

## Timeboxing

A project has the triple constraints of time, scope and cost (there are other parameters of course). If one parameter changes, something else must change to keep the three as a constant. Timeboxing is normally used to complete a defined amount of work in a fixed period.

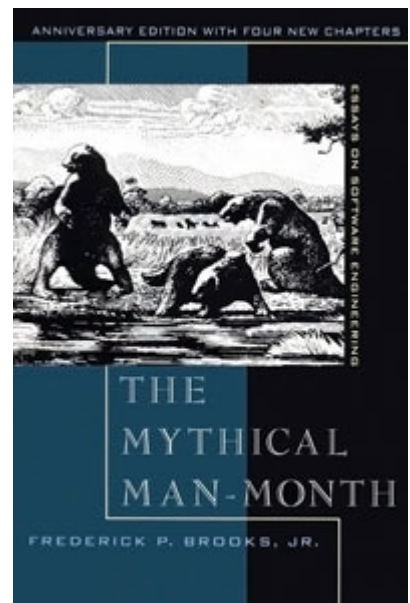
Timeboxing makes time a fixed component and adjust the other two constraints (scope and cost) to fit. This can lead to a number of interpretations:

- We will complete as much as we can in the period available.
- We will apply as many resources as we need to complete the work in the period available.
- Or something in between.

Timeboxing is a very simplistic approach to management but quite common in software development. Some of the problems include:

- The lack of any correlation between resource numbers and productivity. Increasing resources will definitely cost more but there is unlikely to be a commensurate increase in productivity. This problem was elegantly defined by Fredrick P. Brooks in *The Mythical Man-Month*<sup>1</sup>, summarised by the saying adding manpower to a late software project makes it later!
- De-scoping normally reduces value far quicker than any other element. Timeboxing can quickly lead to a software release that has cost as much as planned and taken the time planned but delivered almost no value.

Just because time is easy to measure, it does not mean time is the most important component of a project.



20<sup>th</sup> Anniversary edition – 1995

For more information on scheduling and planning, visit Mosaic's planning and scheduling home page at: <http://www.mosaicprojects.com.au/Planning.html>

---

<sup>1</sup> First published 1975 - *The Mythical Man-Month: Essays on Software Engineering*, Frederick P. Brooks (ISBN 0-201-00650-2)