White paper:

Comprehensive Review and Implementation of Risk Management Processes in Software Development

This paper reviews the principles of risk management in software development of GxP systems, elaborates on the well-known risk management approaches and identifies some of the important concepts of proactive management of risks throughout the software development lifecycle.

PharmOut®

MKT_TMP200_01_r06

Overview

This paper should help audiences interested in risk management in software development, validation and verification to gain an overall understanding of the area.

65% of project failures are accounted by management issues and 35% by technical issues. Managerial issues include problems which can be attributed to project structure, project resources, planning methodologies, customer buy-in, and inadequate risk management.

Software program production often encounters issues such as, over-budget costs, delays in schedule and low quality of product. All of these issues pose a risk to the development of software systems.

Program developers must perform a risk analysis before issues develop to identify the risks to their system, and create an action plan to mitigate the impact of the risks as well as resolve any issues that are unanticipated. The success of projects can be credited to the appropriate management of risks.

This paper reviews the fundamentals of software risk management and the different popular risk management process models.

Principles

One of the important considerations challenging any risk management is the definition of risk. There are several formal definitions of risk available in literature, few of which are presented below:

"Risk is a combination of an abnormal event or failure, and the consequences of that event or failure to a system's operators, users, or environment."

"A risk can range from catastrophic (loss of an entire system, loss of life, or permanent disability) to negligible (no system damage or injury)".

"A possible future event that, if it occurs, will lead to an undesirable outcome".

Software risks

The medical profession's growing reliance on electronic medical databases not only requires security guarantees but introduces safety hazards as well. With the safety of patients at risk it becomes even more critical the software developer must be familiar with risk assessment techniques.

For example, with a distributed database of medical records a risk analysis may identify that incorrect linkage to somebody else's medical record may be considered catastrophic.

The incorrect link is itself not necessarily harmful provided data in the original record does not get corrupted. The hazard occurs when a patient gets incorrectly diagnosed based on the incorrect or incomplete data supplied.

In today's economic climate it might no longer be cost effective to build entire systems from scratch, and more and more software is being constructed from Commercial off the Shelf (COTS) components.

The increasing use of COTS components also introduces a new set of possible risks. Components often provide more functionality than is actually required, and these unneeded services may interfere with intended functions.

As the source code may not be available, it is impossible to check if there is any malicious code such as viruses.

The rapid changes associated with COTS releases and internet and web-based systems makes it impossible to produce "air-tight" requirements.

In addition COTS components are usually designed for other, more generic purposes and are unlikely to have been subjected to the level of verification and validation and vendor audits required for safety critical systems.

Thus exposure to risk is multi-faceted. In addition to project management or business risks, the SMEs software developer must also deal with risk associated with software development.

Context of Risk Management

Risk management is a way to manage risks. In other words, it concerns all activities that are performed to reduce the uncertainties associated with certain tasks or events.

In the context of projects, risk management reduces the impacts of undesirable events on a project. Risk management in any project requires undertaking decision-making activities.

A brief summary of each risk management activity is described as follows:

- **Identify**: Identification surfaces software-related risks before they become actual problems which adversely affect the project. Before risks can be managed, they must be identified.
- **Analyse**: Analysis is the conversion of identified risk data into decision-making information, and provides the quantification and oversight clarity needed to guide the project manager to work on the "right" risks.
- **Plan**: Planning involves developing actions to mitigate individual software risks, prioritizing risk mitigation actions, and integrating these actions into an executable risk management plan.
- **Track**: Tracking consists of implementing the risk management plan and monitoring the status of risks and actions taken to mitigate those risks. Risk metrics and triggering events are monitored as part of the tracking function.
- **Control**: Control corrects for deviations from planned risk mitigation actions; and builds on project management processes to control mitigation plans, respond to triggering events, and improve risk management processes.
- **Communication**: Communication among the appropriate organizational entities must exist for risks to be identified, analysed, planned for, tracked, and controlled correctly. Risk communication lies at the centre of the paradigm to emphasize both its pervasiveness and its criticality.

ANSI/AAMI/IEC 62304 Management Process

This standard requires the use of the risk management process as defined in ISO 14971.

A section of the ISO 14971 pertains to control of identified risks associated with each hazard identified during the risk analysis.

It does not illustrate on details of the risks associated with activities defined in the software development life cycle.

The standard emphasis on the conditions required defining the risk associated with the failure of the software item, which are defining the software items in term of its purpose, interfaces with other software and hardware items in the system architecture, ensuring changes are controlled and finally specifying risks control measures and risk analysis on the system architecture.

It also identifies the software architecture as the earliest point at which software items can be classified definitively according to their safety role.

The standard refers to the three major principles which promote the safety of medical devices software as Risk Management, Quality Management and Software Engineering.

GAMP 5 Risk Management Process

GAMP 5 utilizes the ICH 09 guidelines for a systematic process for the assessment, control, communication, and review of risks. It is an iterative process used throughout the entire computerized system life cycle from concept to retirement and focuses on - identifying functions with Impact on Patient Safety, Product Quality, and Data Integrity

GAMP 5 prominences on the three approaches for managing risks as elimination by design, reduction to an acceptable level, and verification to ensure risk are managed to an acceptable level.

It specifically addresses risk controls of software and placement of risk control in the development life cycle and type / complexity of software .i.e. bespoke, and off-the-shelf.

TIR36:2007 Risk Management Process

TIR36 focusses on identifying potential risk with Impact on Patient Safety, Product Quality, and Data Integrity and demonstration of compliance to an FDA regulation or ISO standard.

It also specifically addresses the risk controls of software and the placement of risk control in the development life cycle and type of software .i.e. bespoke and off-the-shelf and the complexity of the system.

It identifies two methods of the analysis of software failure as Software fault tree analysis and failure modes and effects analysis

The TIR 36:2007 technical report refers to three major process which promote the safety and efficacy of the medical product as Risk of harm to humans, Regulatory risk and Environmental risk.

It also applies the concepts of ISO 14971 risk management process for managing risks.

Purpose of Risk Management

Risk management involves studying a system or process thoroughly to identify concerns or potential risks, analysing them, and developing strategies for mitigation and control of the risks. Risk mitigation does not mean altogether eliminating the activities that create the risk. It may instead result in the reduction of the risk to an acceptable level.

When identifying risks to a software system, it is important to know all the possible risks, the level of severity of each risk and all the potential consequences of each. The action steps to mitigate or control each risk are determined based on a thorough knowledge of all risks. This "preventative" approach to risk management allows software developers to finish projects within their expected timelines and budgets.

Projects with effectively managed risks also tend to produce better quality outputs, in addition to reduced costs and time.

Software risk management approaches assess risks during all the phases of software development, by integrating risk management practices along with the software development process.

As a result, in these approaches, the risk management models depend on the development process.

With increasing complexity in software development, organisations have realized the importance of risk management, because it helps in reducing the uncertainties involved in developing software, and decreasing the chances of project failures.

It is therefore imperative that software developers are aware of the risks and are knowledgeable about the nature of the risks. Risks to software systems should be considered throughout the process, rather than one part of the process.

A risk management training program for software developers should aim to teach developers to:

- identify risks (business, process and integrity),
- calculate risk probabilities for quantitative assessments and to set realistic bands for qualitative assessments,
- calculate quantitative and determine qualitative risk impacts,
- determine when applying qualitative versus quantitative assessment is appropriate,
- perform safety and hazard analysis of a software product,
- prepare and carry out risk mitigation, monitoring and management strategies.

Risks can be identified in all parts of a process. There are several risk models available to assist with risk identification and classification. These models may focus on the management of business risks rather than product risks. However, software products are critical and therefore developers should also be aware of methods used for determining risks to the safety and integrity of their products.

The taxonomy helps in providing with an instrument (questionnaire) to elicit different classes of risks such as Requirements risks, Design risks, Coding and testing risks, Contract risks, Resource risks.13

Taxonomy of Software Development Risks

The taxonomy may be used to classify various factors relating to software development such as development tasks, quality procedures, and sources and consequences of risk.

It is left to the organization to decide on how to establish the classification.

The definitions used in this technical paper have been obtained from Taxonomy Based Risk Identification (CMU/SEI93-TR-006, ADA266992). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

- Product Engineering
- Development Environment
- Program Constraints

Product Engineering

This approach to software risk classification involves defining the software engineering activities and all the possible risk associated with each of the activities is established.

These activities include system and software requirements analysis and specification, software design and implementation, integration of hardware and software components and software and system test.

Components of each of the activities are further broken down into quality, technical, safety and other attributes that may have an impact on the software development activities and putting the necessary controls in place.

The diagram below details the product engineering classification concept:

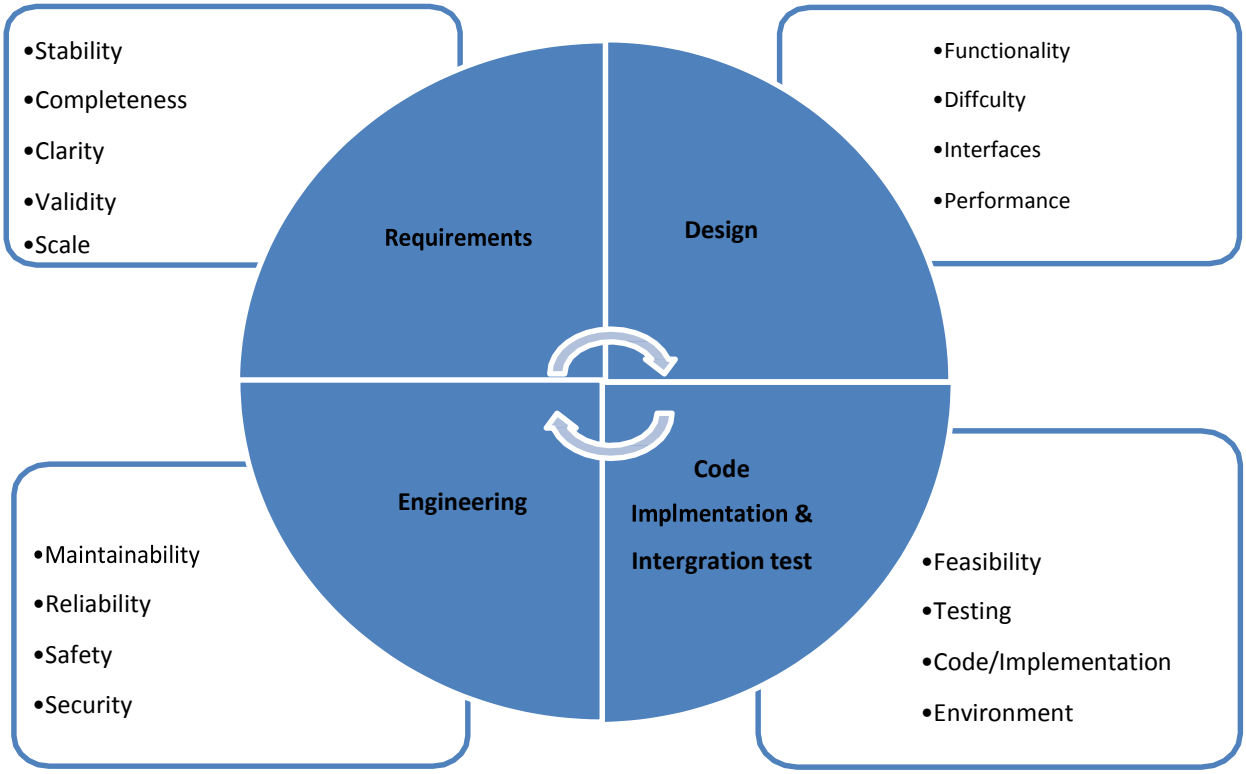

Figure 1 Product Engineering Classification Concept

Development Environment

This approach to software risk classification involves defining the activities of the project environment and the process used to develop a software product. The environment includes the development process, development system, management methods and work environment.

Risk of each of the activities and process is then grouped as necessary and risks controls are put in place to accomplish project goals.

The diagram below details the program engineering classification concept.

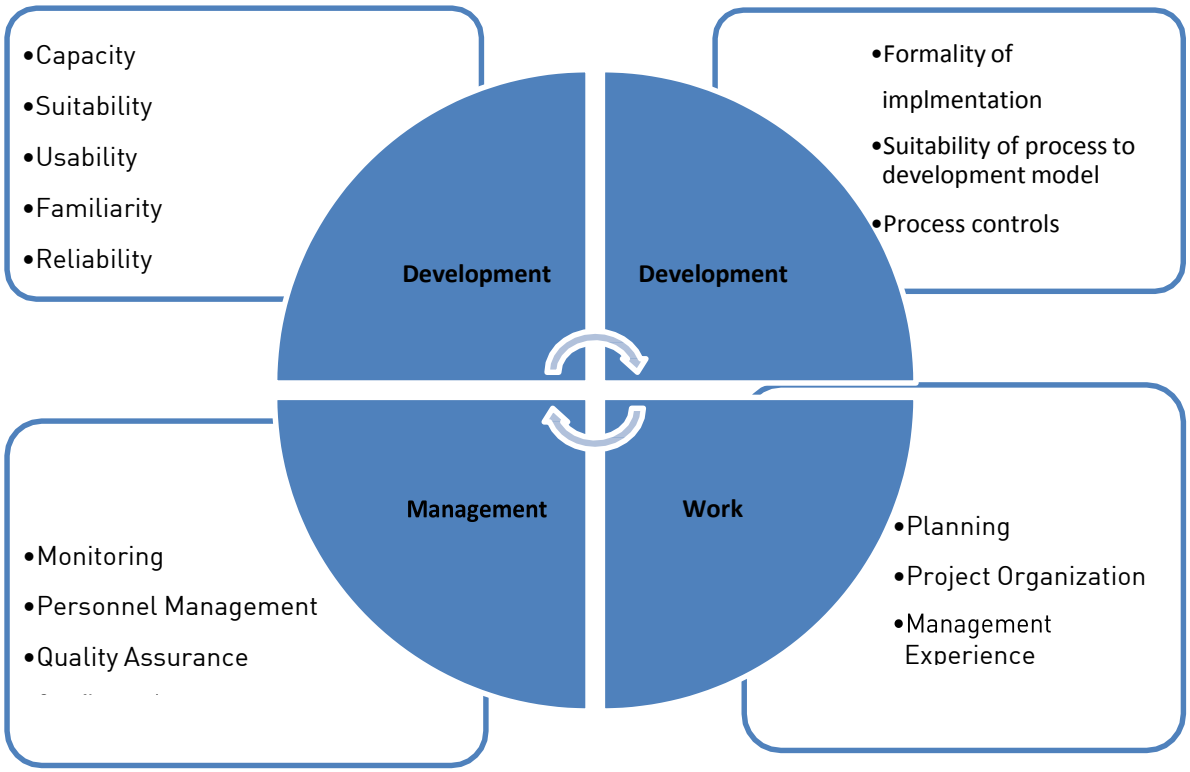

Figure 2 Development Environment Classification Concepts

Program Constraints

This approach to software risk classification involves defining external risk factors. These are factors that may be out-side the control of the project but can still have major effects on its success or constitute sources of substantial risk. These external factors include resources, contract and program interfaces. The risk attributes that may impact on the quality of the software development are then defined and control measures are put in place.

Summary

The ultimate objective of the risk management process is to identify and control risk associated with software development and implementation as soon as possible.

Organisations should establish and implement risk management models that are aimed at achieving project quality objectives.

In order to identify the risks associated with software development and implementation the project quality expectations essentials are to be defined.

The software development model, environment, type of software i.e. bespoke or off-the- self all influence risk management outcomes.

Software audits may be required for off-the-self software focusing on the project quality expectations prior to implementation.

The software development tools, and program language would all have an impact on the quality of software developed hence all the risk attributes associated with these elements needs to be evaluated and have control measures adopted as part of the quality assurance process.

The process of risk management is continuous and applies throughout the systems life- cycle by identifying system vulnerabilities, assigning probability and impact and determining reasonable mitigating strategies.

Terms	Definitions
Activity	A set of one or more interrelated or interacting tasks
Anomaly	Any condition that deviates from the expected based on requirements specifications, design documents, standards, etc. or from someone's perceptions or experience.
Architecture	Organizational structure of a system or component
Change request	A documented specification of a change to be made to a software product
Configuration item	Entity that can be uniquely identified at a given reference point
Deliverable	Required result or output (includes documentation) of an Activity or Task
Evaluation	A Systematic determination of the extent to which an entity meets its specified criteria
Harm	Physical injury, damage, or both to the health of people or damage to property or the environment
Hazard	Potential source of harm
Medical Device	Any instrument, apparatus, implements, appliance, implant, in vitro reagent or c`alibrator, software, material or other similar or related article, intended by the manufacturer to be used, alone or in combination, for human beings for one or more specific purpose(S) of diagnosis, prevention, monitoring, treatment or alleviation of disease, compensation for an injury, investigation, replacement, modification, or support of the anatomy or of a physiological process.
Medical Device Software	Software system that has been developed for the purpose of being incorporated into the Medical Device being developed or that is intended for a use as a Medical Device in its own right.
Process	A set of interrelated or interacting activities that transform inputs to outputs
Safety	Freedom from unacceptable risk
Security	Protection of information and data so that unauthorised persons or systems cannot read or modify them and so that authorised persons or system are not denied access to them.

Serious Injury	Injury or illness that directly or indirectly: • Is life threatening, • Results in permanent impairment of a body function or permanent damage to a body structure, or • Necessitates medical or surgical intervention to prevent permanent impairment of a body function or permanent damage to a body structure
Software development life cycle model	Conceptual structure spanning the life of the software from definition of its requirements to its release for manufacturing, which; • Identifies the PROCESS, ACTIVITIES and TASKS involved in development of a SOFTWARE product, • Describes the sequence of and dependency between ACTIVITIES and TASKS, AND • Identifies the milestones at which the completeness of specified deliverables is verified.
Software Product	Set of computer programs, procedures, and possibly associated documentation and data
Software System	Integrated collection of Software items organized to accomplish a specific function or set of functions
Software unit	Software unit that is not subdivided into other units
System	Integrated composite consisting of one more of the processes, hardware, software, facilities, and people, that provides a capability to satisfy a stated need or objective.

References

McManus J. 2004. Risk Management in Software Development Projects, Elsevier.

Glutch, D. P. 1994. A Construct for Describing Software Risks, Technical Report

Report CMU/SEI-94-TR-14, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Leishman, T. R., and VanBuren, J. 2003. The Risk of Not Being Risk Conscious: Software Risk Management Basics, STSC Seminar Series, Hill AFB, UT.

McCandless M. (1998) Staying Healthy in a Wired World, IEEE Intelligent Systems, Jan/Feb 1998, pp. 2 -3

Anderson R. J. (2000) Privacy Lessons from Healthcare, S&P 2000. Proceedings of 2000 IEEE Symposium on Security and Privacy pp. 78 -79

Smith E. & Eloff J. (2000) Cognitive Fuzzy Modeling for Enhanced Risk Assessment in a Health Care Institution, IEEE Intelligent Systems, March-April 1998, pp. 69-75

Heard S., Grival T., Schloeffel P. & Doust J. (2000) The benefits and difficulties introducing a national approach to electronic health records in Australia. Report to the Electronic Records task Force

Devanbu, P.; Fong, P.W.-L.; Stubblebine, S.G. (1998) Techniques for trusted software engineering. Proceedings of the 1998 International Conference on Software Engineering, pp. 126 –135

Boehm B. (2000) Requirements that Handle IKIWIS, COTS, and Rapid Change. IEEE- Computer, 33:7 pp. 99

Platt, A-B. (1999) The usability risk. Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, pp. 396 –400

Kornbluh, Ken (2000) Technical Software, IEEE Spectrum, 37:1, pp. 58 – 62

McDermid, J.A. (2000) Complexity: concept causes and control. Proceedings. Sixth IEEE International

Foo, S. -W. and Muruganantham, A. 2000. "Software Risk Assessment Model". Proc. of the 2000 IEEE International Conference on Management of Innovation and Technology, 2, 536-544

Fuller, A, Croll P & Dei, L, A new approach to teaching software risk management with case studies, pp. 6

Sources

Links used within this document are prone to change. Please refer to the appropriate source for the most recent information. We endeavour to keep an up-to-date record of information at www.pharmout.net

PharmOut is an international GMP consultancy serving the Pharmaceutical, Medical Device and Veterinary industries. PharmOut specialises in PIC/S, WHO, United States FDA, European EMA, and Australian TGA GMP consulting, engineering, project management, training, validation, continuous improvement and regulatory services.

Our team includes international GMP experts who have previously held leadership roles within regulatory bodies.

For more information please visit www.pharmout.net or contact us at info@pharmout.net.

PharmOut Pty Ltd, ABN: 85 117 673 766, Unit 10, 24 Lakeside Drive, Burwood East, Victoria 3151.
Ph: +61 3 9887 6412, Fax: +61 3 8610 0169, Email: info@pharmout.net Web: www.pharmout.net