Software is hard because we want it to be perfect, even transcendental. The article's author, Kyle Wilson, cites many common mistakes in software development that are why, as he puts it, "most projects end up horribly over-budget or cancelled or both." Most of these mistakes are made in the early planning stages of a project and are rooted in over-optimism about what can plausibly be achieved within a budget and a timeframe.

While optimism is important in planning a software project, each little goal and feature adds to the production cost and time. While in a different industry this might be easy to budget and plan for, in software development "we're never quite sure what a feature is going to cost until it's done." The needs of each software program are different, and not always known upfront. Wilson describes this as "fractal in complexity," meaning that the search for a solution often uncovers hitherto unseen problems.

Furthermore, time estimation in software development is a very inexact measurement. The science of software studies has not found any apparent correlation between a finished program's scope and purpose and the size and complexity of its code. It cannot be accurately measured after the fact, nor can it be predicted beforehand. Guesswork often fills in for reliable metrics of estimation. And guesswork is subject to the same over-optimism that plagues software planning.

One possible reason why so many software projects are finished late is simply because the deadline was never possible under anything but ideal circumstances. The software development world is very aware of this persistent issue, but no universal solution has been discovered. Software, and the problems that software tries to solve, are simply too diverse to have a one-size-fits-all solution.