

# Analysis of Gold Plating: A Software Development Risk

Kirandeep Kaur<sup>1</sup>, Bharti Jyoti<sup>2</sup>, Rekha Rani<sup>3</sup>

<sup>1,2,3</sup>Department of Computer Science, KCCEIT, Nawanshahr, (Punjab), India

<sup>1</sup>Er.kiran2007@gmail.com, <sup>2</sup>bharti88.kc@gmail.com, <sup>3</sup>rekh50@gmail.com

**Abstract:** Software Engineering is a profession to provide high quality software products to its customers. It is an application of systematic, disciplined approach to development, operation, maintenance of software. But there are several risks which influence the quality software like Unrealistic time and cost estimates, Developing the wrong software functions, Developing the wrong user interface, Late changes to requirements, gold plating etc. Risk means the possibility of suffering harm or loss. They are based upon a uncertainties. Gold plating is one of the important risk factor provided by Barry Boehm. It is assumed that gold plating is a major risk which should be analyzed before adding new features to project.

**Keywords** - Software Engineering, Software Risk, Gold

**Plating.**

## 1. Introduction

Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. Software consists of 7 phases and these phases are called Software Development Life Cycle. Fig 1 shows the SDLC:

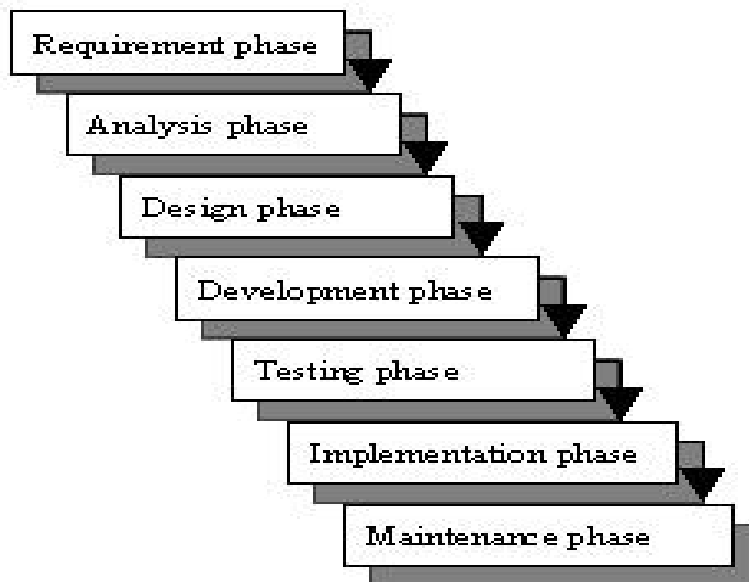


Fig: 1 Software Development Life Cycle.

This is also known as Classic Life Cycle Model. This research work is based on the SDLC model's phase which is related to develop a new methodology to solve risk

management problems with the purpose of determining the project's attractiveness

**Table 1:** Software Development Phase

<b>Software Development Phase</b>	<b>Description</b>
<b>Feasibility study</b>	Feasibility is the analysis of risks, costs & benefits relating to economic, technology, & user organizations.
<b>Requirements Analysis</b>	The elicitation, analysis, specification, and validation of <u>requirements for software</u> . Analyzes end-user information needs. Important activities are: <ul style="list-style-type: none"> <li>• Completing the requirements analysis.</li> <li>• Delimiting the problem domain.</li> <li>• Roughly sketching the components of the target system.</li> <li>• Feasibility study of the planned project.</li> <li>• Creating a rough project schedule.</li> </ul>
<b>System Specification</b>	System Specification is a contract between the client & the software producer.
<b>Design</b>	The process of defining the architecture, components, interfaces, and other characteristics of a system or component. It is also defined as the result of that process. Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.
<b>Coding</b>	The real code is written here.
<b>Testing</b>	The dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite executions domain, against the expected behavior. Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
<b>Acceptance</b>	The final stage of initial development, where the software is put into production and runs actual business

## 2. Software Risk

A “risk” is a problem that could cause some loss or threaten the success of your project, but which hasn’t happened yet. These potential problems might have an adverse impact on the cost, schedule, or technical success of the project, the quality of your products, or team morale. Risk analysis has in its essence uncertainty and impreciseness. Any analysis made ignoring this uncertainty and impreciseness may cause information to be seriously misleading, therefore, contributing to large mistakes. Risk management is the process of identifying, addressing, and controlling these

potential problems before they do any harm. There is need of formal risk management. A formal risk management process provides multiple benefits to both the project team and the development organization as a whole. First, it gives us a structured mechanism to provide visibility into threats to project success. By considering the potential impact of each risk item, we can focus on controlling the most severe risks first. We can carry risk assessment with project estimation to quantify possible schedule slippage if certain risks materialize into problems. This approach helps the project manager generate sensible contingency buffers.

Sharing what does and does not work to control risks across multiple projects helps the team avoid repeating the mistakes of the past. In order to successfully manage software projects, we must learn to identify, analyze and control software risks. Although controlling risks have a cost, but if the risks are not addressed and does indeed bite us. Almost everything we do in today's business world involves a risk of some kind: customer habits change, new competitors appear, and factors outside your control could delay your project. But formal risk analysis and risk management can help you to assess these risks and decide what actions to take to minimize disruptions to your plans. They will also help you to decide whether the strategies you could use to control risk are cost-effective.

### 3. Gold Plating

Gold plating means when you are Adding extra feature in your product to delight your customer (a kind of surprise). Gold plating is not a bargain. It can increase operation and maintenance costs and reduce quality. Gold plating in software engineering refers to continuing to work on a project or task well past the point where the extra effort is worth the value it adds (if any). After having met the requirements, the developer works on further enhancing the product, thinking the customer would be delighted to see additional or more polished features, rather than what was asked for or expected. The customer might be disappointed in the results, and the extra effort by the developer might be futile. The factors are given:

**Table 1: Factors**

S. No.	Factors Name
1.	Personnel shortfalls
2.	Unrealistic time and cost estimates
3.	Developing the wrong software functions
4.	Developing the wrong user interface
5.	Gold plating
6.	Late changes to requirements
7.	Shortfalls of external supplied components
8.	Shortfalls of externally performed tasks
9.	Real-time performance shortfalls
10.	Straining science capabilities

### 4. Conclusion

In Software Engineering, researchers are still working. There are several parameters of risks, which can have great impact on adverse impact on the development of software. The important parameters can be efforts in terms of time, cost and resources. Gold plating is one of the important risk factor provided by Barry Boehm. It is assumed that gold plating is a major risk which should be analyzed before adding new features to project. So, we must balance this cost against the potential loss we could incur if we don't address the risk and it does indeed bite us.

### References

- [1] Barry Boehm, "Software Risk Management: Principles and Practices", IEEE Software, Vol. 8(1), pp: 32-41, 1991.
- [2] T. Addison and S. Vallabh, S. "Controlling Software Project Risks –an Empirical Study of Method used by Experience Project Managers", Proceeding of SAICSIT, pp.128-140, 2002.
- [3] Mark C. Paulk, "Factors Affecting Personal Software Quality", the Journal of Defense Software Engineering, pp: 9-13, 2006.
- [4] Shradhanand, Amarjeet Kaur, Dr. Satbir Jain, "Use of

*fuzzy logic in software development*”, Issues in Information Systems, Vol. 8 (2), pp: 238-244, 2007

[5] Tharwon Arnuphaptrairong, “*Top Ten Lists of Software Project Risks: Evidence from the Literature Survey*”, Proceeding of international multi-conference of engineers & computer scientists, Vol. 1, ISSN: 2078-0966, 2011.

[6] M.H. Zirakja, R. Samizadeh, “*Risk Analysis in E-commerce via Fuzzy Logic*”, Int. J.Manag. Bus. Res., Vol. 1(3), pp: 99-112, 2011.

[7] Xizhu Mo, Xiangzhou Zhang, Yuran Zeng, Jianfeng Du and Kang Xie., “*Intelligent Analysis Model for Outsourced Software Project Risk Using Constraint-based Bayesian Network*”, Journal of software, Vol. 7(2), pp: 440-449, 2012.