

Good Planning – Not Development Methodology – Is the Key to Successful Software Project Delivery

IT development teams have long been searching for a magic formula of success in software development. How can teams find the greatest efficiency and the least cost, without sacrificing quality?

Teams have tried a myriad of methodologies to achieve these goals. Twenty years ago, developers looked to waterfall software development as the answer. Waterfall then gave rise to object-oriented incremental or spiral, Rational Unified Development (RUP) practices.

Today, it's all about agile development. Companies are investing lots of capital to develop agile methodologies and committing significant resources to train employees to work within agile frameworks.

Despite this investment in agile methodologies, many software projects still fail, clients are unsatisfied, and IT departments often miss deadlines. Why?

The method doesn't matter. It's all about planning and how teams use resources.

When software projects get behind schedule, team leaders almost automatically think that adding more staff will help. After all, many hands make light work, right? The more staff a team has working on a project, the faster and shorter development time needed.

The old adage doesn't necessarily apply in software development. In fact, the opposite effect can occur: adding more people to a project can make more work or even slow things down further. The additional person-hours may give the team a short-term boost, but over time, the team will have to manage higher costs and more connection points—each of which creates an opportunity for a mistake or defect, bringing additional risk to the project.

What's more, how teams allocate resources is just as important, if not more, to successful projects as which development method teams use. While methods have changed, the allocation of resources has remained paramount over the last two decades. This finding was underscored in QSM's latest analysis, which leveraged data from the most recent [QSM Software Project Database update](#). The update includes [new insights into agile development processes](#) and the staffing models that agile teams are employing for their projects.

The following table provides a glimpse into how adding staff affects a project's outcome. It compares 390 applications of the same size featuring both 10,000 and 20,000 lines of newly developed code, with a significant portion using agile methods and tools. One sample uses an average of less than four people; the other, nine people or more. While the additional staff reduced the schedule by approximately 30 percent, the project cost actually increased by 350 percent. The additional staff also created 500 percent more defects that had to be fixed during testing.

Agile Team Size Unknown

In the graphs below, the lines represent the average behavior for size developed vs. schedule, effort, defect, and average staffing. Notice the large variation in effort and defects and small variations in schedule.

Agile Team Size

Image not found or type unknown

Image not found or type unknown

Image not found or type unknown

Image not found or type unknown