Why Technical Projects Fail: Avoiding Disaster

By Glenn P. Kessler

Successful project management requires a carefully tailored blend of technical understanding, team building expertise, public relations know-how, political savvy and project management basics mixed with large doses of optimism, energy, creativity, and perseverance. Given this broad range of required project management skills, it's not surprising that many project managers fail to deliver their projects on time, on budget, and with the required functionality. What may be much more surprising is the enormity of the odds against success (over 80% of technical project efforts fail) and the reasons why most technical projects are derailed.

This article does two things: it presents the reality behind technical project management and surveys 6 of the most common and devastating technical project management errors. Managing a technical project is difficult. Managing it to success is orders of magnitude more challenging. The good news is that the major barriers to success are not a mystery. There is substantial agreement on those factors that make project management such a risky enterprise. These factors are reviewed in brief in the context of some fundamental project management principles. In subsequent articles, we'll examine these project management traps in detail, illustrate them with real life examples, and offer specific suggestions and techniques for increasing your own chances for success.

The road less traveled: successful project management

To set the stage, let's survey the project management landscape. Imagine a map of the United States. The Northeast and Mid-Atlantic states (say Maine through Washington) are one color. The rest of the East, South, and Midwest to the nation's heartland are a second color. The entire western portion of the country is a third color, as shown in **Figure A**. This gives us a rough picture of the technical project management landscape.

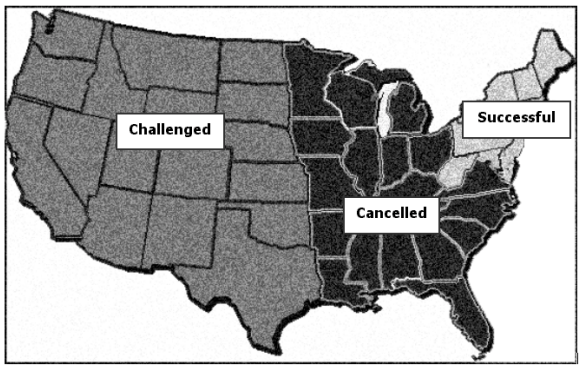

Figure A: *If technical projects were states, only a small portion of the U.S. would be economically viable.*

According to an often-cited report called "CHAOS" from The Standish Group International (see sidebar, Project Management CHOAS), of the technical projects undertaken (small, medium and large across a range of industries) only 16% were fully successful. These few fully successful projects were on time, within budget, and delivered with the full complement of promised requirements. Nearly 53% were "challenged" or completed unsuccessfully. Of these, the average cost overrun was 189%. The average time overrun was 222%. Overall, only 61% of their required features were delivered. The balance, the remaining 31%, was cancelled at some point in the development cycle.

In other words, if technical project starts were distributed evenly across the contiguous United States, and if a region's economic stability depended the success of those projects, most of the country would be in very difficult straits.

The moral is pretty clear. A project leader embarking on a new technical project is facing an extremely difficult journey. He or she is covering territory littered with the evidence of past failure. To the degree to which you're able to read and interpret these signs correctly, your chances of making it to the Promised Land (an on-time, on-budget, fully featured product) will increase. With this information in mind here's an interesting exercise. Make a list of all the projects you've managed over the past few years. Classify them as successful, challenged or "impaired" (i.e., cancelled). Run the numbers and compare your percentages to those in the study.

The basic axioms

Seen against a landscape strewn with project failure, it's striking that there already exists a substantial and very compelling body of knowledge that addresses both *why projects fail* and *how to prevent their failure*. (As noted in another

report from The Standish Group, "Unfinished Voyages," a much more vexing question, which we'll touch only by inference, is why – given what we know – they *continue* to fail.)

To put things in perspective, let's lay down a few fundamental principles from which the bulk of project management knowledge flows. These share a number of traits with other fundamental principles (e.g., Euclid's axioms for geometry, Peano's postulates for elementary arithmetic, the intuitive principles of set theory, etc.). They are intuitively obvious but their consequences are not. They are powerful in their ability both to predict and explain things with which we are already familiar (e.g., that projects are more likely to fail than succeed – and *why*). They point to connections in directions we may not yet have considered. One such connection is the striking similarity between the technical project management and process reengineering "failure maps." Here are the principles with convenient labels:

Fundamental Project Management Principles

Context Principle: Technical projects take place in a broader context

Entropy Principle: The amount of disorder in projects will not, of itself, decrease with time

Society Principle: Technical projects are social undertakings

The specific pitfalls we consider in this article derive from failures to understand or deal with these very fundamental truths about project management. Keeping these straightforward principles in mind can go a long way towards avoiding their most debilitating consequences.

Caution! Obstacles ahead

Any project manager can reflect on his or her own experience and derive a personal list of the most frequent and devastating project pitfalls. Odds are pretty good that these personal favorites are covered in the list developed in this article. The list is drawn from several sources including published research, seminal articles and books, shared war stories with colleagues, a project manager survey conducted for this article, and first-hand experience. Not surprisingly, there's substantial overlap in the results from

these diverse sources. Let's take a closer look at the three main principles and their implications.

The Context Principle

Technical projects take place in a broader context.

Technical projects seldom take place in isolation from the rest of the organization, its customers, or a broader environment. If they do, their value is likely to be pretty limited. Most (and the most important) "technical projects" are not just "technology projects," they are *business* projects. Technical projects, no matter how complex and sophisticated the technology, are undertaken for business reasons. They invariably involve other aspects of the organization and, in the best cases, the customer as well. A corollary is that successful technical project management demands skills that extend way beyond the technology realm. Failure to adequately understand and attend to this broader context, treating a technical project as a "technology issue," underlies a host of common and devastating technical project management pitfalls. Kathy Warden, Senior Vice President of Consulting at Equient, expresses this principle as the need for a "holistic life cycle for managing the project— one which considers process, organization, and technology change as components of the solution."

Although it's fairly obvious, violation of this principle takes first place among reasons why technology projects fail. The most frequent and devastating violation of the context principle is:

Violation 1: Lack of adequate user input and involvement in the project

Inadequate downstream participation is familiar, well documented, and a continuing project management favorite. It's unusual to find a project manager who does not, based upon personal experience, place it near the top of the list.

The principle is simple enough. Those who will *use* a product have a valuable perspective on its design. And the cost of ignoring this perspective is extremely high. In the first place, incorporating end-users early in the process significantly reduces their natural reluctance to embrace a "foreign" solution. This tendency, covered extensively in the change management literature, can easily torpedo a complex implementation effort. Of equal importance, is the fact that errors introduced early in a project, such as a mistake in requirements, get increasingly more expensive to correct as the project moves for-

ward. In his Software Project Survival Guide Steve McConnell observes that an error can cost 50 to 200 times as much to correct late in the project as it does to correct early in the project life cycle. This suggests an equally important consequence of the context principle:

> Downstream input, in the form of requirements validation, must be *iterative*.

Technical project managers are solving user problems rather than building systems. These solutions need to be honestly and frequently validated by users long before the final product is delivered. Without iterative validation the project manager runs a strong risk of delivering a product which either:

- Does not satisfy the requirements as understood by the end-user, or
- Satisfies requirements that are no longer valid due to changes in the broader landscape.

What's the best mechanism for effective downstream input? There's extensive literature to choose from including the disciplines known as "contextual design", as discussed in Hugh Beyer & Karen Holtzblatt Contextual Design, and quality function deployment (QFD).

The next violation of the context principle runs a very close second:

Violation 2: Lack of sustained executive management support

This problem is nearly always catastrophic and often derives from a fundamental but frequently overlooked organizational parameter that I refer to as "*MTBR*": mean time between reorganizations. I've found the following rule of thumb to be a particularly useful consequence of the context principle:

> Elapsed time between significant project milestones should not exceed the organization's MTBR.

In many organizations the expectation of sustained executive support for a complex and lengthy project is simply not realistic. This is particularly true in turbulent economic times. Ask yourself whether today's management is likely to be around to see the completion of the project. Any realistic assessment of project risk

will have to consider the possibility of a negative answer. Creating smaller well-defined project milestones, in addition to keeping the project on track, provides a mechanism for bridging executive level organizational shifts.

Another familiar and debilitating consequence of vacillating executive support is the reshuffling of organizational priorities. This can mean the outright cancellation of a project (making it one of the 31% that never see the light) or reallocation of crucial resources.

The Entropy Principle

The amount of disorder in projects will not, of itself, decrease with time

As with any isolated system, a project, left to itself, will drift toward chaos. The project manager's task is to impose (or extract) order in the face of this natural tendency. This is the point of well-defined and repeatable project management processes.

There is by a now a vast body of knowledge which details the sort of order required for project success and the specific areas in which it must be imposed. One example is the "key process areas," presented in Figure B, and their associated "key practices." These are the core of the second level of process sophistication (there are five levels in all) in the Capability Maturity Model (CMM) developed by the Software Engineering Institute. They offer one effective remedy to project entropy.

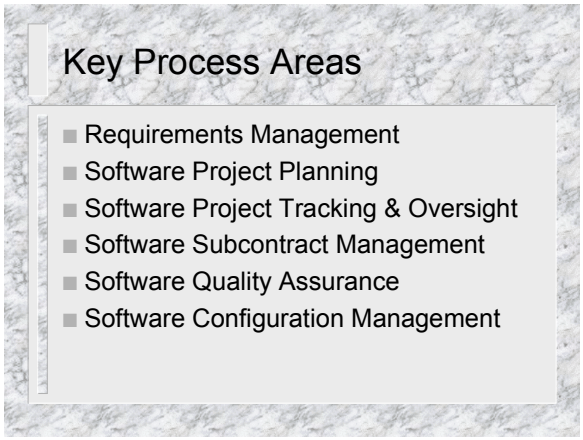

Figure B: *These are the six Key Process Areas at the second maturity level ("repeatable processes") of the Capability Maturity Model.*

The predominant pitfall in the project entropy category will be familiar to any seasoned project manager:

Violation 3: Unclear requirements

We noted above that failures introduced early in the development cycle are the most costly to correct downstream. Unclear, incomplete, and inaccurate requirements are three frequent examples of these early life cycle errors. However, clear requirements are not enough. A related and no less debilitating entropy error is the lack of a well-defined process for *managing* changes to these requirements. Without such a process, the project invariably succumbs to "death by a thousand changes" frequently taking the form of the "feature creep" or even "scope creep." These amount to an unauthorized (and sometimes unnoticed) alteration in the development plan. The requirements management key process area in the Capability Maturity Model provides a solid framework for avoiding this pitfall.

The fourth major violation is probably the most obvious of all:

Violation 4: Lack of proper planning

The most interesting point about this violation is that it's not at the top of the list. While project planning (as embodied in the CMM key process areas listed above) is often considered job #1 of a technical project manager, it is not the reason that technical projects most frequently jump the track. We need to look to *context* rather than *entropy* for the most frequently cited errors. (Whether this is because project managers are particularly good at fighting entropy, particularly poor at context and society issues, or something else is the topic for another article.)

Of the specific problems related to lack of project planning, unrealistic expectations in the form of overly aggressive schedules and inadequate budgets are cited most frequently. Once again, there is a wealth of literature in these areas. For a good informal discussion of realistic project schedules see Chapter 3 of McConnell's Software Project Survival Guide cited above.

When we think of project management and the ways it can fail, items in the entropy category are often first to mind. This is the "technical" side of project management. This side of project management deals with formal methods and practices. While a lack of process is certain death for a project, the context principle tells us that an exclusive focus on these issues is also sure to result in failure.

The Society Principle

Technical projects are social undertakings

Projects are social undertakings. I have never encountered one that did not require the collaborative effort of a group (or, more frequently, multiple groups) of individuals to achieve its defining objectives. Unlike some social enterprises (e.g., a church, a community service organization) they have a very specific duration and objective. But this does not make them any less social nor any less subject to the familiar political and interpersonal dynamics characteristic of the many dysfunctional organizations we have all come to know and love. Technical projects are social organizations in microcosm.

The Society Principle implies that along with organizational agility (the Context Principle) and project management expertise (the Entropy Principle) a successful technical project manager requires appropriate people management skills. This is the "management" side of technical project management. Looking at the most common pitfalls in this area clarifies where the developmental emphasis should lie.

The first violation in this category has both internal and external consequences:

Violation 5: Lack of clear vision & objectives

A precondition for the success of any team is that it knows where it's going. Unclear goals create a domino effect. Without clear goals both the project management process and team members' roles within that process are left ungrounded. Teams members won't understand why they're being asked to do what the project manager requests. The consequence is invariably a divergence among team members about what the project is, what the deliverables should look like, as well as how they will be implemented.

The vision must also make clear the project's relationship to the rest of the organization. Without an understanding of this broader context the project team feels like it's working in a box – the all too familiar "mushroom syndrome." This invariably undermines team member's motivation.

For the same reasons, this vision needs to be understood and embraced by the broader organization as well as the project team. The project team is, *de facto,* a part of a larger group of "stakeholders" – all those with a significant interest in the project's success. It's a primary responsibility of the project manager to ensure

that all the interested parties understand the project's impact and their responsibility in realizing its mission.

Finally, the Society Principle highlights the danger of ignoring or minimizing the social dynamics surrounding the project; expressed as our last violation:

Violation 6: Inadequate attention to project momentum, harmony and rhythm

For project managers with extensive technical training and experience, the "softer" social side of project management is often the least familiar. While this can make it tempting to ignore it does not diminish its importance. This is where the more general management and leadership skills find the most traction. There are three dimensions to consider: the *personal, team,* and *collaborative* dimensions.

The results of neglecting any of them are familiar, well documented, and, invariably, undesirable.

The *personal dimension* realizes that project participants are individuals and need to be recognized for their contributions. If you expect team members to excel they need the tools and conditions required to do an excellent job (e.g., appropriate software, quiet space conducive to focused software development, etc.) They need to be appropriately motivated. They need to see how their participation in the project makes a difference. And they need to understand why the project is important.

The *team dimension* recognizes that the project team is not simply the sum of the individuals that comprise it. The team is an organic entity with needs and a rhythm all its own. A project manager's failure to understand the basic stages of team development or the normal fluctuations in project rhythm can seriously hinder a project team's ability to sustain the required forward motion. Other major pitfalls in this category include uncontrolled problem employees and lack of the right blend of skills or experience on the team. Any project manager would benefit from at least a quick read of Peopleware, Tom DeMarco & Timothy Lister's classic discussion of the personal and team dimensions of project management.

Finally, the *collaborative dimension* of the Society Principle recognizes that the project does not end with the project team. Failure to understand, incorporate, continually attend to, or

manage the needs of the key stakeholders in the project will undermine its success. As Kathy Warden says, "leading the project team and holding the technical vision for the project are only portions of the job of project manager. Working with the business stakeholders to ensure the system meets business needs should be the primary objective." Seen from the perspective of the Society Principle, the primary function of a project manager is *not* to deliver a technical product but to manage and satisfy the commitments to his or her "constituency". As Peter G.W. Keen puts it, "Successful software development and systems integration require commitment management, not project management. Companies should restart their IT processes around technical and organizational commitments and the relationships between them."

Conclusion

Only a small percentage of technical projects succeed, meaning they are completed on time, within budget, and with all required features. But the causes of these failures are well understood. The most common and debilitating project management errors can be captured under three fundamental project management principles: context, entropy and society.

Table A summarizes these pitfalls against a simple project life-cycle model. The chart indicates both the project stage(s) in which a pitfall can most effectively be addressed and the relative importance of attending to it in the stage. A filled circle (●) indicates that significant attention to the pitfall in the stage is critical for project success. A dotted circle (⊙) indicates that appropriate attention to the item in this stage will substantially increase your chances of success. An empty circle (○) indicates "maintenance mode" – check in frequently to ensure that entropy hasn't taken over. Note that the close-out stage of the project demands attention to *all* the pitfalls. Evaluating how these pitfalls were handled in the course of the project is an invaluable part of a project *post mortem*. As with other areas, experience, be it good or painful, is frequently the best teacher for a technical project manager.

Table A: *Summary of Principles and Pitfalls*

Principles & Pitfalls	Initiate	Plan	Imple-ment	Close-Out
***Context Principle*: Technical projects take place in a broader context**				

1. Lack of adequate user input and involvement in the project	●	●	●	⊙
2. Lack of sustained executive management support	●	●	●	⊙
***Entropy Principle*: The amount of disorder in projects will not, of itself, decrease with time**				
3. Unclear requirements	⊙	●	⊙	⊙
4. Lack of proper planning	⊙	●	○	⊙
***Society Principle*: Technical projects are social undertakings**				
5. Lack of clear vision & objectives	●	⊙	○	⊙
6. Inadequate attention to project momentum, harmony and rhythm	⊙	●	●	⊙

.

About the Author

Glenn Kessler is a founding partner of Strategic Technology Partners. He can say with confidence that he has made all of the above mistakes (as well as many others) in his 20 years as an information technology, strategy and project management consultant. He has learned from his experience and can be contacted via e-mail at gkessler@stratpartners.com.

SIDEBAR – PLEASE PLACE THIS NEXT TO THIS ARTICLE IN A CALL OUT BOX IF POSSIBLE. MAYBE A 1-PAGE SIDEBAR TREATMENT WOULD WORK

Project Management CHAOS

The Standish Group "CHAOS" report published in 1995 presents one of the more systematic and comprehensive studies of project management success and failure. This report is based upon both an extensive survey and personal interviews. The survey sample reflects the experience of 365 IT executive managers and represents 8,380 applications. This sample covers a wide spectrum of organizations. Some key Standish Group findings are summarized in **Table B** below. The full report, along with its sequel "Unfinished Voyages", is available on the web at http://standishgroup.com/visitor/chaos.htm

Table B: *The Standish Group Survey: Key Findings*

Rank	Top 10 Reasons Why Projects Succeed	Top 10 Factors That Cause Project to be "Challenged"	Top 10 Factors That Cause Projects to be Cancelled
1	User involvement (15.9%)	Lack of user input (12.8%)	Incomplete requirements (13.1%)
2	Executive management support (13.9%)	Incomplete requirements & specifications (12.3%)	Lack of user involvement (12.4%)
3	Clear statement of requirements (13.0%)	Changing requirements & specifications (11.8%)	Lack of resources (10.6%)
4	Proper planning (9.6%)	Lack of executive support (7.5%)	Unrealistic expectations (9.9%)
5	Realistic expectations (8.2%)	Technology incompetence (7.0%)	Lack of executive support (9.3%)
6	Smaller project milestones (7.7%)	Lack of resources (6.4%)	Changing requirements & specifications (8.7%)
7	Competent staff (7.2%)	Unrealistic expectations (5.9%)	Lack of planning (8.1%)
8	Ownership (5.3%)	Unclear objectives (5.3%)	Didn't need it any longer (7.5%)
9	Clear vision & objectives (2.9%)	Unrealistic time frames (4.3%)	Lack of IT management (6.2%)
10	Hard working, focused staff (2.4%)	New technology (3.7%)	Technology illiteracy (4.3%)
	Other (13.9%)	Other (23.0%)	Other (9.9%)

Selected References

1. Hugh Beyer & Karen Holtzblatt, Contextual Design (Morgan Kaufmann Publishers, 1998)
2. Peter G.W. Keen, "Let's Put Project Management Out of Its Misery" (Computerworld, 12/7/98)
3. Software Engineering Institute, The Capability Maturity Model (Addison Wesley, 1997)
4. Steve McConnell, Software Project Survival Guide (Microsoft Press, 1998)
5. The Standish Group, "CHAOS" (The Standish Group International, 1995)
6. The Standish Report, "Unfinished Voyages" (The Standish Group International, 1996)
7. Tom DeMarco & Timothy Lister, Peopleware (Dorset House Publishing, 1979)