Cyclic Incremental Management-Oriented (CIMO) Model

Azeem Uddin
B.tech (Computer Science & Engineering), Galgotias University, Greater Noida, Uttar Pradesh, India

***Abstract*- Software engineering is a layered technology. Quality, process, method & tools are the key elements that collectively treat software engineering as a layered technology. In this high tech world, where user is so fickle-minded, developing a high quality software is not a cakewalk. Software development lifecycle (SDLC) is a well-defined structured sequence of activities that leads to the production of a software product. This paper has explored an efficacious CYCLIC INCREMENTAL MANAGEMENT-ORIENTED (CIMO) model for the development of a robust, managed & reliable software. The Model is described in a thorough manner covering phases, advantages, usage & approach.**

Index terms- Software Engineering, Software development lifecycle (SDLC), Cyclic incremental management-oriented model, CIMO model.

INTRODUCTION

CIMO model is a systematic, planned & incremental approach of building a software product. In this high tech world where user is so fickle-minded, building high quality software is not a cakewalk. Selection of appropriate SDLC model matters a lot. Every model has its merits & demerits but selection of a good model can be based upon the user requirements & the advantageousness of that model. CIMO model sums this up neatly.

PHASES OF CIMO MODEL

Software development lifecycle specifies an structured sequence of activities that lead to the production of a software product. Likewise, CIMO model being a SDLC model also has certain phases that provide a supporting framework for software development. The various phases of CIMO model are summarised below:

1. Communication : Communication is the first step for the software development process. In this phase, there is a communication between the customer/user & the developer so that customer can approach the development team & can initiate the software development process.

2. Requirement Engineering : Requirements focus on What not how. In the requirement engineering phase, requirements are gathered from the customer as to what system the customer wants. Through interviews, brainstorming sessions etc., requirements are gathered & documented in a proper document termed as Software Requirement Specification (SRS). This document must be unambiguous, consistent, correct, verifiable, modifiable & traceable. Infact, this SRS is the bedrock or the base for the whole process of software creation. The ultimate agenda of this phase is to get the requirements from the customer in a clear-cut way. The developers must be more alert & attentive during the requirement elicitation as sometimes the communication gap or incorrect requirement analysis leads to serious issues in the subsequent stages.

3. Feasibility study: The third stage is feasibility study. Feasibility study determines whether the project is workable or not. An analysis & evaluation is done in order to find that project will be profitable or not. It is possible to build the system as specified by the customer or not. Technical feasibility, economic feasibility, operational feasibility & schedule feasibility are taken into consideration to determine whether the solution for the problem should be implemented or not.

4. Management process: Management process is the process of initiating, planning, executing, controlling & closing the software development & creation to meet specific success criteria at the specified time. It is a temporary endeavor undertaken to create a unique software product.

In the management process phase, there are various activities that are performed so as to ensure better management of the software like staff management, scheduling, risk management, project monitoring & control, resource management, change management etc. Proper planning is done before starting the actual workflow for developing a software as specified by the customer. In a more concised way, this management process deals with various activities that must be performed while developing a software product. Project manager must be skilled to make this management process successful. Developer can also manage various tasks involved in building the specified software. Moreover, this management process goes hand in hand with all the subsequent stages until the software gets developed.

5. Design: After management process, there comes the design phase. It is all about the modelling techniques like use case approach, Entity relationship diagram, data flow diagram etc. The developer thinks which model must be used to portray the functionalities, behaviour & structure of the software system.

6. Coding: The development team decides which programming language is the most suitable for the proposed system. Actual coding is started & a general idea is set regarding the programming methodologies. Pseudo code, rough estimations & code is maintained.

7. Testing & quality assurance: Testing is the combination of verification & validation. The test criteria & test case plan is created. More emphasis is given to the principles & objectives of testing with regard to the software project. Moreover, software quality assurance activities are also taken into consideration. Software quality attributes like portability, efficiency, flexibility, testability, maintainability etc. are some of the crucial elements that software product must possess.

8. Deployment & maintenance: Now, as the CIMO model is cyclic & incremental in nature, hence the final release is out only when the customer gets satisfied & when the whole product gets ready. Till then, there are several releases with limited functionality that are improved & new functionalities are added to the existing release. Ultimately, after few releases the whole system gets ready to be deployed.

After the final release, now there comes the role of maintenance of the software that will go on & on. Change management, release management, promotion management, version control etc. are taken into consideration.

USAGE OF CIMO MODEL

CIMO model can be used for large projects because if requirements are clear, the large project can be broken down into submodules with limited functionalities & ultimately, they can be integrated to produce a whole software product. To make this CIMO model successful, efficient & highly skilled project manager & developer are required.

Approach of CIMO model:

The CIMO model follows BOTTOM-UP approach as several submodules are integrated to produce the entire software product. It is iterative in nature as well. In n number of iterations, the product goes out in the form of submodules that are verified by the customer/user & if customer gets satisfied, then the final release can be made. If not, then again the flow goes back to the start. Again requirements are taken & the whole process goes on. Hence, this CIMO model is both cyclic & incremental in nature.

ADVANTAGES OF CIMO MODEL

1. Customer involvement & customer satisfaction: In CIMO model, customer involvement & customer satisfaction is more as after every cycle (Iteration), the increments or subprojects are shown to the customers to get their feedback. Moreover, the customers get the assurance that the process of the development of their software is going well.
2. Management-orientedness: In CIMO model, the project management is itself the one phase of the development process. Risk management, change management, quality management, project monitoring & control, scheduling etc. all are given utmost attention in the management phase of CIMO model.
3. Quality-orientedness: Software quality is the degree of conformance to the user/customer requirements & expectations. Software quality assurance is a process which works parallel to development of software. In CIMO model, the software testing & quality assurance phase is purely dedicated to several testing techniques & SQA activities so that organisations cannot ignore or underestimate the importance of software quality. Hence, the model is quality-oriented.
4. Incremental approach: In CIMO model, development of a software is broken down into

many submodules/versions. These partial systems are successively built to produce a final total system. However, highest priority requirement is tackled first. Hence, this model is quite appropriate for software in which higher risk features & goals are involved.

5. Better understanding of the system: Since a submodule/version is displayed to the customer/user, the customer gets a better understanding of the system being developed as customer feedback & customer satisfaction come into picture.
6. Reduced time & cost: The CIMO model is less costly to change requirements as compared to other models. It is easier to develop & test when number of iterations is small. Moreover, quick software is released during early phase of software development lifecycle.

CONCLUSION

Software engineering consists of two words: Software & Engineering. Software is a computer program that when executed provide desired features, function & performance. It is more than just a program code & engineering is all about developing products using well-defined scientific methods & principles. When it comes to developing software product, there are many commonly experienced problems like poor estimates & plans, lack of quality standards & measures, lack of proper communication, lack of up-to-date documentation etc. To deal with these obstacles, correct strategies & managerial aspects must be followed so as to maintain the quality of the software. Software development lifecycle (SDLC) is a well-defined structured sequence of activities that leads to the production of a software product. This paper has explored an efficacious CIMO model for the development of robust, managed & reliable software. The two specialities of CIMO model are its management-orientedness & quality-orientedness. In short, CIMO model is a systematic, planned & incremental approach of building a software product. In this high tech world where user is so fickle-minded, building high quality software is not a cakewalk. Selection of appropriate SDLC model matters a lot. Every model has its merits & demerits but selection of a good model can be based upon the user requirements & the advantageousness of that model. CIMO model sums this up neatly.

ACKNOWLEDGEMENT

The grace of God is above all. First & foremost, praises & thanks to the Almighty God for establishing me to complete this research paper. I would like to express my deep & sincere gratitude to my parents for their love, prayers, care & sacrifices for educating & preparing me for my future. I would also like to express my gratefulness to all my academic colleagues & friends for their constant encouragement. It is hoped that the subject matter of this research paper can be of great help in some way or the other. At last, I express my sincere thanks to one & all who directly or indirectly have lent their helping hand in this venture.

REFERENCES

[1] Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
[2] Pankaj Jalote, Software Engineering, Wiley.
[3] M. Cotterell, Software Project Management, Tata McGraw-Hill Publication.