An investigation into the effectiveness and impact of a Continuous Deployment pipeline upon university-level games development teams?

Frost Donovan

***Abstract*—What's the problem? What am I looking at? How does that help solve the problem?**

Opening, Challenge, Action, Resolution

I. Introduction

II. Background & Supporting Literature

What is CD [1], [2]? Best practices [3]? Has this been done before in academic setting [4]–[7]? What benefits is it supposed to have [1]? Drawbacks [8]? Links to other things - CI [9], Unit tests, regular product reviews, stakeholder (supervisor) confidence, git flow [10]

III. Research Question

From the above sources, I have formed **The actual question**

A. hypothesis & null hypothesis

IV. Artifact

A. What will be made

CD pipeline utilising Github Actions

B. How will I ensure Quality

Quality control. Roadmap? Unit Testing? Integration testing?

C. How will I create it

D. Why will this answer the questions

V. Research Methodology

A. Experimental Design

B. Limitations

Time, resources

C. Sampling Plan

Sample size, sampling method

D. Data management plan

Managing, collecting, & storing data

E. Data Analysis

T-test?

F. Ethical Considerations

VI. Appendix

Data analysis code, supporting screenshots, list of unit tests & testing plan

References

[1] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.

[2] Atlasian, "Bamboo best practice - using stages," 2021.

[3] P. M. Duvall, S. Matyas, and A. Glover, *Continuous integration: improving software quality and reducing risk*. Pearson Education, 2007.

[4] S. Krusche and L. Alperowitz, "Introduction of continuous delivery in multi-customer project courses," in *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, (New York, NY, USA), p. 335–343, Association for Computing Machinery, 2014.

[5] S. Krusche and B. Bruegge, "User feedback in mobile development," in *Proceedings of the 2nd International Workshop on Mobile Development Lifecycle*, MobileDeLi '14, (New York, NY, USA), p. 25–26, Association for Computing Machinery, 2014.

[6] B. Bruegge, S. Krusche, and M. Wagner, "Teaching tornado: From communication models to releases," in *Proceedings of the 8th Edition of the Educators' Symposium*, EduSymp '12, (New York, NY, USA), p. 5–12, Association for Computing Machinery, 2012.

[7] K. Kuusinen and S. Albertsen, "Industry-academy collaboration in teaching devops and continuous delivery to software engineering students: Towards improved industrial relevance in higher education," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 23–27, 2019.

[8] L. Chen, "Continuous delivery: Huge benefits, but challenges too," *IEEE Software*, vol. 32, no. 2, pp. 50–54, 2015.

[9] S. Pittet, "Continuous integration vs. continuous delivery vs. continuous deployment."

[10] V. Driessen, "A successful git branching model," 2012.