



A Software Development Solution¹

By David H. Gleason

Managing Consultant, IT Quality Solutions

dgleason@ITQual.com

www.ITQual.com

Copyright © 2001

Rev. 3/18/03

1. Introduction

For small to medium-sized software projects, basic project management (PM) principles and attention to subsumption ethics can significantly reduce the risk of project failure. Additionally, stakeholder impact analysis (SIA) within the framework of the IEEE/ACM Software Engineering Code of Ethics and Professional Practice further reduces risk and, indeed, results in better software. A prototype tool, the *SoDIS Project Auditor*, is available to apply both basic project management techniques and ethical impact analysis to software development projects.

This paper will use three case summaries to demonstrate the effectiveness of these methods. It will suggest how PM practices and SIA can help mitigate problems. It will describe the Software Development Impact Statement (SoDIS) Project Auditor, under development by Don Gotterbarn and Simon Rogerson. Building on these examples and concepts, it will suggest next steps for implementation of these techniques.

2. Software Project Success and Failure

Software projects are improved by two activities: 1) project management and 2) stakeholder impact analysis. The latter is not rigorously included in most standards and methodologies.

In order to succeed at a basic level, software must meet operational objectives and be timely and affordable. To excel, software must also have a net positive effect on stakeholders.

Basic project management, according to the Project Management Institute, means managing nine knowledge areas: project integration; scope; time; cost; quality; human resources; communications; risk; and procurement. Attention to outcomes is paramount: “Prevention over inspection — the cost of avoiding mistakes, is always much less than the cost of correcting them” [PMBOK, 1996, 84].

¹ This Paper was presented at the ETHICOMP 2001 conference in Gdansk, Poland in June, 2001.

The following three case summaries illustrate the value of basic PM and SIA. These are real cases, with identifying facts changed to preserve confidentiality.

2.1. Case summaries

Nirvana Travel

Narrative:

When they embarked on a project to rebuild their operating database, Nirvana Travel had grown from a startup to almost 200 users. The software platform they had originally built to manage operations would no longer accommodate the growth of the firm, so they contracted with a software developer to build a web-based application with a browser interface for both internal and external users. The new application was to be cutting edge technology, designed to move towards an e-business platform. An initial budget was set at \$850,000.

The developer assigned an inexperienced team to the project, consisting of programmers, analysts and a project manager. Nirvana was assured that the developer had done this before, that there was nothing to worry about, and that the project would proceed without problems. The developer was a leader in the industry, and they convinced Nirvana to proceed.

The developer helped Nirvana select a software platform that had a good reputation, but which was new to the developer.

The team began work on the project, and product specification became a problem immediately.

The first specification drafts were late and inaccurate — not representing properly the content of meetings. They were accompanied by a statement that they had to be approved within two business days or the project budget and schedule would slip.

Nirvana was unwilling to approve specifications that were inaccurate, leading to the first of many disagreements between the firms. Finally, an understanding about the specification was reached, and development began. Initial re-estimates climbed to \$1,250,000 during product specification.

The developer was unfamiliar with the software platform, so they were unable to take advantage of the tool to program efficiently. The project slipped to \$1,700,000 and 4.5 months late.

As the first iterations of the software appeared, they did not meet the business requirements of Nirvana. Needed features were missing, and many routine functions were not working properly.

During the project the developer's team shifted as original members were reassigned. However, little practical expertise was introduced in the new team members. As time passed, the developer refused to provide delivery of functioning software and related products until Nirvana made payments. Ultimately the delivered product was unstable, almost a year late, and did not function as intended by the firm. The final cost was \$4,300,000.

Analysis:

Throughout this effort there was insufficient project management to recognize emergent risk events. Because the technology was new, the team was inexperienced, and business requirements were not clear, there were no criteria by which to evaluate progress.

Nirvana would later learn that the developers had not used functionality that was built-in to the software platform, and had instead devised extensive workarounds. These workarounds made the system unstable, increased programming costs, and hurt performance.

Ironically, the developer was financially enriched by this approach. There was little incentive for the developer to keep the project under control. Under a time and materials development effort, the developer *makes more money* if the project goes over time and budget.

Desire for client satisfaction and industry reputation can balance a developer's incentive to allow cost increases. Another incentive to meet obligations can come from a sense of responsibility and integrity — the ethical principles embodied in the Software Engineering Code of Ethics and Professional Practice. In the case of Nirvana Travel, the developer did not follow these principles, leaving at least the appearance of willful ignorance and professional irresponsibility.

Nirvana might have cut off the contract earlier, or been more proactive in their own project management in order to avoid the situation. While the outcome could easily have resulted in a lawsuit, Nirvana felt that they were better off growing their company than spending their resources on a protracted legal battle. Furthermore, the developer's ability to pay out any settlement was doubtful.

Evaluation:

Nirvana's experience is typical of ad-hoc software development. Basic project management is the first step in avoiding such outcomes.

Evaluation, Inc.

Narrative:

Evaluation, Inc. (EI), a small knowledge-management company, needed to move its primary product from CD-Based distribution to Web-Based distribution. EI prepared a Request for Proposals (RFP) before embarking on the project to replace their platform. This RFP required extensive input from many of the people at the firm. An initial project budget was established and included in the RFP, along with business requirements and an outline of required functionality. The delivered product was highly functional, but at three times its original budget of \$150,000.

Analysis:

EI did not do stakeholder impact analysis on the user community because it was seen as too costly. As a result, the scope of the project *had* to change in order to meet belatedly perceived market needs.

The RFP process imposed a project management discipline on EI. The project progressed on-budget until the user interface for the client users needed to be finalized, at which point many suggestions started to arise for how it should be changed. Since project management controls were in place, those requests were adjudicated and all additional work was properly authorized.

Unlike the situation at Nirvana, the PM processes kept the project from going out of control. People knew what the additional costs were going to be, and they had an opportunity to approve changes in advance of work activity. They could then accept the additional cost as a result of their own decision-making.

Evaluation:

Because EI did no stakeholder analysis, they failed to correctly estimate the cost of user-interface development. Stakeholder impact analysis, in addition to basic project management, can help avoid such overruns.

Orion Call Center

Narrative:

Orion Call center, a firm of about 300 employees, needed to automate its information technology help desk, equipment acquisition and inventory control services. A software project was initiated to develop an application using an inexpensive, customizable customer relationship management tool. An in-house developer was assigned to build the product. The product was delivered under-budget and exceeded expectations. It has been in use for 6 months with no significant modifications.

Analysis:

This software was built by a developer who knew the environment, business requirements, technical requirements and software development platform. She had project management skills. A project plan was written that described both desired outcomes and project process. She used the SoDIS software to determine who, inside and outside the organization, would be affected by development and the product. Based on this preparation and an efficient tool set, the developer was able to build a 12-table database in two days.

2.2. Ethical Impact Analysis

Subsumption Ethics

The results of stakeholder impact analysis (SIA), or the lack thereof, are subsumed into software products. The difference between computers and simple machines is the extent to which computer systems subsume design and development decisions over which users have little or no control. Unlike a human operator, there is no point in the cycle where the machine pauses to ask, “Should I do this?” Stakeholder impacts need to have been adjudicated *before* software is put into operation.

Subsumption ethics is the process by which decisions become incorporated into the operation of information technology (IT) systems, and subsequently forgotten. IT systems, by nature, repeat operations over and over. If those operations are faulty, the system will continue to execute them anyway. SIA helps ensure that subsumed decisions represent the needs of stakeholders.

Subsumption Ethics has four axioms. They are:

- A. Information systems subsume design, policy and implementation decisions in programming code and content. Code segments and content become “subsumed objects.” This axiom posits that the decisions themselves, including many subtle factors, are incorporated into systems operation.
- B. Subsumed objects have determinate moral value. A subsumed object that causes a system to physically harm a person has a negative moral value.
- C. Subsumed objects have a high “invisibility factor.” Subsumed objects are invisible to most users.
- D. Subsumptive complexity increases over time. As systems are developed, components become subsumed more and more deeply. Once subsumed, components may be irretrievably buried.

Stakeholder Impact Analysis

Project stakeholders can be broadly defined to include those directly and indirectly affected by both the project and its products. Ideally, the impact of subsumed objects in software development should be assessed for each stakeholder. This method ensures that the subsumed objects in the product will not adversely affect stakeholders. The method requires PM techniques, broad stakeholder identification, application of ethical principles and attention to subsumption ethics.

2.3. Software Project Auditing

Don Gotterbarn and Simon Rogerson have enhanced their work on the Software Development Impact Statement (SoDIS) to include a software tool for project auditing called the *SoDIS Project Auditor*. This tool both addresses basic project management and also provides a framework for SIA based on the Software Engineering Code of Ethics.

SoDIS is the result of careful analysis of software development activities on identified stakeholders. It applies a set of ethical questions derived from the Software Engineering Code of Ethics to work breakdown structures (project task outlines) and stakeholders. By exploring the impact of each task in software development to relevant stakeholders, SoDIS seeks at minimum to avoid reworking software after its completion. At its best, the SoDIS process, can help produce exemplary software.

The SoDIS software under development has three phases: Feasibility analysis; Requirements Analysis and Detailed Analysis. Feasibility Analysis applies basic PM principles to software under consideration. Requirements Analysis identifies major project risks. Detailed analysis applies the Software Engineering Code of Ethics principles to each task in the WBS for relevant stakeholders.

The use of all three phases can help avoid the failures of both the first and second case summaries above. Ultimately, this approach can produce better software.

3. What's new here?

There exists an extensive body of knowledge on project management. Stakeholder impact analysis using the SoDIS and its application of the Software Engineering Code of Ethics is new.

In software development, step one is to get project management practices in place. Step two is to run feasibility analysis in relation to stakeholders. Step three is to do extensive impact analysis of individual project activities and outcomes on each relevant stakeholder.

Today, most software projects are still done at the ad-hoc level. Project management practices, the Capability Maturity Model (CMM) from the Software Engineering Institute at Carnegie Mellon, subsumption ethics and other techniques, bring project management to a level of reliability where expectations and specifications can be consistently met. The next step in improving project outcomes is to apply rigorous stakeholder impact analysis.

4. Next steps

4.1. Rollout the Software Project Auditor

As software Project Auditor becomes available, it must be tested and adopted in actual industry practice.

4.2. Develop international standards for SIA

Stakeholder impact analysis — the SoDIS concept — should be evaluated for revision and adoption by national and international standards organizations. This is a natural evolutionary step from the project management standards already in place.

4.3. Provide training programs on SIA and the SoDIS Software

Concurrently, training in SIA and on the SoDIS in particular should be provided. Marketing efforts are needed to identify potential users of the methods and standards and then to help them with adoption.

4.4. Work with existing project management associations to develop SoDIS methods and standards

Existing project management associations have already identified many of the techniques needed to improve software project quality. SoDIS is a natural and logical extension of the current body of knowledge.

5. Conclusion

As the three case summaries above demonstrate, software project management techniques can significantly improve project outcomes. In the first case, a lack of basic project management led to a development failure. In the second case, the lack of stakeholder impact analysis led to cost and schedule overruns. In the third case, the application of all these techniques led to a project that exceeded expectations.

References:

- Project Management Institute, Project Management Body of Knowledge (PMBOK), Newtown Square, PA, USA, 1996.
- Gleason, David, Subsumption Ethics, Computers and Society, 29,1, March, 1999.