Requirements Engineering as a Success Factor in Software Projects

Hubert F. Hofmann, *General Motors*

Franz Lehner, *University of Regensburg*

Deficient requirements are the single biggest cause of software project failure. From studying several hundred organizations, Capers Jones discovered that RE is deficient in more than 75 percent of all enterprises.[1] In other words, getting requirements right might be the single most important and difficult part of a software project. Despite its importance, we know surprisingly little about the actual process of specifying software. "The RE Process" sidebar provides a basic description.

Based on their field study of 15 requirements engineering teams, the authors identify the RE practices that clearly contribute to project success, particularly in terms of team knowledge, resource allocation, and process.

Most RE research is conceptual and concentrates on methods or techniques, primarily supporting a single activity. Moreover, the rare field studies we actually have do not establish a link between RE practices and performance. We therefore conducted this study to identify the RE practices that clearly contribute to software project success.

Stakeholders and teams

Stakeholders are individuals and organizations that are actively involved in a software project or whose interests the project affects. Stakeholders of any computer system can include customers, users, project managers, analysts, developers, senior management, and quality assurance staff. Table 1 illustrates the wide range of expertise and motivations that stakeholders typically exhibit.[2]

A typical software project team consists of a project manager, analysts, developers, and quality assurance personnel. Often it includes users or their representatives. In the case of commercial off-the-shelf (COTS) software, marketers such as sales representatives and account managers tend to substitute for users and customers.

Field study

Seven field studies have reported on RE in practice.[3–9] Unfortunately, these rare studies have not established a clear link to performance and tend to focus on a narrow set of variables. Our study provides a more integrated view of RE by investigating team knowledge, allocated resources, and deployed RE processes (see Figure 1) and their contribution to project success. In addition, we incorporate the observations of previous field studies.

Fifteen RE teams, including six COTS and nine customized application develop-

0740-7459/07/$10.00 © 2007 IEEE

The RE Process

Requirements engineering denotes both the process of specifying requirements by studying stakeholder needs and the process of systematically analyzing and refining those specifications.[1] A *specification*, the primary result of RE, is a concise statement of the requirements that the software must satisfy—that is, of the conditions or capabilities that a user must achieve an objective or that a system possesses to satisfy a contract or standard.[2] Ideally, a specification enables stakeholders to quickly learn about the software and developers to understand exactly what the stakeholders want.

Despite heterogeneous terminology throughout the literature, RE must include four separate but related activities: elicitation, modeling, validation, and verification. In practice, they will most likely vary in timing and intensity for different projects.

Typically, we first elicit requirements from whatever sources are available (experts, repositories, or the current software) and then model them to specify a solution. Eliciting and modeling requirements are interrelated. Modeling describes a perceived solution in the context of an application domain using informal, semiformal, or formal notations. The gradual normalization of such models in terms of the requirements leads to a satisfactory candidate specification, which then must be validated and verified. This gives stakeholders feedback on the interpretation of their requirements so they can correct misunderstandings as early as possible.

Elicitation

Elicitation is often treated as a simple matter of interviewing users or analyzing documents, but several other elicitation methods are available. Some emphasize group sessions in the form of focus groups or workshops; others are employed primarily to elicit requirements for specific types of systems. For example, developers frequently use repertory grids, sorts, and laddering methods in specifying knowledge-based systems.

Elicitation also includes those activities that explore how software can meet organizational goals, what alternatives might exist, and how they affect various stakeholders.

Modeling

Experts have proposed many modeling methods and specification languages to make requirements precise and consistent. Traditionally these methods have separated the data, functional, and behavioral aspects of requirements and specified software by creating one or more distinct models. Prototypes, for instance, attempt to create an operational model that stakeholders can directly experience.

Paul Ward and Stephen Mellor, followed by many others, proposed extensions to basic models. Most of these extensions focus on modeling real-time systems. Developments in OO programming and design have introduced a more integrated approach to modeling requirements. In addition, advanced modeling methods attempt to establish a closer link between models and "the customer's voice," stakeholders' viewpoints, and business goals.

Validation and verification

The purpose of validating requirements is to certify that they meet the stakeholders' intentions: Am I specifying the right software? In other words, validation examines a work product (for example, a specification) to determine conformity with stakeholder needs. Verification, on the other hand, determines whether a work product conforms to the allocated requirements: Am I specifying the software correctly? That is, it checks a specification for internal consistency through mathematical proofs or inspection techniques.

An important point in validating and verifying requirements is prioritizing them. By addressing high-priority requirements before considering low-priority ones, you can significantly reduce project costs and duration. Moreover, throughout RE you should revisit the priorities assigned, for example, during elicitation to ensure that they continue to adequately reflect the stakeholders' needs. This highlights the recurrent nature of requirements validation and verification.

Methods for validating and verifying requirements are relatively scarce. Peer reviews, inspections, walk-throughs, and scenarios figure most prominently. Moreover, the recording of decisions and their rationales is quite useful.

References

1. H.F. Hofmann, *Requirements Engineering: A Situated Discovery Process*, Gabler, Wiesbaden, Germany, 2000.
2. *IEEE Guide to Software Requirements Specification*, IEEE Std. 830-1998, IEEE Press, Piscataway, N.J., 1998.

Table 1

What Stakeholders Want and What They Offer

Stakeholder	Motivation	Expertise areas
Customer	Introduce change with maximum benefit	Business and information system strategies, industry trends
User	Introduce change with minimum disruption	Business process, operating procedures
Project manager	Successfully complete the project with the given resources	Project management, software development and delivery process
Analyst	Specify requirements on time and within budget	RE methods and tools
Developer	Produce technically excellent system, use latest technologies	Latest technologies, design methods, programming environments and languages
Quality assurance	Ensure compliance to process and product standards	Software process, methods, and standards

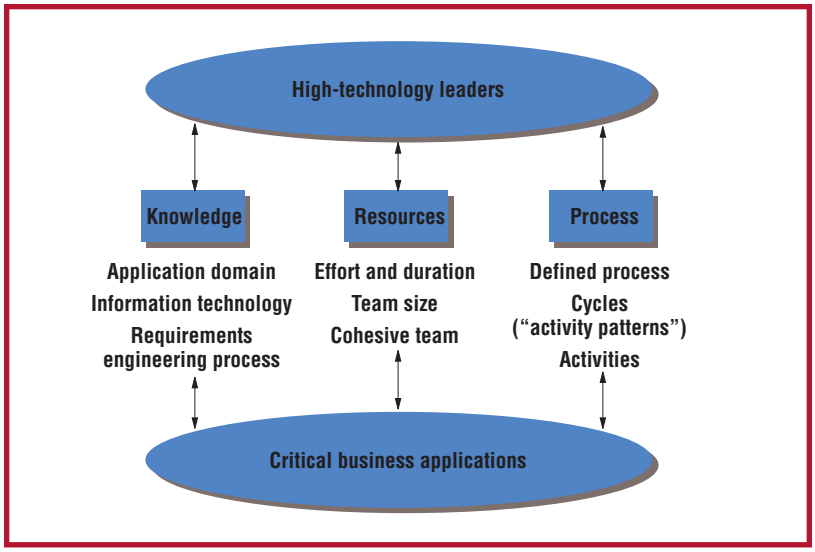


Figure 1. We studied three factors that contribute to project success: team knowledge, allocated resources, and exhibited RE processes.

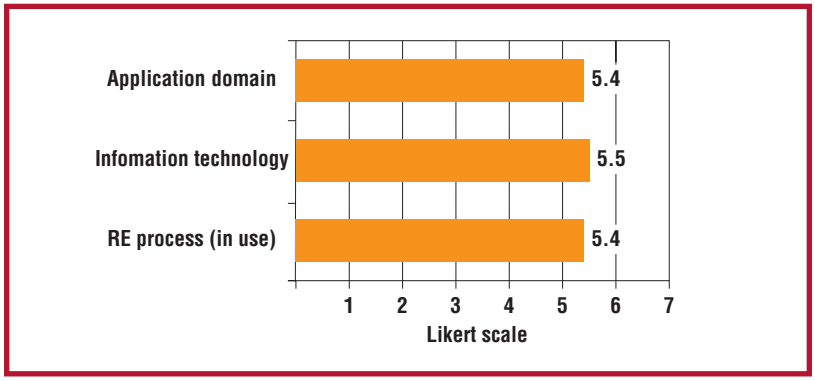


Figure 2. Research results regarding knowledge of application domain, information technology, and the RE process being used.

ment projects in nine software companies and development organizations in the telecommunications and banking industries, participated in our study. There were 76 stakeholders: 15 project managers, 34 team members, and 27 other stakeholders such as customers, management, and quality assurance personnel. The development projects we targeted were recently released critical business applications. On average, the participating projects finished in 16.5 months with an expended effort of approximately 120 person-months.

Through questionnaires and interviews, we collected data directly from each project's stakeholders to avoid an eventual bias and to obtain a more complete understanding of the RE process. We assessed the participants' confidence in their responses by using a 1–7 Likert scale in the questionnaires. That is, individuals rated their own perceived degree of accuracy of statements (such as "the project has a well-documented process for specifying requirements") by assigning a value from 1 (very inaccurate) to 7 (very accurate). The Likert scale let us calculate a total numerical value from the responses.

Our research approach

We decided to study knowledge because, as Bill Curtis, Herb Krasner, and Neil Iscoe have emphasized, deep application-specific knowledge is required to successfully build most large, complex systems.[4] We investigated team knowledge with regard to application domain, the technology needed to implement the proposed solution, and the RE process used. To gather the participants' perceptions of team knowledge, we employed a 7-point Likert scale for each focus area.

Allocated resources included team size, expended effort in person-months, and duration (in months) of RE. With regard to team size, the project managers provided data on full- and part-time members of the RE and project teams. We also gathered perceptions about the teams' coordination and interaction capabilities during RE. In the questionnaire, we used the construct of "cohesiveness," measured on the Likert scale, to address this aspect of RE. During follow-up interviews, we further investigated communication breakdowns, quality of interaction, and conflict resolution.

We gathered the stakeholders' perceptions of the defined RE process by considering the extent of standardization (for example, is it well-documented? is it tailored from an organizational standard?) and the configuration of work products and their changes, as well as by obtaining independent reviews of RE activities and deliverables. We measured these constructs on the Likert scale.

To enable the participants to characterize the practiced RE process, we gave them a list of typical elicitation, modeling, verification, and validation activities, some of which we identified by surveying the RE literature. We also applied the construct of *RE cycles* to distinguish activity patterns over time. An RE cycle is a set of activities that contain at least one each of elicitation, modeling, validation, and verification activities. Generally, a specific deliverable (for example, a prototype or data model) also characterizes a completed RE cycle.

We gathered stakeholders' perceptions of RE performance in terms of process control, the quality of RE service, and the quality of RE products. Process control addresses the team's capability to execute according to plan; thus, we gathered data to compare planned and actual cost, duration, and effort. We evaluated the quality of RE service in terms of the stakeholders' satisfaction with the RE process and the perceived "organizational fit" of the proposed solution. The stakeholders rated the quality of RE products using these quality attributes: correct, unambiguous, complete, consistent, prioritized, verifiable, modifiable, and traceable.[10] Requirements *coverage*, the functional and nonfunctional requirements addressed by a project's RE products, also influences quality. We assessed functional requirements from the perspective of functions, data, and behavior, and nonfunctional requirements by gathering perceived coverage of product, process, and external requirements.[11]

Findings

We focused on three factors that contribute to project success—knowledge, resources, and process—and analyzed their contribution to a project's success (performance).

Knowledge

Group research emphasizes the impact of experience and expertise on team effectiveness. For instance, in the study by Curtis, Krasner, and Iscoe, project managers and division vice presidents consistently commented on how differences in individual talents and skills affected project performance.[4] The authors identified the thin spread of application domain knowledge as one of the most salient problems in software projects.

In this study, stakeholders perceived the team's domain knowledge as relatively good. It reached 5.4 on a 7-point Likert scale, where 7 indicates an RE team with a high degree of knowledge (see Figure 2). Several teams repeatedly referred to their "good use of domain experts." (The quotes here and throughout this article indicate a direct quote from a participant in our study.) They also included "end users and customers from the very beginning" and "tried to get as much feedback as possible from team members when defining requirements." Some teams, however, worked with marketing personnel rather than the actual customer, assuming that marketing knew what was best for the customer. This produces, in some instances, "unrealistic" requirements due to the fact that the marketing staff's understanding often differs from the application-specific information about the software use that is needed for design.[4]

On half of the participating projects, senior management, project managers, and system analysts defined at least the initial requirements. Only a few software teams involved technically knowledgeable stakeholders such as software developers, quality assurance, test, and configuration management personnel early in the project. In the rare case that they were involved in RE, they were selected because they had greater application domain knowledge than their colleagues.

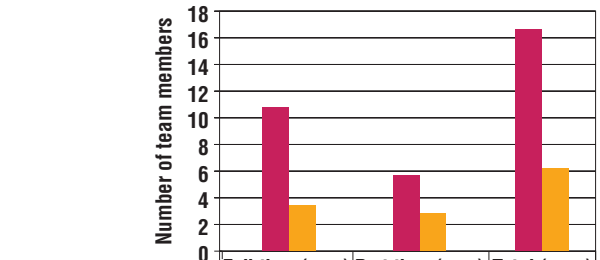
Involving stakeholders early also resulted in an increased understanding of the RE process being used. On the other hand, a lack of training and "weak project management" led to teams that were less familiar with the RE process. For instance, some organizations assign project managers based on their availability rather than their capabilities.[5] In those cases, both project managers and team members must learn the basics as they work. Participants in this study confirmed that this can result in severe budget overruns, frequent schedule adjustments, and canceled projects.[3]

Resources

Traditionally, RE receives a relatively small percentage of project resources throughout the software life cycle. For example, in 1981 Barry Boehm found that 6 percent of a project's cost and 9 to 12 percent of project duration are allocated to specifying requirements.[12] Over the last 20 years, the resources allocated to RE activities have increased. In this study, perhaps due to the projects' high-tech leaders, the resource allocation to RE was significantly higher. Project teams expended on average 15.7 percent of project effort on RE activities. The average amount of RE time equaled 38.6 percent of total project duration.

Patricia Guinan gives an average RE team size of 5.6 members in the 66 projects she studied.[7] For the 15 projects in our study, we calculated an average RE team headcount of 6.2 (see Figure 3). The varied

Some teams worked with marketing personnel rather than the actual customer, assuming that marketing knew what was best for the customer.



	Full-time (avg.)	Part-time (avg.)	Total (avg.)
Project team size (avg.)	10.9	5.7	16.6
RE team size (avg.)	3.4	2.8	6.2

Figure 3. Comparing the resources spent on RE by team size and part-time versus full-time allocation of people.

uses of full- and part-time resources during RE were interesting: Some projects allocated only dedicated resources ("product teams") to RE, whereas others relied on RE "experts" whom various projects "shared."

We also investigated participants' perceptions of team cohesiveness. The teams rated themselves as relatively cohesive—5.5 on a 7-point Likert scale. Overall, stakeholders gave higher scores to RE teams that frequently communicated with customers and other teams involved in the development project. In other words, such teams executed more "boundary management activities."[7] While some teams leveraged "a lot of confidence and trust between customers and developers," others struggled to achieve "buy-in from all stakeholders" or to secure "the necessary participation of other teams to stay on track and complete the specification."

Several RE teams used specification templates to facilitate communication among stakeholders. They also included "comprehensive examples" to improve specification readability. The most common tool used during RE was an internal Web site, accessible to all stakeholders, where the project team posted and maintained the requirements. Some RE teams experimented with commercially available RE tools. In all but one case, these tools interfered with rather than supported RE activities. We believe that either a lack of well-defined RE processes or the RE team members' lack of training in the selected tools caused this undesired effect.

Process

Only some projects defined their RE process explicitly or tailored an organizational ("standard") RE process (see Figure 4) to their needs. A tailored RE process uses or adapts a collection of RE process "assets" such as methods, templates, and tools to better fit a specific project's characteristics. Although several RE teams executed a documented RE process, with quality assurance providing insight into the compliance of a team's actions to their plan, most stakeholders perceived RE as an ad hoc process.

Most projects specify software in a dynamic environment and therefore struggle with the classic problem of rapidly fluctuating requirements.[4] This requires "flexible requirements" that "can be clarified and changed as the product progresses." In other words, the RE process has to account for stakeholders' learning curves and for requirements negotiation throughout the development project. Early on, some projects lamented the lack of "a detailed enough system architecture to adequately specify requirements," while others "froze the specification early" only to face customers that "kept changing requirements late in the cycle." The average ratings of configured RE work products (for example, prototypes and object-oriented models) and configured requirement changes reflect the stakeholders' struggle to balance the "drivers for change and the desire for stability." The most successful projects, however, recognized elusive requirements early in the process (with such statements as "the specification is a living document"). They managed requirements change explicitly rather than "freezing the whole specification."

Most of the RE teams performed multiple RE cycles. This is consistent with Pradromas Chatzoglou's research finding that more than half of the 107 projects he studied exhibited three or more RE cycles.[3] In our study, RE teams focused significantly more on eliciting and modeling requirements than on validating and verifying them (6.4 and 6.2 percent compared to 3.1 percent of project effort).

All teams performed some level of document analysis. Some, for instance, analyzed business plans and market studies while others derived requirements from contracts and requests for proposals. Most of the projects also used unstructured interviews, brainstorming, and focus groups. Only two projects held workshops to elicit requirements.

Most RE teams did not use "textbook" modeling methods,[8] but they did adopt some of those methods' principles, such as

abstraction and partitioning. The majority of projects developed prototypes ranging from simple mock-ups to operational prototypes. A third of the RE teams developed OO models.

Most projects verified and validated requirements with multiple stakeholders. More than half the projects performed peer reviews or walk-throughs to verify or validate requirements. Repeatedly, participants emphasized the importance of including customers and users in peer reviews. Moreover, several teams created scenarios to validate requirements. Only five of the 15 RE teams explicitly tracked requirements throughout their projects' life cycles.

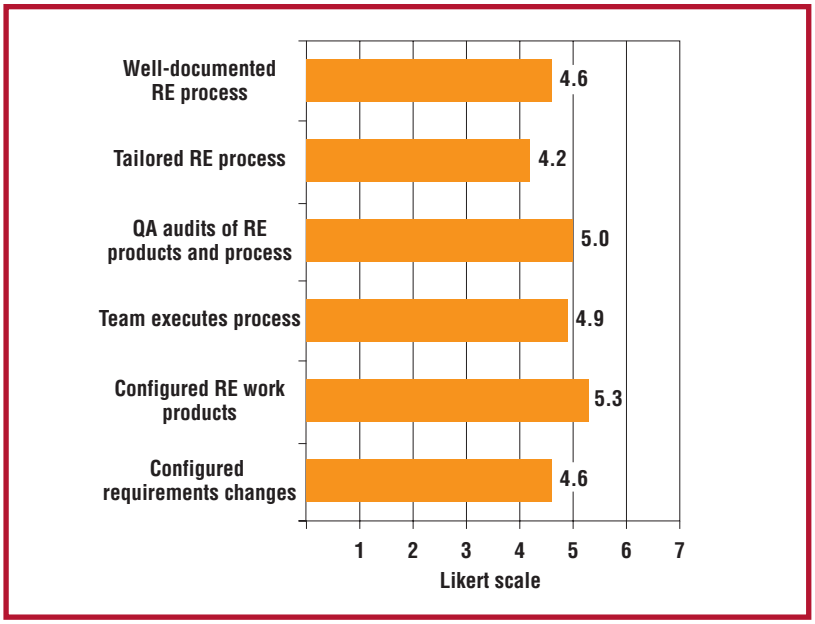


Figure 4. Evaluation of the participating teams' RE processes, based on a 7-point Likert scale.

Performance

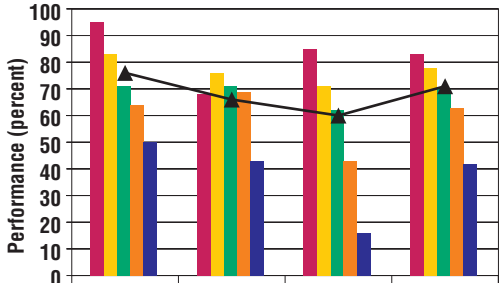
We considered three dimensions of performance: quality of RE service, quality of RE products, and process control. Using existing preference measures,[5] we calculated the product of weighted performance dimensions to obtain a total performance indicator (TPI). That is, quality of RE service is most important (with a weight of 1.438), followed by the quality of RE products (1.126) and process control (0.438).

Stakeholders rated the quality of service at 76 percent on average. Several stakeholders mentioned their early and frequent involvement in RE activities and the "good interaction between all groups" as important aspects that influenced their rating. They were more satisfied with the fit of the recommended solution than with RE, however. Frequently, this was due to difficulties in project planning (for example, "planning for high-priority items slipped" or "maintaining requirements in later development phases not planned").

The quality of RE products considers both requirements coverage and specification quality. Stakeholders gave it an average rating of 66 percent; the customers' perception of requirements coverage was relatively low, particularly with regard to nonfunctional requirements. Management seemed satisfied with data requirements, whereas the team and project managers focused on functions. Stakeholders emphasized, however, that concentrating on functions and data resulted in a "lack of total system requirements attention" and in "incomplete performance, capacity, and external interface requirements."

The stakeholders also scored specification quality. Overall, they were most satisfied with the specification's consistency and the ability to modify it as necessary, but they emphasized that the lack of traceability hurt their project. Prioritization of requirements, however, caused the most difficulty for RE teams. The average stakeholder rating for this quality attribute was significantly lower than any other attribute. RE teams struggled with adequately involving customers to identify high-priority requirements, management's inattention to resolving "unknowns," and nonfunctional requirements that were not managed "in the same process or as tightly." Several teams also mentioned the inability to consistently execute RE according to stakeholders' priorities rather than their own interpretation of "what's important," and some reported difficulties in keeping everybody on the project informed of changing priorities.

Process control, the third dimension of RE performance, was rated at 59 percent on average. The less variance between actual and planned cost, duration, and effort, the better the process control of a particular project. On average, project teams contained cost overruns within less than 2 percent and effort overruns within less than 6 percent. In addition, their cost and effort performance was predictable, with a standard deviation of less than 25 percent. Project duration, however, showed an average overrun of 20 percent, with a standard deviation of 47 percent. Managers frequently focused on cost to



	Quality of RE service	Quality of RE products	Process control	TPI
Projects (TPI > 75%)	95%	68%	85%	83%
PT/model-based process	83%	76%	71%	78%
PT-based process	71%	71%	62%	70%
Model-based process	64%	69%	43%	63%
Projects (TPI < 50%)	50%	43%	16%	42%
▲ Total (avg.)	76%	66%	59%	70%

Figure 5. Comparing the deployed processes and their total performance indicator (TPI), the quality of RE service, the quality of RE products, and process control.

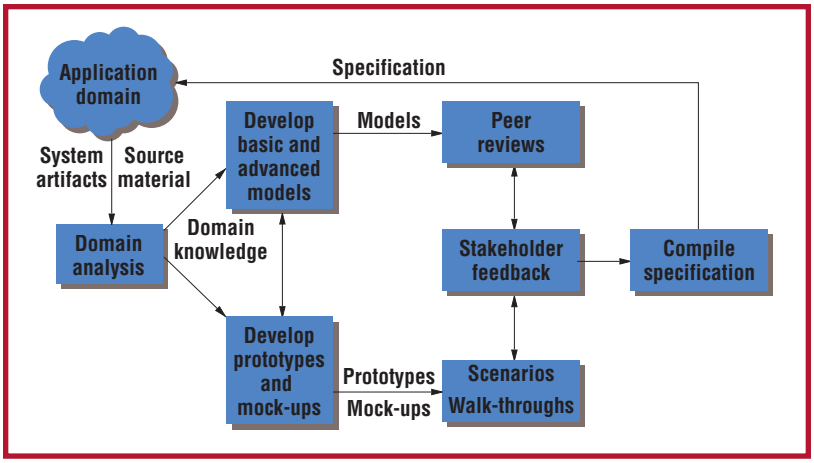


Figure 6. A successful RE process.

control the RE process, leading to undesired changes of expended effort and predicted duration. Moreover, some project managers perceived "pressure" to underestimate effort and duration to meet cost targets determined outside their control.

The "top" performers in this study (TPI > 75 percent) dynamically balance knowledge, resources, and process. For example, as Figure 5 shows, RE teams that used either prototypes or models exclusively faired better on average than projects with a TPI below 50 percent. A combined prototyping and model-based (PT/M) process resulted in higher ratings in all performance dimensions. The most successful projects also expended twice the effort to specify requirements. They performed RE activities throughout the project's duration, while lower-rated RE teams (TPI < 50 percent) were primarily involved during the "front end" of the project. For both types of projects, about four full-time team members participated in the RE team. On the most successful projects, however, several part-time members also supported the team.

For projects with a higher TPI, stakeholders reported that RE teams were more knowledgeable about the application domain, IT, and the RE process. Whereas Mitchell Lubars suggests that a "better atmosphere" contributes to project success,[8] Guinan concludes that an environment where team members share positive and friendly feelings is unrelated to team performance.[7] In our study, we found low cohesiveness only on projects with a TPI of less than 50 percent. This suggests that cohesiveness is a "trouble" indicator rather than a performance predictor. In other words, an increase in cohesiveness reduces the risk of failure but does not guarantee success.

Successful teams performed on average three iterations of the RE process (see Figure 6). They identified major stakeholders and domain boundaries, examined system artifacts and source material from current and previous systems, and frequently obtained stakeholder feedback and expert guidance on how to proceed. This resulted in much more explicit "win conditions" for stakeholders.

The successful RE teams used advanced modeling methods such as OO models, knowledge models, and quality function deployment matrices that translated "the voice of the customer" into quantitative technical requirements. However, they did not abandon basic models (for example, the entity-relationship model or state transition diagrams). They simply tried "to create a more complete model of the system." Moreover, the teams developed (basic and advanced) models and prototypes together to clarify "known" requirements and to guide the discovery of new ones. A combination of models and prototypes helped the stakeholders, especially customers and users, envision the proposed solution.

During peer reviews, the RE team, technical and domain experts, and customers and users examine the models. The resulting feedback enables the RE team to create acceptable models that accurately specify the

Table 2

Best Practices

Focus area	Best practice	Cost of introduction	Cost of application	Key benefit
Knowledge	Involve customers and users throughout RE	Low	Moderate	Better understanding of "real needs"
Knowledge	Identify and consult all likely sources of requirements	Low to moderate	Moderate	Improved requirements coverage
Knowledge	Assign skilled project managers and team members to RE activities	Moderate to high	Moderate	More predictable performance
Resources	Allocate 15 to 30 percent of total project effort to RE activities	Low	Moderate to high	Maintain high-quality specification throughout the project
Resources	Provide specification templates and examples	Low to moderate	Low	Improved quality of specification
Resources	Maintain good relationships among stakeholders	Low	Low	Better satisfy customer needs
Process	Prioritize requirements	Low	Low to moderate	Focus attention on the most important customer needs
Process	Develop complementary models together with prototypes	Low to moderate	Moderate	Eliminate specification ambiguities and inconsistencies
Process	Maintain a traceability matrix	Moderate	Moderate	Explicit link between requirements and work products
Process	Use peer reviews, scenarios, and walk-throughs to validate and verify requirements	Low	Moderate	More accurate specification and higher customer satisfaction

requirements. Scenarios and walk-throughs further guide the discovery of requirements. The breakdowns experienced while using prototypes and mock-ups lead to an evolutionary improvement of the specification.

Best practices

Successful RE teams have in-depth knowledge of the application domain, IT, and the RE process. In other words, successful projects have the "right combination" of knowledge, resources, and process. Table 2 summarizes the best practices exhibited by the most successful RE teams.

Stakeholder feedback plays a decisive role from the beginning to the end of successful RE projects. The most successful teams always involve customers and users in the RE process and maintain a good relationship with stakeholders. They have an ongoing collaboration with stakeholders to make sure that requirements are interpreted properly, to deal with fluctuating requirements, and to avoid communication breakdowns. Research supports this best practice: according to one study, user participation is one of the most important factors contributing to RE success.[5]

Successful RE teams identify the boundaries of the application domain and of the major stakeholders. To validate their understanding of the application domain, they identify and consult all likely requirements sources. They examine, for example, system artifacts and source material from current and previous systems.

As other research points out,[4] some individuals perform "10 times better than others." Thus, managers of successful RE teams should

- carefully select team members skilled in the application domain, IT, and the RE process;
- always assign experienced, capable project managers to RE; and
- consult domain experts and stakeholders early on to augment and validate the team's knowledge base.

Successful projects allocate a significantly higher amount of resources to RE (28 percent) than the average project in this or previous field studies, and they expend these resources according to a well-defined process. Successful teams also maintain a balance between RE activities. That is, they allocate on average 11 percent of project effort to elicitation, 10 percent to modeling, and 7 percent to validation and verification. To streamline RE activities, successful teams frequently leverage specification templates and examples from previous projects.

Requirements prioritized by stakeholders drive successful RE teams. This allows the

RE team to decide which requirements to investigate when and to what degree of detail. To specify prioritized requirements, the RE team develops various models together with prototypes. Moreover, they maintain a requirements traceability matrix to track a requirement from its origin through its specification to its implementation. This lets the team show how its work products contribute to satisfying the requirements. In addition, successful teams repeatedly validate and verify requirements with multiple stakeholders. They use peer reviews, scenarios, and walk-throughs to improve the specification throughout the software's life cycle.

Teams often struggle with fluctuating requirements, communication breakdowns, and difficulties in prioritizing requirements. RE goes through recurrent cycles of exploring the perceived problem, proposing improved specifications, and validating and verifying those specifications. It is a learning, communication, and negotiation process; to succeed, you must integrate your technical, cognitive, social, and organizational processes to suit your project's particular needs and characteristics. In other words, you must progressively discover your project requirements to specify software successfully.

Acknowledgments

We thank the participating companies and the outstanding executives managing their software development and procurement processes for making this study possible. We also thank Steve McConnell and the reviewers, especially Karl E. Wiegers, for their helpful comments and suggestions. Further thanks go to Theresa Hofmann and John Overmars.

References

1. C. Jones, *Applied Software Measurement: Assuring Productivity and Quality*, McGraw-Hill, New York, 1996.
2. L.A. Macaulay, *Requirements Engineering*, Springer, London, 1996.
3. P.D. Chatzoglou, "Factors Affecting Completion of the Requirements Capture Stage of Projects with Different Characteristics," *Information and Software Technology*, vol. 39, no. 9, Sept. 1997, pp. 627–640.
4. B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Comm. ACM*, vol. 31, no. 11, Nov. 1988, pp. 1268–1287.
5. K.E. Emam and N.H. Madhavji, "A Field Study of Requirements Engineering Practices in Information Systems Development," *Second Int'l Symp. Requirements Eng.*, IEEE CS Press, Los Alamitos, Calif., 1995, pp. 68–80.
6. N.F. Doherty and M. King, "The Consideration of Organizational Issues during the System Development Process: An Empirical Analysis," *Behavior & Information Technology*, vol. 17, no. 1, Jan. 1998, pp. 41–51.
7. P.J. Guinan, J.G. Cooprider, and S. Faraj, "Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach," *Information Systems Research*, vol. 9, no. 2, 1998, pp. 101–125.
8. M. Lubars, C. Potts, and C. Richter, "A Review of the State of the Practice in Requirements Modeling," *First Int'l Symp. Requirements Eng.*, IEEE CS Press, Los Alamitos, Calif., 1993, pp. 2–14.
9. S.R. Nidumolu, "A Comparison of the Structural Contingency and Risk-Based Perspectives on Coordination in Software-Development Projects," *J. Management Information Systems*, vol. 13, no. 2, 1995, pp. 77–113.
10. *IEEE Guide to Software Requirements Specification*, IEEE Std. 830-1998, IEEE Press, Piscataway, N.J., 1998.
11. H.F. Hofmann, *Requirements Engineering: A Situated Discovery Process*, Gabler, Wiesbaden, Germany, 2000.
12. B.W. Boehm, *Software Engineering Economics*, Prentice Hall, Englewood Cliffs, N.J., 1981.

About the Authors

Hubert F. Hofmann is manager of information systems and services for General Motors. His professional interests include strategic planning, enterprise-wide program management, software development, and system delivery processes. He received a PhD in business informatics from the University of Regensburg and an MBA from the University of Linz and the University of Zurich. He is a member of the IEEE Computer Society. Contact him at 7000 Chicago Rd., Warren, MI 48090; hubert.hofmann@gm.com.

Franz Lehner is a professor of management and information systems at the University of Regensburg, Germany, and head of the Department of Business Informatics. His fields of interest are software engineering and reengineering, knowledge management, and distance education. Contact him at the University of Regensburg, Business Informatics, D-93040 Regensburg, Germany; franz.lehner@wiwi.uni-regensburg.de.