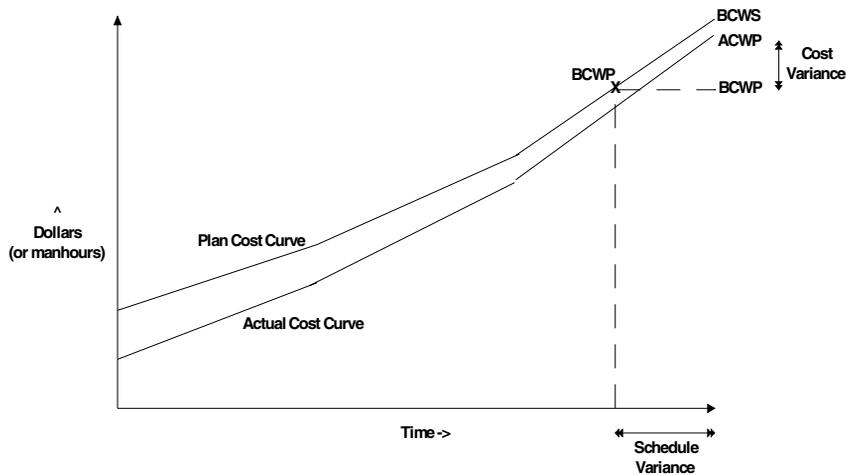
Figure 9.2. Book content and success factor

Project Control		
Success Criteria	Metrics	Book Chapter
Completion Factors		
Project Management		
Schedule (Time)	EVA SPI	XIV
Cost	EVA CPI	XIV
Progress	Percent Complete, EVA CR	XIV
Scope	CO Approved/CO Submitted	XI
Team Morale	Surveys & Interviews, Staff Attitude, Turnover	XIII
Stakeholder Morale	Surveys & Interviews	XIII
Risk	Planned vs Unplanned Mitigation	VIII
Methodology		
Change Introduction	\$ CO/ \$ Budget	V, X
Change Resolution	CO Completed/CO Approved	V, X
Commitment to Perform		
Sponsor Support	Interviews	XIII
Upper Management Support	Interviews	XIII
Ability to Perform	Current Budget, Budget Horizon	VI, X
Verification		
Defect Introduction	Defects per 1000 LOC (by phase)	V, X
Defect Resolution	Defects Reported/Defects Corrected	v
Technology		
Process	Productivity (LOC per person hour)	V
Product	Product-related metrics	V
Satisfaction Factors		
Business Justification	ROI, IRR, Payback Period	III, X, XVI
Validation		
Stated Requirements	Requirements Document Approval	VI
Unstated Requirements	Preliminary Product Manifestation Review	X
Acceptance Testing	Acceptance Exceptions (Changes & Defects)	X
External Quality Dimensions	Product-related Metrics	X
Workflow & Content	Preliminary Product Manifestations Review	X
Standards	Compliance Audits and Inspections	V, X
Maintainability & Support	Comment Ratio, LOC per CO, Code Walkthrough	V, X
Adaptability	Reuse %, Avg Time per CO, Code Walkthrough	5, 10
Trust/Security	Security and background checks	XII, XIII
Process Security	Security incidents, lost time	VI
Product Security	Intrusion testing	VI, VIII, X

Project management addresses the use of proper project management skills and methods in dealing with each of the nine PMI knowledge areas. The following is a list of subdivisions

- *Schedule, cost, and progress* control has been traditionally handled via Gantt chart analysis and budget-vs.-actual cost metrics. A much better method of measure-

Figure 9.3. Earned value analysis



ment and analysis is earned value analysis (EVA), detailed in Chapter XIV. EVA uses three measurements and provides three key metrics: cost performance index (CPI), schedule performance index (SPI), and critical ratio (CR). These metrics can be used to estimate project time to complete and cost at completion. The three measurements are budgeted cost (BCWS), actual costs (ACWP), and earned value (BCWP); these are illustrated in Figure 9.3. Earned value is a function of the progress measured in terms of “percentage complete” of each WBS task.

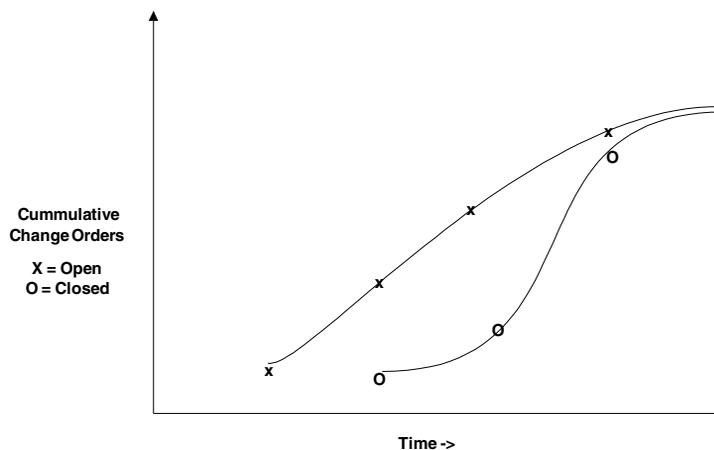
- *Scope control* comprises the management of project change for the benefit of the project and resulting product. A PM must avoid *scope creep* and *gold plating*, and depending upon the contract situation, must make sure that all changes are estimated, priced, and billed. A key metric is the ratio of approved change orders to the total requested change orders. If the majority of change order requests are being approved (as opposed to being denied or deferred to another phase/version), then either changes are being accepted that are outside of the initial scope or the initial scope was ill defined. Change control is discussed in detail in the chapter on change management.
- *Team morale* and the morale of other stakeholders can be monitored through various types of interviews or surveys. The quality stage gates technique, discussed throughout this book, provides one opportunity for such monitoring. Other indicators of team morale problems include poor attitudes and complaints about the project and assignments, staff turnover, unplanned overtime, and low productivity.
- *Risk management* comprises the identification, quantification, and mitigation of potential risks to the project. This was discussed in detail in Chapter VIII, including specific metrics and techniques for IT risks. The key measure of risk planning effectiveness is the number of planned versus unplanned mitigation (workarounds).

Methodology comprises the selection of specific IT software engineering processes (requirements analysis, systems analysis, design, development, documentation, testing, etc.) and how these processes will be organized, utilized, and integrated, both among themselves and with the project management processes. (Chapter V discussed these subjects.) If the methodology chosen is inappropriate for the project, then there will be a high percentage of the project cost for changes and corrections. For measurement purposes, these costs can be subdivided into the following areas:

- *Change introduction* can be measured by the money (or person hours) in change orders relative to the total budget. Values over 10% may indicate the need for a different methodology or smaller increments if an incremental methodology is already in use. As discussed previously, large IT projects have lower success rates than smaller projects because the complexity grows according to the square of the number of items (people, requirements, technologies, etc.) involved. Excess change causes the project to grow and become more complex.
- *Change resolution* can be measured by the ratio of completed change orders to total change orders submitted and approved. As a project draws towards completion, the curve of completed change orders should approach the curve of approved changed orders, as is shown in Figure 9.4. If these two curves do not start to converge, then the change process is out of control due to an inappropriate choice of methodology.

Commitment to perform comprises upper management support for the project from both the project sponsor and upper management. It can best be measured by personal interviews and dialog as the project proceeds. In many organizations, this involves “politics,” to which either the PM and/or the project sponsor must be a party.

Figure 9.4. Change orders

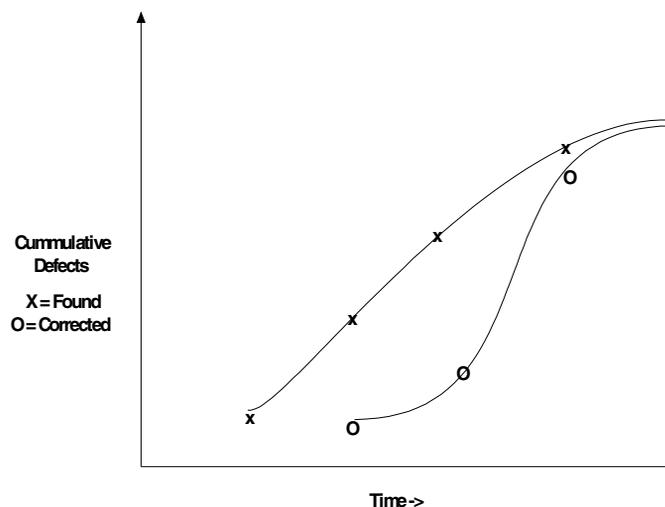


Ability to perform is having the amount of resources needed and the correct resources to carry out the project plan. The key measurement in this area involves budgetary matters including both the current fiscal period budget and the budget horizon for longer term projects and multi-phase projects. In this area the PM must work with the project sponsor and upper management to make sure that continued funding for the project is included in strategic planning forecasts.

Verification involves built-in quality or defect prevention and concerns the quality of the development *processes*, thus answering the question, “Have we built the product right?” Formally, verification is proof of compliance with requirements, specifications, and standards. Verification processes usually result in exception (bug) reports, in which compliance is not achieved. For measurement purposes, this is implemented via internal testing and inspection and can be subdivided as follows:

- *Defect introduction* measures the defects relative to the size of the development effort, the common measure being defects per 1,000 LOC (KLOC). The basis for the KLOC may be the total LOC or just the LOC in new code, not including any “reused” code (“included” code). If the basis is the total code, greater reuse will mean a lower defect introduction rate. Current industry values average between 2 and 5. Defect introduction should also be analyzed based upon where the defect was originally introduced (requirements, design, coding, testing, etc.); this is discussed in more detail in Chapter X.
- *Defect resolution* measures the number of defects reported versus the number of defects corrected, and may be measured in terms of numbers or cost (i.e., in money or in person hours). Similar to the situation with change orders, as a project draws toward completion, the curve of corrected defects should approach the curve of found defects, as is shown in Figure 9.5. If these two curves do not start to

Figure 9.5. Defects



converge, then the defect situation is out of control due to the lack of built-in quality or an inappropriate choice of technology.

Technology comprises the proper selection of applicable technology for use in both the product and in the process of building the product. It covers architecture, platform, language, and supporting technology selection as well as issues of each, including the maturity, stability, and support thereof. It can be subdivided into the following:

- *Process* comprises the technology used to construct the product, and a key measure is productivity rate, usually measured in function points and/or KLOC per person, per hour, or per day. Design processes typically use function point-based metrics, and coding processes typically use KLOC metrics. Depending upon the chosen programming language, most programmers average anywhere from 25 to 100 KLOC per day (code developed prior to integration testing), with an average value of 50 to 60.
- *Product* comprises the technology built into the product, including the choice of platforms and other dependent software. There are a number of product-related metrics, including recoverability, scalability, and portability. Another key metric is whether the product is (or can be) based on open source components (runs on or utilizes open source products, such as Linux, Apache, MySQL, etc.).

Measurement of Satisfaction Factors

In a similar manner, the satisfaction factors can be examined to determine the necessary key measurements:

Business justification comprises the selected type of cost-benefit model(s), as was discussed earlier in this book. The business justification should be revisited and measured during the project to ensure that the assumptions in those models are still correct and relevant. This is part of the quality stage gate analysis process.

Validation comprises the *product*, which is the subject of the project and checks all user requirements (both stated and expected) and answers the question, “Have we built the right product?” Formally validation is proof that the customer and end users are satisfied with the system. Validation processes usually result in change orders when the user is not satisfied with an aspect of the product. Proper user involvement is vital to this aspect of the development and/or integration process.

- *Stated requirements* (needs) are initially manifested in some type of requirements document, as was discussed earlier in this book. Additional requirements are formalized via the change order approval process. The requirements are also included in more tangible forms in other documents, such as a users manual.

- *Unstated requirements* (wants and expectations) are best discovered via user review of preliminary product manifestations, such as use cases, design drawings, paper prototypes, and live prototypes. Unstated requirements were discussed in detail in Chapter V, and are part of the quality stage gate analysis.
- *Acceptance testing* is the final and formal measure of user satisfaction with the product (see Chapter X for more details on acceptance testing).
- *External quality dimensions* are a part of validation, even though they may not be a formal part of the contract or the acceptance testing. External quality dimensions must be addressed for the long-term success of the product and the performing organization. Included herein are measures such as usability, reliability (does the product do it right all the time), robustness (product can handle invalid/unusual data and usage), responsiveness and efficiency (with respect to speed, storage, clicks, keystrokes, and other resources), testability, auditability, and capacity/scalability.

Workflow and content comprises the effective integration of the new product into the organization's (and each user's) workflow. Content includes all deliverable information, including documentation, help system, data, and media content (especially in the sense of modern and Internet applications). The measures for this aspect of the project are the degree to which the deliverables corresponding to these items have been completed and the degree of the customer's satisfaction with such manifestations of the product.

Standards relate to compliance with applicable industry, corporate, and user (customer) standards in regard to both external (i.e., user interface) and internal issues (i.e., coding standards). (See Chapter X for more details on standards.) The measures for this item are compliance audits and inspections, which may be exhaustive or selective ("spot" checks).

Maintainability and support involves the inherent maintainability of the developed product and the willingness and timeliness of the developing (or support) organization in responding to the customer's concerns about usage or integrity (real or perceived) issues. For IT projects that succeed, 70% of the total life-cycle cost of the product is spent in the maintenance phase; thus, maintainability is extremely important. Maintenance programmers also spend about half of their time studying the existing code (Standish, 1984). Therefore, if the code is easy to read and easy to understand, support costs are lower. One traditional but important metric is the ratio of comment statements to executable statements in the code. For well-written 3GL code, this ratio should be about one comment line per every one to three code lines. Periodic code walkthroughs are also necessary to ensure the quality of the comments and the quality (and standards compliance) of the code. Another measure of maintainability is the lines of code affected per change order. More comprehensive metrics in this area include both the lines of code affected and the lines of code examined per change order.

Adaptability relates to the flexibility of the product to be adapted (successfully modified) for evolving changes in the environment in which the product is deployed; this includes both technical changes and business changes. Design and code walkthroughs are necessary to ensure proper object-oriented techniques. One key measure is the ratio of

the total code that has been reused (from object-oriented libraries and packages). The percentage of reused code should be over 50 for most projects in modern IT environments. This metric can be misleading if one imports simply because that code is part of the overall imported module, but that code is not actually used. Modern object-oriented implementations usually have ways to measure only the code that is part of the executable program. One cannot assume that the cost for reusing code is always negligible. Another important measure is the average time (person hours) per change order, because adaptable systems are able to be changed faster.

Trust and security relates to both the security built into the product and to the security of the process for building the product. Security must start with the project stakeholders, particularly the people involved in designing, building, and testing the product; these individuals (both employees and contractors) should undergo complete security and background checks.

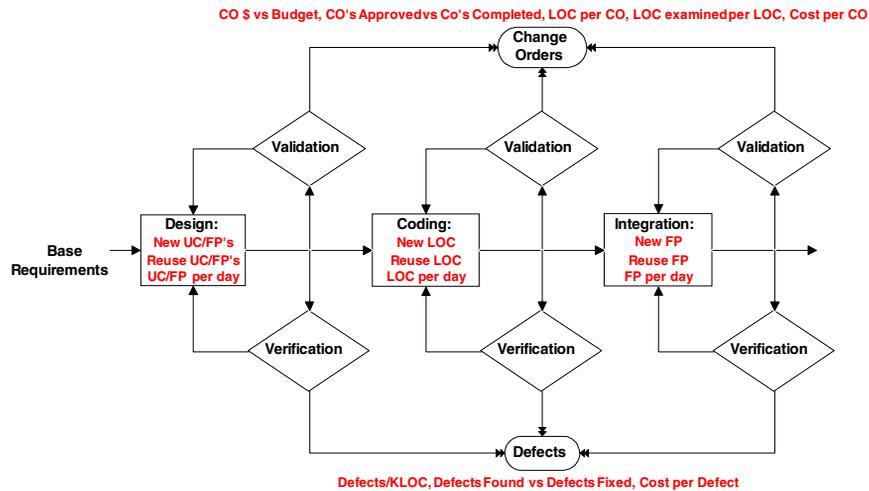
- *Process security* metrics would include counts and severity of security incidents that occur during the project. A related metric would be lost time due to security problems.
- *Product security* involves the customer's willingness to fully utilize the system in all necessary modes without concern for compromising any of the customer's assets, including information assets. This type of security must be built into the product, and the metrics involve special intrusion testing in regard to security holes.

Measuring and Reporting

To avoid spending more effort on measure and control than can possibly be gained in return, appropriate measures are necessary. The metrics for a project must be selected carefully and appropriately for the size and complexity of the project. Once the desirable measures are selected, the next step is to find the best way to obtain those measurements. It is important, particularly for projects involving professionals, to have measurements that are noninvasive and consume very few additional resources. If possible, the measures should be a byproduct of normal work or of other normal and required processes, such as basic time/attendance and/or status reporting. For IT projects, many of the measures can be obtained automatically by the choice of methodology and technology and the supporting tools thereof; this is illustrated in Figure 9.6. For most IT organizations, the chosen system of measurements is put into place for all projects performed by that organization; certain metrics for small or simple projects may be excepted. A project management office (PMO) often coordinates this overall measurement system and its supporting tools and techniques (PMOs are discussed in Chapter XVI).

The reporting process consists of taking the information from established metrics and summarizing some of that information for some stakeholders; not all measures are

Figure 9.6. IT project metrics



reported to all stakeholders. A communication plan indicates specific information that is reported to certain stakeholders, and included how and when that information is reported. (Communications and stakeholder management is discussed in Chapter XIII.) Although not all information is reported to all stakeholders, there may be organizational policies and procedures or contract provisions that dictate certain reporting requirements; however, the PM should always follow the following key principles in reporting progress and other measures:

- Honesty is the best policy!
- Bosses hate surprises!
- Bad news does not get better with age!
- Document all issues and problems!

A simple progress report form is shown in Figure 9.7. Problematic projects require closer monitoring than nonproblematic. Some warning signs of “runaway” projects were given in PM Network (Block, 1999):

- Inadequate project planning
- Faulty task management
- Poor reporting and communications
- Infrequent status reports
- Insufficient documentation
- Abrupt schedule changes

Figure 9.7. Project status

<u>Project Status</u>	
Project Code: _____	Report Date: _____
Project Name: _____	
Benefiting Organization: _____	
Performing Organization: _____	
Project Manager: _____	
Business Evaluation: _____	

Milestones Met:	
1. _____	
2. _____	
N. _____	
Earned Value Metrics	
Actual Cost: _____	Planned Cost _____ % Complete: _____
Budget Variance: _____	Earned Value: _____ Critical Ratio: _____
CPI: _____ BAC: _____	EAC: _____ ETC: _____
SPI: _____	Estimated Completion Date: _____
Risks Status: _____	

Explanations/Notes _____	

Corrective Actions Necessary: _____	

Approvals	
Benefiting Organization	Performing Organization
By: _____	By: _____
Date: _____	Date: _____

- Project disorganization
- Muddled business objectives
- Extreme project complexity
- Escalating costs
- Too many project team meetings

Stage Gate Implementation

As suggested earlier, the use of a dual stage gate approach to project performance reporting and control. This is illustrated again in Figure 9.8. Multiple quality stage gates may be within one management stage gate or vice versa. This dual gating process minimizes the time that upper management and the project team spends in status reporting meetings by splitting the review process into separate completion and satisfaction reviews with the occurrence of each, based upon the need thereof. However, it ensures that customer involvement is sufficient in the project matters that most concern the users.

Figure 9.8. Dual stage gates

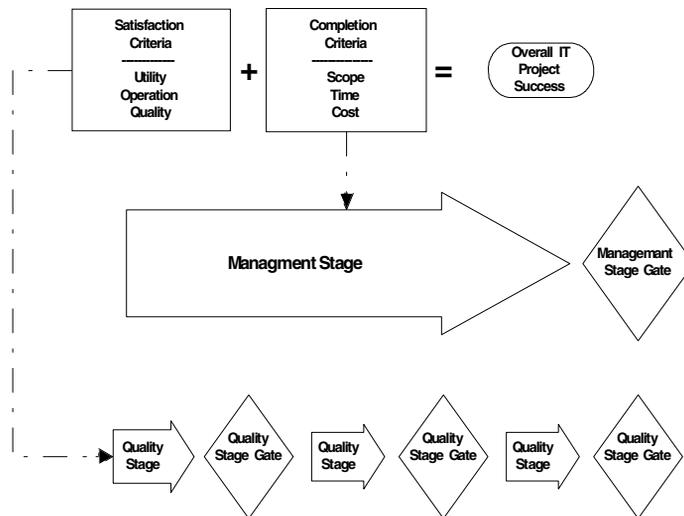
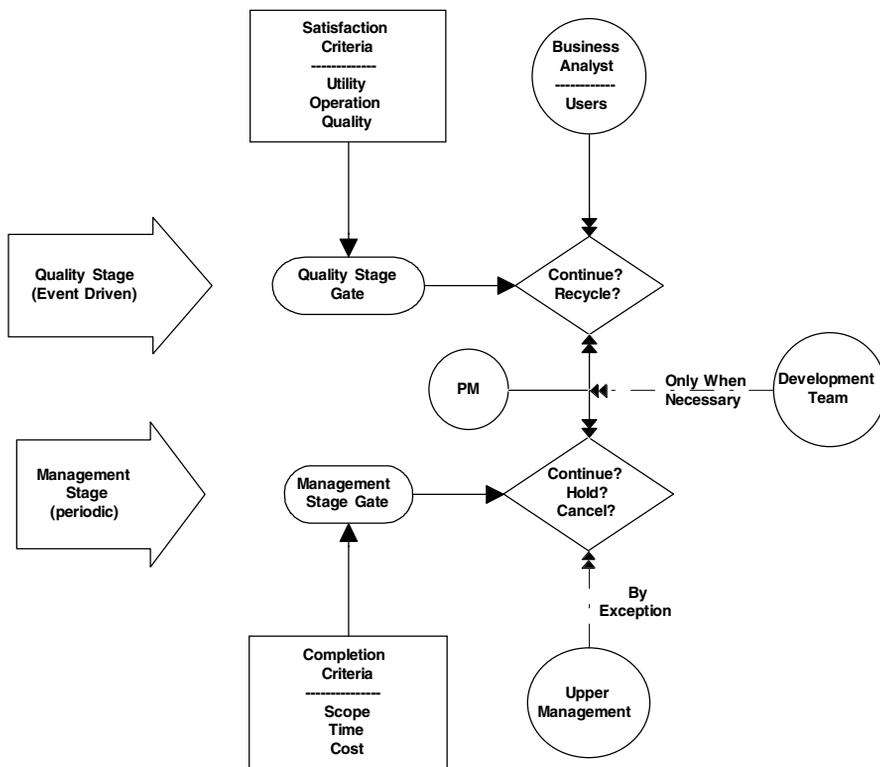


Figure 9.9. Stage gates and stakeholder involvement



Management stage gates handle the completion criteria, and these stage gates can be set at fixed time intervals or upon completion of a major project phase. For methodologies in which phases overlap, or when using incremental or iterative approaches, the management stage gates can be set between increments or iterations, or upon fixed-time increments, such as 1 month. To minimize the time and cost associated with formal management reviews, it is recommended that the management stage gates happen at regular time intervals but that they take place only on an exception basis, when key metrics (such as EVA indexes) indicate problems. At the management stage gate, completion metrics are reviewed (either exhaustively or by exception) and a go/kill/hold decision is reached. This is illustrated in Figure 9.9.

Quality stage gates address the satisfaction factors that are examined by relevant stakeholders for that gate via focusing on a particular preliminary product manifestation; the review does not make a go/kill/hold decision but rather a go-forward or recycle type decision. Figure 9.9 illustrates these distinctions. These concepts are discussed in more detail in Chapter X.

Corrective Actions

Measurements are compared to some desired values or standards to determine deviations and differences. If the difference between the measurement and the desired value exceeds a threshold, corrective action of some type may be necessary. In project management, the need and amount for corrective action depends upon the direction and magnitude of the deviation. Deviation rates and accumulations may also factor into the type and magnitude of the chosen corrective action as well as the point in the project that the deviation is discovered.

Controls need to be appropriate for the type, size, complexity, and organizational environment of the project, or one may spend more on the effort to control than can possibly be gained in return. A PM cannot assume the job is finished when he or she has implemented the controls. In addition, accurate results may not be obtained unless the project team and other subordinates also buy into the controls. Some controls are needed, particularly in IT, and one should:

- Keep them as simple as possible
- Control only what needs to be controlled
- Minimize the additional work of data acquisition and processing for the controls
- Let the project team and other employees in on the controls and the overall purpose thereof

In general, the PM uses the measures and controls to track and review but does not intervene directly unless there is a problem. Corrective actions are management prerogatives that are available to a PM (and upper management), based upon the type of

organization (functional/hierarchical, projectized, or matrix, as is discussed in a later chapter), the position of the project manager, the organization culture, HR policies, possible union rules, and the governing laws of the state or country. Some corrective actions may be operational, some tactical, and some strategic.

Managing a project involves the trade-off of some key variables, in particular, scope, time, and cost. There is an old expression in project management that says, “Do you want it good, fast, or cheap—pick any two.” If the project is behind schedule, one can consider adding additional resources (people, money, etc.). This will cost additional money, which may be acceptable if the project is below cost or if the customer is more concerned with schedule than cost (contract permitting). Changing resources and obtaining better-cheaper-faster resources may also be considered, but there will be a cost and delay for the changeover (and added risk), and if it is late in the project or if it is a relatively short project, the net benefit may be minimal. If the customer is more concerned with cost than schedule, the schedule can be extended by adding more time, contract permitting. Another alternative is to reduce the scope by deleting features, by moving features into a future phase (or version), or moving features into the maintenance phase. Still another alternative is to reduce quality by:

- Increasing tolerances (safety margin, backup, etc.)
- Reducing testing
- Forcing the testing onto users
- Not fixing all the defects

These measures usually increase the long-term cost of a product, but some software companies often use these techniques.

Duration compression is another technique that can sometimes be used to correct schedule problems. There are two types of duration compression: crashing, which may be combined with reductions in scope and/or quality, and fast tracking. Both of these methods usually increase project risk. Crashing involves allocating more resources to the critical path tasks; according to PMBOK, “Taking action to decrease duration by analyzing a number of alternatives to get the maximum compression for the least amount of cost” (PMI, 2000). Resources may be taken from noncritical path tasks if resource types

Figure 9.10. Task crash evaluation

Task	Duration	Crash Time Savings	Crash Cost	Crash Risk
A	15	2	3000	HIGH
B	10	2	9000	LOW
C	4	1	1000	LOW
D	8	1	2000	HIGH
E	12	3	8000	NONE

are same or similar. This may result in higher costs; however, for many tasks (particularly IT tasks), increasing resources may not speed up those tasks. A common analogy applies: It takes one woman 9 months to have a baby; the process cannot be accelerated by putting nine women on the job for 1 month. According to Brook's Law, adding manpower to a late IT project makes it later (Brooks, 1995).

As an example, consider the project-critical path tasks shown in Figure 9.10. If one had to cut 3 months time off the project and the schedule was most important, followed by cost and then risk, then tasks A and C would be crashed. Instead, if one had to cut 3 months time from the project, and schedule was most important, followed by risk and then cost, tasks E would be crashed.

Fast-tracking is starting tasks that fall later in the project schedule before their predecessors are completely finished and/or reducing lag times. This also typically increases project risks. For traditional waterfall methodologies, fast-tracking may include starting one phase before its predecessor phase is completed (i.e., starting coding before the completion of detailed design). Fast-tracking to some extent is already a part of overlap, iterative, and incremental methodologies, which are discussed in Chapter V. Consider the small project schedule shown in Figure 9.11. Any task on the critical path that has a predecessor task could be fast-tracked by starting it before the predecessor was completed (if it was physically possible to do so); here, task B, C and/or G.

Many other corrective actions may be available to PMs that may improve the productivity or resourcefulness of the PM's project team members and thus reduce time or cost, or improve quality. These actions include employee/contractor disciplinary actions (negative reinforcement), employee/contractor incentives (positive reinforcement), and "pep talks" and other motivational techniques. A PM must maintain the strong commitment of stakeholders throughout the project, and these human resources issues are discussed in detail in Chapter XIII. Of course, upper management may also choose to change the PM.

There may be systematic problems in the project due to the choice of methodology, technology, or tools. Defects introduced early in the overall development process are more expensive to correct than defects introduced later. For this reason it pays to focus defect prevention during the early processes, such as requirements and design. An analysis, such as the one shown in Figure 9.12, examines where the defects are introduced versus where they appear.

Figure 9.11. Network PDM diagram

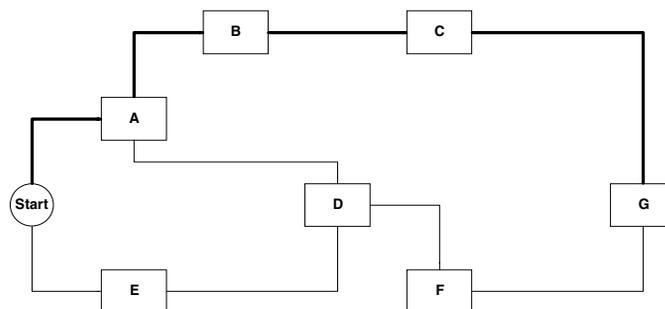


Figure 9.12. Defects by stage

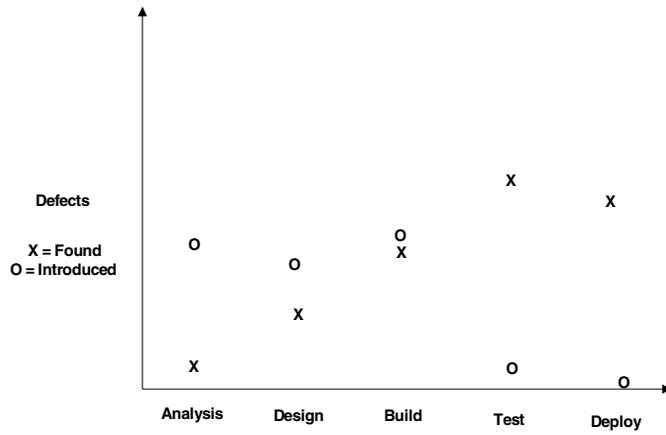
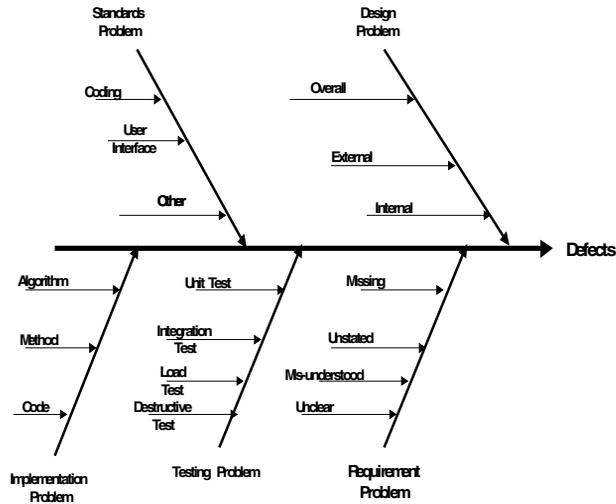


Figure 9.13. Cause and effect diagram



For systematic problems, one first needs to find the root causes. Fishbone diagrams, as is shown in Figure 9.13, are a good way to analyze these types of problems in detail so that an eventual solution will be discovered.

Chapter Summary

In this chapter, project performance control and corrective action techniques were defined and discussed. Performance metrics for each critical success factor were identified and illustrated. PMs must often be clever and innovative to solve difficult project control problems. A common analogy is that upon running into a “brick wall,” a PM must examine all alternatives and find a way to go over, under, or around that wall. Cockburn (as cited in King, 2004) stated in *Computerworld*: “A core part of the job of project manager is coming up with inventive ways to get out of incredibly constrained situations.”

References

- Block, T. (1999, April). The seven secrets of a successful project office. *PM Network*.
- Brooks, F. (1995). *The mythical man month: Essays on software engineering, 20th anniversary edition*. Boston: Addison-Wesley.
- King, J. (2004, February). Project management in the trenches. *Computerworld*, p. 16.
- PMI. (2000). *The project management body of knowledge (PMBOK)*. Newton Square, PA. ISBN 1-880410-22-2.
- Royce, W. (1998). *Software project management—A unified approach*. Addison-Wesley.
- Standish Group. (2004). *Chaos chronicles*. Retrieved from www.standisgroup.com
- Standish, T. (1984, September). An essay on software reuse. *IEEE Trans on Software Engineering, Vol.SE-10(5)*.