How to de-risk your technology projects including your **GRC systems**

As featured in IT Pro Portal & Information Age

Rich Eddolls, Chief Product Officer and Co-Founder

Introduction

Around 80% of IT projects are considered failures by businesses

Recent reports reveal that the success rate for IT and software projects remains alarmingly low. According to Gartner, **around 80% of IT projects are considered failures by businesses, often due to cost overruns, missed deadlines, and unmet expectations.** The Standish Group's Chaos report indicates that fewer than 1 in 3 software projects produce successful outcomes, with 66% ending in partial or total failure. This low success rate, based on an analysis of 50,000 projects worldwide, has remained largely unchanged over the years. Whatever the reasons for failure, it appears that project teams are not learning from their mistakes.

Of course, success is a relative term. It can be defined and measured in many ways and often depends on context – and on what the story needs to be. The Standish Group **defines success as projects being on time, on budget, and producing a satisfactory result, examining value, user and sponsor satisfaction, and meeting target requirements.** Regardless of how you define it, anyone involved in technical projects knows that far too many fail to deliver the intended benefits.

Extrapolating from these experiences, it's likely that billions of dollars and millions of hours are wasted annually on projects that either don't add value or end up being cancelled altogether. Clearly, there are significant gains to be made if we can avoid some of the common factors that contribute to project failure.

Some prerequisites for a successful project are well-established and obvious:

- **Getting the requirements right**
- **Providing effective leadership**
- **Ensuring full support and engagement from sponsors and users**

Without these in place, no project is likely to succeed. This article explores some of the less obvious ways to reduce risks to your technology projects.

Scope and timetable

This is a matter of methodology and development mindset. A purely waterfall or purely agile approach is rarely the best choice; the most effective method is often somewhere between these extremes. Understanding requirements and business benefits is essential, but spending months – or longer – creating reams of documentation is not the answer. Besides being difficult to digest, this documentation is often outdated by the time it's completed.

A word of warning: don't let project teams cherry-pick the easiest elements from each methodology, as this can become an excuse for skipping documentation altogether.
The ideal starting point is a set of fundamental requirements with enough detail to develop against. The rest can be delivered iteratively, ensuring that business benefits are not overlooked while realizing the key benefit of iterative approaches: engaging stakeholders and acting on their feedback.

Iterative doesn't necessarily mean agile. It's entirely possible to **have well-defined key requirements for each phase while proceeding iteratively,** although prioritizing requirements becomes essential. A major benefit of this approach is that the project scale becomes more manageable, and the timescales are more immediate, allowing for greater focus.

If the first deliverables for any project component are more than a few months away, you need to question your approach. You may be tackling the problem incorrectly, using inappropriate technology, or even addressing the wrong issue altogether – not everything has a solution rooted in technology. Clearly, the less time spent doing it wrong, the better, so aim to deliver something as soon as possible.

Delivering early doesn't just allow users to begin evaluating and providing feedback sooner; it also provides a usable tool for the business. **The sooner it goes live, the sooner the benefit is realized – and a fraction of the final benefit is better than none at all.**

How and what to deliver?

In house

Given the choice, many organizations prefer to develop in-house. This is usually because they believe internal projects will produce a solution tailored to their specific needs rather than one compromised by others' requirements, or because they think it will offer greater control or lower costs. However, these assumptions don't always hold up under scrutiny.

Recruiting and training new staff takes time and money, and there is always an opportunity cost. **Staff turnover** frequently results in the loss of expertise and project control. How many 'in-house' technology projects are managed and staffed by contractors? Managing third-party suppliers, bound by commercial contracts, can be easier than managing in-house teams, and **third-party vendors bring valuable experience, saving time and money while increasing the likelihood of a well-designed, futureproof product.**

> *If coding is required, the buyer should ensure they understand which elements are configurable and which require code-based changes.*

Go external

If the decision is made to go outside the organization, should the requirements be met with an off-the-shelf product, a bespoke solution, or a platform? While all products are customizable to some degree, it's rare for one-size-fits-all solutions to perfectly meet every organization's needs. Additionally, **the future direction of your solution will be at the mercy of the third party's product roadmap.** However, it's also rarely necessary to start from scratch – almost any new requirement can use common components, and if the work has already been done, it makes no sense to reinvent it. Therefore, **a platform-based solution, with reusable components and a custom business logic layer, often makes the most sense.**

By using such a solution, time and cost are saved, as is much of the risk inherent in new development. **If coding is required, the buyer should ensure they understand which elements are configurable and which require code-based changes.** This is not to say that coding is problematic, but it inevitably extends timeframes and increases project risk.

A platform-based solution, with reusable components and a custom business logic layer, often makes the most sense.

Designing and implementing the solution

When determining requirements, the capabilities of the technology should not be the starting point. The purpose of the technology is to support the best way of running your business; it should not dictate how the business should operate. If this is occurring, the first priority should be to change the technology, not to adopt suboptimal requirements and lower expectations.

Adequate testing is a non-negotiable element of any technology project, yet an alarming number of software vendors lack a formal testing function. While some features can be tested automatically, most require a dedicated test team. Testing activities should mirror development efforts, with testing occurring throughout the project and extending beyond its completion. **If testing is left to the end, as in traditional methodologies, delays in production can lead to cuts in testing time, increasing the risk of a flawed end product.** Additionally, there is less opportunity to identify design flaws or missing requirements early on – User Acceptance **Testing (UAT) alone is not a sufficient testing methodology.**

Prioritize simplicity and performance

The success of a technology project depends on more than just its technical components. Developers often view external elements as just 'cosmetic', but **the user experience is crucial to success**. This doesn't just mean generating wireframes and design guidelines; it also involves **considering storage, network requirements, and overall performance before starting**. The key should be that if users have to wait more than a second or two for information to load, there needs to be a valid reason for the delay, with consideration given to how it affects their experience.

Ultimately, **a journey through the product should be smooth and intuitive,** with tools and alternative routes logically placed without being intrusive. While the process itself might be complex, completing it should be as simple as possible. This is usually the rationale behind the project in the first place: **simplifying and improving the efficiency of a process is what adds value.** Remember, developers are experts in software development, not in user experience, and should not be responsible for this aspect of the project.

In summary, successful projects will:

1. **Focus on delivering early rather than extensively scoping out requirements.**
2. **Choose a platform solution with reusable components and flexibility for custom business logic.**
3. **Ensure requirements drive technology choices, not the other way around.**
4. **Incorporate continuous testing throughout the development process.**
5. **Prioritize making the user experience as intuitive and enjoyable as possible.**

5.0
FAQ

FAQ

What are the key risks associated with GRC technology projects?

- Key risks include **scope creep, unclear requirements, lack of stakeholder engagement, and inadequate testing.** Managing these risks is essential to ensuring that GRC technology projects deliver the expected benefits and adhere to governance and compliance requirements.

What is the relationship between GRC technology and project testing?

- **Testing is critical** to ensure that the GRC technology meets compliance and regulatory requirements. Continuous testing throughout the project's development ensures that the final product is robust, compliant, and risk-free, **mitigating the chances of failure or delays.**

Why is it important to choose the right GRC platform for a project?

- Choosing the right GRC platform is crucial because the wrong platform can lead to inefficiencies, increased risk, and failure to meet compliance standards. **A well-chosen platform provides the flexibility, scalability, and integration capabilities** needed for successful technology project execution.

What are some common mistakes when implementing GRC technology in projects?

- Common mistakes include inadequate planning, not engaging key stakeholders early on, insufficient testing, and underestimating the complexity of integration with existing systems. Additionally, failing to **continuously assess risks** throughout the project can lead to missed compliance or governance gaps.

Looking to Elevate Your GRC Initiatives?

Want to learn more

Join our expert-led, 1-hour Governance, Risk, and Compliance (GRC) workshop—tailored specifically to your team's needs. Whether you're at the beginning of your GRC journey or managing a mature program, there's always room to elevate your impact.

This value-driven session is designed to challenge your thinking, spark innovation, and help your organization unlock new potential in its GRC practices.

Book your workshop here

About Rich

Richard is a co-founder and Chief Product Officer at CoreStream GRC, where he's redefining the way organizations approach governance, risk, and compliance. With 20 years of experience in business-driven GRC system design and a background at Deloitte, Richard is all about challenging the status quo and delivering technology that actually works. As the visionary behind the CoreStream GRC platform, he's committed to building solutions that don't just promise change, but deliver it.

Follow Rich on LinkedIn here

About CoreStream GRC

The intuitive, flexible GRC platform that delivers efficiency and value – your way.

Driven by the belief that technology should be an enabler—not a barrier—we created the CoreStream GRC platform: a flexible, no-code solution that empowers organizations to design their perfect GRC system with our expert team. You tell us what you need, and we deliver it—quickly and without unnecessary complexity. Using pre-built, customizable features, it's as intuitive and versatile as building with Lego bricks – the solutions are limitless.

With seamless scalability, an intuitive interface, and rapid implementation, CoreStream GRC turns GRC from an administrative burden into a powerful enabler for your business. Trusted by leading organizations like the BBC, Deloitte, NHS, PwC Middle East and Shell Energy, CoreStream GRC consistently delivers real, measurable value for all your risk, and compliance management needs.

Learn more