

**Risk Management for IT Projects:
How to Deal with Over 150 Issues and Risks
By Bennet P. Lientz & Lee Larssen
(A book review by R. Max Wideman)**

Introduction

According to authors Bennet Lientz and Lee Larssen:

"The rate of failure of IT (information technology) projects has changed little in survey after survey over the past 15-20 years – approximately 40-50%. This has happened in spite of new technology, innovative methods and tools, and different management methods. Why does this happen? Why can't the situation be better? One reason is that many think of each IT effort as unique. Therefore, while you might be able to use basic methods and tools again, the situation is different. In reality many IT projects are very similar at a high strategic level. Where they differ is in the people, exact events – and the details."¹

I have news for the Authors. Project people in other project management domains make the same claim with the same reality.

The authors go on to observe:

"Risk is a fuzzy term – it can mean different things to different people. Here, work in IT has a high risk if significant issues remain unresolved or are solved in a way that is negative towards the work. Issues can be either negative problems or positive opportunities. Risk is often characterized as the product of exposure or loss and the likelihood of the problems occurring. By tracking issues, establishing an issues database, and taking other similar steps detailed in this book, you can reduce the likelihood."²

Therefore, the authors' objectives are to:

- Provide their readers with a proven, modern method for dealing with IT-related issues and risks
- Supply an approach identifying and tracking issues and risk
- Enable the reader to perform a wide range of analysis of issues, and
- Address specific commonly encountered issues in different areas that include or are related to IT³

As you will gather from the book's title, after describing their approach to Effective Issues Management and Coordination, the authors present over 150 examples of issues and risks. This they do in a very organized way, but warn in a concluding note to their preface:

"As you read this book, you may find some of the examples unrealistic. You might ask, 'Do such companies and organizations exist that are this screwed up?' They do. We and you have knowledge of them or even worked with them. Is this bad? No. It is a fact of life."

Armed with that caveat, scan the Table of Contents for the title that most closely matches your current predicament, turn to the corresponding page and read up the authors' experiences and their recommendations for resolution.

Book Structure

The book is divided into four parts containing eighteen chapters altogether as follows;

Part I – Issues and Risk Management

1. Introduction
2. Effective Issues Management and Coordination
3. Analysis and Measurements of Issues and Risk

Part II – Internal Issues and Risk

4. Teams
5. The Work
6. Business Units
7. Management
8. Projects
9. Resistance to change

Part III – External Issues and risks

10. Vendors, Consultants, and Outsourcing
11. Headquarters
12. Technology

Part IV – Issues and Risks in Specific IT Activities

13. IT Strategic Planning
14. Analysis
15. Software Packages
16. Development
17. Implementation
18. Operations and Support

In addition, there are three Appendices: The Results of a Survey in IT Issues; The Magic Cross-Reference; and Websites.

The first three chapters in Part I are by way of introduction to the book generally and describe the authors' thoughts on various aspects of Issues Management. For example, they lament that:

"Lessons learned are gathered, if at all, at the end of a project, when most of the people have vanished to work on other projects and tasks. The experience and lessons learned that were collected were not organized, used, or updated."⁴

Contrary to this condition, the authors recommend that *Effective Issues Management and Coordination*⁵ require tracking Issues in three types of databases:⁶

- General Issues
- Project Issues, and
- Issues Tracking

In fact, the authors show that tracking the number of open Issues on an IT project is a valuable metric for determining the health of that project. That is because dealing with Issues openly and upfront will improve the probability of project success.⁷

From Part II onwards, the contents consist entirely of the authors' Issues and Risks descriptions organized as shown in the Table of Contents above. Each Issue in these chapters is set out under the following five headings: Title; Discussion; Impact; Detection; and Actions and Prevention. For example, Chapter 4, the first of the series, deals with Teams and, as with the remaining chapters, starts with an

Introduction. The issues that follow include such Issues as Lack of Teamwork; Team Members That Are Difficult to Manage; Project or Work Leader Who Lacks Experience; and Too Much Time Spent in Meetings.

These sorts of problems will be very familiar to many project people, and readers will no doubt study the authors' recommendations with great interest.

What we liked

Unlike the myriad of other project risk management books on the market that provide long shopping lists to choose from, this book provides a select number of common project risk issues culled from the authors' personal experience in Information Technology. More importantly, as we described under *Book Structure*, each Issue is discussed in some detail explaining how it may come about, its impacts and how it may be detected. The authors then describe what you have to do about each, or what you can do to avoid them in the first place.

But at the outset, the authors set out to show how *IT Differs from Other Types of Business Work*, and do so by presenting the following Table to prove it.⁸

Factor	Business	IT
Nature of Work	Task or Transaction	Group of tasks or project
Focus	Short term	Longer term
Concentration	Limited	Problem solving
Activity	Routine work	More variable work
Duration of work	Transaction, minutes	Longer term, days
Multi-tasking	Limited	May be extensive
Creativity	Work is routine	Often needed
Measurement	Easy	Difficult

Figure 1: Some Key Differences Between IT and Business Work

Some of the authors' comments on this Table include:

"A business unit employee typically shows up and starts work. During the day the person completes groups of transactions. Each transaction is a single task. The transactions often are not related. In IT the staff member can be interrupted by questions, crises, and issues.

In IT the work is more variable . . . For business staff the work tends to be routine, . . ."⁹

Well, we are not sure that "business staff" would necessarily quite agree with all of that, but you get the general point.

As an engineer, we always like things that are methodical, and this book is very methodical as we showed in describing the structure of the book's contents. With "Over 150 Issues" to choose from, most of which will be familiar to IT people, this book does not have to be read progressively from cover to cover. Instead, you can pick and choose today's particular area of interest, and dive right in. You can even open the book at random and start by reading the "Discussion" of the Issue on that page.

To prove this point, we've just opened the book at random (Yes, honestly!), and the *Issue* happens to be "Team Members or Departments That Do Not Get Along with One Another". The *Discussion* of this topic starts out with:

"Some people see themselves in competition with other IT staff. As such, they do not want to share knowledge. Another factor that contributes to this problem is that through training and culture, the individual is favored over the group. Also, some organizations have no significant turnover of staff. For example, at a university with tenure, the same faculty work (or do not work) together for decades. In these situations it is not surprising to find personality conflicts and even hatred."¹⁰

This section then goes on to describe typical examples under varying conditions.

We couldn't wait to get to the Actions and Preventions section, to see what the authors' recommended. These recommendations offer five different approaches – but you'll have to read the book to find out what they are. Interestingly, the one recommendation that is not included, and that we would have been inclined to add, is: "Fire the recalcitrant combatants or, at least, get them transferred to someone else's project!"

Small Projects and Support Work

An interesting question is the authors' position on the matter of how small projects and support work should be treated. In many IT shops, the same set of people is responsible for both projects of some size as well as support work. In this case, support work may be defined as short-run, project-like non-discretionary work necessary to keep normal operational work going. An example might be a discrete task of, say, less than 25 man-hours to fix an outage, or hardware or software failure.

From a management perspective, it would make more sense for the two types of work to be separated and staffed by different groups of people. At least that would permit the project work to proceed without frequent interruption, and increase the probability of completing the project on time and with less ambiguity as to the project's time (or cost) investment. However, two things militate against this option. If the shop is relatively small, the "support work", even with "small" projects included, may not be sufficient to make full time work for dedicated personnel, especially if several distinct skills are involved. Secondly, where only a few people have specialist knowledge, of a database or system for example, such people must be available to work on both support-type work as well as on new projects.

Indeed, the authors go further. In the *Discussion* of this Issue they observe:

"In IT, small projects that involve a short period of time and very limited resources are often not treated as projects at all. Instead, they are treated as normal work. If an IT group has a rigid, bureaucratic project management methodology in place, then managers and staff will potentially take efforts not to label the work a project. They know that if it is called a project, a huge burden will come down on top of the work [delaying] the work and the results. [However,] the impact of having small projects not being projects is that they are almost totally uncontrolled."¹¹

The authors emphasize this point by including a graphic displaying the following distribution of effort: Large projects 40%; Small projects 20%; and Support work 40%. The point of the graphic is that in this scenario, 60% of the work is not being effectively controlled. Or, to put it another way, 60% of the work is escaping under the radar.

"Huge burden" may be a bit of an exaggeration, but relative to the size of a "small" project, it is probably not. Perhaps more importantly, but not mentioned by the authors but nevertheless that we have encountered, is the opportunity for individuals to "escape". If switching from project work to "Other work" is indiscriminate, and if the project work becomes tough sledding, then it is possible for individuals to escape by engaging in more routine and comfortable work under the guise of necessary or urgent maintenance.

So, the authors are unequivocal on this point. They say that:

"Our experience indicates that the best long-term approach is to treat all work as projects. This will give you more control of the work. However, that said, it can only work if you have a flexible and – this is important – scalable project management methodology. If you have project templates, a standard issues database, and lessons learned, then this can be done. Any project has a list of open issues. You can scale up documentation, reviews, etc. based on the number, type, and importance of the project or work. If you cannot do this action, then at least impose some minimal structure on small projects [by having]:

- Standardized list of tasks
- Shared issues with larger projects
- Regular but streamlined project reporting
- Planning at the start on issues, purpose, and scope."¹²

In the final chapter the book, Chapter 18, the authors explain the Issue of *How Operations and Maintenance Should Be Managed*.¹³ At the end of the chapter, they conclude:

"A basic point of this chapter has been that support, operations, and maintenance have to be managed as well as projects. It is best if they are managed like small projects. This is easier than you think, since IT is already geared up for projects."

All good advice indeed.¹⁴

Downside

In Chapter 1, Authors Bennet and Lee make a great play as to *Why IT Efforts Fail*¹⁵, some of them spectacularly. The public record on this phenomenon is well known. However, most of these reports take the criteria of time and cost as the basis for the record of success or failure. In our view, this is a very narrow perspective. Surely the most important thing is whether the project delivered a product that actually added satisfactory value (i.e. realized benefits) to the organization in one way or another so that the project was worthwhile? In other words, unless you first define what constitutes project success and failure, the numbers are meaningless. After all, most IT shops operate on a fiscal budget basis, rather than a complete project budget basis. So, if the final cost is even known, an overrun is simply an entry in last year's financial statement.

The time parameter, on the other hand, is probably more significant. But from the descriptions of how work is handled in most IT shops, failure of IT projects to complete on time is probably more a matter of how the organization is set up and run, than it is of the failure of project management. (See discussion earlier under *Small Projects and Support Work*.)

The authors do not provide a Glossary of Terms and we had some difficulty grasping the authors' distinction between an *Issue* and a project *Risk*. In our view, at least in project management

terminology, Issues may be defined as: "Events that are currently affecting the ability of a project to produce the required deliverables."¹⁶ By comparison, Project Risks may be defined as "Those factors that may cause a failure to meet the project's objectives."¹⁷ In other words, an Issue is a Risk Event that has already occurred, i.e. whose time has come. So technically, the "Issues" described are really potential Project Risks and do not become Issues until the risk events actually occur.

As the authors point out, an issue can be either a problem or an opportunity. However, they choose to focus only on the negative issues since these often lead to IT failure.¹⁸ While that is true, we would have been pleased to hear of the authors' experience in which they managed to turn some typical issues or problems into opportunities. After all, the greatest successes are born of opportunities – not failures.

Summary

As the back cover of this book says:

"In Risk Management for IT Projects, IT Management experts Ben Lientz and Lee Larssen show you how to identify and track the recurring issues leading to failure in IT projects and provide a proven, modern method for addressing them. By following the recommendations in this book, readers will significantly reduce the risk of IT failures and increase the rate of success.

While offering a systematic approach to managing risk and issues, Lientz and Larssen demonstrate:

- How the issue arises and with what frequency
- What the impacts of the issue are if it does occur
- How to detect specific issues at an early stage
- How to prevent individual issues from occurring
- Detailed actions to take if an issue does arise"¹⁹

Between them Lientz and Larssen have over 75 years of experience dating back to the 1960s. They have worked with over 150 organizations in over 25 countries and have seen all of the issues they describe many times.²⁰ So, for IT projects people, this is a very handy and practical reference book.

The authors are to be congratulated on sharing their experiences.

Footnote

In an Email subsequent to preparation of this review, Bennet Lientz observed:

"I would like to point out that we (the authors) believe that the standard critical path is highly overrated, since it is not sensitive to the issues and their severity related to tasks. It only deals with duration and dependencies. That is why at the start of the book we spent considerable time on issues analysis. Issues are problems which give risk to the actual or potential impact (risk)."

And

"We define an issue in terms of being potential, active but not actionable, active, and resolved. When it becomes active, there is actual risk. When it is active but not actionable, it has potential or likely risk. If you resolve the issue, then the risk is mitigated. Risk can only arise due to a problem or issue(s) not being resolved. To keep the issues tied to the project, we link all issues to tasks (summary or detailed)."

R. Max Wideman
Fellow, PMI

-
- ¹ Lientz, Bennet, P., & Lee Larssen, *Risk Management for IT Projects*, Elsevier, Oxford, UK, 2006, Preface p xv
 - ² Ibid
 - ³ Ibid, pp xv-xvi
 - ⁴ Ibid, p3
 - ⁵ Ibid, Chapter 2, p17
 - ⁶ Ibid, pp91-20
 - ⁷ Ibid, pp 4, 17-18
 - ⁸ Ibid, p7
 - ⁹ Ibid.
 - ¹⁰ Ibid, p56
 - ¹¹ Ibid, p141
 - ¹² Ibid, p142
 - ¹³ Ibid, p308
 - ¹⁴ Ibid, p310
 - ¹⁵ Ibid, p5, & Appendix A: *The Results of a Survey on IT Issues*, p311
 - ¹⁶ Westland, J., *The Project Management Life Cycle*, Kogan Page, London, 2006
 - ¹⁷ Wideman, R. M., Project Management Glossary, version 4.2 [D01502]
 - ¹⁸ Lientz, Bennet, P., & Lee Larssen, *Risk Management for IT Projects*, p9
 - ¹⁹ Ibid, see description on the back cover.
 - ²⁰ Ibid, p xvi