How do Incremental Software Development practices affect the overall stability of the project code base?

COMP130 - Game Architecture & Engineering

1804356

February 24, 2019

1 Presenting the problem

Incremental Software Development (further referred to as IDS) is a model for developing software that relies on incrementally prototyping, designing and implementing pieces of software in steps towards an end product. However, one pitfall commonly discussed [1]–[4] is the overall reliability of the newly-implemented pieces of software in the code base. In other words, as the project advances, there is a tendency to re-visit previously implemented code in order to re-factor because of incompatibilities or poor design decisions that only became visible later on in the development cycle. In the present case (the author's current project), for instance, it was noted that such impediments led to several code-base-wide refactoring and re-designs that obviously slowed the development process. However, it can be argued that more time spent designing instead of prototyping

or actively developing would have been sufficient to prevent such outcomes, both in the presented case and in general [5], [6]. Consequently, the author would like to present the paper's question in the following manner: Using IDS practices, how would one best balance the 3 stages of development -designing, prototyping, developing- each cycle, in order to reduce the maximum number of times specific pieces of code need to be revisited and adapted to new changes?

The approach this paper will take to answering the question is to study how different aspects of project development come into play to influence productivity and the overall quality of work over the course of the development cycle. Additionally, the paper aims to identify the role played by effort estimation [7], as well as alternative software development approaches [8], and finally, feedback control [9]. In the cited sources, all these concepts appear to be individually researched, yet a comprehensive study of how they all affect each other in the context of IDS has not yet been produced, to the author's best knowledge.

In other words, the paper is to be comprised of several sections identifying core aspects of the software development cycle, together with their most common reported issues, complemented by the author's personal experience with their latest software-focused project. It is estimated by the author that more case studies will be required to yield relevant data for the paper, and will therefore make use of resources (in this case, of a non-academic nature) such as the Games Developer Conference Vault (Further referred to as GDC Vault) in order to provide relevant data for the paper.

References

[1] T. Fujii, T. Dohi, and T. Fujiwara, "Towards quantitative software reliability assessment in incremental development processes", in *2011 33rd International Conference on Software Engineering (ICSE)*, May 2011, pp. 41–50. DOI: 10.1145/1985793.1985800.

[2] D. R. Graham, "Incremental development and delivery for large software systems", in *IEE Colloquium on Software Prototyping and Evolutionary Development*, Nov. 1992, pp. 2/1–2/9.

[3] T. Tan, Q. Li, B. Boehm, Y. Yang, M. He, and R. Moazeni, "Productivity trends in incremental and iterative software development", in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, Oct. 2009, pp. 1–10. DOI: 10.1109/ESEM.2009.5316044.

[4] D. Liu, S. Xu, and W. Du, "Case study on incremental software development", in *2011 Ninth International Conference on Software Engineering Research, Management and Applications*, Aug. 2011, pp. 227–234. DOI: 10.1109/SERA.2011.43.

[5] S. Ikemoto, T. Dohi, and H. Okamura, "Estimating software reliability with static project data in incremental development processes", in *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, Oct. 2013, pp. 219–224. DOI: 10.1109/IWSM-Mensura.2013.38.

[6] P. Trivedi and A. Sharma, "A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation", in *2013 2nd International Conference on Information Management in the Knowledge Economy*, Dec. 2013, pp. 188–194.

[7] P. Mohagheghi, B. Anda, and R. Conradi, "Effort estimation of use cases for incremental large-scale software development", in *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.*, May 2005, pp. 303–311. DOI: 10.1109/ICSE.2005.1553573.

[8] P. E. Ross, "The exterminators [software bugs]", *IEEE Spectrum*, vol. 42, no. 9, pp. 36–41, Sep. 2005, ISSN: 0018-9235. DOI: 10.1109/MSPEC.2005.1502527.

[9] S. D. Miller, "A control-theoretic aid to managing the construction phase in incremental software development (extended abstract)", in *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, vol. 2, Sep. 2006, pp. 341–343. DOI: 10.1109/COMPSAC.2006.92.