

BY LINDA WALLACE AND MARK KEIL

SOFTWARE PROJECT RISKS AND THEIR EFFECT ON OUTCOMES

How to identify the risks that interact to pose the most significant threats to successful project outcomes.

While many software projects deliver the functionality and performance promised by their developers on time and within budget, some result in systems that fail to deliver as promised. The Standish Group, an IT consulting firm, reports \$275 billion is spent on software development projects each year in the U.S. alone [7]. More than 70% of these projects suffer total failure, cost overruns, schedule overruns, or deliver fewer functions than promised [6]. Examples of software project failures have been described in several books devoted to the subject [3, 4], and reports of troubled projects appear regularly in the business media. Failing to understand and manage the related risks can lead to project failure [1, 2], a costly problem that hasn't been completely addressed in the almost 30 years since such outcomes were first described in the literature. Software project managers would thus benefit from a better understanding of how software project risks affect project outcomes, leading to fewer project failures.

ILLUSTRATION BY ORESTE ZEVOLA

Risks are factors that can, when present, adversely affect a project, unless project managers take appropriate countermeasures. While a number of risk checklists have been developed, few firms have effectively incorporated them into their risk-management strategies. Managers need a simple approach for categorizing software project risks and providing insight into the relationship between different types of risk and project outcomes. At least one software-project-risk framework has classified individual risk factors according to their perceived importance and whether project managers view them as controllable [8]. Here, we explore the results and implications of its use in a multi-industry study of more than 500 software development projects managed by members of the Project Management Institute (see the sidebar “How the Study Was Done”).

Identifying the risks complicating software development projects and incorporating them into a coherent risk management strategy is clearly a challenge [9]. While the various risk checklists have been proposed [11], relatively little effort has gone into organizing the risks. Moreover, only a handful of studies have examined the effects risk factors might have on the outcomes of projects [5, 12]. As a result, software project managers have few formal procedures to guide themselves in identifying the relative effects of the various risk factors and the trade-offs needed to manage them.

The framework in [8] for identifying software project risks represents one of the earliest attempts to create a useful risk management tool for software development managers (see Figure 1). Based on an international Delphi study of software project managers published in 1998, the framework organizes software project risks into four categories based on perceived importance (in the project manager’s view) of the risk and perceived level of control project managers are likely to have in managing each one.

Customer mandate (quadrant 1, or Q1, in Figure 1) focuses on risk factors relating to customers and users, including lack of top management commitment and inadequate user involvement; though important to the success of a project, such factors are

often beyond the project manager’s control. Scope and requirements (Q2) focuses on risk factors associated with a project manager’s inability to judge a system’s scope. It also includes the risks associated with required functionality. Project managers should be able to control many of the risks associated with Q2. Execution (Q3) focuses on such risk factors as inadequate project staffing,

inappropriate development methodology, failure to define roles and responsibilities, and poor project planning and control. Because most project managers are confident they can control these risks, they regard them as producing moderate rather than strong effects [8]. Finally, environment (Q4) focuses on risk factors in both

internal and external environments, including changes in organizational management, that might affect a project.

Although the framework proposed in [8] is intuitively appealing, it remains untested, prompting project managers to ask: Do the risks embodied in each of the four quadrants in Figure 1 affect project outcomes? And do the risks in one quadrant interact with or offset the risks associated with any other quadrant? The framework’s practical value could be established more clearly by exploring such questions. In addressing the first, we hope to give managers a better understanding of the relationships among different types of risk and project outcomes. In addressing the second, we hope to provide insights into the interactions that may exist among different types of risk and how to manage affected projects.

How different risks affect process outcomes. We first analyzed how the different types of risk affect process outcome—whether a project is completed on schedule and within budget. For process outcome, we found only scope/requirements risk (Q2) and execution risk (Q3) were significant and no interactions among quadrants. These results make sense intuitively, as poorly executed projects or projects involving unstable scope or requirements generally exceeding their budgets and schedules. We also found that execution risk is twice as important as scope and requirements risks in explaining process outcome. Because execution risk embodies factors associated with project teams, project complexity, and project planning and control, management must focus on

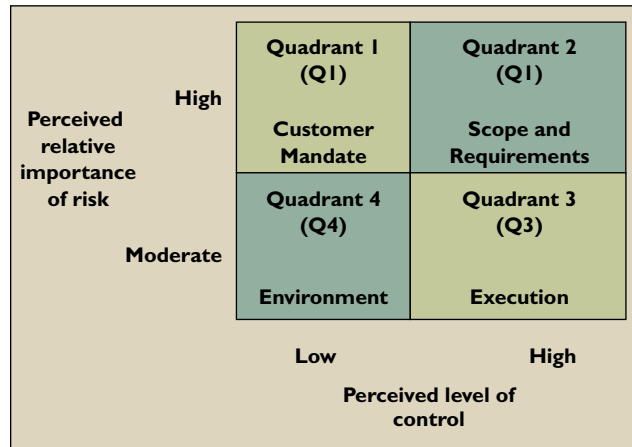


Figure 1. A risk categorization framework.

budget and schedule concerns. Risks associated with customer mandate (Q1) and environment (Q4) do not significantly affect process outcome. In other words, it may be possible to deliver software on schedule and within budget without much regard for risks associated with customer buy-in or the organizational environment. However, these risks may still affect project outcome.

How different risks affect product outcomes. Our statistical analysis showed that customer mandate (Q1), scope and requirements (Q2), and execution risks (Q3) have a significant relationship with product outcome. Environment risks (Q4) did not significantly affect product outcome. As pointed out in [8], although environment risks, including changes in organizational management during a project, are relatively rare, they are almost always unpredictable. Although the project managers in the study reported they experienced some environment risks, these risks might not have been significant enough that they couldn't be handled by well-defined requirements and superior project execution. Perhaps because environment risks occurred less frequently, they were not identified as significant in our sample.

We found that several interactions among these first three quadrants also influence product outcomes (see Figures 2 and 3), prompting several observations. The most notable is that when execution risk is low, high levels of customer mandate or scope/requirements risk have little effect on project outcome. On the other hand, when execution risk is high, the effect of customer mandate risk and/or scope/requirements risk on project outcome is significantly greater—almost two times if customer mandate risk is high and nine times if scope/requirements risk is high. Practically speaking, this means that project managers who know execution risk is high and are unable to lower it must develop a risk-mitigation strategy focusing on minimizing the risks associated with scope/requirements and customer mandate. On the other hand, if execution risk is low, the effect of a high level of customer mandate or scope/requirements risk is minimal. This relationship suggests that if execution risks are minimized, the effect from the other types of project risk will likewise be minimized.

These results seem reasonable given that execution risk deals with such issues as project-team experience, project complexity, and project planning and control. Managing them effectively may compensate for, prevent, or neutralize the risks associated with

scope/requirements, including scope creep, volatile requirements, and customer mandate. If execution risk factors are out of control (high risk), then the addition of high levels of risk in customer mandate and/or scope and requirements are sure to interact in ways that

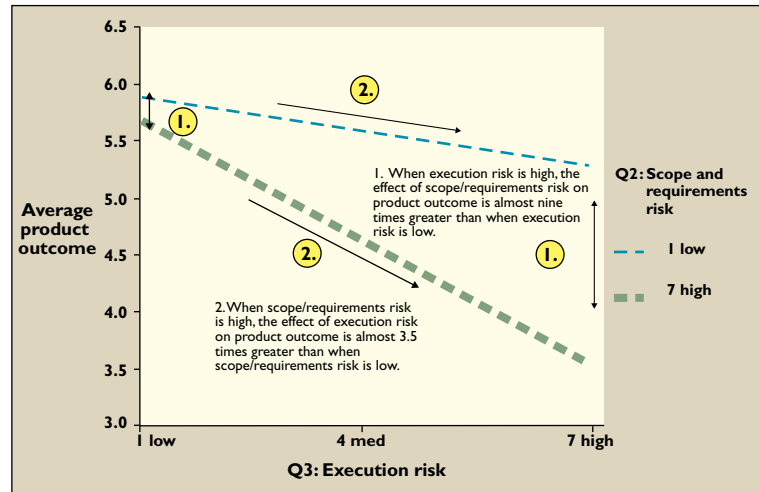


Figure 2. Interaction between quadrant 2 and quadrant 3 in Figure 1 and effect on product outcome.

increase the project's execution complexity and difficulty.

The interactions also show that a low level of scope/requirements risk helps compensate for high levels of execution risk. If scope/requirements risk is kept low, then execution risk will have only a minimal effect on product out-

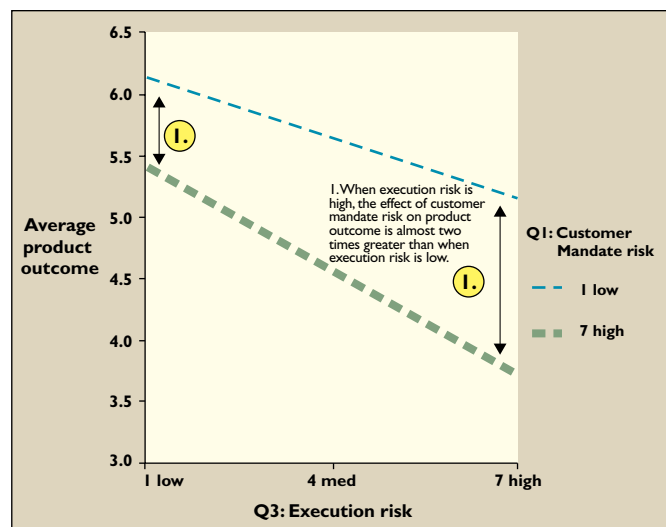


Figure 3. Interaction between quadrant 1 and quadrant 3 in Figure 1 and effect on product outcome.

come. On the other hand, if scope/requirements risk is high, execution risks must be managed to prevent a less than desirable product outcome—up to 3.5 times worse than when scope/requirements are under control, execu-

How the Study Was Done

In order to test the framework in Figure 1, we compiled a list of software project risk factors known to affect development efforts. We used previous research and interviews with practicing project managers in the U.S. to write and refine 53 statements reflecting the range of software project risks project managers are likely to encounter [12]. We mapped the 53 risk factors into the four quadrants in Figure 1 (see the table here). The two researchers engaged in the project performed this mapping independently; while they generally agreed on the placement of most risk factors in the quadrants, the occasional disagreement prompted them to discuss factor placement until they reached a consensus.

We included the 53 items in the Web-based survey, and 507 software project managers—all members of the Project Management Institute's Special Interest Group on Information Systems—indicated the extent to which each risk factor was present during their most recently completed projects. We used a seven-point Likert scale; higher numbers represent a higher presence of a risk factor, lower numbers a lower presence. We also asked about two types of project outcomes:

Product. Measuring the success of the application produced through the development effort [10]; and

Process. Focusing on the success of the development process, or whether the project was delivered on schedule and within budget.

For each project, we calculated a score for each quadrant using an average of the values (from 1 to 7, indicating the degree to which a risk factor was present) of the risk factors belonging to the quadrant. We then used multiple regression analysis to determine the relationships among the different types of risk and

| Quadrant | Factor Description |
|----------|--|
| 1 | Lack of user participation |
| 1 | Users resistant to change |
| 1 | Conflict between users |
| 1 | Users with negative attitudes toward the project |
| 1 | Users not committed to the project |
| 1 | Lack of cooperation from users |
| 1 | Lack of top management support for the project |
| 1 | Lack or loss of organizational commitment to the project |
| 2 | Undefined project success criteria |
| 2 | Conflicting system requirements |
| 2 | Continually changing system requirements |
| 2 | Continually changing project scope/objectives |
| 2 | System requirements not adequately identified |
| 2 | Unclear system requirements |
| 2 | Incorrect system requirements |
| 2 | Ill-defined project goals |
| 2 | Users lack understanding of system capabilities and limitations |
| 2 | Difficulty in defining the inputs and outputs of the system |
| 3 | Inadequately trained development team members |
| 3 | Lack of commitment to the project among development team members |
| 3 | Inexperienced team members |
| 3 | Frequent conflicts among development team members |
| 3 | Frequent turnover within the project team |
| 3 | Development team unfamiliar with selected development tools |
| 3 | Team members not familiar with the task(s) being automated |
| 3 | Negative attitudes by development team |
| 3 | Team members lack specialized skills required by the project |
| 3 | Project involves the use of new technology |
| 3 | High level of technical complexity |
| 3 | Highly complex task being automated |
| 3 | Project affects a large number of user departments or units |
| 3 | One of the largest projects attempted by the organization |
| 3 | Large number of links to other systems required |
| 3 | Immature technology |
| 3 | Project involves use of technology that has no been used in prior projects |
| 3 | Lack of an effective project management methodology |
| 3 | Inadequate estimation of project schedule |
| 3 | Lack of people skills in project leadership |
| 3 | Project progress not monitored closely enough |
| 3 | Inadequate estimation of required resources |
| 3 | Poor project planning |
| 3 | Project milestones not clearly defined |
| 3 | Inadequate estimation of project budget |
| 3 | Ineffective project manager |
| 3 | Inexperienced project manager |
| 3 | Ineffective communication |
| 4 | Resources shifted from the project due to changes in organizational priorities |
| 4 | Change in organizational management during the project |
| 4 | Corporate politics with negative effect on project |
| 4 | Unstable organizational environment |
| 4 | Organization undergoing restructuring during the project |
| 4 | Dependency on outside suppliers |
| 4 | Many external suppliers involved in the development project |

The 53 project risk factors mapped to the quadrants in Figure 1.

project outcome (measured in terms of product and process), as well as among the interactions among the different types of risks. **C**

tion problems will likely have less of an effect on the product. Requirements define what the product should be and how it should perform. Thus, a clear understanding of these requirements could be responsible for producing a good product outcome, even as problems with process outcome persist.

These inevitable effects on product outcome suggest that in order to produce a successful application,

software project managers must learn to control execution risks. If execution risks are managed effectively, then the effect of any level of customer mandate and scope/requirements risks will be minimal. If execution risks are not managed effectively, project managers must adjust their risk-mitigation strategies to minimize the risks associated with both scope/requirements and customer mandate.

Conclusion

We have explored how different types of risk influence both process and product outcomes in software development projects by analyzing input from more than 500 software project managers representing multiple industries. Our results reflect the importance of three of the four types of risk identified in the framework in [8]; the only one missing was environment (Q4). From the perspective of process outcome, managing the risks associated with Q2 and Q3 is critical. Thus, managers chiefly concerned with meeting schedule deadlines and budget limitations must find ways to reduce the risks associated with project execution, as well as the risks associated with scope and requirements.

The good news is that achieving successful process outcomes hinges on managing the risks associated with the two quadrants—Q2 and Q3—over which project managers feel they have the most control. However, every project manager knows that process isn't everything.

From the perspective of product outcomes, managing the risks associated with Q1, Q2, and Q3 is critical. Moreover, significant interactions take place among the different types of risk associated with these quadrants and influence product outcomes. As with process outcomes, the factor with the greatest influence on product outcomes is how the project is executed. Managing the risks associated with project execution requires project managers enlist experienced project team members who work well together and use proven project planning and control techniques. Scope and requirements, as well as customer mandate, also affect product outcomes.

The interaction effects we observed also suggest that good project execution can, to some extent, compensate for shortcomings in other areas, including customer mandate and the tactics employed for managing project scope and requirements. Similarly, if scope and requirements are identified, then execution problems are less important, though they could still affect the process and the likelihood of the project being completed on time and within budget.

Dealing with the triple constraints—scope, cost, and schedule—of project management almost always requires trade-offs. Our results suggest that projects emphasizing cost and schedule, or process goals, must be managed differently from projects emphasizing scope, or product goals. Ideally, both product and process outcomes should result in successful projects. Producing a project on schedule and within budget is of little use if the resulting product lacks the features and functions users thought they were paying for. However, in situations where budget

and schedule are the top priorities, Q2 and Q3 must be the most closely managed. In other situations, product may be the most important outcome, so problems with budget and schedule may be overlooked if the resulting system is highly functional. Thus, Q3 risk must be minimized; to a lesser extent Q1 and Q2 risks must also be minimized.

Perhaps the most important conclusion to be drawn from the study is that project execution matters more than any other type of risk in terms of shaping both process and product outcomes. We cannot overemphasize the importance of employing experienced team members who work well together, managing project complexity, and exercising good project planning and control methods. Though other types of risk are important, managing the risks associated with project execution must be management's main focus. ■

REFERENCES

1. Boehm, B. Software risk management: Principles and practices. *IEEE Software* 8, 1 (Jan. 1991), 32–41.
2. Charette, R. *Software Engineering Risk Analysis and Management*. Intertext Publications, New York, 1989.
3. Ewusi-Mensah, K. *Software Development Failures*. MIT Press, Cambridge, MA, 2003.
4. Glass, R. *Software Runaways*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1998.
5. Houston, D., Mackulak, G., and Collofello, J. Stochastic simulation of risk factor potential effects for software development risk management. *J. Syst. Soft.* 59, 3 (Dec. 2001), 247–257.
6. Johnson, J. *Chaos in the New Millennium: The Ghost of Christmas Future*. The Standish Group, West Yarmouth, MA, 2000.
7. Johnson, J. Turning chaos into success. *Software Mag.* 19, 3 (Dec. 1999), 30.
8. Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. A framework for identifying software project risks. *Commun. ACM* 41, 11 (Nov. 1998), 76–83.
9. Longstaff, T., Chittister, C., Pethia, R., and Haimes, Y. Are we forgetting the risks of information technology? *Comput.* 33, 12 (Dec. 2000), 43–51.
10. Nidumolu, S. The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable. *Inform. Syst. Res.* 6, 3 (Sept. 1995), 191–219.
11. Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. Identifying software project risks: An international Delphi study. *J. Mgmt. Inform. Syst.* 17, 4, (Spring 2001), 5–36.
12. Wallace, L. *The Development of an Instrument to Measure Software Project Risk*. Doctoral Dissertation, Georgia State University, 1999; see www.cob.vt.edu/wallace/dissertation.pdf.

LINDA WALLACE (wallacel@vt.edu) is an assistant professor in the Department of Accounting and Information Systems at Virginia Polytechnic Institute and State University in Blacksburg, VA.

MARK KEIL (mkeil@gsu.edu) is a professor in the Department of Computer Information Systems in the J. Mack Robinson College of Business at Georgia State University in Atlanta.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.