A Systematic Literature Review: Software Release Management and Delivery

Amiruddin Saddam
Pasca Sarjana Fakultas Teknologi Informasi,
Universitas Budiluhur,
2111600827@student.budiluhur.ac.id

Abstract—Software release management is a process in delivering a product to users. The process consists of planning, software development, testing, and deployment process until the software can be used. The purpose of writing this literature review is to conduct a review of several other papers that conduct research on evolution and also the techniques used in software release management. This paper gives information about the processes that can be used and also the evolution of software release management.

Keywords—Software Release Management; Software Development; Paper Review.

I. INTRODUCTION

Nowadays the software release management starting from traditional techniques towards new phenomenon such as agile and continuous delivery via DevOps. Release and Deployment Management is the process of planning, building, testing, deploying, version controlling and distributing software products. A well-documented and well-designed process for release and Deployment Management indicates organizational maturity and the major concern of a software organization is to deliver software releases with maximal services with greatest quality within the boundaries of the resource constraints. Ineffective release management process leads to lack of control over the delivered changes and their quality which can affect any software organization badly. There are many reasons behind poor quality releases e.g. growing volume and frequency of application releases, lack of control over the release procedures, insufficient relationship between dev team and release team, complexity in version control, manual deployments, etc[1] .

Some research works are devoted to decision making about the release time of software. In past few decades, various software release policies have been proposed to calculate the optimal release time for software which are mainly based on the two evaluation criteria: total development cost and reliability. The development cost of software includes mainly three components: failure cost, appraisal cost and prevention cost. The maintenance cost after the software release also causes increment in total cost of software. In the present market, maintenance accounts for approximately 50% of total revenue. This revenue stream is well growing and highly profitable for software industries than license revenue. Maintenance also enhances the existing functionality of the software system. It is possible that the maintenance process accounts for the removal of the fault in the operational phase including the faults which lie outside the warranty period or warranty definitions. In other words, warranty action may be included in maintenance action, but not the converse[2] .

II. SOFTWARE RELEASE MANAGEMENT PROCESS

The management of software releases is all about the release process control, so companies can promptly provide quality software. The distribution of applications is a complex undertaking involving various parties concerned, including product managers, developers, testers, construction and release engineers, senior management, and the management team[3].

The release management process has a structure that contributes greatly to the increase of service quality and continuity in IT, to decrease the risks, costs, time of product release, and to increase the consistency of the live environment. Many different technologies and tools continue to emerge to increase this contribution and efficiency. Especially release automation systems and process automation systems contribute to increasing the efficiency of release management. However, some important principles and technologies emerge throughout the entire life cycle before the release management process[4] .

Release management contains five major processes to plan and deploy the approved releases into an IT production environment.

A. Release Planning

Access the importance of the Release. Select the most relevant Release and Deployment plans, and get them agreed with customers and stakeholders. It is nothing but planning the releases in line with requirements resulting from approved changes. Analyze all the affected CIs (Configuration Items) and review what software or hardware is required to accomplish the plan, and determine the cost. This phase needs the plan and design approval before proceeding to the next step.

B. Release Build and Testing

Build effective release-packages for the deployment of one or many changes into IT production environment i.e. collect and integrate the release components. The release mechanisms should be tested to ensure that there is only minimum disruption to the production environment.

C. Service testing and pilots

Test service requirements and operational requirements by employing the most suitable method for the release.

D. Plan and Prepare for Release deployment

Make sure everything is ready for the release i.e. all the required approvals, and documents are available to execute the release.

E. Deploy Release

1. Execute - deployment of release
2. Training - Identify and fulfill KT or Training requirements for different users about the service
3. Deployment/Rollout - Introduction of a hardware or software release into the IT environment
4. Retire or decommission the unused assets - Remove the hardware and software that is no longer required
5. Back-out/Rollback - A detail plan explaining how a specific release can be taken back after the changes being applied, if required.

F: Verification

Verify with the users, service operations and stakeholders are capable of using or operating the service. If any issue found, take necessary corrective action, verify and review the release to ensure the maximum succesfull deployments.

Release Management process is supervised by the Release Manager whose role is principal to the success of a release. The Release manager guides to enable the readiness of the releases, and to proficiently ascertain the deployment targets for the deployment stages of a release. This level of control guarantees that, the Release Manager can deliver changes to the IT production environment successfully, to all relevant stakeholders on time also to achieve business value [5] .

III. SOFTWARE RELEASE METHOD

There are many models exist in the literature for the software life cycle and release management which describe the series of steps the system goes through starting from realization of need, through construction, maintenance and retirement. Brief description for some of these models will be mentioned in the following sections[6].

1.Waterfall / Adhoc Method

Before agile software developments projects were relying on waterfall approach to software development. The waterfall method was like a series of logical phases in which the progress use to flow from one phase to another [7].

This methodology focus only on planning the contents of the next direct release using manual approach. Ad hoc methods are used to determine solution plans but are far from objective demands.

2. Agile Method

Agile has its origin in project management methodology and particularly in software development. Agile approach is contrary to the waterfall approach a continuous change approach, which incorporates many small changes also reflecting the client demand. From a scientific method approach, agile would qualify as an inductive approach, while waterfall follows a deductive methodology [8]. However, agile project managers have an important role to play in ensuring a project's success. This relates to the kind of behavior required to achieve the success. Therefore, seven agile project manager behavior in determining project success have been identified such Leadership, Openness, Results Orientation, Ethics, Communication, Strategic and Creative and Innovative [9].

3. Incremental Method

Incremental software development is the process in which software product is developed in incremental manner such that additive components and/or faults correction are produced through the sequential product releases. This will enable the end customers to receive parts of the system early to get higher business value and gain early feedbacks.

4. Devops and Continuous Delivery

The continuous approach goes beyond the borders of traditional software development to reach the operational side as well. In this scenario, DevOps stands for a continuous integration between software development (Dev) and its operational deployment (Ops). DevOps efficiently integrates development, delivery, and operations, thus facilitating a lean and fluid connection of these traditionally separated silos [10].

With the continuous development and expansion of the electricity dispatching control system business, the traditional way of understanding the real-time operation status of the power grid through the dispatching workstation is increasingly unable to meet the requirements of modern power grid control for monitoring, operation and maintenance of real-time, timeliness, and anywhere. In this context, how to develop mobile applications that meet customer needs with minimal investment in resources and efficient development efficiency has become our research direction[11] .

IV. RESULT AND DISCUSSION

From the various studies conducted on software release management, each research has a goal of analyzing the data. Table 1 displays state of the art of literature that discusses software release management and delivery processes for their research topic, methodology, and finding. In table 2 display state of the art of literature that discusses software release management and delivery processes for their limitation, and also area for future research and related features quantitative research on software release management and delivery. From the various studies that have been done that displayed in

Table 3, deployment methods will be made and also the advantages of using this method.

Table 1.. State of The Art (Research Topic, Methodology, And Finding)

Source Information	Research topic /Question	Methodology	Findings
Inaki Prieto, et.al (2017)	The continuous deployment strategy and tools used in software development, but adapted to the problem of creating, maintaining and deploying CityGML models	Literature	• CD Advantage is becoming increasingly common in software development, it may be easier to find people who already know about it when hiring people to work on the maintenance and deployment of City GML models.
Malena. Z (2021)	The framework consolidates gathered data to demonstrate how logistics startups apply agile methods and practices based on the results of the Delphi study and the survey, and how the usage of agile methods changes over the age of logistics startups.	A Delphi study and survey	Agile methods are used by startups of all ages to improvecommunication within the team. More direct communication streams between the develop-ment team and the customer may help to increase the level of customer satisfaction
Colomo Palacious, et.al (2018)	use knowledge management tools in the adoption of DevOps practices by a traditional software vendor as away of efficiently integrating development, delivery and operations of cloud solutions	Grounded Theory (GT)	knowledge management practices and beyond tools are the main enablers of continuous software engineering adoption and success
Harkarirat, AK (2021)	find out about the waterfall methodology and search that whether today we still use this method	Observation	Waterfall project is best when there is clear picture of the final product and the requirements are well defined which will not change frequently. When the time is not an issue and final product is main concern. • Traditional methods used to focus on project scope using them to determine cost and time schedule
Syahbaz Ali, et.al (2018)	discuss various aspects and issues of Software Release	Qualitative	CI will allow the developers to integrate software code into a shared repository several times a

	management and to propose a framework to promote Continuous Integration and other advanced tools in order to have better control over software releases		day, each check-in by developers to code repository is verified by the automated build process and allow the developers to know the problem, its location earlier
Rosero R, et.al (2017	present an approach to RT for software products with DB access under an incremental software development context. It is based on the clustering of theDB access code along with the DB schema.	Literature	consider iterative and incremental development environments in which the software is developed and delivered in short cycles, the execution of the complete suite of testcases is an unacceptable practice given the volume of testcases to execute (the goal of pursuing agility is lost).

Table 2. State of The Art (Limitation, Area For Future Research, And Related Features Quantitative Research On Software Release Management And Delivery)

Source Information	Limitation	Area For Future Research	Related Features Quantitative Research On Software Release Management And Delivery
Inaki Prieto, et.al (2017)	If there is a significant number of actors, there will probably be more conflicts among their changes. This could require a different Git workflow instead of the centralized one used her	continue completing the pipeline withmore checking tools. The use of CityGML deployment pipelines should make it easier to set up large, collaborative CityGML maintenance scenarios	need tools that can connect between git and automated testing. besides that also need tools for code checker.
Malena. Z (2021)	it can be argued that the use of a framework in the field of agile software development in the logistics industry might be too inflexible in its usage and/or increase the over all effort applying agile methods.	Future research could involve case studies to explore how and what traditional logistics companies and logistics startups can learn from each other, particularly in terms of agile methods but also how they can profit from each other's ways of approaching customer-value creation	Need a comparison between methods according to company needs
Colomo Palacious, et.al (2018)	Knowledge management is seen as one of the corner-stones for	The need for implementing DevOps emerged when Meta4 moved from	Need a comparison for know a gap between two traditionall

	software quality.	on premise products toservice-oriented cloud solution	y independent departments: software development and IT operations
Harkarirat,A K (2021)	concentrated on the traditional methodology namely waterfall model as if it can be used in this new e-business era and for which type of businessthis model is suitable. It explains about the waterfall model and how this model works	In today world scenario waterfall methodology is used very less because they are often teams which spend a long time on critical tasks which leads the project progress behind, and it ends up with a long list of unfinished tasks at the end of the project, Is it still can be used ?	Need a method that can increase the speed in releasing a software
Syahbaz Ali, et.al (2018)	This approach has used configuration as minimum as possible to achieve the final goal. One sample test case is also used to validate the results.	As a future work, more complexity to this process can be added like if different teams consider different versions of the software build, and the developers are desirable to add third-party components integrations, for example, open source applications	Need tools that can integrate CI with team collaboration tools or other tools that can help CI
Rosero R, et.al (2017	Although results shed light on the useof the RT approach on a small and a medium size software product.	it is necessary to perform studies with large scale software products	need tools that can connect between git and automated testing. besides that also need tools for code checker.

Table 3. Method and Benefit

Paper Title	Author	Release Method	Result / Benefit
A continuous deployment-based approach for the collaborative creation, maintenance, testing and deployment of cityGML models	Inaki Prieto, et.al (2017)	Continuous Delivery	The main benefits of this method are the automation of model testing, every change to the model is tested in a repeatable way and the automation of the model deployment, every change to the model can reach its end users as fast as possible.
A Framework on the Use of Agile Methods in Logistics Startups	Malena. Z (2021)	Agile	process optimisation, improve the customer relationship or bring improvements within the organisation, and improve communication within the team.
A case analysis of enabling	Colomo Palacious, et.al (2018)	Continuous Delivery	Increased customer satisfaction,

continuous software deployment through knowledge management			shorter time-to-market, higher developer productivity and efficiency, continuous rapid feedback and,finally, higherquality and reliability.
Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis	Harkarirat,A K (2021)	Waterfall / AdHoc	provides a structure for organizing and controllinga software development project. The design details and errors are captured by the method before any software is written so, we can save time in developing process
Achieving Software Release Management and Continuous Integration using Maven, Jenkins and Artifactory	Syahbaz Ali, et.al (2018)	Continuous Delivery	Simplify the process from development to the successful bug free delivery.
Regression Testing of Database Applications under an Incremental Software Development Setting	Rosero R, et.al (2017)	Incremental	Developed and delivered in short cycles

From the various studies above, it can be seen that the use of the release method that has been widely used for the last few years is to use continuous delivery.

V. CONCLUSSION

From the discussion of the review paper regarding software release management and delivery, the author can conclude that there are various methods used in software releases. One of the commonly used methods is continuous delivery.

In addition, there are many tools that can be used in continuous delivery, especially in the deployment process. Based on this review paper, the author plans research on the analysis of the effectiveness of using deployment tools using the C4.5 algorithm.

REFERENCES

[1] U. Ayman Koana, F. Hasan, S. Rahman, A. M. Ismael, and K. Hussein, “SW release challenges: A case study with mitigation plan using semi-automated process Local Receptive Fields based Extreme Learning Machine for Face Recognition View project Two-stepped majority voting for efficient EEG-based emotion classification View project SW Release Challenges: A Case Study with Mitigation Plan Using Semi-Automated Process,” *Article in International Journal of Advanced Science and Technology*, vol. 29, no. 8, pp. 4942–4949, 2020, [Online]. Available: https://www.researchgate.net/publication/342530303

[2] S. Chatterjee and A. Shukla, “An Ideal Software Release Policy for an Improved Software Reliability Growth Model Incorporating Imperfect Debugging with Fault Removal Efficiency and Change Point,” *Asia-Pacific Journal of Operational Research*, vol. 34, no. 3, 2017, doi: 10.1142/S0217595917400176.

[3] S. M. Mohammad, “DevOps Automation Advances I.T. Sectors with the Strategy of Release Management,” *International Journal of Computer Trends and Technology*, vol. 67, 2019, [Online]. Available: http://www.ijcttjournal.org

[4] F. Bildiri and Ö. Akdemir, “From Agile to DevOps, Holistic Approach for Faster and Efficient Software Product Release Management,” *AYBU Business Journal*, vol. 1, no. 1, pp. 26–33, 2021, [Online]. Available: https://aybu.edu.tr/isletme/tr/sayfa/5218/AYBU-BUSINESS-JOURNAL

[5] P. Shanmugasundaram and B. Sarojini, “An Overview on Release and Deployment Management Strategy,” *International Journal of Pure And Applied Mathematics*, vol. 118, no. 9, pp. 5–12, 2018, [Online]. Available: http://www.ijpam.eu

[6] S. I. Mohamed, “Software Release Management Evolution-Comparative Analysis across Agile and DevOpsContinuous Delivery,” *International Journal of Advanced Engineering Research and Science (IJAERS)*, vol. 3, no. 6, 2016, [Online]. Available: www.ijaers.com

[7] H. K. Aroral, “Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis,” *International Journal of Applied Business and Management Studies*, vol. 6, no. 1, p. 91, 2021.

[8] D. Beerbaum, “Applying Agile Methodology to regulatory compliance projects in the financial industry: A case study research,” Kamath, 2019. [Online]. Available: https://ssrn.com/abstract=3834205

[9] K. Sutling, Z. Mansor, and S. Widyarto, “Agile Project Manager Behavior :The Taxonomy,” *2014 8th Malaysian Software Engineering Conference*, pp. 234–239, 2014.

[10] R. Colomo-Palacios, E. Fernandes, P. Soto-Acosta, and X. Larrucea, “A case analysis of enabling continuous software deployment through knowledge management,” *International Journal of Information Management*, vol. 40, pp. 186–189, Jun. 2018, doi: 10.1016/j.ijinfomgt.2017.11.005.

[11] J. Dai, D. Li, B. Gao, L. Zhou, and W. Chen, “Research on the release management technology of the mobile application for electricity dispatching,” in *Journal of Physics: Conference Series*, Jul. 2021, vol. 1971, no. 1. doi: 10.1088/1742-6596/1971/1/012055.