International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Timeboxing: Hurdles and Solutions

Basant Namdeo

International Institute of Professional Studies, Devi Ahilya University, Indore, India

Abstract: ***Time boxing is beneficial to industry and end users in terms of saving time, money and human resources. Time boxing model posses some serious issues like (1) Structural dependency, (2) Deliverable dependency, and (3) Uneven logic in work distribution. In this paper we are proposing solutions to these problems. Thus this paper may be useful to software development industry, users of that software, scientists and students.***

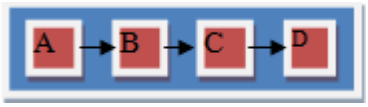
Keywords: Timeboxing, pipelining, software process, iterative development, team communication.

1. Introduction

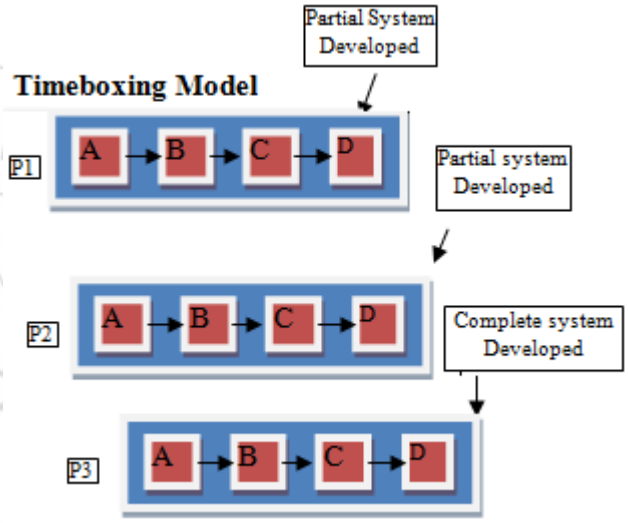
Today's competitive world, companies are required to develop new product in speed and flexible way. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done [1]. So the companies are interested to develop software very fast with the existing resources. Software development is not a single part. This thing can be divided in to various parts of task and each part of task can be assign to the team of employees who are expert in that particular task. Jalote et al. [2] proposed the timeboxing process model that takes the concept of time boxed iterations further by adding pipelining concepts to it for permitting overlapped execution of different iterations. In the timeboxing process model, each time boxed iteration is divided into equal length stages, each stage having a defined function and resulting in a clear work product that is handed over to the next stage[2]. With this division into stages, pipelining concepts are employed to have multiple time boxes executing concurrently, leading to a reduction in the delivery time for product releases[2].

Waterfall Model

For example we have
A= Requirement Analysis
B =Design
C= Implementation
D=Testing



Timeboxing Model



By this above timeboxing model diagram we can say that once **P1** phase has completed its **Requirement Analysis** then **P2** will starts its **Requirement Analysis** Phase with same resources by which **P1** has completed its **Requirement Analysis** Phase. Like this when other phases of **P*i*** *are* completed then **P*i+1's*** will start with **P*i's*** resources.

2. Background

Norbjerg experiences from a real life project that used timeboxing as the basic organizing principle in an incremental and iterative design and construction process in [4]. The project's experiences show that such a process is indeed both flexible and manageable but that it requires periodic planning and re-planning, explicit concern for coordination and synchronization activities, high process discipline and organizational readiness to accept fluctuating requirements [4]. Miranda proposes the use of a modified set of Moscow rules which accomplish the objectives of prioritizing deliverables and providing a degree of assurance as a function of the uncertainty of the underlying estimates [5].

The things are not easy as seen. There are lots of issues in using timeboxing in software. Gerogiannis et al.[3] addressed the problem of optimizing the schedule of a software project that follows an iterative, timeboxing process model. Gerogiannis et al.[3] proposed multi objective linear

programming technique to consider multiple parameters, for example the project length of time, the work discontinuities of development teams in successive iterations and the release (delivery) time of software deliverables. There are more problems which can be arises if we apply timeboxing as instruction pipeline.

3. Timeboxing Hurdles and Solutions

In general, there are 3 major difficulties that cause the timeboxing to deviate from its normal operation.

3.1 Structural dependency

There are more than one team who are working in different time box(say one is **Requirement Analysis** Team and other is **Design** Team in same phase P1) and each team required the same set of resources and we have one less than set of resources that are required, then this difficulty arises. This problem delays the output of particular phase by time that a maximum of team uses the set of resources.

Solution could be increase the number of resources required simultaneously.

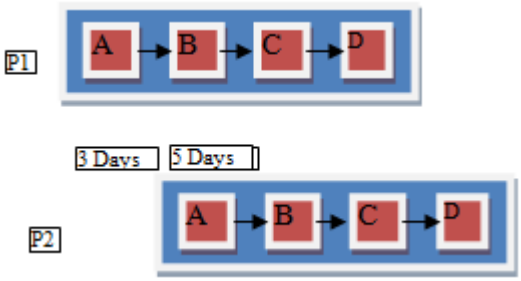
3.2 Deliverable dependency

There may be a possibility when one team completes its task, but its previous team from which it need next task has not completed its task yet (say P1 phase's **Design** Team completed its task, but P2's **Requirement Analysis** Team not completed its task, so P1's **Design** Team has to wait for the output of P2's **Requirement Analysis**.)

Solutions could be **Partial Data Forwarding** (**PDF**). In the PDF method we forward the partial data which are produced by the previous phase and is consistent, is forwarded to the next phase's Team. For example by our example we can say, P2's Requirement Analysis Team will give the partial data which must be consistent to the P1's Design Team. By this **P1's** Design Team at least do some work.

3.3 Uneven logic in work distribution

There may be a possibility when each of the phase's time box will not be of same time frame. For example phase **P1's Requirement Analysis** will take 3 days to complete and Design will take 2 days, but **P2** phases **Requirement Analysis** takes 5 days to complete. So at the end this will create a problem that, when P1's phases **Requirement Analysis** Team completed its task's and assign it to **Design** Phase and take the next phases Requirement Analysis task. Now P1's **Requirement Analysis** took 3 days and then in 2 days design will complete its task, but phase P2's Requirement Analysis has not completed its task, this required more 3 days to complete its tasks. In these 3 days **Design Team** has no work to do. Solution could be when we see this type of uneven time box, we have to restructure the team of each time box. We can increase the number of employees if a particular time box is taking more time unit or we can decrease the number of employees if a particular time box is taking less time units.



4. Conclusion

We can say that, in timeboxing model of software development, there are chances of some problems due to time box size on each iteration, number of resources required at a particular time by each team and timing, but they can be solved by applying various strategies described in this paper.

References

[1] [1] Takeuchi, H and Nonaka, I. The new new product development game, Harvard Business Review,1986, 64(1), 137-146.
[2] [2] Jalote, Pankaj, Aveejeet Palit, and Priya Kurien. "The timeboxing process model for iterative software development." Advances in Computers 62 (2004): 67-103.
[3] [3] Gerogiannis, Vassilis C., and Pandelis G. Ipsilandis. "Multi Objective Analysis for Timeboxing Models of Software Development." In ICSOFT (ISDM/EHST/DC), pp. 145-153. 2007.
[4] [4] Norbjerg, Jacob. "Managing incremental development: combining flexibility and control." ECIS 2002 Proceedings (2002): 74.
[5] [5] Miranda, Eduardo. "Time boxing planning: Buffered Moscow rules." ACM SIGSOFT Software Engineering Notes 36, no. 6 (2011): 1-5.