Software engineers are optimists in general, and they incorrectly assume that the project will go well. In the book “The Mind of the Maker”, creativity is divided in three stages – the idea, the implementation and the interaction. Since programming medium is tractable, a programmer assumes few difficulties in implementation. But the inconsistencies of the idea become clear during implementation of a large programming project.

The assumption that more manpower will make the project finish faster doesn’t hold for projects that require communication among the workers. In tasks that can be partitioned but require communication among subtasks, the training and intercommunication can actually lengthen the schedule.

Software managers often don’t negotiate project delivery timeline if the project schedule overruns. Sharing productivity, bug, estimation data can help customers accept extended delivery time.

Because of optimism, programmers often assume the number of bugs to be smaller than it usually is. A good rule of thumb is to assign one-third of the time for planning (larger than normal but still insufficient), one-sixth for coding (as it’s easy to estimate), half of the time for debugging (larger than normal but practically needed). Since debugging part comes at the end, the secondary costs of project overrun is often significant.

In case of project overrun in the first phase, it can firstly be assumed that the underestimation only applies for the first phase, and increase manpower to cover the overrun. This in turn will add training and communication requirement that can further overrun the project. Secondly, it can be assumed that the overrun represents actual time requirement, and the total project will require longer schedule. So, enough time can be taken so that further rescheduling is unnecessary. Lastly, the project can be trimmed in a planned way to reduce time requirements.