ISSN No. 0976-5697

Volume 4, No. 10, September-October 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A comprehensive study of Software Risk Management

Akshay Sharma*[1], Deepak Basora[2], Nikita Chhillar[3] & Deepika Yadav[4]
Department of Computer Science, Dronacharya college of engineering
Gurgaon, India
sharma.akshay781@gmail.com[1], deepak1990basora@gmail.com[2],
nikitachhillar@yahoo.com[3], dipssaggi16@gmail.com[4]

Abstract: The challenges and realities in applying effective software risk management processes are difficult, in particular integrating the risk management processes into software development organizations. However, the benefits of implementing effective risk management tools and techniques in software development project are equally great. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. Current perceptions and emerging trends of various software risk management practices are reviewed and risks specific to software development projects are identified. Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process. If risk management process is in place for each and every software development process then future problems could be minimized or completely eradicated. This paper addresses lessons learned from implementing project risk management practices in software development environment and recognizes the increasing role of risk management in present software projects and aims at providing more support in this area.

Keywords: Risk management tools, Risk management planning, Risk management process, Software development

I. INTRODUCTION

The software industry is one of the largest manufacturing industries in the world, with $350 billion in off-the shelf software sold each year and over $100 billion in customized code on top of that. Project failures are the result of the multiplicity of risks inherent in software project environment. Software development projects are collections of larger programs with many interactions and dependencies. It involves a creation of something that has never been done before although the development processes are similar among other projects [1].

Risk management is an investment; that is, there are costs associated with identifying risks, analyzing those risks, and establishing plans to mitigate those risks. Software risk management is a software engineering practice with processes, methods, and tools for managing risks in a project. The main objective of Risk Management is to identify potential problems before they occur so that risk handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. It should begin at the earliest stages of project planning and continue throughout the total life cycle of the project [2]. Different types of risks are found that will affect budget, user satisfactions, and system performance. Studies indicate that 15 to 35% of all software projects are cancelled outright, and the remaining projects suffer from schedule slippage, cost overruns, or failure to meet their project goals [3, 4].

Software project risk management is an ethic in which the project team continually assesses what may negatively impact the project, determines the probability of such events occurring, and determines the impact of such events. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process [2].

However the project success is difficult to predict because project scope is changed by continuous market requirements and resources are constantly being reallocated to accommodate latest market conditions. Projects for specific customers also have a large degree of uncertainty for requirements due to the customized technical attributes. As a result, software development engineers have high turnover rates among software development firms. For example, software managers in India perceived personnel turnover as their biggest source of risk [5].

Many software projects and programs involve many entities such as companies, divisions, etc., that may have certain interests. There is often a feeling of disconnection between software developers and their management, each believing that the others are out of touch with reality resulting in misunderstanding and lack of trust. Research shows that 45% of all the causes of delayed software deliverables are related to managerial issues [6].

II. RISK

A risk is a potential future harm that may arise from some present action, such as, a schedule slip or a cost overrun. The loss is often considered in terms of direct financial loss, but also can be a loss in terms of credibility, future business, and loss of property or life.

Risk in itself is not bad; risk is essential to progress, and failure is often a key part of learning. But we must learn to balance the possible negative consequences of risk against the potential benefits of its associated opportunity. (Van Scoy, 1992) [7]

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization [8]. In IT systems, risk can be introduced from the internet, servers, networks, malicious insiders and even lapses in physical security.

Risk is the possibility of loss. It is a function of both the probability of an adverse event occurring and its impact; the impact manifests itself in a combination of financial loss, time delay, and loss of performance. A risk is the precursor to a problem; the probability that, at any given point in the software life cycle, the predicted goals cannot be achieved within available resources. Risk cannot be eliminated from a software project, but it can be managed.

Risk management in software engineering is related to the various future harms that could be possible on the software due to some minor or non-noticeable mistakes in software development project or process. Risk management is critical to the success of any software effort and is a strategic aspect of all software projects. This issue is generally managed by Software Project Management. During the life cycle of software projects, various risks are associated with them. These risks in the software project is identified and managed by software risk management. Some of the important aspects of risk management in software engineering are software risk management, risk classification and strategies for risk management. [9]

A. Factors creating risks:

Current perceptions about risk management from majority of software project organizations contributes to the lack of project stability in addition to the inherent challenges posed by the nature of software projects. Forces that contribute to loss or damage constitute elements of risk. Some influences are external to the enterprise and other influences are internal to the enterprise. These forces cannot be completely eliminated, and, hence, the enterprise has to take a calculated risk on its IT investment.

Risk can be classified into systematic and unsystematic risks [10]. Systematic risk refers to that portion of risk caused by external factors; this is common and may affect all firms. Virus, hacking, fire, natural disasters and power loss are sources of systematic risk. Their effect is felt by many of the companies that are placed in the same position. For example, a loophole in the Internet browser that is vulnerable for hacking affects all of the firms that use the browser. Unsystematic risk is the portion of total risk that is unique to the firm. The factors such as misuse of data, loss of data, application error, and human interaction, inside attack and equipment malfunction can be cited for unsystematic risk. Unsystematic factors are largely independent of factors affecting the IT industry in general.

The proportion of systematic and unsystematic risk denotes degree of vulnerability of the firm to the external or internal factors. Systematic risk is also known as generic risk, and unsystematic risk is also known as specific risk. Even though systematic risk is common for all firms of similar nature, its effect is not the same across all firms. This may be due to differences in the level of exposure and counter measures taken by firms.

a. Further there are three key software risk factors and concerns of both executives and software managers.

a) ***Estimation errors:*** Some tasks are harder to estimate than others because of the lack of experience of similar tasks or because of the nature of the task. Producing a set of user manuals is reasonably straightforward and, given that we have carried out similar tasks previously, we should be able to estimate with some degree of accuracy how long it will take and how much it will cost. Estimation can be improved by analysing historic data for similar activities and similar systems.

b) ***Planning assumptions:*** At every stage during planning, assumptions are made which, if not valid may put the plan at risk. In the planning process it is important to list explicitly all the assumptions that have been made and identify what effect they might have on the plan.

c) ***Eventualities****:* Some eventualities might never be foreseen and we can only resign ourselves to the fact that unimaginable things do. They are however very rare. The majority of unexpected can be identified- the requirements specification might be altered after some of the modules have been coded, the required hardware might not be delivered on time. Such events do happen from time to time.

B. Risk classification:

The key purpose of classifying risk is to get a collective viewpoint on a group of factors. These are the types of factors which will help project managers to identify the group that contributes the maximum risk.

Risk classification is considered as an economical way of analyzing risks and their causes by grouping similar risks together into classes. Some of most important risks in software engineering project described in Table I are categorized as software requirement risks, software cost risks, software scheduling risk, software quality risks, and software business risks. [4]

Software risks could be classified as internal or external. Those risks that come from risk factors within the organization are called internal risks whereas the external risks come from out of the organization and are difficult to control. Internal risks are project risks, process risks, and product risks. External risks are generally business with the vendor, technical risks, customers’ satisfaction, political stability and so on. In general, there are many risks in the software engineering which is very difficult or impossible to identify all of them.

Table I Types of Risks

RISKS	DESCRIPTION
Software requirement risks	Lack of analysis for change of requirements. Change extension of requirements Lack of report for requirements Poor definition of requirements Ambiguity of requirements Change of requirements Invalid requirements
Software cost risks	Lack of good estimation in projects Unrealistic schedule The hardware does not work well Lack of testing Lack of monitoring Complexity of architecture Management change, technology change, and environment change Lack of reassessment of management cycle
Software scheduling risks	Inadequate budget Change of requirements and extension of requirements Human errors Inadequate knowledge about tools and techniques Long-term training for personnel Lack of employment of manager experience Lack of enough skill Lack of good estimation in projects
Software quality risks	Inadequate documentation Lack of project standard Inadequate budget Human errors Unrealistic schedule Poor definition of requirements Lack of enough skill Lack of testing and good estimation in projects

III. RISK ENGINEERING

The objective of risk management is to avoid or minimize the adverse effects of unforeseen events by avoiding the risks or drawing up contingency plans for dealing with them. There are number of models for risk management, in that they identify two main components- risk identification and risk management. An example of often used model viewed in Fig 3.1 shows a task breakdown structure of risk engineering.

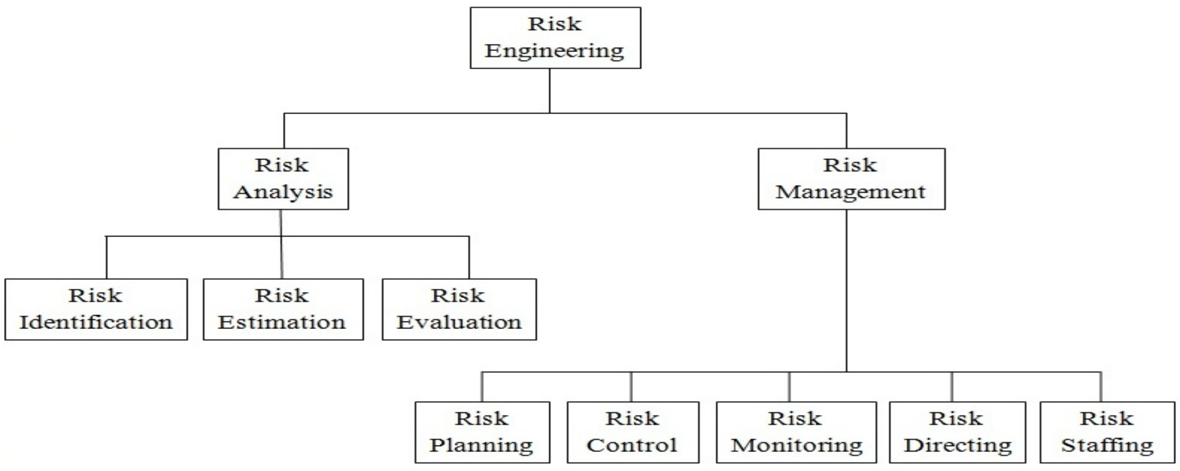

Figure 3.1 Risk engineering task breakdown structure

A. Software development risk management processes:

As shown in Fig. 3.2, Software development management is an eight-step process during the initial phases of the project. When any new risks are identified throughout the project, a five-step inner process is used to improve earlier estimates and judgments continuously. Despite the inherent risks associated with software development projects, there are strong indicators that these risks can be managed successfully. Research of failed software projects showed that "their problems could have been avoided or strongly reduced if there had been an explicit early concern with identifying and resolving their high-risk elements". Effective risk management is the most important management tool a project manager can employ to increase the likelihood of project success. Since risk management is not widely used and understood, this could be a significant competitive advantage to those that implement the risk management processes in their projects. A large number of processes have been generated in recent years to address the need for more effective risk management.

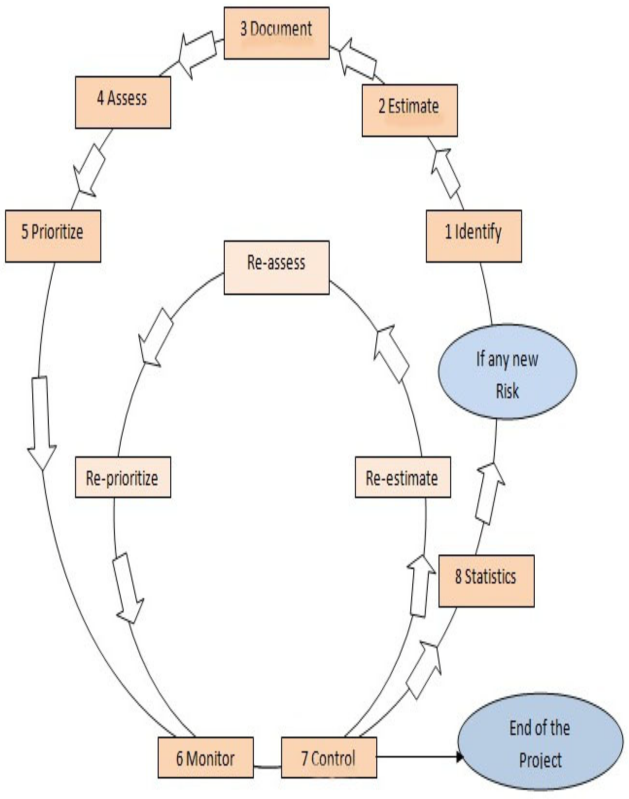

Figure 3.2 Soft Risk Model

B. *Risk Identification:*

Risk Identification consists of listing all of the risks that can adversely affect the successful execution of the project. In the risk identification step, the team systematically enumerates as many project risks as possible to make them explicit before they become problems. The first stage in any risk assessment exercise is to identify the hazards that might affect the duration or resource costs of the project. A hazard is an event that might occur and will, if it does occur, create a problem for the successful completion of the project. For example, the illness of a team member is a hazard that might result in the problem of late delivery of a component. The late delivery of that component is likely have an effect on other activities and might, particularly if it is on the critical path; put the project completion date at risk [11].

A common way of identifying hazards is to use a checklist listing all the possible hazards and factors that influence them. Some hazards are generic risks- that is, they are relevant to all software projects and standard checklists can be used and augmented from an analysis of past projects to identify them. Some risks are identified in Fig 3.3.

GENERIC RISKS	PRODUCT-SPECIFIC RISKS	
PROJECT RISKS	PRODUCT RISKS	BUSINESS RISKS
Factors to Consider: **People, size, process, technology, tools, organizational, managerial, customer, estimation, sales, support**		

Figure 3.3 General Categories of risk

Generic risks are potential threats to every software project. Some examples of generic risks are changing requirements, losing key personnel, or bankruptcy of the software company or of the customer. It is advisable for a development organization to keep a checklist of these types of risks. Teams can then assess the extent to which these risks are a factor for their project based upon the known set of programmers, managers, customers, and policies.

Product-specific risks can be distinguished from generic risks because they can only be identified by those with a clear understanding of the technology, the people, and the environment of the specific product. An example of a product-specific risk is the availability of a complex network necessary for testing. Generic and product-specific risks can be further divided into project, product, and business risks.

Project risks are those that affect the project schedule or the resources (personnel or budgets) dedicated to the project. *Product risks* are those that affect the quality or performance of the software being developed.

Finally, *business risks* are those that threaten the viability of the software, such as building an excellent product no one wants or building a product that no longer fits into the overall business strategy of the company.

a. The categories of factors that will need to be considered include the following:

a) Application factors:

The nature of the application- whether it is a simple data processing application, a safety critical system or a large distributed system with real-time elements- is likely to be a critical factor. The expected size of the application is also important-the larger the system, the greater is the likelihood of errors and communication and management problems.

b) Staff factors:

The experience and skills of the staff involving are clearly major factors-an experienced programmer is, less likely to make errors than one with little experience.

c) Project factors:

It is important that the project and its objectives are well defined and that they are absolutely clear to all members of the project team and all key stakeholders. Any possibility that this is not the case will pose a risk to the success of the project.

d) Hardware/Software factors:

A project that requires new hardware for development is likely to pose a higher risk than one where the software can be developed on existing hardware. Where a system is developed on one type of hardware or software platform to be used on another there might be additional risks at installation.

e) Supplier factors:

The extent to which a project relies on external organizations that cannot be directly controlled often influences the project's success. Delays in, for example, the installation of telephone lines or delivery of equipment may be difficult to avoid.

f) Environment factors:

Changes in the environment can affect a project's success. A significant change in the taxation regulations could have serious consequences for the development of an application.

C. Risk estimation:

Having estimated the risks that might affect our project we need some way of assessing their importance. Risk Estimation consists of assessing the likelihood and impact of each hazard. The probability of a hazard's occurring is known as the risk likelihood; the effect that the resulting problem will have on the project, if it occurs, is known as the risk impact and the importance of the risk is known as the risk value or risk exposure. The risk value is calculated using (1).

Risk Exposure = risk likelihood x risk impact (1)

Ideally the risk impact is estimated in monetary terms and the likelihood assessed as a probability. The risk exposure for various risks can then be compared with each other to assess the relative importance of each risk and they can be directly compared with the costs and likelihoods of success of various contingency plans.

D. Risk Evaluation:

Risk evaluation consists of ranking the risks and determining risk aversion strategies. Many risk managers use a simple scoring method to provide a quantitative measure for assessing each risk. Some just categorize likelihoods and impacts as high, medium or low, but this form of ranking does not allow the calculation of a risk exposure. A better and popular approach is to score the likelihood and impact on a scale of, say 1 to 10 where the hazard is more likely to occur receives a score of 10 and he least likely a score of 1.

Impact methods, must take into account the total risk to the project. This must include the following potential costs:

a. The cost of delays to scheduled dates for deliverables;
b. Cost overruns caused by using additional or more expensive resources;
c. The costs incurred or implicit in any compromise to the system's quality or functionality;

Managing risk involves the use of two strategies. First there is reducing the risk exposure by reducing the likelihood or impact; secondly, Drawing up contingency plans to deal with the risk should it occur;

The analyzed risks are organized into a risk table. The template for a risk table is shown in Table II. The information that is to be provided in each of the columns is now explained.

a) *Rank* of the risk.
b) *Risk* is the description of the risk itself.
c) *Probability* is the likelihood of the risk occurring, using either a numeric or categorical scale, as discussed in the last section.
(1)
d) *Impact* is the magnitude of the loss if the risk were to occur, using either a numeric or a categorical scale.
e) *Rank last week* and the number of *weeks on list* are documented so the team can monitor changes in priority, to determine if actions are being taken that cause changes in the stature of the risk.
f) *Action* documents what the team is doing to manage the risk. The action field is often not completed until the risks have been prioritized [11].

Table 2. Risk Table

Rank	**Risk**	**Probability**	**Impact**	**Rank Last week**	**Action**

Some risks, once recognised, can be reduced or avoided immediately with very little cost or effort and it is sensible to take action on these regardless of their risk value. For other risks we need to compare the costs of taking action with the benefits of reducing the risk. One method for doing this is to calculate the risk reduction leverage (RRL) using (2).

$$RRL = \frac{RE_{before} - RE_{after}}{risk reduction \cos t} \qquad (2)$$

Where RE_{before} is the original risk exposure value, RE_{after} is the expected risk exposure value after taking action and the risk reduction cost is the cost of implementing the risk reduction action. If the values are expected monetary values then an RRL greater than one indicates that we can expect to gain from implementing the risk reduction plan because the expected reduction in risk exposure is greater than the cost of the plan. In either case the higher the leverage value for a risk then the more worthwhile it will be to plan the risk reduction action.

E. Risk Planning:

Risk planning consists of drawing up contingency plans and, where appropriate, adding these to the project's structure. With small projects, risk planning is likely to be the responsibility of the project manager but medium or large projects will benefit from the appointment of a full-time risk manager. Following are some examples of the kinds of risk planning actions that can take place.

(2)

***a. Information buying*:**

Perceived risk can be reduced by obtaining more information through investigation. For example, in a project in which the use of a new technology has created risk, the team can invest some money to learn about the technology. Throw-away prototypes can be developed using the new technology to educate some of the staff on the new technology and to assess the fit of the new technology for the product.

***b. Contingency plans*:**

A contingency plan is a plan that describes what to do if certain risks materialize. By planning ahead with such a plan, you are prepared and have a strategy in place do deal with the issue.

***c. Risk reduction*:**

For example, if the team is concerned that the use of a new programming language may cause a schedule delay, the budget might contain a line item entitled "potential schedule" to cover a potential schedule slip. Because the budget already covers the potential slip, the financial risk to the organization is reduced. Alternately, the team can plan to employ inspections to reduce the risk of quality problems.

***d. Risk acceptance*:**

Sometimes the organization consciously chooses to live with the consequences of the risk [12] and the results of the potential loss. In this case, no action is planned.

F. Risk Control:

Risk Control concerns the main functions of the risk manager in minimising and reacting to problems throughout the project. This function will include aspects of quality control in addition to dealing with problems as they occur.

There are five strategies for risk controlling.

a) Hazard Prevention: Some hazards can be prevented from occurring or their likelihood reduced to insignificant levels.
b) Likelihood reduction: Some risks, while they cannot be prevented, can have their likelihoods reduced by prior planning. The risks of late changes to a requirement specification can, be reduced by prototyping.
c) Risk avoidance: A project can be protected from the risk of overrunning the schedule by increasing duration estimates or reducing functionality.
d) Risk transfer: The impact of some risks can be transferred away from the project by contracting out or taking out insurance.
e) Contingency planning: Some risks are not preventable and contingency plans will need to be drawn up to reduce the impact should the hazard occur. A project manager should draw up contingency plans for using agency programmers to minimise the impact of any unplanned absence of programming staff.

G. Risk Monitoring:

Risk monitoring must be an ongoing activity, as the importance and likelihood of particular risks can change as the project proceeds. After risks are identified, analyzed, and prioritized, and actions are established, it is essential that the team regularly monitor the progress of the product and the resolution of the risk items, taking corrective action when necessary. This monitoring can be done as part of the team project management activities or via explicit risk management activities. Often teams regularly monitor their "Top 10 risks." Risks need to be revisited at regular intervals for the team to revaluate each risk to determine when new circumstances caused its probability and/or impact to change. At each interval, some risks may be added to the list and others taken away. Risks need to be reprioritized to see which are moved "above the line" and need to have action plans and which move "below the line" and no longer need action plans. A key to successful risk management is that proactive actions are owned by individuals and are monitored. [13]

As time passes and more is learned about the project, the information gained over time may alter the risk profile considerably. Additionally, time may make it possible to refine the risk into a set of more detailed risks. These refined risks may be easier to mitigate, monitor, and manage.

H. Risk Directing:

Risk Directing and staffing are concerned with day-to-day management of risk. Risk aversion and problem strategies frequently involve the use of additional staff and this must be planned for and directed.

IV. CONCLUSION

The most important thing for a software project to do is to get focused on its critical success factors. Although software risk management is a daunting task, organizations that implement effective processes proved to be successful, while those that fail in this effort will be unsuccessful. The nature of software projects creates many risks that must be managed diligently to avoid the common drawback of many projects. For various reasons, including the influence of previous document-driven software management guidelines, projects get focused on activities which are not critical for their success. We can take some steps such as, ranking the project's most significant risk items, establishing a regular schedule for higher management reviews of the project's progress and so on to keep tracking on major risk factors. Formal risk management process is recommended to manage complex issues associated with software development projects. Many risk management processes have been created to aid organizations, but integrating the processes into organizations was not successful. Effective risk management process will succeed by changing the organizational culture to motivate the individual. We observed, still now software risk management reside in back seat but we should keep more focus on it. To handle all of the complex people-oriented and technology-driven success factors involved in software projects, a great measure of human judgment is required.

V. REFERENCES

[1] http://www.engpaper.com/project-risk%C2%A0management.htm

[2] http://www.ijcit.org/ijcit_papers/vol2no1/IJCIT-110740.pdf

[3] Boehm, B.W., 1991. Software risk management principles and practices. IEEE Software 8 (1), 32–41.

[4] Klein, S.A., 1998. Putting methodology in perspective from a project risk viewpoint. In: IEEE Power Engineering Society 1999 Winter Meeting, vol. 1,, pp. 362–365.

[5] Boehm, B.W., DeMarco, T., 1997. Software risk management. IEEE Software 14 (3), 17–19.

[6] Van Genuchten, M., 1991. Why is software late? An empirical study of reasons for delay in software

development. IEEE Transactions on Software Engineering 17 (6), 582–590.

[7] Van Scoy, R. L., "Software Development Risk: Opportunity, Not Problem," Software Engineering Institute, Pittsburgh, PA CMU/SEI-92-TR-030.

[8] NIST Risk Management Guide for Information Systems Special Publication 800-30. July, 2002

[9] www2.latech.edu/.../Risk%20Management%20in%20Software%20Engineering...

[10] Reilly, F.K.; K. Brown; Investment Analysis and Portfolio Management, Harcourt College Publishers, 2002

[11] Bob Hughes and Mike Cotterell; Software Project Management, Tata Mcgraw hill, 2010.

[12] Hall, E. M. (1998). Managing Risk: Methods for Software Systems Development, Addison Wesley.

[13] Larman, C. (2004). Agile and Iterative Development: A Manager's Guide. Boston, Addison Wesley.