UNIVERSITY OF MINNESOTA

SENG 5852

The Race to Continuous Integration, Delivery, & Deployment

RESEARCH PAPER OUTLINE

LUE XIONG

March 19, 2019

1 Introduction

1.1 Thesis Statement

The software industry is transforming at a rapid pace to accommodate the dynamic nature of the market and as a result, it continues to struggle to find process-identity with continuous software development.

1.2 Purpose Statement

Software engineering has for two decades, contemplated and experimented with the concept of distributing software faster and better, without sacrificing reliability and security. There is a widespread movement in the technical community to advocate for using Agile practices to achieve such a feat, and in particular: continuous integration, delivery, and deployment. The traditional methods of software development no longer meets the need of businesses that want to proactively engage and retain customers. The organizational transition to Agile practices demands a large mentality change, understanding software as small increments developed with cross-collaboration of small team units as opposed to large modules developed by extensive siloed units.

2 Body

2.1 Differences of Interpretation & Implementation

2.2 What is Continuous Integration, Delivery, & Deployment

2.2.1 Inherently Agile

2.2.2 Continuous Integration

2.2.3 Continuous Delivery

2.2.4 Continuous Deployment

2.3 Struggles of Traceability

2.4 Paradigm Shift in Leadership

naturally forming resistance from individuals from within.

3 Conclusion

4 Bibliography

References

[1] Atkinson, B., & Edwards, D. (2018). Generic Pipelines Using Docker: The DevOps Guide to Building Reusable, Platform Agnostic CI/CD Frameworks. Berkeley, CA: Apress. doi: https://doi.org/10.1007/978-1-4842-3655-0

[2] Bosch, J. (2014). Continuous Software Engineering. Cham: Springer International Publishing. doi: https://doi-org.ezp1.lib.umn.edu/10.1007/978-3-319-11283-1.

[3] Continuous Delivery, Deployment & Integration: 20 Key Differences. (2018, June 04). Retrieved from https://stackify.com/continuous-delivery-vs-continuous-deployment-vs-continuous-integration

[4] Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access, 5, 3909-3943. doi: 10.1109/access.2017.2685629

[5] Ståhl, D. (2017). Large Scale Continuous Integration and Delivery: Making Great Software Better and Faster. [Groningen]: University of Groningen.

[6] Ståhl, D., Hallén, K., & Bosch, J. (2016). Achieving traceability in large scale continuous integration and delivery deployment, usage and validation of the eiffel framework. Empirical Software Engineering, 22(3), 967-995. doi: 10.1007/s10664-016-9457-1