International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2063215

Smart Development Model - To Overcome Cons of Existing Models

Sivaramakrishnan S
M.C.A, Sysbiz Technologies, Chennai, Tamilnadu, India

ABSTRACT

This document tries to analyze and get a look at various cons of existing models for software development and tries to evolve a new smart development model for software, in order to make better quality software with key issues and concepts handled in the life cycle model itself. It is going to help in building business processes and business intelligence at the same time as the software evolves through a smart lifecycle model.

Keywords : Software, Lifecycle, Smart, Business, Process, Intelligence, Model

I. INTRODUCTION

The Software Development Life Cycle(SDLC) has been key concept from the time of building a software. As the industry evolved and the business requirements changed, the lifecycle has to evolve and such changes in the software development methodologies lead to various models in the software industry, making them as the standard to follow.

Perhaps, there are such models and key ideas of building software, still the end product is not what you assume at the starting or even when you are into an iteration, as the predictions and assumptions does not always follow the changing business requirements.

And we have cons in every model that we have been following, so we are going to take a look at the cons of such software models, and try to understand where we can improve in the SDLC to get a better quality software, even though we cannot predict end product, as requirements are not known fully at the beginning. From such cons, we try to evolve a new smart software lifecycle and try to approach it practically in this paper.

This paper tries to solve the lifecycle issues of current scenario of software development.

II. MODELS OVERVIEW

A. Waterfall Model

The Waterfall Model follows a single iteration technique with SDLC key ideas included into the Software Development.

Cons:

- Predicted and static Business requirement,
- Simple Projects,
- Single Iteration.

B. Spiral Model

This model added iterations and dynamic software development to the previous waterfall model and

helped to make softwares for changing business requirements.

Cons:

- The collaboration within team may not be good at times, especially within large teams.
- The issues and backlogs keep on popping up and the iterations just keeps on building towards infinite time, less interaction with clients.

C. Agile Model

This model deals with divide and conquer methodology to the spiral model as it divides the problem and makes it into smaller pieces and tries to give the solution.

It adds more Client interaction.

Cons:

- The collaboration within team may not be good at times, especially within large teams.
- The issues and backlogs keep on popping up and the iterations just keeps on building towards infinite time.
- Due to more interaction with client, the business needs and the challenges increase.
- No proper Documentation.

D. SAFe Model (Scaled Agile Framework)

This model covers entire software cycle from budgeting to release at various phase and suits cloud based software releases (CI/CD pipelines) and various level within the enterprise. It helps to develop complex systems.

Cons:

- Following Kanban and Scrum, the requirements are just expressed not explained, it leads to confusion in development, as more time spent in trying to identify what has to be done among developers, or to explain itself.
- Documentation not done, As well as no pictorial representations, only words. As we all know picture speaks more than words.
- It is Enterprise friendly but not developer friendly, it gets issues popping up and backlogs more.

III. SMART DEVELOPMENT MODEL

From above overview, We could find lot of cons in our existing models that could affect the quality of software and budget of the software.

So, Why don't we come up with some smart system that has all advantages from this system and fixes cons, such that we can develop software with assured quality.

A. Iteration

Handling iteration we clearly know, what has to be done in a particular iteration, so it has clear idea on requirement and design, so it could be documented as well to avoid confusion.

Also, it helps to collaborate a bigger team as your documented requirement going to require less interaction within organisation, than so many meetings to understand what has to be done.

B. Responsibility

From documentation, assigning responsibility to the team becomes easy and it helps to cause lesser confusion among the team members.

Identify skill set to assign responsibilities, through test tasks as part of planning, make sure it is done in lesser time or from previous tasks, for a particular iteration.

Have a leader for each iteration based on performance, as it improves team members mindset towards performing well.

C. Time

Time plays a huge role in the business requirement of today.

So, it is always good to forecast the feasibility for a particular piece of requirement within the time, if it is huge, make sure it is made into pieces and made as tasks for different people within the team.

Based on time, try to split as simpler tasks as possible, if it is not feasible, try to move it to next iteration but start with the task with some bench people in the team, such that the pressure is handled very well. But complete such tasks on time, as you planned.

The Model Contains Two Inputs (Backlog Pool and Client Requirements) and One Issue pool for issues after release.

IV. FIGURES AND TABLES

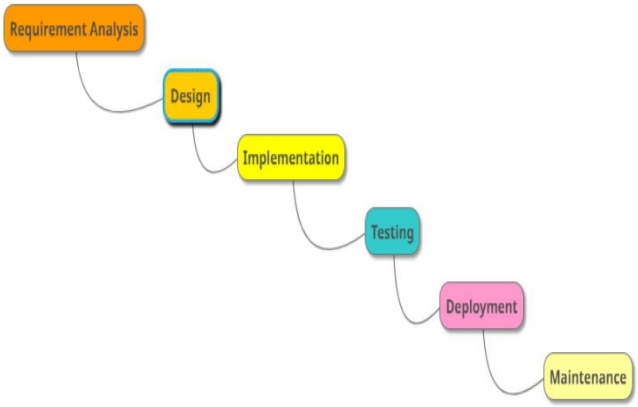

Figure 1. Waterfall Model

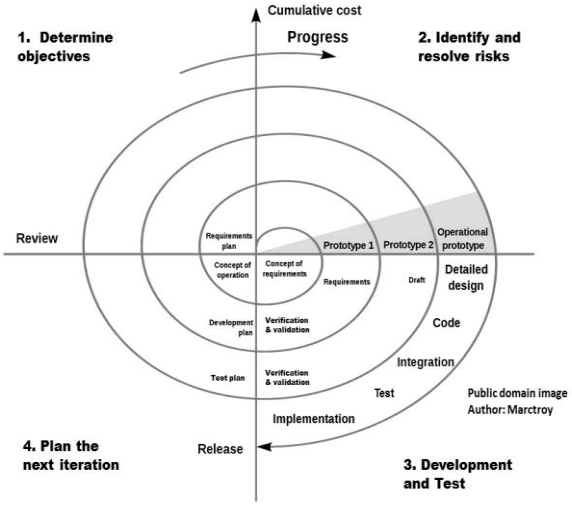

Figure 2. Spiral Model

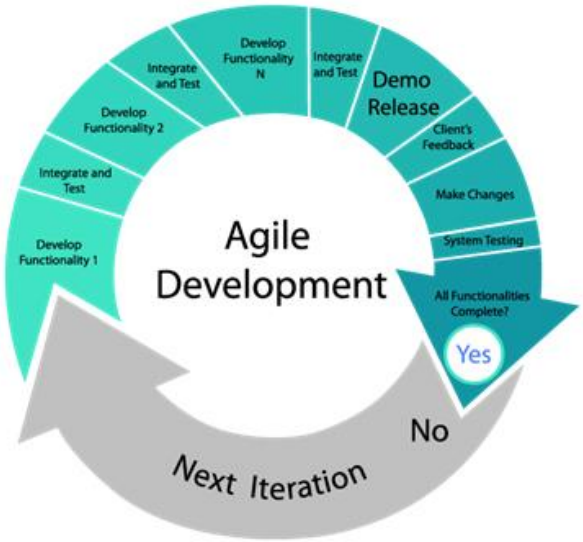

Figure 3. Agile Model

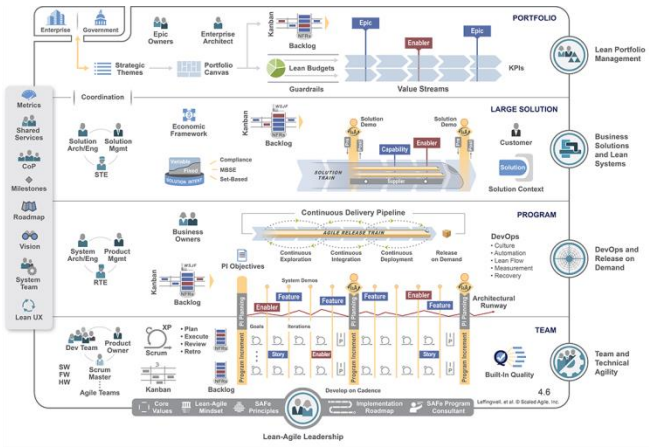

Figure 4. SAFe Model

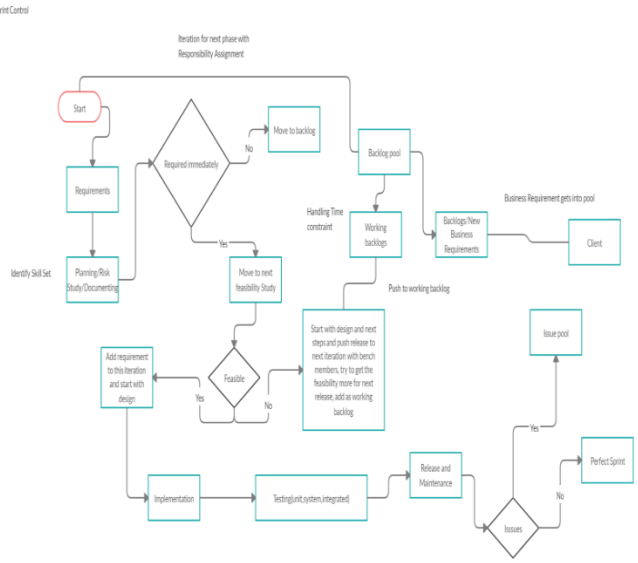

Figure 5. Smart Development Model

V. CONCLUSION

The Smart Development Model tries to address key issues in current software development and cons in the models being used for development.

The idea is to create a software life cycle model which can reduce cost and time and include risk and working backlogs as a concept for handling time constraints for the next sprints.

Making Perfect Sprints more with quality software is the idea for the end product with more client interaction and changing business requirements.

VI. REFERENCES

[1] https://en.wikipedia.org/wiki/Waterfall_model
[2] https://www.scaledagile.com/whats-new-in-safe-4-6/
[3] https://en.wikipedia.org/wiki/Spiral_model
[4] https://en.wikipedia.org/wiki/Agile_software_development

Cite this article as :

Sivaramakrishnan S , "Smart Development Model - To Overcome Cons of Existing Models", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6, Issue 3, pp.988-991, May-June-2020. Available at doi : https://doi.org/10.32628/CSEIT2063215
Journal URL : http://ijsrcseit.com/CSEIT2063215