

A hand is shown reaching up from the water, with the fingers spread. The hand is positioned on the left side of the page, and its reflection is visible in the water below. The background is a blue, textured surface that looks like water with a halftone pattern.

White Paper

Five Indicators Of A Failing Project (And What You Can Do)

Providing you with:

- ✓ Early warning signs of trouble in your project
- ✓ Tips to get back on track
- ✓ Guidance on ensuring success from the start



Introduction

Failing projects are a fact of business...

Having a piece of custom software built can deliver numerous advantages to organisations trying to solve a specific business problem; making processes more efficient, increasing customer satisfaction and even opening up brand new revenue streams. However, if these kinds of projects are not planned and managed correctly they can start to spiral out of control, resulting in swelling costs, a slower time to market and more defects, all of which can have a serious impact on return on investment. Additionally, if these issues aren't addressed in a timely manner it can lead to project deliverables not meeting initial requirements, whether through an inability to scale, critical features not working as expected or the product not being aligned with core business needs. In some cases teams may even be forced to abandon the project entirely; often at significant expense, as research has found that organisations typically wait for nine months after a project becomes a problem to kill it off (source: Oxford University and Simon Fraser University).

You don't have to go far to find examples of digital projects failing to deliver against expectations; just look at Healthcare.gov, which is still dogged by the problems that accompanied its launch back in 2013. Statistics show that it's not just the big hitters who suffer either, with studies revealing that just 9% of organisations rate themselves as excellent at successfully executing initiatives to deliver strategic results. Consequently, only 56% of strategic initiatives meet their original goals and business intent, with this poor performance leading to organisations losing an average of \$109m for every \$1bn invested in projects and programmes (source: PMI).

...but there is hope

There are ways to tell if a project is heading for rough waters, and spotting these signs early on enables action to be taken before major issues arise, including calling in 'project rescue' specialists who can minimise any negative impact and enable the delivery of software that not only works but actively supports business goals. These signs can be grouped into social subsystem risks, project management risks, and technical subsystem risks, also translated as 'people', 'process' and 'product', with research finding that the majority of common project issues fall into the people and process categories (source: IT Today). In this white paper we'll cover some of the most important things to watch out for, along with suggested actions should you recognise any of the signs.





No one is communicating

Communication is fundamental to the success of all digital projects, especially where numerous individuals may be working together on a single product, so if your teams aren't talking something may be seriously wrong. In the early stages, a lack of communication can make it difficult to create a shared vision of your aims and objectives, leading to stakeholders not properly understanding the value the project will deliver and therefore not buying into it as fully as they might have. As a result, you may find it harder to get them to provide all necessary input, from the time required to give feedback on iterations to sign-off on budget requests. Additionally, if the project's direction isn't agreed at the outset this may cause issues later on that will hinder progress or distract from your ultimate goal; for example, without a clear brief company politics may dominate the project, leading to power struggles and a shift of focus from the needs of the users onto the opinion of senior stakeholders. It's important, therefore, to look out for individuals who are unsure about the objectives of the project, questioning the need for the work or refusing to commit to necessary actions, and make sure your intentions are clear, reasoned and visible at all times.



Ensuring a good understanding of the project early on is important not only to secure buy-in, but also to complement any written information you may have. It's crucial that this documentation is kept up-to-date too,

with the most recent versions clearly signposted so that anyone working on the project can easily assess its current status. Consequently, documentation that is outdated, unclear or incomplete should be treated with caution, as if this is being relied upon by team members it could lead to misunderstandings, confusion and complications.

The importance of high levels of communication should be reinforced throughout a project, especially with regards to your development team where a lack of transparency may indicate that things aren't going to plan or prevent you from resolving problems that may have already arisen. Little communication at this stage could also be a sign of potential personality conflicts or a lack of trust, something that managers of abandoned projects report much more frequently than those working on successful builds (source: Oxford University and Simon Fraser University). If these issues prevent knowledge from being shared this could slow progress, as team members take longer to get to grips with unfamiliar tools, systems and processes. Worse, it could even lead to the wrong tasks being worked on if updates to the priorities of different features are not communicated to developers or known market changes not reflected in the project plan. As such, be sure to provide all team members with regular opportunities to deliver updates and also raise any concerns; if this is not currently happening then leaving it could cause major problems.

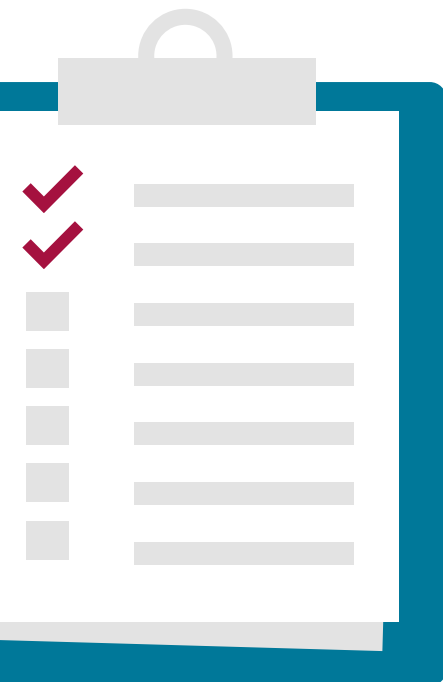
Tips

- Involve all stakeholders in the initial stages of your project to secure early buy-in
- Communicate your documentation procedures so that everyone knows how and where to access the latest versions
- Schedule in regular meetings to guarantee participation from representatives of all relevant

2

There's little infrastructure or process

One of the benefits of a shared understanding of a project's vision is that it facilitates the creation of a unified and optimised technical infrastructure that will effectively underpin all development. Not outlining this infrastructure upfront could lead to disparate and even conflicting technologies, platforms and environments being employed, potentially increasing the complexity of any required integrations and forcing teams to learn multiple tools and skills unnecessarily. If you have a number of unconnected systems or technologies that don't seem to add value to your project, make sure they're not likely to cause these kinds of problems and take steps to consolidate where possible.



Remember too that tools on their own can't deliver value, and can even hinder project progress and quality if used incorrectly, making it vitally important to also have the right people and operational procedures in place. Guaranteeing that all output has passed through a series of testing and validation phases before being released to live, robust processes minimise the likelihood of issues being found in your software and, as everyone on the project is referring to the same set of guidelines on how to approach different tasks, also provide confidence that delivery is as streamlined as possible. If your stakeholders haven't agreed upon a standard process or your processes aren't adequately defined and documented, consider how this may affect the successful delivery of your project.

However, even if you do have a unified infrastructure in place that is supported by process, your project could still fail to deliver against its objectives if your approach hasn't been planned in line with your business goals. In these cases you may find that your software, while functional, doesn't do what you need it to, is unable to scale according to your growth projections, or is not supported

by the platforms your users access most frequently. For example, you may have built a mobile application while a responsive solution may have been more effective as the majority of your users own tablets or laptops. This is a particular concern if your infrastructure and process weren't defined at the same time as your requirements; if this the situation with your project, check that all considerations have been incorporated and look for ways to accommodate any vital requirements that aren't already covered.

Tips

- Involve IT specialists in the technology vendor selection process to ensure technical concerns are addressed
- Clearly document your process to make it easy for all team members to comply to required standards
- Consider an initial consultancy phase that validates your proposed solution before significant resource is committed



Your project keeps expanding...

So far this report has focused on those issues that may arise in the early stages of your project and, while they should ideally be addressed before development begins, if caught quickly they can often be resolved with little or no negative impact. However, if these issues aren't raised and resolved promptly they can easily snowball and lead to more significant problems that manifest in additional symptoms. One such consequence is an expanding project, which may lead to development going vastly over budget, and can continue to cost you even when live, as a result of the frequent updates required to meet the real-world requirements not accounted for during the build. A noticeable sign of this issue is 'scope creep', where requirements are constantly added to the initial specification without the project's budget, timescales or functionality being updated accordingly. If this is happening on your project it could indicate dangerous bloating, as could major overtime being needed or an increasing number of people working on your build to meet agreed deadlines.

It's also important to look at the reasons why your project is expanding; a likely cause is a poor requirements gathering phase that hasn't incorporated both quantitative and qualitative research, focused on real-world users, and involved all necessary stakeholders. Without this activity it's incredibly difficult to reach agreement on a set of requirements, which can lead to delivered specifications that are incomplete or unclear. As a result, critical features and functionality can easily be overlooked and subsequently not factored into initial estimates. If your project management process does not allow for new requirements to be managed during the development phase, raising these once the project's underway could have a major impact on timescales, budget and quality. It's also been found that teams have greater difficulty

establishing requirements when working on IT projects that are eventually abandoned than when working on those that make it through to completion (source: Oxford University and Simon Fraser University), so if possible assess the quality of these crucial early phases to ensure all business and user requirements have been covered, and conduct further research or workshop activities if necessary.

Along with ill-defined requirements, not having metrics in place that confirm the project meets its objectives can lead to expectations not being set and managed effectively. Consequently, additional features may be added outside of the existing scope, as early deliverables don't provide stakeholders and end-users with the products and services they need or want. This is perhaps because the scope of the project was not defined clearly at the outset, or because the development team have had to make assumptions as to a feature's goal and purpose in lieu of detailed information. As IT Today succinctly concludes: "projects with undefined success criteria by definition cannot succeed", so be aware if your project lacks this critical information.

Tips


- Set up a process for change requests that ensures when new tasks are added others are de-prioritised, or budget and timescales are increased
- Incorporate a wide variety of research activities into your requirements gathering, such as competitor benchmarking, user surveys and stakeholder workshops
- Consider capturing requirements in the form of user stories ("As [user], I want to [task], so that [motivation]") as this can provide useful initial acceptance criteria



4

...or your project is never-ending

A project expanding in size can obviously also increase timescales, and when unclear requirements and poorly-defined success metrics compound this problem your project may always seem '90% complete', as no one is sure what constitutes 'done' and sign-off is subsequently hard to achieve. This can be alleviated by the appointment of an executive sponsor who has an in-depth understanding of and investment in the build, and can help drive development forward by assuming ownership of the project and providing final sign-off at key stages. This sponsor can also act as a project advocate within senior management, representing the interests of the team and communicating the business value in order to secure the necessary budget and resources. If you don't currently have someone in your management team or C-suite who fulfils this function then this may impact decision-making processes and, consequently, project timeframes; in particular watch out for a lack of understanding around your long-term roadmap and major decisions continually being put off, as this may indicate uncertainty as to where you are and where you're going.



Delays to a project can also be caused by inefficiencies in your process, especially if it hasn't been tailored to your specific requirements (the importance of which has already been covered). These issues can be further exacerbated by excessive levels of bureaucracy and administration, leading to reduced productivity, or project management issues such as a lack of leadership and collaboration. If your team is experiencing inefficiencies, this could manifest itself as a reluctance to commit to

project deadlines, long periods where you don't see demonstrable deliverables, issues that aren't resolved in a timely manner and, of course, a lack of regular communication, typically in the form of project updates.

Another potential reason why a project may be taking longer than anticipated is that it was too big to start with, illustrated by the fact that large projects are reportedly 10 times more likely to fail outright than smaller ones (source: IT Today). Breaking a vast project down into smaller tasks or setting regular milestones can make it easier to keep control, as it enables objectives and outputs to be defined at a more granular level. Issues can also be identified at an earlier stage, as not meeting initial deadlines highlights the need to take immediate action to get the schedule back on track. If your project is particularly large or complex but doesn't have any intermediate delivery markers, look at adding milestones to your schedule; you may also want to consider splitting development into distinct phases to enable core functionality to be delivered as a priority, with future phases providing valuable enhancements.

Tips

- Clearly communicate your sign-off process and highlight the individuals who need to be involved, to avoid delays
- Train relevant members of your team to enable them to effectively fulfil their duties on the project
- Be firm about what functionality is vital for the success of your project, and what are desirable extras

5

You keep finding issues

Another critical measure of a successful project is the quality of any outputs, and there are a number of things to look out for that may suggest this is at risk of being compromised. If you're finding issues when integrating legacy systems, for example, this may indicate that they weren't factored into initial plans and may negatively affect the stability, performance and extensibility of your finished solution as a result. Finding problems in features classed as completed may be a sign, too, that your specification is not clear enough for the development team to understand; this could potentially leave you with a product that doesn't deliver against your core goals. In both these cases if you spot the issues early enough you may be able to review your specification documents and rectify the project's direction to ensure it meets your true requirements and avoids the never-ending scope creep discussed earlier.

Finding issues in the later stages of a project could also indicate areas for concern in your process, such as testing not being performed effectively or required subject matter experts not having been consulted early enough in the project for specific concerns, including compliance and security, to be addressed. These oversights could lead to major issues and blockers not being spotted until the final stages of delivery when they can't be easily rectified, and consequently may result in inferior workarounds and, of course, delays. With a siloed and fragmented process like this you may also find that on-going work causes issues in deliverables that have already been signed off and leads to live software breaking; providing a poor user experience, potentially negatively impacting your reputation and even costing you conversion opportunities. If possible, ensure all stakeholders have been included in the requirements gathering phase of your project before the specification is signed off.

Additionally, you should review your project schedule along with the delivery process to confirm enough time has been allocated for full testing to be carried out, and provide testers with the information they need to conduct their duties effectively.

If your project is already underway you can still take steps to prevent its failure, through the introduction of regular retrospectives that not only help teams identify issues as early as possible but facilitate the continuous improvement of your processes to further increase project efficiency and quality. Reportedly already conducted by over 90% of organisations, if your project is lacking these kind of retrospectives setting them up is a fast and powerful way to safeguard success, while if they're already in place you could still benefit from assessing their current effectiveness and optimising accordingly.

Tips

- Agree with your development team how requirements are to be delivered, from format and labelling to what information is required
- Familiarise yourself with the issue tracking systems employed on your project so bugs can be reported efficiently
- Encourage team members to both raise immediate concerns and highlight general areas for improvement during your retrospectives

Recognise yourself?

If you've experienced any of these issues in your own projects, hopefully this guide has encouraged you to look at remedying them as quickly as possible. As discussed earlier, having good levels of communication is a great way to prevent problems occurring in the first place, so speaking to the other members of your team should be an obvious first step; if you ensure people are talking to each other regularly you should be able to identify issues before they become major obstacles and get your project back on track with minimal fallout. However, if your build is already bigger, longer or more complex than you envisaged then independent assistance may be required, and in these situations an external supplier can be useful in delivering the extra resource you need within short timeframes.

It's important to remember, though, that the best results will come when processes have been optimised from the outset. Working with a third party could still prove valuable in these cases as they can provide a fresh pair of eyes to uncover new approaches, particularly if you select a supplier experienced in problem-solving. Whether developed internally, by a third party, or in partnership, you should aim to accommodate the following within your project plan to ensure success:

- A detailed requirements gathering phase, involving all stakeholders, that informs the creation of a well-research and clearly-defined specification
- Agreed measures of success at the outset – if you don't know what done looks like it's much harder to get there!
- Cross-functional teams, along with mentoring and training frameworks, to ensure you have all necessary skills in place from the start
- An infrastructure and process that support rapid, high-quality delivery

- Regular project updates, ideally including the demonstration of deliverables
- Testing and review activities that enable feedback to be gathered and addressed well in advance of go-live
- Effective change control procedures that prevent vital new additions from negatively affecting budget, timescales and features

You can also increase your chance of success by employing an Agile project management approach. Through the delivery of working software at the end of every two-week development 'sprint', Agile enables you to gather stakeholder and user feedback on your product from the very beginning, making it easier to spot when you might be going off course and as such both reducing the likelihood of your project becoming unmanageable and minimising the risk of wasting investment on the wrong tasks. Additionally, with the most important features delivered first you can go live at virtually any point, reducing the chance of your project never being released to the public.

More important than getting the project successfully delivered, though, is ensuring that what is delivered offers real benefit to your organisation and, importantly, your users. The question of how beneficial a project is has been highlighted by leading analysts such as The Standish Group, who have introduced new criteria covering project value and alignment with strategic corporate goals into their annual CHAOS Manifesto, alongside the traditional attributes of time, budget and requirements completed.

As such, ensuring your project is focused on the most valuable opportunities is vital, and Agile can help here too, with its iterative approach enabling a backlog of tasks to be continually reassessed and reprioritised in response to feedback as well as changing market and business conditions.

At Box UK we're experienced in helping organisations rescue out-of-control projects, drawing on our expertise in Agile methodologies, software consultancy, user-centred design and quality assurance to support the successful

delivery of strategic software solutions that generate measurable return on investment. To find out more about how we might be able to help you visit our Project Rescue page or [get in touch](#) with us today.

Further Reading

[IT project failure: Read the early warning signs and act fast](#)

Oxford University and Simon Fraser University

[Chaos Manifesto 2013](#)

The Standish Group

[Pulse of the Profession: The High Cost of Low Performance](#)

PMI



Simply brilliant thinkers making software brilliantly simple



**Strategy &
Planning**



**User Experience
& Design**



**Software
Consultancy**



**Software
Development**

At Box UK our team of simply brilliant thinkers, consultants and application developers mastermind simply brilliant solutions to the world's toughest web and software projects.

It's why blue chip organisations around the world, from airlines to car manufacturers, international banks to global charities, choose to trust us with the execution of their digital strategies, from the development of enterprise-level custom software platforms to the delivery of award-winning multi-channel experiences.

So if you want an effective, compelling software solution, talk to us.

info@boxuk.com • +44 (0)20 7439 1900