

# An Approach on Software Readiness Workflow with Change Impact Analysis (SRW-CIA)

Sathyarajasekaran k, Ganesan R

**Abstract :** *Software readiness means to say that the software is to be up to date to the current technology. The readiness determines the application needs to handle issues such as getting recent updates and patches and at the same time determining the working conditions and looking into any vulnerability that affects the software. Change Impact Analysis (CIA) plays a vital role in enhancing the system to adapt to the latest updates without any issues when it's been deployed after accepting the change. Software readiness is well supported by Change Impact Analysis by predicting the changed behavior on the existing system through the impact analysis of the change so when the change is been accepted through the impact analysis study the software won't be having any vulnerability when it's updated by the change. This paper depicts the workflow and management of periodical change and impact of the software life cycle which includes deployment and compliance with the software patches used. This approach can integrate the impact analysis over the software readiness to provide a best possible way to provide change workflow.*

**Keywords:** *Impact analysis, software readiness, change workflow,*

## I. INTRODUCTION

Over the period any software requires a change or upgrades to best suit the technical needs. Application assessment and rationalization give a clear picture of software readiness as well as compliance[1]. Change Impact Analysis is a growing technology in today's era, it saves the project especially the software's when they are put upon on a minor change or major change in the requirements directly or any other supporting modifications [12]. Impact analysis plays a major role and helps as a supporting index for the project which has deployed and in service when a change is been requested and the software is been in service the change can't be performed as it is, as it may cause a major break down in the system. The change has to be analyzed and the environment of software service is to be validated before adopting or updating the software system. CIA helps in analyzing the impact caused by the change and through that analysis, the software acceptance over the change and its reaction of other functionalities in the system can be studied which helps in determining the change won't disturb or create an unusual problem to the system[13]. Same time when the change analysis study been performed an update or patch to existing system is to be made means the

**Revised Manuscript Received on May 06, 2019**

**Sathyarajasekaran K**, School of Computing Science and Engineering, VIT University, Chennai, India.

**Ganesan R**, School of Computing Science and Engineering, VIT University, Chennai, India.

time of patch update over the software environment has to analyzed as the system is already deployed and it is in service means the update to be made over the system is to planned so that it won't create any problem during the service and at the same time the update can't be put hold for a long time it has made within a time span.

Over the concept of lifecycle management, we are upgrading the software to latest patches which makes the environment compliant and overcome the vulnerability this brings new challenges and impact that need to be foreseen and mitigated if not documented for failover or fallback. CIA helps in managing the change in the system through its entire software lifecycle and with a different form of impact analysis study it paves the environment condition of the software service and delivers the time of undergoing the update to accept the service.

This paper address the readiness of the software with the impact analysis through which deployment can be done with the change workflow mechanism discussed in this work.

## II. BACKGROUND STUDY

Readiness is essential to confirm correct operations before software is free to the market or integrated into the products. Operational readiness refers to an idea that tries to quantify the "probability that, at any purpose in time, the system is prepared to be placed into operation on demand once used beneath explicit conditions" [1]. Since Gibson and Nolan (1974) initial introduced a four-stage maturity model, [2] more assessment models are developed over the years: the aptitude maturity model (CMM) [3], the ISO/IEC 15504 norm for package method improvement and capability determination, a defect trailing methodology mistreatment prophetic modeling approaches or the IT Capability Model Framework (IT-CMF) [4]. Deciding the readiness of package "requires continuous and customized measurement". The aim of software readiness is that the identification of a discrepancy, which can be resolved by ensuant improvement actions [5,6]. Common approaches to assess SR are checklists, business standards and academically developed methodologies [7]. Thereafter, procedures to predict Grieger et al. / Adding lightness to package Readiness Assessment Twenty-Sixth European Conference on info Systems in ECIS2018, Portsmouth, UK, 2018 four operational readiness are investigated and developed by researchers and practitioners within the style of package maturity in software maintenances [8], unleash readiness and package readiness for software modules [7]. All of those analysis streams examine the assessment of package by



## An Approach on Software Readiness Workflow with Change Impact Analysis (SRW-CIA)

sure criteria so as to support the choice between emotional package too early or too late. The terms “maturity” and “readiness” are used synonymously [9] notwithstanding, we have a tendency to use the term “software readiness” while not the intention to exclude relevant publications on “software maturity” or “release readiness”.

In the other hand, change impact analysis transform sway dissection foreseeing those possibility results of a recommended change; Furthermore, change proliferation actualizing a change by propagating transforms with administering consistency inside the software product [10]. Past effort need recommended a schema that backs change proliferation in the Development of agent- turned configuration models [11-13]. That skeleton need also been stretched out on managing different model sorts [14,15], endeavor construction models[16], or service- turned building design models[17]. The keep tabs for this paper may be looking into transform effect dissection about availability frameworks. Change impact examination [18] as a rule begins for the software product maintainer looking at those transform ask for furthermore figuring out the substances at first influence. Eventually, Tom's perusing the essential changes. The software maintainer then determines other substances in the framework that bring the possibility of reliance connections for those beginning ones and manifestations. The individuals affected parts also identify with different

substances Furthermore consequently those impact dissection proceeds this procedure until a complete effect situated is gotten. Change impact analysis assumes a major aspect of arranging and Creating the possibility of a transform as far as foreseeing the expense. Furthermore, the multifaceted nature of the progress the vitality of the progress effect dissection issue need to be prompted significant fill in around proposing particular impact analysis strategies. In spite of the fact that ideas and thoughts from an extensive group of work tending to progress effect dissection for traditional programming frameworks [18] might make adapted, genuine frameworks for their different qualities. This work addresses the software readiness framework which can guide the maintenance persons to keep track of their activities.

### III. SRW-CIF FRAMEWORK

Considering an IT organizations framework every resource or data is considered as a CI (Configuration item) each CI can be mapped under multiple architecture or domain. Getting into details with software CI often referred to as application or package is configured under the company's architecture. This package flow is narrated below which comprises of mainly 5 phases, starting from Application Request Initiation to Application Availability in production.

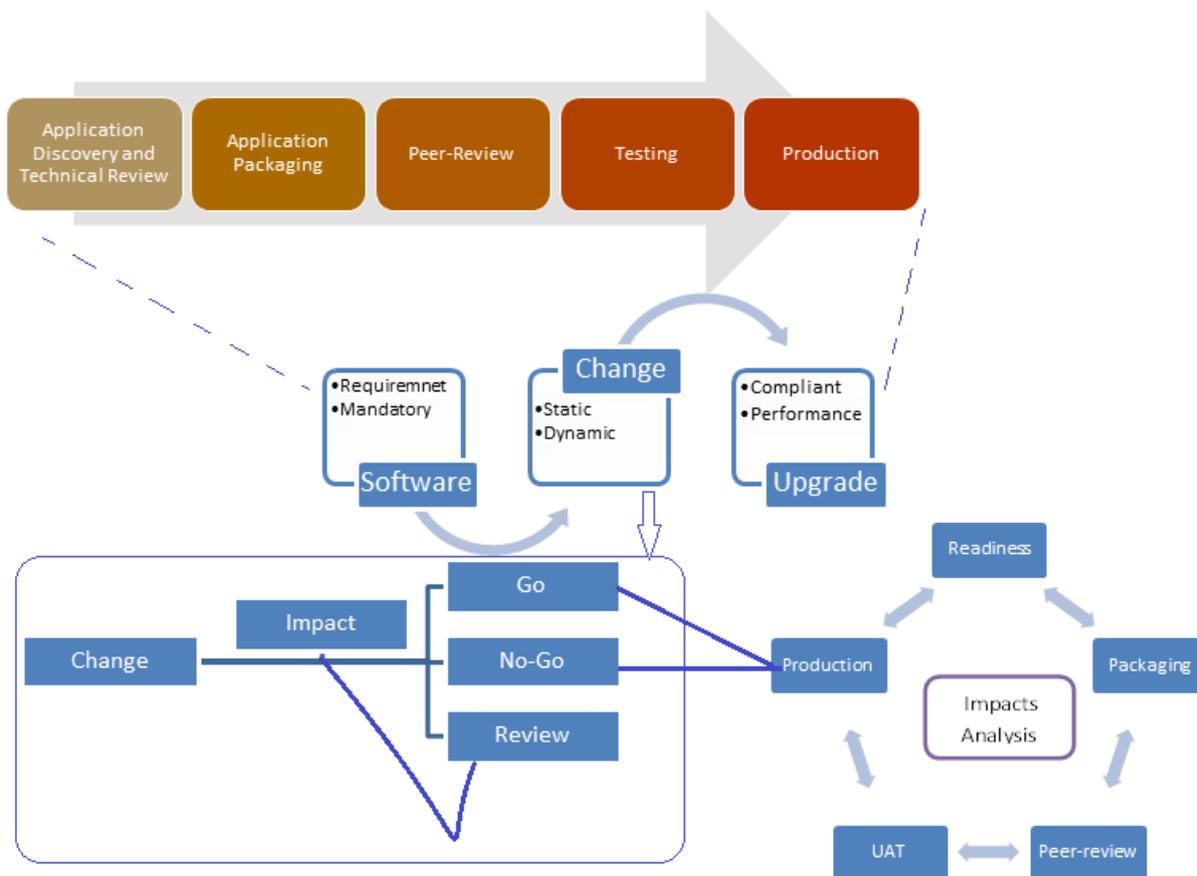


Fig 1. SRW-CIA Framework

**Discovery and Review:** Rationalizing software to remove unwanted files and folders

**Packaging:** Using packaging tools to create a single click executable

**Peer-Review:** Validating Company and best practices are followed without any conflicts

**Testing:** Verifying whether the package is behaving a source



**Production:** Ready for deployment

Once an application reaches the production phase it meets the end of the cycle and not to be concerned about changes. This will open a wide treat to the environment as well as affect the performance of the client. What we are trying to accomplish from the lifecycle method is continual improvement and upgrade using a change process and its impact.

Over the lifecycle process, we can calculate the merits and demerits throughout the cycle. The change comes with a risk factor this risk needs to be calculated with an impact value which needs to be thoroughly analyzed for which workflow need to be designed and approved with a backup/failover module

We should also consider the external factors such as cost involved, application functionality and conflicts with dependent modules.

**Step 1: Change can be classified to below categories**

- *Static change* - which can be termed as a pre-approved/governed process
- *Dynamic change* - requires on-demand fulfillment.

With a well-defined and organized process in place like a lifecycle module, we can analyze and estimate the impact and its back-up/fallback with a precision.

**Step 2: Change will go through CIA request which will be iterated and questioned on multiple levels with a fall back approach in case of any issues.**

Impact analysis can be classified with the below outputs:

- Go: ready for deployment
- No-Go: No change
- Review: with modification for latency approach

**Step 3: Software Readiness and deployment process**

Software deployment process goes through certain workflow which consists of multiple phases

This upgrade can be perceived in two ways, one way to bring the software to the latest patches being compliant and overcome the vulnerability. The second way is to comprehend the software with hardware/firmware with enhanced features and to unlock its full potential abilities.

**Sample Scenario1: Enterprise-wide software patched on demand**

Going with the above cases we have multiple scenarios with each and differing requirement. Let's talk about three of such requirement's which require a major contribution.

1. On-demand software update/install

This request can be to through any media or means of following

- 1.1 User requirement
- 1.2 Project requirement

1.3 Performance issue

2. Core apps to be upgraded: With growing demand, each software need to be revisited and updated with its latest precedes
- 2.2 Licensing
- 2.3 Performance
- 2.4 Regional value

**Sample Scenario2: Mandatory security patches and anti-virus updates to be deployed.**

To consider the case study about the virus in May 2017, an attack called WannaCry was able to infect and encrypt more than a quarter million systems globally. The malware uses asymmetric-encryption so that the victim cannot reasonably be expected to recover the (private and undistributed) key needed to decrypt the ransomed files [19]. One of the major entries was through a vulnerable protocol for Windows 7 which was exploited. The most server of the vulnerabilities could allow remote code execution if an attacker sends specially crafted messages to a Microsoft Server Message Block 1.0 (SMBv1) server.

To be the compliant and non-vulnerable environment, we need to bring systems up-to-date with the latest security/vulnerable patches. With on-demand request or business requirement we can bring the change to closure and over time, it will be faded without a track or reference.

**IV. FRAMEWORK VALIDATION WITH SURVEY APPROACH**

The practice areas of the ten interview participants included software industry (including analyst, developer, deployer and tester. Many of the participants represented multiple practice areas, however, for the purpose of this summary, only the primary practice area was noted. The analysis of the interviews identified the software readiness over impact analysis themes as the focused review. Study instrument given online will be utilized to accumulate poll on survey questions SQ1 – SQ7 to assemble information from the investigators, area specialists, engineers, and Testers. It utilizes a five-point scale with the accompanying classifications: "1" being "poor/not relevant" and "5" "critical" for the expressed goal. Result dialog of this methodology is given in the execution assessment of section 6.

This evaluation is based on the input from the different stack holders in the software life cycle. Based on their input, the following figure shows the survey results of proposed software readiness workflow using change impact analysis (SRW-CIA). Here, SQ1-SQ7 refers to the questionnaires' addressed by the stakeholders.



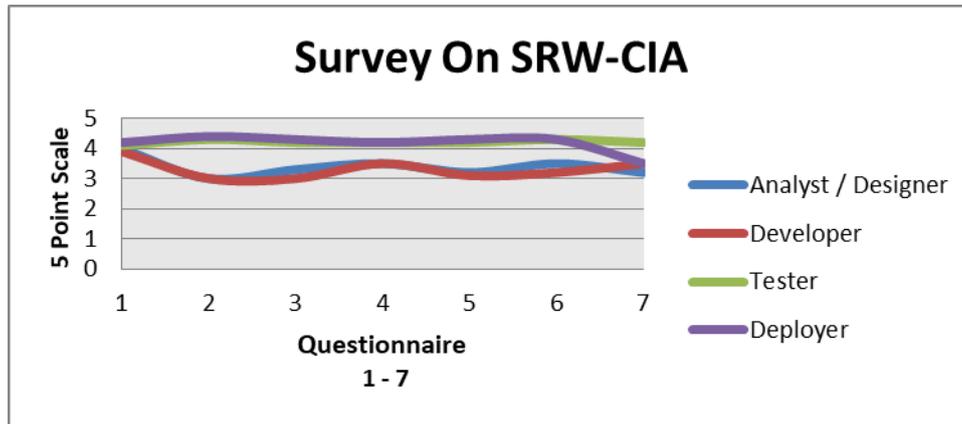


Fig 2. Result Analysis

A survey result of software readiness workflow using change impact analysis (SRW-CIA) is provided in Figure 2. This demonstrates that the association of impact analysis towards tester and analyst/designer are average because of their role primarily focuses on analysis and requirement changes in early stages. It ought to overcome comprehension of members and their own translation of the inquiry. In this module statistical examination of the input data can be implemented and verified using the survey method.

## V. CONCLUSION AND FUTURE DIRECTIONS

Any change will have a certain amount of impact that needs to be channeled with a systematic and failover approach. Considering the growing demand with technology and hardware requirement with potential features few risk factors to oversee for advancement and futuristic endeavor. On the other hand, few IT industries are still fall back on older settings which imposes a huge threat for the data and information loss. So, this framework can help to maintenance phase and useful for software readiness. This work can be extended to identify the software patterns for readiness and also neural network concepts can be used to assess the software packages along with the survey approach. Even more, agent-based workouts can be helpful to do further improvement.

## REFERENCES

- Kececioglu, D. (2003). Maintainability, availability, and operational readiness engineering handbook. Lancaster, Pa.: DEStech.
- Gibson, C. F., & Nolan, R. L. (1974). Managing the four stages of EDP growth (pp. 76-87). Harvard Business Review.
- Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). The capability maturity model for software. Software engineering project management, 10, 1-26.
- Curley, M., Kenneally, J., & Ashurst, C. (2012, December). Design science and design patterns: a rationale for the application of design-patterns within design science research to accelerate knowledge discovery and innovation adoption. In European Design Science Symposium (pp. 29-37). Springer, Cham.
- Mettler, T. (2011). Maturity assessment models: a design science research approach. International Journal of Society Systems Science (IJSS), 3(1/2), 81-98.
- Pfeffer, J., & Sutton, R. I. (1999). Knowing "What" to Do Is Not Enough: TURNING KNOWLEDGE INTO ACTION. California management review, 42(1).
- Asthana, A., & Olivieri, J. (2009, May). Quantifying software reliability and readiness. In Communications Quality and Reliability, 2009. CQR 2009. IEEE International Workshop Technical Committee on (pp. 1-6). IEEE.
- Mankins, J. C. (2009). Technology readiness assessments: A retrospective. Acta Astronautica, 65(9-10), 1216-1223.

- Gove, R., & Uzdinski, J. (2013). A performance-based system maturity assessment framework. Procedia Computer Science, 16, 688-697.
- Dam, H. K., & Ghose, A. (2013). Supporting change impact analysis for intelligent agent systems. The science of Computer Programming, 78(9), 1728-1750.
- Ryder, B. G., & Tip, F. (2001, June). Change impact analysis for object-oriented programs. In Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering (pp. 46-53). ACM.
- T.Mens, Introduction, and roadmap: history and challenges of software evolution, in T.Mens, S.Demeyer(Eds.), Software Evolution, Springer, Berlin, Heidelberg, 2008.
- H.K.Dam, M.Winikoff, An agent-oriented approach to change propagations of software maintenance, Journal of Autonomous Agents and Multi-Agent Systems 23(3)(2011)384-452. <http://dx.doi.org/10.1007/s10458-010-9163-0>.
- Padgham, M. Winikoff, Developing Intelligent Agent Systems: a practical guide, John Wiley & Sons, Chichester, ISBN:0-470-86120-7, 2004.
- H.K. Dam, M. Winikoff, Supporting change propagation in UML models, in Proceedings of the 26th IEEE International Conference on Software Maintenance, ICSM'10, IEEE computer society, Washington, DC, USA, 2010, pp.1-10.
- H.K. Dam, L.-S. Le, A. Ghose, Supporting change propagation in the evolution of enterprise architectures, in Proceedings of the 2010 14th IEEE International Enterprise Distributed Object Computing Conference, EDOC'10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 24-33.
- <http://dx.doi.org/10.1109/EDOC.2010.23>.
- H.K.Dam, A.Ghose, Supporting change propagation in the maintenance and evolution of service-oriented architectures, in Proceedings of the 2010 Asia Pacific Software Engineering Conference, APSEC'10, IEEE computer society, Washington, DC, USA, 2010, pp.156-165.
- R.Arnold, S.Bohner, Software Change Impact Analysis, IEEE Computer Society Press, 1996.
- Web link: <https://searchsecurity.techtarget.com/>

## APPENDIX

### Survey mechanism for analysts, developers, testers, and deployers are given below:

#### Survey on SRW-CIA

Nowadays computing assumes a noteworthy part of everyday life. In this presentation, software readiness framework is proposed for the system, which will be useful for various partners in the software engineering life cycle. The study is started to confirm whether the recognized framework is helpful in the product business environment. The study can be filled and for any elucidation, the researcher to be reached. This study, as a guide, is to evaluate the suitability of impact analysis distinguished in the setting mindful framework. This examination is significantly devoted to



ceaseless quality change and the survey is a key bit of the assessment process. By taking two or three minutes to complete and give back the study using google-docs will be of much help and the Researcher is much obliged for the coordinated effort. Execution investigation of the SRW-CIA an approach requires specialists in the business, who must exhibit certain skills in wide ranges of programming advancement. The

accompanying things are the coveted parameters to evaluate and distinguish the patterns in the predefined requirement situation. An input on the significance of the things in the present position and the rate of instruction bestowed by these examples is an appreciated sign anticipated. The most suitable number for everything might be tick marked to evaluate the general significance in the present position. (5 - Extremely vital and 1 - not essential)

		5	4	3	2	1
SQ1	<b>Importance of change impact analysis in the software lifecycle</b>					
SQ2	<b>CIA usage in software readiness phase</b>					
SQ3	<b>Being an expert your role in the CIA usage</b>					
SQ4	<b>Different event/functionality involved in the system and its clarity</b>					
SQ5	<b>Importance of available service usage with SRW-CIA</b>					
SQ6	<b>Impact analysis method and its procedure usefulness</b>					
SQ7	<b>Role of policy rules of CIA maps with expected action</b>					