Improving business performance

May 2002 Volume 4, Number 1

ESI Horizons

The Complete Source for Project Management Solutions

Software Requirements Engineering Ensures Success

By George Sifri, PMP and Karl Wiegers, Ph.D.

This is the second of a three-part series discussing software requirements engineering. Part 1 appeared in the April issue and Part 3 will appear in the June issue.

The majority of the material in Part 1 and Part 2 of this article is a direct synopsis of the first two parts of a reference that tackles requirements engineering comprehensively, Karl E. Wiegers's book "Software Requirements," published by Microsoft Press in 1999.

Business requirements specify the vision and scope for the software system and are the most important aspect of the requirements chain. The user requirements and the software functional requirements must align with the context and objectives established by the business requirements. Requirements that do not support achievement of the project's business objectives should not be included.

A project that lacks a clearly established and well-communicated direction is an invitation for disaster. The requirements will never stabilize if the project stakeholders do not share a common understanding of the business needs the product must satisfy and the benefits it will provide.

A symptom of ill-defined business requirements is that certain features are initially included, then deleted, and then added back later. Well-defined business requirements will ensure that:

- Vision and scope issues are resolved before detailed functional requirements are fully specified
- Scope and limitations are documented and communicated
- A well-defined project vision and scope document provide a baseline for making decisions about proposed requirements changes

Continued on Page 2

Managing in the Zone: The Art of Managing Software Developers During Projects

By Donna Fitzgerald

Over the last 20 years I have had the unique opportunity to work with a number of high-tech, extremely successful Silicon Valley firms and their resultant spin-offs. Most of the time, during those years, money was plentiful and engineering ruled the companies.

Dealing with developers in this environment has been both a challenging and uniquely rewarding experience. As a result of these experiences I've come up with certain rules of thumb that have helped me to successfully manage teams of software developers working on leading-edge development projects.

Technology is God

In well-funded, high-tech environments, developers are interested in solving problems using the most interesting, cutting-edge solution they can find. The tougher the problem, the more interesting most of them will find the project. The needs of the end user of the product rarely enter into their equation.

The average developer, unlike an IT programmer, has virtually no firsthand knowledge of the business problem an end product is designed to solve.

Most technical project managers are comfortable with this mind set, since most had it

Continued on Page 6

Contents

ESI International
4301 Fairfax Drive, Suite 800
Arlington, VA 22203 U.S.A.
(703) 558-3000 • (703) 558-3001fax
horizons@esi-intl.com
www.esi-intl.com

President Larry Seeley, PMP
Executive Vice President J. LeRoy Ward, PMP
Editor Claudia Payne

Continued from Page 1 — Requirements Engineering

Every software development organization should also adopt a standard Software Requirements Specifications (SRS) template for its projects. Figure 1 shows an SRS template that was adopted and extended from the IEEE 830 standard for software requirements specifications. This documentation should be accomplished as the team works through the requirements development process that starts with the vision and scope document.

Vision and Scope Document

The vision and scope document collects the business requirements into a single, concise document that sets the stage for all subsequent development.

This document should include at minimum the following sections.

Business Requirements

The business requirements identify the primary benefits that the new system will provide to the customers and to the organization that is developing the product. This section should answer the following questions:

- Why are we undertaking the project?
- What are the benefits that the project will provide to its builders and buyers?

This major section is divided into the following sub-sections:

Background — summarizes the rationale for the new product. It provides a general description of the history or situation that led to the decision to build this product.

Business opportunity — describes the market opportunity that exists or the business problem that is being solved.

Business objectives — summarizes the important business objectives the product will satisfy.

Customer or market requirements — describes the needs of typical customers, including needs that are not being met by products already in the marketplace or by existing information systems.

Value — defines the value the customers will receive from this product and indicates how the product will lead to improved customer satisfaction. For example, improved productivity or reduced rework, cost savings, streamlined business processes, automation of previously manual tasks and compliance with pertinent standards and regulations.

Business risks — summarizes the major business risks associated with developing, or not developing, this product, such as marketplace completion, timing issues, user acceptance, implementation issues or possible negative impacts on the business.

Software Requirements Specifications

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience Suggestions
- 1.4 Product Scope
- 1.5 References

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 Assumptions and Dependencies

3. External Interface Requirements
- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

4. System Features
- 4.x System Feature X
- 4.x.1 Description and Priority
- 4.x.2 Stimulus/Response Sequences
- 4.x.3 Functional Requirements

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes
- 5.5 Business Rules
- 5.6 User Documentation

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To-Be-Determined List

Figure 1. Template for Software Requirements Specification

Vision of the Solution

This section describes the long-term vision for the system that will address the business objectives. It provides the context for making decisions throughout the course of the product development life cycle. This section should include:

Vision statement — summarizes the long-term purpose and intent of the new product. It should reflect a balanced view that will satisfy the needs of diverse customers.

Major features or user capabilities — defines, in a numbered list, features that distinguish it from previous or competing products.

Assumptions — includes those made when conceiving the project and writing the vision and scope document.

Dependencies — outlines specific technologies to be used, third-party vendors, development partners or other business relationships.

Scope and Limitations

The project scope defines the concept and range of the proposed solution. The project limitations identify certain capabilities that the product will not include. This section should include:

Scope of initial release — summarizes the major features that will be included in the initial release of the product. It describes the quality characteristics that will enable the product to provide the intended benefits to its various customer communities.

Scope of subsequent releases — indicates which major features will be deferred and the desired timing of subsequent releases.

Limitations and exclusions — define the boundary between what's in and what's out. This is a way to manage both scope creep and customer expectations.

Business Context

This section summarizes some of the project's business issues, including profiles of major customer categories and management priorities for the

Continued on Page 3

Continued from Page 2 — Requirements Engineering

project. It is divided into the following sub-sections:

Customer profiles — identifies some essential characteristics of different categories of customers for this product. Each profile will list the following information for each customer category:

- Major benefits to the customer
- Likely customer attitudes toward the product
- Key product features of interest
- Success drivers
- Any known customer constraints that must be accommodated

Project priorities — are clearly established so that the project participants can focus on a common set of objectives. One method is to consider the five dimensions of a software project: features, quality, schedule, cost and staff.

In any given project, each of these dimensions can fit in one of three categories:

- Driver: a top-priority objective
- Constraint: a limiting factor that the project manager must operate within
- Degree of freedom: a factor that the project manager can balance against the other dimensions to achieve the drivers within the known constraints

Product Success Factors

This section documents how success will be defined and measured for this product. It describes the factors that are likely to have the greatest impact on achieving that success.

Some examples of these factors are market share, sales volume or revenues, customer satisfaction measures, transaction-processing volume and accuracy.

Customer Involvement

The following steps are essential in securing customer involvement, a critical factor in delivering excellent software.

Identify Sources of Requirements

Software requirements can come from many places, depending on the nature of the product and the development environment. The following are several typical sources for software requirements:

- Interviews and discussions with potential users
- Documents that describe current or competing products
- System requirements specifications
- Problem reports and enhancement requests for a current system
- Marketing surveys and user questionnaires
- Observations of users at work
- Scenario analysis of user tasks

Identify Different User Classes

Each user class will have its own set of functional and nonfunctional requirements. Some of the people affected by the application might not be direct users of the product. Instead, they are accessing the product's data or services through reports or other applications.

These indirect users can have their own requirements and thus constitute additional user classes. In addition, user classes can be applications or hardware components with which the product interfaces. User classes are based on:

- Use of the product
- Application domain and computer system experience
- Features used
- Business processes performed
- Geographic location
- Access privilege levels

Assign User Representatives

User representatives should be involved throughout the development life cycle. A diverse user participant group is required to represent different user classes and levels of expertise. To identify user representatives, managers can:

- Engage people from beta-testing or early-release sites
- Set up focus groups that comprise current users of your products or your competitors' products
- Include a mix of user types, with knowledgeable customers as well as those who are less experienced

The product champion approach provides an effective way to structure and formalize the customer-developer partnership. Each product champion represents a specific user class and serves as the primary interface between users in that class and developers. The product champion must be an actual user and collect requirements from other members of their user class.

Agree on Decision Makers

Developers must identify the decision makers for requirements issues and conflict resolution. The following points should be considered:

- How well does the current decision-making approach work?
- Where does the current decision-making approach break down?
- Are the right people making decisions?
- Identify who should be involved in decision making and suggest processes they should use for reaching agreement on the requirements issues before developers have to resolve these issues themselves.

Classifying Customer Input

To most effectively use the information collected, it should be classified into categories. Classification categories might include:

Business requirements — anything that describes the financial, marketplace or other business benefit that either customers or the developing organization can gain from the product.

Use cases or scenarios — general statements of user goals or business tasks.

Business rules — operating principles about a business process.

Functional requirements — define what the system will do and usually include statements that begin with "The user must be able to …[perform some function]."

Quality attributes — statements that indicate how well the system performs some behavior or lets the user take some action.

External interface requirements – describe the connections between

Continued on Page 5

Upcoming events

May 13–16 ProjectWorld Boston 2002, Hynes Convention Center, Boston, MA. For more information visit: www.projectworld.com. **Visit ESI at Booth #819**.

May 13–17 GTC West 2002 16th Annual Government Technology Conference, Sacramento Convention Center, Sacramento, CA. For more information visit: www.govtech.net/events. **Visit ESI at Booth #1419**.

May 22–24 WelcomWorld 2002 "Project Success — Making the Connection," Le Meridien Hotel, New Orleans, LA. For more information visit: www.welcomworld.com.

May 29–30 "Mastering the Project Management Office: An Asian Perspective." Grand Hyatt, Singapore. For more information visit: www.esi-au.com.

June 4–6 IPMA 2002 16th World Congress on Project Management in Berlin, Estrel Convention Center, Berlin, Germany. For more information visit: www.worldcongress.de.

June 19–20 PMI Europe 2002: The Project Management Festival, Noga Hilton, Cannes, France. For more information visit: www.pmieurope2002.com.

June 24–27 E-Gov 2002, Washington Convention Center, Washington, DC. For more information visit: www.e-gov.com.

July 14–17 PMI Research Conference 2002 "Frontiers of Project Management Research and Applications," Elliott Grand Hyatt Seattle, Seattle, WA. For more information visit: www.pmi.org.

July 15–17 IIR's ProjectIMPACT, The Sheraton Hotel & Towers, New York, NY. For more information visit: www.iir-projectimpact.com.

July 22–24 NCMA World Congress 2002, Long Beach, CA. For more information visit: www.ncmahq.org. **Visit ESI at Booth #217.**

July 23–24 IIR's Developing Leaders and Planning Succession, New York, NY. For more information visit: www.iirusa.com/lead.

Industry News and Notes . . .

updates from the project management field

Bonus Pay Rising For IT and PMP® Certification

According to an article in the March 20, 2002 issue of InformationWeek, bonus pay increased for IT certifications last year but steadily declined for individual technical skills that had not been certified, according to a survey by Foote Partners LLC, an IT workforce research company.

Certification bonus pay rose 4% from the fourth quarter of 2000 to the fourth quarter of 2001, reaching a median average of 8.3% of base pay.

The Project Management Professional certification provides the best bonuses, averaging 14% of base pay.

For the complete article visit www.informationweek.com.

Genuity Attains the Industry's First ISO 9001 Certification for Project Management Practice

Genuity Inc., a provider of Internet infrastructure services and eBusiness Network Platforms, announced in April that it has received ISO 9001:2000 Certification for the quality management system established and followed by its eServices Delivery professionals — program and project managers that manage all aspects of complex eBusiness implementations using a structured approach through all phases of the project life cycle.

Genuity is the first Internet services provider to achieve this level of certification for a professional service.

Genuity was awarded this certification by TUV Management Service, an international certification organization and leading ISO 9000 registrar.

For more information visit www.genuity.com.

PMI® Chapters to Sponsor Risk Management Symposium

The PMI® Risk Management SIG is partnering with the PMI®Los Angeles, Orange County and San Diego chapters to organize the *SoCal Risk Management Symposium*.

The event, scheduled for September 12–14, 2002 in Long Beach, CA, features three prominent speakers from the project management community: **Dr. Barry W. Boehm**, TRW Professor of Software Engineering, Computer Science Department — Director, USC Center for Software Engineering; **Dr. Robert N. Charette**, President of ITABHI Corporation; and **Dr. Brian Hagen**, Managing Partner of Strategic Decisions Group's Strategy Implementation Practice. For more information visit www.RiskSIG.com/socal.

Continued from Page 3 — Requirements Engineering

the product and the rest of the universe. Examples are interfaces and communication mechanisms for users, hardware and other software systems.

Constraints – conditions that limit the options available to the designer or the engineers. They represent another type of nonfunctional requirement that should be documented.

Data definitions – customers describe the format, allowed values or default value for a data item or the composition of a complex business data structure. These definitions should be collected in a data dictionary to provide a master reference that stakeholders can use throughout the product's development and maintenance.

Solution ideas – a customer's suggestion as to a specific way to interact with a system to carry out some action.

Completing the Process

The following cues suggest stakeholders and team members have reached the point of diminishing returns on requirements elicitation.

- Users cannot think of any more use cases
- Users begin to repeat issues they already discussed previously
- Suggested functional requirements are all deemed to be out of scope
- Proposed new requirements are all low priority
- Users start proposing capabilities that might be included "sometime in the lifetime of the product," rather than "in the specific product we are talking about right now"

Now that the software requirements have been developed and documented, the next step involves the principles and practices of requirements management. Don't miss the conclusion of this series, to appear in the June issue.

George Sifri has more than 15 years of experience in managing information technology projects. To contact Sifri e-mail him at gsifri@ccc.gr.

Karl Wiegers, Ph.D., is Principal Consultant with Process Impact in Portland, OR, where he specializes in requirements engineering, peer reviews, process improvement and project management. Wiegers is the author of "Peer Reviews in Software: A Practical Guide," "Software Requirements," and "Creating a Software Engineering Culture" as well as more than 150 articles on many aspects of software, chemistry and military history. For more information visit www.processimpact.com.

References:

Andriole, Stephen. Managing Systems Requirements: Methods, Tools, and Cases. New York: The McGraw-Hill Companies, 1996.

Dorfman, Merlin, and Richard H. Thayer. Standards, Guidelines, and Examples on System and Software Requirements Engineering. New York: IEEE Computer Society Press, 1999.

McConnell, Steve. Software Project Survival Guide. Redmond, WA: Microsoft Press, 1997.

Wiegers, Karl E. Software Requirements. Redmond, WA: Microsoft Press, 1999.

WebReview: Site Targets IT Project Managers

By Claudia Payne

If you manage projects in today's complex IT marketplace, you know that information on how to succeed is invaluable. You're looking for information that doesn't cost a bundle but still offers help you can use. One Web site where you can find extensive help with managing IT projects is Gantthead.com.

To make the most of this informational site, you'll have to register and set up your profile, which is a little time consuming. However, the benefits of completely filling out the registration form will become apparent as you start using the site.

This site uses your profile to customize the information and links that appear. You can also sign up to receive a weekly e-mail newsletter and information from sponsors.

Like most good sites, however, you can also opt out of any and all of these "mailings." And although the site does sell sponsorships and advertising, the ads are not intrusive.

If you're looking for specific information, you can search the site, link to the gantthead-at-a-glance feature or go directly to the topic of your choice. The site is organized by departments including Business Intelligence, Career Development, Customer Relationship Management and Knowledge Management.

The site also offers a "Project Accelerator" section that provides help with specific project management functions like estimating ROI, getting management buy-in or measuring results.

You can also look for specific processes and go directly to the download area. In the download section, you can find templates, checklists and other documents.

Many of the downloads are for premium members only and you can subscribe to become a member at a reasonable fee. But if you're interested only in free information, the site still offers many of its templates at no cost.

Other links from the home page include daily featured articles from project management experts, industry news stories and a calendar of project management or IT industry conferences and seminars.

If you've written articles, white papers or have research to share with the community, you can submit these to the site. You can earn PDUs and even monetary compensation, depending on the content of your submission.

Check out this valuable site at **www.gantthead.com**.

Continued from Page 1 — Managing Software Developers

themselves, but ultimately, this attitude builds bad software.

More end-user-oriented project managers can offer a valuable contribution to their development teams by providing the business perspective that is often lacking. Since this often results in a limitation on "cool" technology, developing a shared vision with the team is critical.

Keep Developers Involved and Interested

When I worked for a small software company, we started down a path that was completely cutting-edge. After spending almost 50% of our time with the whole, small team working out the true user requirements, we concluded that the application couldn't be built cost effectively at that time, no matter how cool it was.

Most of the technology wasn't ready and the technology that did exist was cost-prohibitive. Normally, canceling a project in a development shop will automatically lead to resignations. We didn't lose anyone.

By letting the development team explore the technology enough to know themselves that it wasn't going to work, we managed to keep the team intact. We also kept them in the loop, in somewhat painful detail, as to cost impact if they went down a certain path.

Of course, the team was frustrated when we stopped the project. But, they were also proud of themselves that they had done the right thing for the company and for the customer. For some of them it was the first time they had participated in making a real business decision.

On projects with a high level of technology risk, keep the initial team small and composed of predominantly senior people.

Make sure everyone agrees on the problem you are trying to solve as well as the impact and ramifications of either failure or success. Technology will still drive their initial decisions but in many cases this is good. Just make sure the team is capable of knowing when certain technology isn't ready yet and whether it will be better, faster and cheaper a year from now.

Most Developers Are Geniuses

I'm surprised at how many project managers have trouble with this concept. I follow the general rule that developers are as brilliant as they say they are until proven otherwise.

I also strive to be as tolerant as possible of either poorly developed social skills or high levels of arrogance. Most people will work with you if you embrace their own positive concept of themselves.

Most developers are also smarter at what they do than I can ever be. It's my job to make sure that they are free to perform without impediments.

Removing impediments for developers translates to an extra level of care and feeding that IT programmers don't demand and will never get. Over the years, to keep teams happy, I have purchased espresso machines, set aside square footage for a video arcade game, worked with facilities to build additional offices and handed out the beer at the fooz ball tournament.

In general, unless it's illegal, immoral or just too expensive, keeping the team in their choice of perks is a requirement of the project.

Of course standard rules apply in the case of a "bad apple." No project can thrive with disruptive team members. One of the most commonly cited problems on projects is that one individual is considered indispensable no matter how many problems he or she causes. This is not and never has been true. Just because they're geniuses doesn't always make them the right individual for your project team.

Ego in a developer is good. Divisive behavior on a project is bad. If they share the project vision and are committed to the goal, don't sweat the small stuff.

Respecting the Zone

About a year ago there was a lively exchange on the NewGrange list serve about "stupid project managers who interrupt developers for a status check while they're in the zone."

The "zone" is that place of ultimate technical creativity where code flows from on high, through one's body and into the computer with almost magical ease.

Developers in the zone have been known to hang sheets over their office windows or string yellow crime scene tape across their cubicle door. According to developers, it is a project manager's job to respect the time they're in the zone by not inter-

Continued on Page 7

Continued from Page 6 — Managing Software Developers

rupting, not demanding meeting attendance and by generally keeping the world away.

Most project managers, however, have some problems with this perspective. Sometimes the "zone" doesn't produce the quality of magical code that the developer hopes it will. How is a project manager to maintain any sense of schedule if the project deliverable is dependent on the will of the cyber muse? The answer is simple: Know your team. Know their working styles. Get a good sense of the quality of their work product as early in the project as possible.

There are good developers and mediocre developers, although both think they're geniuses. The good ones may do all their work between 4 p.m. and midnight but you'll know by their first milestone that they're delivering what they promised.

Mediocre developers may work twice as hard and be great people but based on research studies, no amount of time, patience or training will ever improve their skills.

Remember developers do need to listen to their muse and creative work doesn't always come on demand, but signs of excessive stress and missed deadlines always indicate a problem. Adjust the work load to fit the true productivity of the developer or make staff changes.

Respect Your Team

Working with software developers is the same as working with any other project team. Respect and understanding for who they are and what they do solves 90% of the problems before they even start.

By spending time up front developing a shared vision of the end goal of the product, it's possible to balance the developers' need to work with cutting-edge technology against the true, often mundane requirements of the end customer.

Developing state-of-the-art software systems isn't like managing an assembly line. A good project manager can work with the team to understand how to adjust scope, time and resources to produce the best product within the predefined constraints.

Donna Fitzgerald is a senior partner with Knowth Consulting, specializing in portfolio and project management. She is also co-founder of the NewGrange Center for Project Management, www.newgrange.org, a worldwide nonprofit project management organization. Fitzgerald has more than 20 years of experience managing projects at Oracle, Bell Atlantic, SUN, Intel and Nortel.