

# Categorization of Software Release Risks and Its Abatement Strategy

Arti Rana<sup>1</sup>, S. P. Singh<sup>2</sup>, Rachna Soni<sup>3</sup>, Ashish Jolly<sup>4</sup>

<sup>1</sup>Amity Institute of Information Technology, Amity University, Noida (UP), India

<sup>2</sup>Birla Institute of Technology, Extension Centre, Noida, India

<sup>3</sup>Department of Computer Science, DAV College for Girls, Yamuna Nagar, India

<sup>4</sup>Department of Computer Science, Govt. College, Bherian (Pehowa), Haryana, India

Email: [artirana@yahoo.com](mailto:artirana@yahoo.com)

Received 20 September 2014; revised 15 October 2014; accepted 9 November 2014

Academic Editor: Yashwant K. Malaiya, Colorado State University, USA

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Growing competition in the software industry with the persistently changing needs and the usual problems associated with software release, which have made acceptance of a new software in market, are extremely important for the success. Volatility in the software developmental processes is generally difficult to handle. The change request at any arbitrary point of time leads to the inevitable change and rework request. The software release process which broadly includes all the process that starts after the completion of development till the final deployment. This complete phase is exposed to various risks which may hamper the final result. This paper presents threat associated with software release activities and their possible mitigation and exploring the role played by the change management in controlling or reducing those risks. For the effective survival in ever changing software industry needs, Software Release Management takes a holistic view of the change and configuration relationship and work on the improvement strategies for the effective release with zero defect potential.

## Keywords

Software Release Management, Change Management, Release Deployment

---

## 1. Introduction

The release management is a link between software development and its final deployment. The software release

management handles the continuous pressure to have the release plan ready as early as possible and release processes including proceeds with progressions all through the release cycle as indicated by the unpredictability of accessibility of human asset and sudden changes in the technology or requirements because of moving business necessities. If not overseen prudently and timely, these repeating progressions can prompt a challenging relationship between release management and software development process prompting to sub-ideal choice making [1].

The major objectives of the release management may include:

- To plan deployment of software on time;
- To give due thought to the standpoint of the customer throughout the planning and rollout of new release;
- To ensure that software changes are traceable;
- To make an effective utilization of available resources to optimize the cost;
- To ensure that proper measures are taken to acquaint the customer with usage of new software;
- To design a proper rollout plan for the effective software release;
- To ensure the availability of back out plan in case of failure of new release for the smooth running and ensure stability;
- To be aligned with the change management and configuration management [2].

## 2. Software Release Process Guidelines

Release planning is an extremely unpredictable issue as it incorporates the alternate points of view of diverse stakeholders, contending goals and having distinctive types of constraints [3] [4]. In the software having numerous of volatile features which are associated to each other with the precedence and/or coupling constraints between them. Moreover, resources, effort and budget constraints must be fulfilled for each one release. The general objective is to discover a “most guaranteeing” release plans such that the general worth and the level of fulfilment of distinctive stakeholders is to be augmented and produce feasible and valuable release plans according to the current business necessities inside time, plan and budget:

- 1) Identify the requirements;
- 2) Prioritize the multifaceted requirements and stakeholders;
- 3) Clustering of similar requirements;
- 4) Identification of constraints;
- 5) Extent of stakeholders’ involvement;
- 6) Time and cost;
- 7) Identification and abatement planning of Risks: Consideration of risk should be part of the planning process. Although hard to quantify, risk consideration helps to generate plans that are more likely to be feasible and operational;
- 8) Planning of back out plan.

## 3. Activities of Release Management

### 3.1. Release Planning

Planning a release includes designing release schedule for the entire project life cycle taking into consideration all the constraints and dependencies and once the plan is designed then gaining the consent on release content. Release planning outputs include the plan for a particular release and acceptance criteria for the release. Software release management should work in association with the change management to meet the changing needs of the stakeholders from the new release.

### 3.2. Design, Build and Configuration of New Release

All the design procedures should be planned and properly documented for building release, the proper documentation helps in reusing defined procedures where possible. A configuration of a particular release of software may be some of which may be developed in-house and others bought in or the combined approach. All software constraints, parameters, test data and all the required software needed for the release should be under the control of Configuration Management Control. To ensure quality, various quality control checks should be performed on every phase of release of software. A complete record of the build results should be stored in the

CMDB. Master details of the software installation and related instructions, documentation must be maintained in the DSL.

### 3.3. Release Acceptance

The testing of the release before deployment ought to be performed completely. The back out methodology ought to additionally be tried as a piece of the testing action to guarantee the preparation of the reinforcement arranges if there should be an occurrence of disappointment. There ought to be a documentation of each one close down phase of testing. The deciding after-effects of the acknowledgement action should be an improvement on the overall software release. The end result of the acceptance activity should be a sign-off on the completeness and accuracy of the whole release.

### 3.4. Rollout Plan

Rollout plan provides the details of complete release plan produced so far for exact installation process developed and the agreed implementation plan. Rollout planning deal with detailed schedule of release mentioning who will do, how to do and when to do.

### 3.5. Communication, Preparation and Training

The clients support staff & connection building staff, need to recognize what is arranged and how it is to completed for guaranteeing its acceptability by the clients, which is trailed by the preparation session for the client and hand on experience to them by demonstrating learning period to them. The outputs of this phase should comprise of the final versions of the software for the user and supporting documentation and training material.

### 3.6. Distribution and Release Deployment

Distribution of the software release from the build environment to the controlled test environment and then into the live. After deploying software, it is essential to check that the release ensures the acceptance of the stakeholders and considering the post release defects [4] [5].

**Table 1** summarizes all the software release phases with their sub activities based on the study of past release literature.

## 4. Software Risk Management

A software releases could be influenced by the substantial assortment of risks. With a specific end goal to comprehend risks and arrangement for its reduction it is essential to recognize the critical risks which may influence the software releases and sorted risks into distinctive classes. The release manager can then inspect the risks and recognize the danger connected with distinctive periods of release procedure [6].

The risk might be extensively arranged in two primary classes which can influence a software project.

### 4.1. Project Risks

Project risk concern is risk danger of calendar slippage. Since software is a non physical, conceptual created by a gathering of individuals located at different places. As it is produced by engineers placed at different places dealing with distinctive modules, emulated by testing lastly release by the different teams. So it is extremely hard to control something which can't be seen through and through. The imperceptibility of the product being produced is an important reason behind why numerous software projects experience the ill effects of the risk of timetable slippage.

### 4.2. Technical Risks

Technical risks are identified with software outline, execution, interfacing, testing, release and post release maintenance issues. Technical risks likewise incorporate vague particular, partial determination, fluctuating specifications and technical obsolescence. Most Technical risks happen because of the absence of endeavors by project teams on different aspects of advancement and release and eventually simply focus on one aspect and

**Table 1.** Summary of Software Release activates at various phases.

Major Steps of Software Release	Sub Steps
Release planning	Determine objectives and constraints
	Determine stakeholders
	Determine schedules & baselines
	Determine available resources human & technology
Design, build and configuration	Redefine model and process design
	Improve user interface design
	Formally specifications of design
User acceptance testing	Independent release testing
	Guarantee the working of back out arrangement in the event that crisis
	Guarantee accuracy of the release
Rollout, communication, preparation and training:	Creating detailed timetable of events and who will do what
	User and support training
	Manuals
Release deployment	Final release product
	Track defects
	Characterize enhancements
	Iterate back to repair and enhance
Validate release	Determine the product is same as craved

leaving different aspects totally [7].

### 4.3. Risk Assessment

The risks are assessed on the premise of their probability of event and harm it causes. The probability of a risk working out as expected (signified as RS). The result of the issues connected with that risk (meant as CS).

In light of these two elements, the necessity of each one risk might be registered:

$$R = RS * CS$$

This gives the R priority with which the risk is to be taken care of associated with the probability of event and level of outcomes brought on by the risk. In the event that all recognized risks are prioritized, then the no doubt and harming risks might be taken care of first and more extensive risk reduction systems could be intended for these risks [8].

### 4.4. Risk Abatement

After all the recognized risks of a task are evaluated, plans must be made to hold the most harming and the undoubtedly risks. Distinctive risks require diverse abatement techniques. Actually, most risks oblige creativity from the project manager in handling the risk.

There are three fundamental systems to anticipate risk restraint:

**Evade the risk:** The risk can be avoided by considering the customer as a prime; all the changes proposed by the user must be implemented as desired. The proper incentive schemes should be provided to the engineers to avoid end time manpower shortages at time of completion. There should be good employee retention policies.

**Transfer the risk:** This can be done by hiring specialized skilled people to avoid any kind of discrepancy between requirement and implementation, the critical components where possible should be outsourced to

transfer the risk to third part, buying insurance cover, etc.

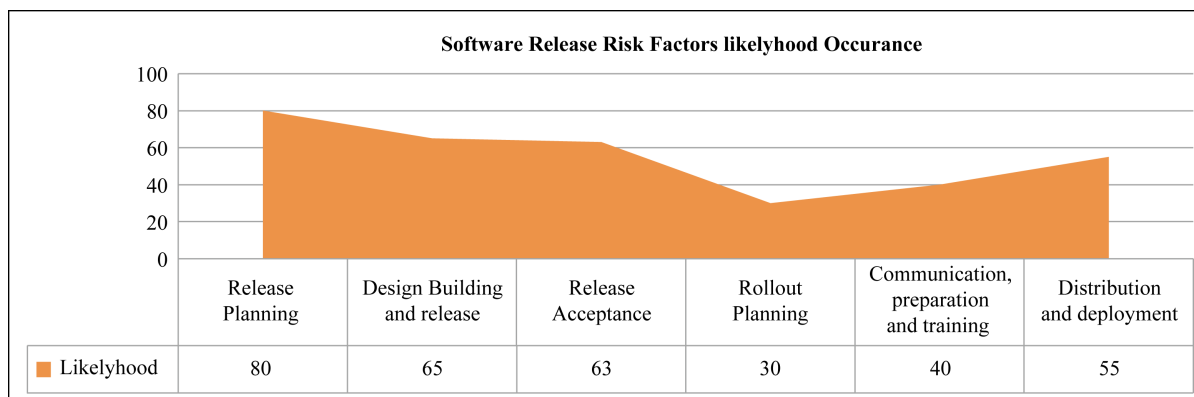
**Risk reduction:** The risk reduction should be ensured by maintaining all the check at every sing off, maintain the effective backup plans to meet up the worse situations in case of emergency [9]-[15].

## 5. Risk Abatement Strategy

Following is a table considering all the software release activities along with the risks involved and its mitigation strategy. The investigation of the past papers and dialog infers that the significant likelihood of the fall in Release Planning, as this phase is the base of the entire process of release and has an impact over the complete release process. This could be quantified below in **Table 2** on the study premise.

**Table 2.** Risk & mitigation factors of Software Release at Various Phases.

SR Activity	Risk Factor	Likelihood	Mitigation
Release planning	Produce the accurate schedule of various phases and baseline results. Release does not work as the customer expects. Due to extended timeline, requirements changes lead to problem in understanding the customer requirements.	80%	Acceptance criteria linked to requirements assessment. Increase customer involvement throughout use version control tools.
Designing and building a release	Release design validation against the requirement for the new or changed service offering. Selection of a suitable release mechanism for the change. Prioritize the issues and actions to ensure that they can be resolved in a timely manner test where the release package delivers the change effectively in line with requirements.	65%	Aligned with change and configuration management. Use of automation tools. Increased stakeholder interaction through involvement or feedback.
Release acceptance	Release meets the requirements in regard to operability, service and product.	63%	User acceptance tests aligned with change management.
Rollout planning	Staff reluctance to adapt changes. Less weight age over the documentation.	30%	Sign-off documentation for each phase. Software traceability. Use of automation tools.
Communication, preparation and training	Lack of timely training, poor & inaccurate communication. Change in business processes is not completely tended to by user training bring on unsuccessful execution.	40%	Provide training prior to implementation. Remove employee reluctance to adapt new system by making them aware of the need of change.
Distribution and deployment	Timely deployment.	55%	Ensure the acceptance testing before installation. Recording any incidents, unexpected events, issues or deviations from the plans. Ready the backup plan in case emergency.



**Graph 1.** Software Release Risks Representation.

## 6. Conclusion & Future Scope

Change management plays a vital role in handling the rework requests at any arbitrary point in time as the software development processes generally cannot handle it. The software developmental processes are exposed to certain risks which may hamper the end result. Here common software release activities and their associated risks while exploring mitigation strategies in controlling or reducing those risks are stressed upon. The risk in release planning and release acceptance is high as shown in **Graph 1**, so the user confidence is built through mitigation strategies. It is observed that change management is aligned with configuration management for better release building. Further release risk mitigation contingency plan can be developed for the cost effectiveness using stochastic system modeling.

## References

- [1] Penny, D.A. (2002) An Estimation-Based Management Framework for Enhance Maintenance in Commercial Software Product. *Proceedings of the international Conference on Software Maintenance (ICSM 02)*, Washington DC, 2002, 122-130.
- [2] IBM Software Thought Leadership (2013) 7 Proven Practices to Strengthen Release Management. IBM Corporation, USA.
- [3] Greer, D. and Ruhe, G. (2004) Software Release Planning Software Release Planning: An Evolutionary and Iterative Approach. *Information & Software Technology*, **46**, 243-253.
- [4] Ruhe, G. and Sailu, M.O. (2005) Art and Science of Software Release Planning. *IEEE Software*, **22**, 47-53.  
<http://dx.doi.org/10.1109/MS.2005.164>
- [5] Sailu, M.O. and Ruhe, G. (2005) Supporting Release Planning Decision for Evolving Systems. *Proceedings of 29th Annual NASA Software Engineering Workshop*, Washington DC, 6-7 April 2005, 14-26.
- [6] Curtis, P. and Carey, M., Thought Leadership in ERM (2012) Risk Assessment in Practice. The Committee of Sponsoring Organizations of the Treadway Commission (COSO), USA.
- [7] Shahzad, B. and Al-Mudimigh, A.S. (2010) Risk Identification, Mitigation and Avoidance Model for Handling Software Risk. 2010 *Second International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, Liverpool, 28-30 July 2010, 191-196.
- [8] Karlsson, J. and Ryan, K. (1997) A Cost-Value Approach for Prioritize Requirements. *IEEE Software*, **14**, 67-74.  
<http://dx.doi.org/10.1109/52.605933>
- [9] Carlshamre, P., Sandahl, K., Lindvall, M., Rengnell, B. and Noch Dag, J. (2001) An Industrial Survey of Requirements Interdependencies in Software Product Release Planning. 5th *IEEE International Symposium on Requirements Engineering*, Toronto, 27-31 August 2001, 84-91.
- [10] Jung, H.W. (1998) Optimizing Value and Cost in Requirements Analysis. *IEEE Software*, **15**, 74-78.  
<http://dx.doi.org/10.1109/52.687950>
- [11] Lindren, M., Nordstrom, C., Wall, A. and Land, R. (2008) Importance of Software Architecture during Release Planning. *Seventh Working IEEE/IFIP Conference on Software Architecture*, Vancouver, 18-21 February 2008, 253-256.
- [12] Wohlin, C. and Aurum, A. (2005) What Is Important When Deciding to Include a Software Requirement in a Project or Release? 2005 *International Symposium on Empirical Software Engineering*, Queensland, 17-18 November 2005, 10.
- [13] Li, Q., Yang, Y., Li, M.S., Wang, Q., Boehm, B.W. and Hu, C.Y. (2010) Improving Software Testing Process: Feature Prioritization to Make Winners of Success-Critical Stakeholders. *Journal of Software: Evolution and Process*, **24**, 783-801.
- [14] Wall, A. (2008) Key Aspects of Software Release Planning in Industry. 19th *Australian Conference on Software Engineering*, Perth, 26-28 March 2008, 320-329.
- [15] Lu, Y.K. (2012) An Information Systems Design Product Theory for Integrated Requirements, Test and Defect Management Systems. 47th *Hawaii International Conference on System Sciences*, Maui, 4-7 January 2012, 5516-5525.



Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either [submit@scirp.org](mailto:submit@scirp.org) or [Online Submission Portal](#).

