

## Chapter 25

# A Proposed Pragmatic Software Development Process Model

**Sanjay Misra**

*Covenant University, Nigeria*

**M. Omorodion**

*Federal University of Technology – Minna, Nigeria*

**Amit Mishra**

*Federal University of Technology – Minna, Nigeria*

**Luis Fernandez**

*University of Alcala de Henares, Spain*

### ABSTRACT

*The rapid growth in technology and the dynamism in our society today poses a lot of problems for Software Engineering practitioners. The result is a series of software development process methods that can be used to combat or meet up with the problems. What we can do is evolve, grow, and adapt to the changes that come along with development. This is the dynamism inherent in man—to adapt to change and improve ourselves and our existing systems—since the world is a far cry from what it was a few decades ago. On this basis lay the need to develop the model proposed in this chapter to meet the variations that exist as a result of technological development.*

### INTRODUCTION

In the process of developing software, as a result of rapid advancement in technology, there always seems to be difficulties in meeting up with the demands of consumers and time constraints. As such, various development models and techniques seem to have mutated from the three fundamental models viz; The Classic Software Lifecycle Model which is sequential, The Evolutionary Process Model which is iterative and the Rapid Application Development Model which is incremental in nature (Crnkovic, Larsson & Chaudron, 2006).

DOI: 10.4018/978-1-5225-3923-0.ch025

Fowler (2005) stated in an article that obtaining accurate sets of requirement does not necessarily solve the problem. Good sets of requirements become obsolete within short periods of time and fixing requirements doesn't mean the world will come to a halt because of that; many changes in the business world are completely unpredictable: and if you cannot get stable requirements you cannot get a predictable plan.

Consequently, a research conducted on various development models and techniques revealed that there is an average of about 40 models already available (Misra, Omorodion & Fernandez, 2012) and yet a consensus has not been reached to agree on one perfect model. Although it could be quite impossible to arrive at a one perfect model, but from the research conducted, software developers seem to have come to terms with the Agile Software Development Methods which is a web of several development techniques applying similar methods to software development such as developing a team, lack of bureaucracy and debugging as the code is being developed so as to minimise time wasted in software checks.

This project/work in the present paper is carried out for the purpose of designing a development model that will put into consideration most of the benefits and limitations of all the models that were found and make their weak points its strong points while also taking advantage of the benefits of those models. Its objectives are:

- To design a software process based on practical activities carried out by developers.
- To carry out a comparative analysis between the benefits of the model and those of past models.
- To design and evaluate a functional model that can and possibly will be implemented for the benefit of all humanity.
- To propose metrics that will be useful for proper product and project evaluation after using the model.

In the next section, several development methods are discussed including a bit of their pros and cons. In forthcoming section we discuss the materials and methods applied in the process of developing the model including the processes applied. A comparative analysis between the model and a few others are discussed in later section. A brief conclusion is given in the last section.

## **LITERATURE SURVEY AND EXISTING MODELS**

The software process models history begins with the introduction of a model called "Build and Fix Model". The model has only two steps:

1. Write the Code.
2. Fix problems in the code.

Thus, the main theme of the model was to write some code first and then think about different phases of development (Boehm, 1988; Gull et al, 2009).

Sometimes, software developers integrate these models so that they can utilize the benefits of each process and discard those attributes that may not be of use to them. Thus, there are several alternative models and methods used for software development.

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/a-proposed-pragmatic-software-development-process-model/192895](http://www.igi-global.com/chapter/a-proposed-pragmatic-software-development-process-model/192895)

## Related Content

---

### Psychophysiological Applications in Kansei Design

Pierre Lévy, Toshimasa Yamanaka and Oscar Tomico (2011). *Kansei Engineering and Soft Computing: Theory and Practice* (pp. 266-286).

[www.irma-international.org/chapter/psychophysiological-applications-kansei-design/46403](http://www.irma-international.org/chapter/psychophysiological-applications-kansei-design/46403)

### Programming Interfaces for Realtime and Cloud-Based Computing

Gregory Katsaros and Tommaso Cucinotta (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 959-975).

[www.irma-international.org/chapter/programming-interfaces-realtime-cloud-based/62491](http://www.irma-international.org/chapter/programming-interfaces-realtime-cloud-based/62491)

### True Color Image Segmentation by MUSIG Activation Function Using Self-Supervised QMLSONN Architecture With Context-Sensitive Thresholding

Pankaj Pal and Siddhartha Bhattacharyya (2018). *Quantum-Inspired Intelligent Systems for Multimedia Data Analysis* (pp. 213-261).

[www.irma-international.org/chapter/true-color-image-segmentation-by-musig-activation-function-using-self-supervised-qmlsonn-architecture-with-context-sensitive-thresholding/202549](http://www.irma-international.org/chapter/true-color-image-segmentation-by-musig-activation-function-using-self-supervised-qmlsonn-architecture-with-context-sensitive-thresholding/202549)

### Novel Fuzzy Preview Control With Granular Computing for Underactuated Systems: Underactuated System Case Studies

Srikanth Kavirayani (2020). *Handbook of Research on Emerging Applications of Fuzzy Algebraic Structures* (pp. 352-368).

[www.irma-international.org/chapter/novel-fuzzy-preview-control-with-granular-computing-for-underactuated-systems/247662](http://www.irma-international.org/chapter/novel-fuzzy-preview-control-with-granular-computing-for-underactuated-systems/247662)

### Cloud Security Engineering Concept and Vision: Concept and Vision

Shadi Aljawarneh (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 93-101).

[www.irma-international.org/chapter/cloud-security-engineering-concept-and-vision/203499](http://www.irma-international.org/chapter/cloud-security-engineering-concept-and-vision/203499)