

White Paper

Six Signs Your Software Is Failing You

Providing you with:

- ✓ Key indicators of software that's no longer fit for purpose
- ✓ An understanding of the upgrade options available
- ✓ Tips on defining an effective strategy for improvement



Introduction

Regular software reviews are crucially important for organisations of all industries and sizes.

Failure to undertake them puts your systems at risk of becoming inefficient, unreliable and insecure - which can have significant negative consequences for the performance of your business. A staggering nine out of ten IT decision makers claim that their legacy systems are preventing them from harnessing the technologies they need to grow and become more efficient, and 44% of large organisations claim that legacy issues affect most or all of their projects (source: Hitachi Consulting).

44%

But when conducting a review that is intended to inform the future of your software, how do you make sure you're using the right criteria? After all, you need to be certain that any subsequent investment will deliver tangible long-term returns, and that you haven't made your decisions in response to industry fads or ill-informed stakeholder requests.

In this paper we'll look at some of the key characteristics of software that's no longer fit for purpose, to help you identify exactly where you should be directing your IT budget.

And should any of the signs sound familiar, there's also advice on defining an effective strategy for improvement.



The technology it relies upon is outdated

The software that powers your core business processes is likely to be deeply embedded within your organisation and, as such, reliant not just on its own technology but that of a whole host of other systems too.

Not keeping up-to-date with changes to these systems (and their supporting technologies) can impact the performance, reliability and security of your primary system, as important fixes and enhancements are missed, or critical new dependencies overlooked. It's important therefore to stay aware of planned changes, to keep your software aligned and up-to-date.

Of course it may be the case that the technology underpinning your software stops being supported entirely; in these situations you may be forced to make rapid changes to keep everything up and running, or continue to work with the defunct versions - a risky move as you could well find yourself on your own if things go wrong. Keeping yourself informed about technology roadmaps will help mitigate this risk, by giving you the necessary time and flexibility to prepare a migration strategy.

Tips

- Appoint an individual or team to take responsibility for reactive upgrades. They'll need to research the roadmaps of any supporting technologies and regularly check for updates and out-of-date dependencies (in many cases, automated tools are available to help with this), before defining an appropriate course of action.

- Ensure time and budget is allowed for a rolling plan of upgrades that can take place alongside more strategic activity. Make sure all roles, responsibilities and tasks associated with this are made clear to minimise the risk of disruption.
- Optimise your setup by investing in additional environments where upgrades can be tested before they're made live, allowing you to test thoroughly yet safely. Automated deployment and testing capabilities can further speed up this process.

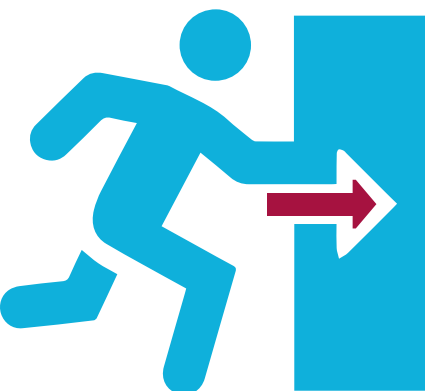


It was developed by a team that has since moved on

The issue of skills shortage is a very real concern among CIOs, with six in ten believing that the lack of available technology specialists will prevent their organisation from keeping up with the pace of change. With development skills a particular concern for 27% of CIOs (source: Harvey Nash), it's not surprising that when the team members who initially commissioned, built and maintained your software move on the management of your software system can become challenging.

Whether your software strategy is managed in-house or through a third-party supplier, losing team members who are experienced in working with your systems can have a significant impact on development productivity, not to mention developer morale. Extra time, effort and budget will likely be required to fix bugs as developers get up to speed with the system - if those bugs can be fixed at all. Without specialist knowledge and skills you may be unable to resolve complex issues; this can lead to the implementation of less-than-perfect solutions that contribute to the buildup of dangerous technical debt (more on this later), or result in serious performance problems and potentially fatal breakages if left alone.

Furthermore, if no one is available to work on fixing bugs, there probably won't be anyone available to extend or optimise your software with new features and functionality (something that's also covered in more detail later in this white paper). In these cases, choosing a new solution or rebuilding your software from the ground up allows you to select technologies with which your team is familiar, enabling you to manage on-going upkeep and enhancement more easily. Of course, you'll need to invest time and effort in ensuring that this knowledge is maintained, to avoid the same situation from arising again!



Tips

- Support knowledge-sharing among your team to avoid development silos. This could be achieved through a shared wiki (or similar internal tool), regular presentations and round-tables, or partner programming/mentoring initiatives.
- Consider linking up with external suppliers who can augment your team with specialist development expertise whenever it's required. Be sure to quiz them on the technologies that they support, and how they themselves manage the sharing of knowledge and keeping up-to-date.
- If you determine updates are required, take action swiftly. Not doing so could be less cost-effective in the long-term.



Hidden technical debt is causing performance issues

Often, the software underpinning an organisation (especially one that's large and well-established) has grown organically over time, as incoming additional stakeholders and new situations have introduced digital requirements and conditions. However, when software has been developed in this way it runs the risk of experiencing high levels of 'technical debt' - hidden issues and inefficiencies that can have a serious impact on the performance not just of your software, but your business too. Some studies suggest that there can be as much as one million dollars' worth of technical debt in the average software system (source: Deloitte).

Because the symptoms of technical debt often go unnoticed, you may not realise that anything's wrong until it's too late, and you've already started to experience problems when trying to scale, add new functionality or make other important changes. At this stage, the worst-case scenario is that a large-scale rebuild is the most sensible option available.

It's important to remember though that technical debt is often caused by shortcuts taken to resolve problems 'for now', without consideration for the longer-term development strategy. As a result, the major changes described above may be avoided if you ensure that any interim upgrades are of the highest possible quality - preventing the debt from accumulating in the first place.



Tips

- Avoid misunderstandings that can contribute to the buildup of technical debt by communicating any requirements clearly. Start with collaborative workshops and continue this shared understanding throughout the project with clear documentation that's centrally stored.
- Enforce stringent quality standards for all code in order to discourage the implementation of lower-quality temporary solutions. Many development languages have recommended standards, but it's also important to embed quality in your own processes by following practices such as test driven development and continuous integration.
- When introducing technical debt is unavoidable, make sure that it is logged. You can then factor known issues into your future development plans, and will also have a record of areas to investigate and optimise at a later date.
- Schedule regular code reviews so that any potential issues can be identified and addressed before they become more significant. It's a good idea to review your code every time something's checked in during development - balancing this with wider reviews every few months to make sure you don't miss out on the bigger picture.

4

There are usability and accessibility issues

As your software's main purpose is to support business processes efficiently and effectively, there is definitely a case to be made for an overhaul where it isn't doing this.

For example, when routine tasks are taking too much time away from revenue-generating activity it may be worthwhile to invest in streamlining the tools that power these processes, or exploring whether any tasks could be automated. Alternatively, if users are currently required to access numerous different systems to achieve their goals, you may want to look at integrating these in order to consolidate common journeys.

To identify the most valuable solutions for your particular requirements it's crucially important to listen to those in the know - that is, the people who are actually using your software. They'll be able to give you an indication of how well it's fulfilling its purpose, and by involving them in the process of defining a solution you'll also be able to secure greater buy-in for any upgrades you end up implementing.

Remember too that the usability and accessibility of your software can significantly impact adoption levels - so if it's not up to scratch you could be paying for something no one's using!



Tips

- Involve user experience specialists in the development process to ensure that user needs, requirements and constraints remain at the heart of your software. Be sure to conduct regular usability reviews too - even the most cutting-edge software will lose its edge over time and it's important to check in at intervals.
- Provide channels for users to share their thoughts about your systems (Usabilla and SurveyMonkey are two popular online tools). This explicit feedback can then be used support an iterative approach to development based on continual review and refinement, augmented with more implicit behavioural insight gathered from tracking tools such as Hotjar and Decibel Insights.
- Clearly communicate the impact of any upgrades so that users can quickly get up to speed. This might be in the form of release notes, newsletters or demos, depending on your audience and their level of technical skill.

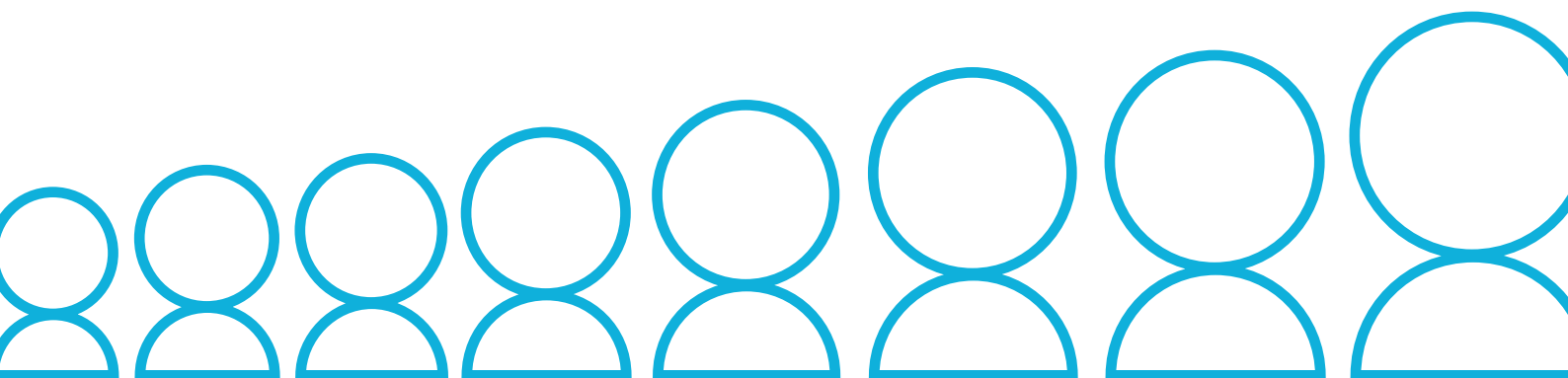


Your business has outgrown it

Even if your software was once suitable for your business requirements, that may no longer be the case. As time goes on it's natural for your business goals and objectives to change, and your software will need to evolve accordingly in order to stay relevant.

A common driver of this kind of evolution is organisational growth, as having more users and more data will put ever-more strain on your existing systems, causing them to slow down or even crash completely. Moving your systems to the cloud is a likely solution in this scenario; the inherent scalability it provides is proving increasingly popular, with 86% of organisations now spending at least part of their IT budgets on cloud services (source: Cloud Security Alliance).

Another important issue organisations have been grappling with for some time is that of mobility. Team growth can lead to greater levels of distribution, making an adaptable approach to the delivery of critical information and services a priority. This is especially pertinent when you consider the volume and diversity of devices on offer today, and the growing trend for Bring Your Own Device (BYOD) strategies, which are currently being planned or executed by an estimated 74% of organisations (source: Tech Pro Research).



Tips

- Align your development and wider business strategies to make sure you can keep up with planned growth and expansion. Ensure your tech team is kept abreast of any relevant arrangements so that you can make best use of their knowledge and skills.
- Look at flexible solutions such as cloud technologies and sophisticated content delivery platforms, so that you are better placed to rapidly respond to new demands.
- Don't forget security considerations if you're making sensitive information available remotely. This is another area where you'll want to formalise a policy with clearly-defined roles, responsibilities and processes, to ensure that there is sufficient capacity for monitoring and responding to issues.

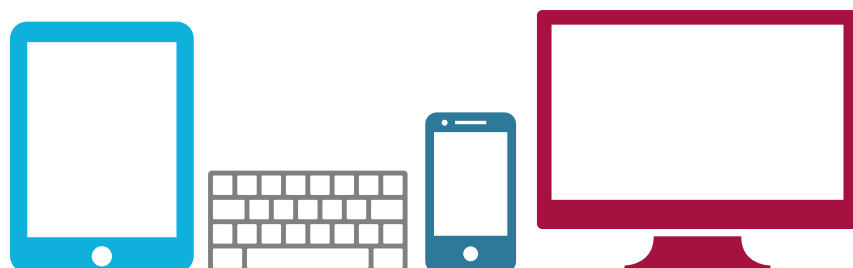


You're unable to take advantage of new advances

As your business grows, it's likely that your digital strategy will evolve too. If your software doesn't keep pace however you'll be left unable to leverage new trends and behaviours.

A desire to do more with digital is a common one, with 56% of UK organisations executing digital transformation as a co-ordinated strategic programme - and achieving profit growth that's two and a half times higher than those making only an average investment in digital transformation (source: CA Technologies).

Leveraging innovative new technologies is an obvious way to enhance your digital capabilities and extend your reach into lucrative new channels and platforms, so it's important to make sure that your software can support these kind of integrations. Not doing so could put you at a serious competitive disadvantage - particularly as organisations across all industries are currently racing to respond to the threats and opportunities presented by digital's disruptive potential. Only one in ten CIOs believe their organisation will remain unaffected by this digital disruption in the coming years, with more than a third reported to be grappling with the issue now and a similar amount expecting to have to manage it within 24 months (source: Harvey Nash).



Tips

- Formalise your commitment to exploring new trends and technologies, by providing digital teams with the time and budget needed to test - and learn from - new initiatives.
- Make sure all opportunities are assessed carefully though, to ensure they'll deliver lasting benefit and aren't simply a passing fad.
- Execute any agreed initiatives as part of a well-defined programme of work that incorporates business-as-usual too. Think about defining a model that balances these streams of work, embedding this approach into the culture of your team.

Next steps

If this white paper has got you wondering whether your software is holding you back from achieving greater business results, don't worry. Conducting a comprehensive systems audit will provide you with a complete list of current assets that can be analysed for the above risks - enabling you to identify areas for improvement and prioritise development initiatives according to business value. And if there are any upgrades to be made, hopefully we've shown that a wholesale replacement isn't the only option available to you. You could also:

- Make specific adjustments to give you the enhanced functionality, capacity and performance you need
- Give your systems a 'lick of paint' by refreshing front-facing elements such as user interfaces and APIs
- Explore existing Software as a Service (SaaS) and Commercial Off the Shelf (COTS) solutions - customising these as necessary to fit your specific requirements and business logic

At Box UK our business and technical analysts are highly experienced in guiding clients through these critical business decisions, to find the software solutions that will best help them achieve their goals. To learn more visit the [Software Consultancy](#) section of our site, and [get in touch](#) with a member of our team to find out what we can do for you.



Further Reading

[Legacy systems holding back 90% of businesses](#)

Hitachi Consulting (via Computer Weekly)

[2015 CIO Survey](#)

Harvey Nash

[Tech Trends 2014: Inspiring Disruption](#)

Deloitte

[Cloud Adoption Practices & Priorities Survey Report](#)

Cloud Security Alliance

[Wearables, BYOD and IoT: Current and Future Plans in the Enterprise](#)

Tech Pro Research

[Exploiting the Software Advantage: Lessons from Digital Disrupters](#)

CA Technologies



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