

Effective Risk Management Techniques in Development of IVR Software

¹Srivastava Kumar Devesh, ²Dr. Hada Singh Priyank

¹Dept. of CSE, Mnipal University, Jaipur, Rajasthan, India

Abstract

Every software project is exposed to adverse internal as well as external influences, called project risks that affect the cost and the duration of the project and possibly the quality of the products. With a risk analysis it can be determined for a specific project what the risks are. These risks then should be included in a systematic and formal manner in the project in order to obtain a realistic and reliable project estimate and a realistic project plan. Currently, software quality assurance does not apply completely on the development of software industry and this leads to some challenges in the industry of software, specially, concerning risk, cost and time consuming. While the success of software development project depends on the application quality assurance standards which starts by the pre-project and continue through the project till it reaches the user at the end. The aim of this paper is the challenge the risk factors incurred during the development of IVR software projects without defects but also the customer's acceptance and satisfaction of the software. Thus to prevent software development projects failure, the risk should be analysed and managed at an early stage.

Keywords

Interactive Voice Response (IVR) Software Project Management (SPM)

I. Introduction

A risk analysis involves identifying the most probable defects to IVR software and analysing the related vulnerabilities of the software to these defects. A risk assessment involves evaluating existing physical and environmental security and controls, and assessing their adequacy relative to potential threats of the software. Software quality assurance is "A planned and systematic product conforms to established technical requirements. And a set of activities designed to evaluate the process by which the products are developed or manufactured [1]. Mostly, the success of any software development projects depends on three major factors, which are, quality, cost and time. As it was found, that one of the reasons for the failure of software development projects is that, some organizations hesitate to place a portion of the project's budget on the quality assurance process [2]. Software quality assurances are responsible to reduce the defect, and repair this defect before it reaches the production environment, because repairing the defect in the production environment is more expensive than repairing it, in the test environment [3]. There are basic risks that are generic to almost all software projects. These risks are - designed to focus on risks unique to each project, Looks at potential problems and plans for contingencies, Evaluates what could happen and looks for ways to minimize the damage, Plans to manage and mitigate potential causes of failure. Boehm defines four major reasons for implementing software risk management [4]:

- Avoiding software project disasters, including run away budgets and schedules, defect-ridden software products, and operational failures.
- Avoiding rework caused by erroneous, missing, or ambiguous requirements, design or code, which typically consumes 40-

50% of the total cost of software development.

- Avoiding overkill with detection and prevention techniques in areas of minimal or no risk.
- Stimulating a win-win software solution where the customer receives the product they need and the vendor makes the profits they expect.

Therefore, the software quality assurance is trying to reach the best product for users by reducing the cost and time. In this paper the IVR software are taken into account to reduce the maximum risk incurred during the development. Interactive Voice Response systems represent a powerful means for automating business and customer-facing processes. IVR systems process phone calls, play pre-recorded messages, provide callers with real-time data from any number of databases and potentially route calls to service agents. IVR technology requires virtually no human interaction over the telephone, as the user's interaction with the database is predetermined by what the IVR system will allow the user access to. For example, banks and credit card companies use IVR systems so that their customers can receive up-to-date account information instantly and easily without having to wait to speak with someone directly. IVR technology is also used to gather information, as in the case of telephone surveys or tele-votes in which the user is prompted to answer questions by pushing the numbers on a touch-tone telephone. IVR systems can combine touch-tone input, speech recognition and text-to-speech capabilities, resulting in high customer satisfaction and operational effectiveness. Figure 1.0 demonstrates the risks factors between the phase of software deployment and IVR voice system.

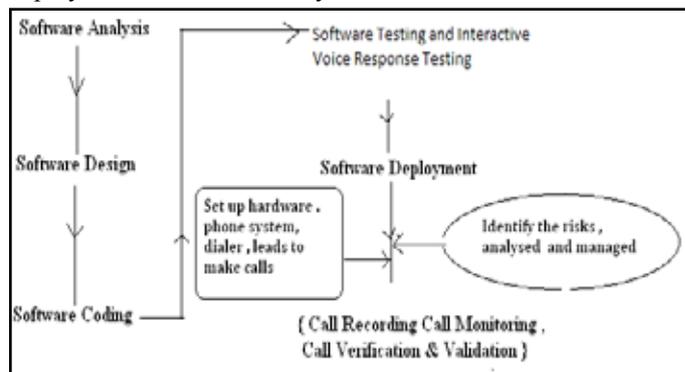


Fig. 1: The Life Cycle of IVR Software Development

IVR systems are complex. While they offer a wide range of features and functionality such as tone detection and voice recognition, they also provide different interfaces to other applications, literally rendering countless transactions possible:

- Bank account information and payments management
- Customer Relationship Managements (CRM) systems
- Personal voice mail systems
- All reservation systems, including airline reservation and check-in systems. Fig. 2, demonstrates the IVR architecture system.

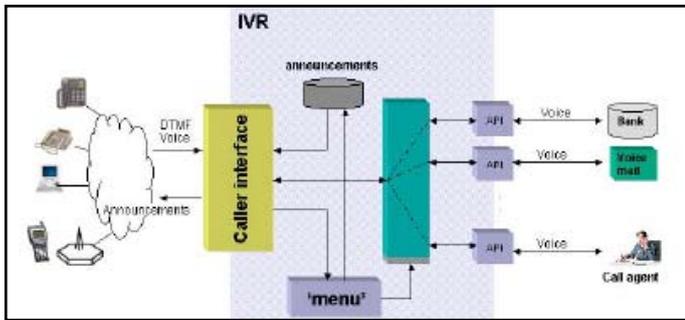


Fig. 2: IVR Architecture System

II. Software Risk

A software project may encounter various types of risks [20]:

Technical risks include problems with languages, project size, project functionality, platforms, methods, standards, or processes. These risks may result from excessive constraints, lack of experience, poorly defined parameters, or dependencies on organizations outside the direct control of the project team.

Management risks include lack of planning, lack of management experience and training, communications problems, organizational issues, lack of authority, and control problems.

Financial risks include cash flow, capital and budgetary issues, and return on investment constraints.

Contractual and legal risks include changing requirements, market-driven schedules, health & safety issues, government regulation, and product warranty issues.

Personnel risks include staffing lags, experience and training problems, ethical and moral issues, staff conflicts, and productivity issues.

Other resource risks include unavailability or late delivery of equipment & supplies, inadequate tools, inadequate facilities, distributed locations, unavailability of computer resources, and slow response times. It is important to note that the developed software risk methodologies have three fundamentally different objectives to manage risk factors by –

- Risk prevention
- Risk mitigation and correction
- Ensuring safe system failure

A. Risks Factors in SPM

Risk management focuses to assess the probability of risk occurring, risk event drivers, risk events, the probability of impact and the impact drivers before the risk actually takes place. The top software risk items can be

1. 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
- Inadequate of requirements
- Impossible requirements
- Invalid requirements

2. Software Cost Risks

- Lack of good estimation in projects
- Unrealistic schedule
- The hardware does not work well
- Human errors

- Lack of testing
- Lack of monitoring
- Complexity of architecture
- Large size of architecture
- Extension of requirements change
- The tools does not work well
- Personnel change
- Management change
- Technology change
- Environment change
- Lack of reassessment of management cycle

3. Software Scheduling Risks

- Inadequate budget
- Change of requirements
- Extension of requirements change Human errors
- Inadequate knowledge about tools
- Inadequate knowledge about techniques
- Long-term training for personnel
- Lack of employment of manager experience
- Lack of enough skill Lack of good estimation in projects
- Lack of accurate system domain definition
- Lack of goals specification
- Difficulty of implementation
- Disagreement between members
- Lack of tools Shortage of personnel
- Tools failure Technology change
- Lack of agreement between customer and developer
- Slow management cycle
- Supply budget in inappropriate time
- Environment change
- Lack of a good guideline

4. Software Quality Risks

- Inadequate documentation
- Lack of project standard
- Lack of design documentation
- Inadequate budget
- Human errors Unrealistic schedule
- Extension of requirements change
- Poor definition of requirements
- Lack of enough skill Lack of testing
- Lack of good estimation in projects
- Inadequate knowledge about techniques
- Lack of employment of manager experience
- Lack of accurate system domain definition
- The simulator is to be destroyed Lack of reassessment
- Inadequate knowledge about programming language
- Inadequate knowledge about tools
- The hardware does not work well Lack of analysis for change of requirements
- The tools do not work well Loss technical equipment
- Lack of stability between personnel
- Personnel change
- Weakness of management Lack of commitment
- Disagreement between members Ambiguity of requirements
- Complexity of architecture incomplete requirements
- Lack of roles and responsibilities definition
- Inadequate training of personnel
- Management change Technology change
- Lack of collaboration between developers

- Environment change
- Lack of a good guideline

5. Software Business Risks

- The products that no one wants them
- The products those are not suitable with total strategy
- The products that sellers do not know how to sell them
- Failure in total budget
- Failure in commitment
- Failure in management because of change in different people

III. Risk Management Activities

The risk management activities depict the different activities involved in the management of risk associated with software development. Risk management activities are controlled by Risk Assessment and Risk Control as shown in figure 3 .A brief summary of risk management paradigm process is described below [12-22].

A. Identification

In this activity common risk area are identified and then managed. The Risk Identification process identifies and documents risks that might affect the project. subject matter experts, customers, end users, other project managers, stakeholders, and risk management. There are many techniques for identifying risks, including interviewing, reporting, decomposition, assumption analysis, critical path analysis, and utilization of risk taxonomies.

B. Analyse

Risk analysis involves how the risk can be minimized by adopting risk management plans. Analysis is the conversion of risk data into risk decision-making information. Analysis provides the basis for the project manager to work on the “right” and most critical risks [19].



Fig. 3: Risk Management Activity Diagram

C. Prioritization

In this activity loss due to risks are measured and risks are assigned priorities on some scale.

D. Control

Risk control corrects deviations from planned risk actions. Once risk metrics and events have been chosen, there is nothing unique about risk control. Risk control melds into project management and relies on project management processes to control risk action plans, corrects for variations from plans, responds to triggering events, and improves risk management processes [21].

E. Planning

Planning turns risk information into decisions and actions. Planning involves developing actions to address individual risks, prioritizing risk actions, and creating an integrated risk management plan.

F. Tracking

Tracking consists of monitoring the status of risks and the actions taken to ameliorate them. Appropriate risk metrics are identified and monitored to enable the evaluation of the status of as well as of risk mitigation plans. Tracking serves as the “watchdog” function of management.

G. Resolution

Risk Resolution is the execution of the risk management plans for dealing with the risk.

H. Communication

Without effective communication, no risk management approach can be viable. While communication facilitates interaction among the elements of the model, there are higher-level communications to consider as well. In order to be analysed and managed correctly, risks must be communicated to and between the appropriate organizational levels. This includes levels within the development project and organization, within the customer organization, and most especially, across that threshold between the developer, the customer and where different the user.

IV. Software Risk Management Techniques

Table 1 describes all the risk management techniques during the development of IVR software.

Table 1: Risk Management Techniques

Source of Risk	Risk Management Techniques
Personnel shortfalls	Staffing with top talent; key personnel agreements; team-building; training; tailoring process to skill mix; walkthroughs.
Schedule, budget , Process	Detailed, multi-source cost and schedule estimation; design to cost; incremental development; software reuse; requirements de -scoping; adding more budget and schedule; outside reviews
External Components	Benchmarking; inspections; reference checking; compatibility prototyping and analysis.
Requirements mismatch	Requirements scrubbing; prototyping; cost-benefit analysis; design to cost; user Surveys
User Interface Mismatch	scenarios; user characterization (functionality; style, workload); identifying the real Prototyping;
Architecture, performance, quality	Simulation; benchmarking; modeling; prototyping; instrumentation; tuning.
Requirements changes	High change threshold: information hiding; incremental development

Legacy software	Reengineering; code analysis; interviewing; wrappers; incremental deconstruction.
Externally-performed tasks	Pre award audits, award-fee contracts, competitive design or prototyping.
Straining computer science	Technical analysis; cost-benefit analysis; prototyping; reference checking.

IV. Conclusion and Future Work

With ever-increasing complexity and increasing demand for bigger, better, and faster, the software industry is a high risk business. When teams don't manage risk, they leave projects vulnerable to factors that can cause major rework, major cost or schedule over-runs, or complete project failure. Adopting a Software Risk Management Program is a step every software manager can take to more effectively manage software development initiatives. Risk management is an on-going process that is implemented as part of the initial project planning activities and utilized throughout all of the phases of the software development lifecycle. Risk management requires a fear-free environment where risks can be identified and discussed openly. Based on a positive, proactive approach, risk management can greatly reduce or even eliminate the need for crisis management within our software projects. Thus one can conclude that formal risk management analysis and formal project assessment are effective and useful approaches that are starting to add rigor to the phrase "software engineering". Not every risk factor is fully controllable, and several risk factors exceed the authority of software managers. Nonetheless, risk analysis and assessment methods are quite effective in the identification of significant problems. Once problems are identified and examined, solutions can often be developed.

References

- [1] Abdullah T, Mateen A, Sattar A. R, Mustafa T, "Risk Analysis of various Phases of Software Development Models", *European Journal of Scientific Research*, Vol. 40(3), pp. 369-376, 2010.
- [2] Barry W. Boehm, "Tutorial: Software Risk Management", Les Alamitos, CA, IEEE Computer Society, 1989.
- [3] Alex Down, Michael Coleman, Peter Absolon, "Risk Management for Software Projects", London, McGraw-Hill Book Company, 1994.
- [4] Barry W. Boehm, "Tutorial: Software Risk Management", Les Alamitos, CA, IEEE Computer Society, 1989.
- [5] B. Gotterbarn, "Enhancing risk analysis using software development impact statements", *Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop*, 2001, pp. 43-51.
- [6] B. Shahzad, S. Safvi, "Effective Risk Mitigation: A User Prospective", *International Journal of Mathematics and Computers in Simulation*, Vol. 2, No. 1, 2008, pp. 70-80.
- [7] Costaa R, Barros Marcio, "Evolutionary Software Project Portfolio Risks", *Journal Systems and Software*, Vol. 80 (1), pp. 16-31, 2007.
- [8] Chittister, Clyde, Haimes, Yacov Y., "Risk Associated with Software Development: A Holistic Framework for Assessment and Management", *IEEE Transactions on Systems, Man, and Cybernetics* 23, 3 (May-June 1993), pp. 710-723.
- [9] Chittister, Clyde, Haimes, Yacov., "Assessment and Management of Software Technical Risk", *IEEE Transactions on Systems, Man, and Cybernetics* 24, 2 (February 1994), pp. 187-202.
- [10] Debbie Tesch, Timothy J. Kloppenborg, Mark N. Erolick, "It Project Risk Factors: The Project Management Professionals Perspective", *Journal of Computer Information Systems*, Summer 2007, pp. 61-69.
- [11] Levine, R., "Risk management systems: understanding the need", *Information Systems Management*, spring 2004, pp. 31-37.
- [12] Schmidt, R., Lyytinen, K., Keil, M., Cule, P., "Identifying software project risks: An interactional Delphistudy", *Journal of Management Information Systems*, (17:4), Spring 2001, 5-36.
- [13] Gang Xie, Zhan Jin, "Risk avoidance in bidding for Software Projects based on the Cycle Management Theory", *International Journal of Project Mangement*, Vol. 12(4), pp. 516-521, 2006.
- [14] Geoffrey G. Roy, "A Risk Management Framework for Software Engineering Practice", *Australian Software Engineering Conference*, pp. 60, 2004.
- [15] Gillian Adens, "The Role of Risk in a Modern Software Development Process", *TASSCTechnical Paper*, 2008.
- [16] G. Tate, J. Verner, "Case Study of Risk Management, Incremental Development and Evolutionary Prototyping", *Information and Software Technology*, Vol. 32, No. 3, 1990, pp. 207-214.
- [17] Hall, Elaine M, "Managing Risk – Methods for Software Systems Development", Addison Wesley, 1998
- [18] JNY fjord, M. Kajko-Mattsson, "Outlining A Model Integrating Risk Management and Agile Software Development", *Proceedings of the 34th Euro micro Conference Software Engineering and Advanced Applications*, 2008, pp. 476-483.
- [19] J. Miller, J. Grski, "A Method of Software Project Risk Identification and Analysis", Ph.D. Thesis, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 2005.
- [20] Linda Westfall, "Software Risk Management", PMB 383, 3000 Custer Road, Suite 270 Plano, TX 75075.
- [21] Rasmita Dash, Rajashree Dash, "Risk assessment techniques for software development", *European Journal of Scientific Research*, Vol. 42, No. 4, 2010, pp. 629-636.
- [22] Ronald P. Higuera, Yacov Y. Haimes, "Software Risk Management - Technical Report CMU/SEI-96-TR-012 ESC-TR-96-012 June 1996
- [23] S. Murthi, "Preventive Risk Management for Software Projects", *IT Professional*, Vol. 4, No. 5, 2002, pp. 9-15.
- [24] Zardari, S., "Software Risk Management", *International Conference on Information Management & Engineering* IEEE Computer Society, 2009, pp. 375-379.