

14 Common Reasons Software Projects Fail (And How To Avoid Them)



Forbes Technology Council
COUNCIL POST | Membership (Fee-Based)

POST WRITTEN BY

Expert Panel, Forbes Technology Council

Successful CIOs, CTOs & executives from **Forbes Technology Council** offer firsthand insights on tech & business.

Mar 31, 2020, 01:15pm EDT | 28,219 views



Photo: GETTY

Tech teams often plunge into new software projects with high hopes, making it all the more frustrating if the project gets derailed. Tech leaders need to be aware of potential project pitfalls ahead of time to avoid wasting time and budget dollars.

The experts of **Forbes Technology Council** have overseen many projects in their professional tenures. Below, 14 of them share common reasons software projects flounder and what tech teams can do to avoid falling into a trap.

1. Not Understanding The Needs Of The Business

One of the reasons software projects fail is the lack of understanding of the business' needs. The business must clearly articulate the requirements in detail. There needs to be a precise mapping of features and functions to the business' needs. Assigning a seasoned business leader to the project team is essential for success. - [Wesley Crook, FP Complete](#)

2. Inability To Reach Consensus On Priorities

There are various reasons why software development projects fail, but a common one that has a big impact is when the project sponsors and project teams are not clearly aligned on top priorities for the project. Decomposing these priorities into “must-haves,” “should-haves” and “could-haves” can provide a solid framework for the iteration and delivery of particular features. - [Jahn Karsybaev, Prosource IT](#)

[Forbes Technology Council](#) is an invitation-only community for world-class CIOs, CTOs and technology executives. ***Do I qualify?***

3. Lack Of Clarity And Execution Strategy

The primary goal of a software project is to solve a business' problems. It requires not only effective and efficient project management and stakeholder-expectation management but also a clear consensus by the entire group of stakeholders on the definition of the business' problem and a robust execution strategy to deliver software that solves the business' objectives. Failure to address any of the aspects outlined above results in a derailed project. - [Kartik Agarwal, TechnoSIP Inc.](#)

4. Not Starting With The End Customer

Sometimes software projects begin with a great idea that is implemented (on time or late) and delivered only for developers to discover that the problem they solved wasn't actually the problem their customer needed to be solved. Doing the hard work of deeply understanding your customers, what they need and what they're willing to pay for sets the ceiling on project performance and can help refocus a team when things derail. - [Guy Yalif, Intellimize](#)

5. Unclear Requirements

One of the most common reasons software projects fail is unclear requirements and the lack of a detailed explanation. Very often clients themselves are not sure exactly what they want to see, and as a result, the project cannot move forward. Communicating with your clients and asking them for their detailed vision of the future of the product is the key to ensuring that the project will not fail. - [Daria Leshchenko, SupportYourApp Inc.](#)

6. Expecting A ‘Silver Bullet’

Too often, enthusiasm arises from the false belief that a proverbial “silver bullet” will solve a given problem. However, proper solutions are rarely so simple—they are a blend of methodology, strategy and team support, not the result of a single action, technology or idea. Tech leaders should encourage open communication and leverage participatory group decision-making to solve challenges. - [Christopher Yang, Corporate Travel Management](#)

7. Working In A Silo

The biggest reason software projects fail is because teams embark on a journey to build something that is either not a business need or does not address the right problem. Both reasons are a result of misalignment between the business and tech. To avoid this, it's crucial to identify the problem the business is trying to solve and then work collectively with the business and not in a silo. - [Tanvir Bhangoo, Freshii inc.](#)

8. Thinking That Scope Can Be Defined Upfront

While it is important to understand the problem and define the use cases upfront, almost no project can be considered successful if it does not adapt to changing business requirements during development. Unfortunately, some tech teams still insist on hitting the original goal, thus rendering their effort ineffective or even a failure. - [Song Bac Toh, Tata Communications](#)

9. Lack Of Coordination And Detailed Planning

Many software projects are late or fail due to a lack of good coordination and detailed planning. Teams need to implement a bottom-up planning process that identifies dependencies between deliverables and includes estimates from the engineers

themselves. After the release plan is set, I run daily 15-minute stand-up meetings where issues are surfaced and new risks are identified and managed. - [Dave Mariani, AtScale](#)

10. Friction Caused By Undefined Roles

Undefined roles often create friction on project teams. Try using a DACI framework from the start to clearly define who has authority on what. For stuck projects, recalibrating on who is the Driver, Approver, Contributor and Informed within the project can act as a hard reset, inspiring renewed collaboration and autonomy. - [Leore Avidar, Lob.com Inc.](#)

11. Expecting Overcustomization Of Software

Oftentimes, we believe that software can be customized to a level that will tailor to all needs. That's a misconception. Being realistic is important. Define the requirements regarding the software's capability. Making change requests as you go requires adjustments, but that's the hat that will need to be worn to avoid frustrations. - [Bhavna Juneja, Infinity, a Stamford Technology Company](#)

12. Lack Of Discipline

If we were to build a house and keep changing the blueprint, the project budget would spiral out of control and deadline after deadline would be missed. Create a vision of what project success looks like. Lock it down and execute. Every other great idea and detour can be considered for a later phase of the project. - [Sam Polakoff, Nexterus, Inc.](#)

13. Too Many Hands In The Dev Pot

Establish (and limit) who's involved from day one, whether you're building in-house or not. This can be difficult for larger tech companies with complex processes and communication channels. But in the app development world, such complexity is detrimental to crafting a fully realized product that matches everyone's unique vision without falling prey to scope creep and a never-ending project timeline. - [Joshua Davidson, ChopDawg.com](#)

14. Not Enough Emphasis On Soft Skills

A clear and meaningful focus on managing the change process is often lacking or insufficient. I've seen many software projects in various categories and in an array of different types and sizes of organizations run into challenges because they are super-focused on the technical work but not applying enough energy toward training, coaching, team building and soft skills. - [Amith Nagarajan, rasa.io](#)