

Experimentus insight guide

THE SILENT ASSASSIN: HOW HIDDEN SOFTWARE QUALITY ISSUES PUT YOUR IT PROJECTS UNDER THREAT

Silent failure surrounds IT development projects. Software defects that cause projects to be delayed, come in hundreds of thousands of pounds over budget or fail after a system has gone live are still rife. This insight paper from Experimentus looks at:

- Why overlooking even small defects has such a major impact down the line.
- The role of testing – and how this must be re-assessed.
- The six most common project pitfalls that lead to quality issues.
- How to build quality into the whole software development lifecycle, so defects are prevented.

www.experimentus.com
info@experimentus.com
0207 871 2300

Experimentus Ltd
17a Dorset Square
London NW1 6QB

There are silent assassins within most, if not all, organisations: software quality issues that nobody is aware of, but which put the success of key IT projects at significant risk.

Quality issues may not come to light until the end of the development process when testing identifies or crystallises the problem, or even until the system is live and performs badly.

The consequences of this are delays, and hidden – often major – costs in the software development lifecycle arising from the stops and rework required to fix defects found during testing, or from managing change late in the lifecycle.

If defects find their way into production and failure occurs post-project, the organisation will face a heavy system support cost. There's also a credibility issue: missed delivery points and a defective system will erode the business's confidence that new systems can be delivered to time and to budget.

Quality is non-negotiable

The role that testing plays in development needs to be rethought, so defects are prevented during the lifecycle rather than detected at the end.

Quality assurance must be built into the whole software design and development process. Instead of a pass/fail 'gateway' at the end of the project, testing should be validation that the software or hardware meets requirements – and if the right quality processes are in place early in the lifecycle no critical defects should be found.

We typically find businesses that improve their quality process can save 40% on testing costs.

The quality issues that lead to functionality defects and failure can be people and process issues as well as software defects. When organisations approach us to help them improve their software development lifecycles it is these six quality issues we discover more often than any others:

1. **The focus is on speed, not quality.**

If all eyes are on the short term goal – the launch date – rather than the quality, there's a high probability that the system won't work the way users need it to. It costs much more to fix poor quality once a system is live because it will need 'dismantling' before it can be rebuilt. A lot of time and effort is expended as a result of defects not being caught earlier in the lifecycle.

Development teams must check regularly that the solution being created is of the right quality to deliver the required outcomes. While the organisation will naturally be reluctant to add steps that increase the time needed for the early stages of the lifecycle, focusing on quality from the start will enable timescales to be met with no surprises: the time needed for testing at the end will be reduced as it will simply confirm the quality of the code.

2. **Blindness to the true cost of quality.**

In the rush to get a system live, the 'invisible' cost of supporting and maintaining the software can get forgotten. Once the project has finished and the team disbands, the software may be used for another 30 years. Most organisations realise too late that their maintenance costs are taking up far too much of their yearly IT budget, which in turn reduces the resources and budget available for new developments. It's important to be aware of and prepared for the costs involved in keeping the system live.

Before the automotive industry took the quality revolution to heart, BMW was shown in a study called *The Machine That Changed the World** to have devoted a greater area of the floor of its manufacturing plant to fixing what the production line had failed to get right first time than was devoted to the production lines themselves.

*James P. Womack, Daniel T. Jones and Daniel Roos, 1990

The professional's perspective:

This organisation's test team was never involved in projects until it was time to start testing, and was only ever given two weeks to test – no matter how big the project. It never hit its launch dates. We convinced it that the testers should be involved from the start, and needed better product knowledge. This reduced timescales by getting quality right up front.

Software development house

3. Going by 'feel', not facts.

Poor measurement of IT projects is extremely common. Businesses simply don't know what they don't know, and that's how the 'silent assassins' sneak by unnoticed. The project may seem to be going smoothly, and it might even be ahead of schedule – but only with a proper framework for monitoring, controlling and analysing progress, quality levels and improvements can the business have confidence in the outcome of an IT project.

The professional's perspective:

“We worked with a system integrator that had no metrics. It needed to introduce the right ones so it could pass a customer audit. Without them, it would have lost the business.”

System integrator

Three quarters of defects found in user acceptance testing (UAT) and live can be directly attributed to poor requirements definition. This also emphasises the need for early testing!

4. Kicking off a project before the business is ready.

There's a culture within IT of never standing still, and a drive to 'just start doing' before the business has clarified its requirements. Understanding the problem the project needs to address simply isn't enough, however: if development is based on what IT thinks the business wants, the solution will not be fit for purpose.

The professional's perspective:

“75% of this telecoms operator's code had to be re-written during test. They started too early and didn't have full requirements or design – they thought their developers would know what they should do!”

Global telecoms operator

5. Lack of buy-in from users.

Users often don't know a new system is coming, let alone understand what it is or how to manage it, which leads to poor adoption and performance. Insufficient preparation – information about the change, or training in how to use the system for instance – will result in a lack of buy-in, so organisations need to communicate with and engage users well in advance of launch.

The professional's perspective:

This business defined high level requirements which were handed to the development team. It drove out the detailed requirements, and built and tested the application against them. The application was handed back to the business to do user acceptance testing. The users' response was 'what is this?!' – the developers had focused on driving the detailed design rather than meeting the needs of and communicating with the business."

Pharmaceutical company

6. Faulty design.

Software design defects compound as they go. If a piece of code is 10% wrong in the first few steps of the design process it will be 50-60% wrong a few layers later. This has a direct impact on cost: the industry maxim is that the cost of fixing a defect multiplies by an exponential factor for each step of the development lifecycle it goes through. If a defect is only caught in production, that impact can be significant – not just on cost, but also reputation.

Disarm the assassins: processes, people and tools

Building quality into the entire software development lifecycle requires more than process improvement – organisations also need to focus on the human contribution.

The processes:

Testing and software quality management form a significant cost of any project – up to 50% – so improving the process results in substantial time and cost savings, as well as boosting quality.

Process improvement begins with an assessment of current practices, measured across the whole software development lifecycle within the organisation, to identify the improvements required and provide a start point against which to monitor progress.

From there, a Process Improvement Plan can be developed, with processes that are aligned to good practices to develop their efficiency and effectiveness. The improvements should then be rolled out in parallel with training and/or mentoring or coaching.

Process improvement actions might include:

- The definition of procedures for a test approach that will ensure a common and clear understanding of objectives, responsibilities and activities.
- Risk-based prioritisation to enable the targeting of tests based on the real impact to the business and users.
- Rationalising and standardising documentation, with the aim of arriving at a set of informative, flexible and pragmatic documents that are fit for purpose, with less bureaucracy.
- Establishing standard methods to capture and validate business requirements.
- to capture and validate business requirements.

The people:

Process improvement requires a commitment to quality and a willingness to change within the testing teams, and in every other function across the software development lifecycle. The people involved at each stage need to take real ownership of the quality of their deliverables.

The key to securing stakeholder commitment is good communication and change management. People need to be taken on the journey, and empowered by a culture of quality rather than one that values and rewards speed of delivery. Implementing robust, repeatable qualitative and quantitative measurements will enable clear progress and quick wins to be demonstrated, which will help to foster engagement.

Resources will also need to be deployed in a different way. The people who previously had to focus on finding defects should be reallocated to the earlier lifecycle stages, to ensure quality and testing can be effectively and efficiently handled.

The change will require test managers and test analysts who know how to avoid the six project pitfalls described above, and have the capability to take both a strategic and tactical approach to testing.

IT projects are carrying risks they shouldn't. The need for consistent testing and quality assessment has still not been fully recognised by the industry – but it's no longer acceptable to make decisions based on cost or speed alone. The market and users alike are increasingly intolerant of delays and functionality defects, while trends like the cloud, mobile banking and BYOD bring security and interoperability challenges that make rigorous quality assurance more critical than ever.

The right approach is one that is designed to prevent software defects rather than detect them. It involves good, consistent processes, controls and management, with people working as cohesive units to ensure their individual deliverables are of the best possible quality.

And what do we mean by best possible quality? Good enough to give IT and the business absolute confidence to go live – meeting all stakeholders' expectations and avoiding the 'silent assassins'.

How Experimentus can help:

We help organisations build and implement software quality processes that will enable them to better measure and manage new and existing apps or systems. We deliver creative solutions – transforming not only your testing procedures, but fundamentally stepping up the quality of your ongoing software deliverables. Alongside traditional test management methods we strive very hard to embed a culture of quality that crucially aligns IT objectives with your business bottom line.

Experimentus Ltd
17a Dorset Square
London NW1 6QB

www.experimentus.com
info@experimentus.com
0207 871 2300