A Simulation Model for Application Development in Enterprise Data Platforms

A

Nayem Rahman
Portland State University, USA

INTRODUCTION

In spite of continuous refinement over the last four-five decades, the software development industry is still not free from many genuine criticisms in terms of providing the best or perfect possible products & services up to the level of high expectations of the increasingly growing software/application users around the world. On the basis of the experiences of working on or being involved in different important software application development projects as an industry professional, and an academician respectively, the authors have been searching for suitable answers to these criticisms.

In the process, from the similar research, or related studies done by the authors in the recent past, they identified the major critical areas which need to be addressed relentlessly, and more & more thoroughly always. Those are i. Controlling Cost, ii. Maintaining Schedule, iii. Ensuring Quality (Khanam and Rahman, 2019). There are many tits and bits to deal with to address these major issues. The authors attempted to develop a simulation model showing the possible solutions to the root problems in each of these major areas by investigating the resolved and unresolved challenges faced by the software development projects undertaken by the renowned supplier of software application programs in the world. The prepared simulation model for application development in Enterprise Data Platforms (EDPs) aims to analyze all the related & root causes of criticism to ensure an efficient life cycle of software development, optimized cost throughout the process, and improved performance of the launched & marketed applications.

It is important that code defects at each stage of software development need to be scrutinized very strictly. This paper illustrates it by adding two extra phases viz. code inspection and code score carding to the usual phases of software application development life cycle in the Enterprise Data Platforms (EDPs). With the proven experiences gained from other similar studies done by the authors in the past, it can be claimed that defect removal rate is much higher in the process having these two extra phases compared to the usual process not quite caring about code inspection and code score carding in every major stage of software development. It is highly expected that the simulation model presented in this paper would certainly enable the developers to improve their existing practices in giving quality and timely output of application software in EDPs (Dabab et al., 2018).

The authors have revised, enhanced, and updated their related research papers for the purpose of preparing this paper to present the said model. The due effort was given to bring appropriate answers to the following questions in the simulation process:

1. How much improvement may be made in process performance by introducing source code inspections?
2. Does the error detection capability of source code inspection leave any impact on the software development process in EDPs?

DOI: 10.4018/978-1-7998-9220-5.ch171

3. What are the risks of not going beyond the lowest rate of error detection even after adding the phase of code inspections?
4. How to choose only one between source code inspection and unit test?
5. How to justify the benefit of source code inspections if the starting source code shows more errors?
6. How to compare the existing cost of any incorrect unit test done in a software development process in EDPs?

BACKGROUND

From the results of the similar studies done by the authors, the authors got more or less conclusive answers to these questions as apprehended, to justify the inclusion of code inspection in the planned simulation model for application development in EDPs (Dabab et al., 2018). Software development is laborious, expensive, and unreliable. Hence, software development projects quite often encounter schedule slippage, cost overruns, and poor quality software in both commercial and government sectors (Fatin and Rahman, 2020; Raffo and Wernick, 2001). To address this potential issue, thoughts of bringing changes to the software development process got momentum. Smith and Rahman (2017) observe that "without efficient processes through which Information Technology (IT) builds and supports the technology, the full business-value potential will remain unrealized." Bringing the software project lifecycle under the radar of simulation models could be a good effort (Kellner et al., 2001). For the last four decades, systems dynamics modeling and simulation techniques were applied in diverse disciplines of science, engineering, and manufacturing processes (Richardson, 2013; Rashidi, 2016). According to the Merriam-Webster Online Dictionary, "simulation is the imitative representation of the functioning of one system or process by means of the functioning of another." Simulations run in simulation time, an abstraction of real-time (Rahman, 2018b; Imagine That Inc., 2014).

Simulation models are used to solve problems that arise in manufacturing (Barra Montevechi, 2016), business process design (Liu and Iijima, 2015), inventory management system (Cobb, 2017) and health care decision-making (Chick, 2006; Chen and Zhao, 2014). Martinez-Moyano and Richardson (2013) and others (Morrison, 2012; Mould and Bowers, 2013) listed 41 best practices of systems dynamics modeling and categorized them in terms of problem identification and definition, system conceptualization, and model formulation. Hughes and Perera (2009) argue that simulation could be integrated as a daily tool to solve problems. They present an easy-to-follow framework – consisting of five key stages, such as foundation, introduction, infrastructure, deployment and embedding - for enabling companies to embed simulation technologies into their business processes (Hughes and Perera, 2009). The work of Eatock et al. (2001) indicates that describing the dynamic behavior of IT could be very helpful for business process modelers in predicting the impact on organizational processes (Dabab et al., 2018; Eatock et al. 2001). Software process simulation is suggested to be helpful to achieve higher Capability Maturity Model (CMM) levels in software development (Rahman, 2018b; Raffo et al., 1999).

In software engineering, simulation modeling has attracted considerable interest during the last decade (Rahman, 2018b; Ahmed et al. 2008). Software process simulation is used mainly to address the challenges of strategic management of software development and to support process improvements (Raffo and Kellner, 2000). In this work, we are making an attempt to leverage simulation modeling in a data warehouse application development (Rahman, 2018a). We developed a simulation model based on defined processes for the application development of a data warehouse reporting environment called Next Generation Capital Reporting (NGCR). In our recent project, we developed and implemented a

Comparison of Brainwave Sensors and Mental State Classifiers
Hironori Hiraishi (2022). *International Journal of Artificial Intelligence and Machine Learning (pp. 1-13).*
www.irma-international.org/article/comparison-of-brainwave-sensors-and-mental-state-classifiers/310933