# Improving Success
# **For Agile and DevOps Projects**

blueprint

> *43 % of respondents felt their* ***organization’s agility*** *was low – up from 39% in 2015.*

- Project Management Institute

# The Hard Truth About **Failure**

Every organization faces the unfortunate reality that, even with diligent planning and work, some software projects fail. It's a fact of life for businesses of all sizes and every industry.

Software projects have grown in visibility and are now the core driver of corporate competitiveness and innovation. Nevertheless, failure rates persist.

Recent data from the Project Management Institute (PMI) indicates that project success rates are currently at a 4-year low.

The main problems are all-too-familiar:

- Changing priorities,
- Unclear vision, and
- Poor requirements.

Each of these problems reflects poor Business-IT alignment.

# The Pillars of Organizational Maturity

When companies examined other measures of organizational maturity – strategy alignment, benefits realization and agility – the results were equally discouraging. In all three areas, organizational maturity has either stayed the same or decreased.

Despite valiant efforts to move the needle positively, organizations continue struggling with these business-critical best practices:

> *Without the ability to quickly respond to threats and opportunities, organizations can't keep up with the competition, much less get ahead of it.*

## Aligning efforts to business strategy.

Without strategic alignment, successful projects are often irrelevant. The PMI study found that less than half of respondents (44%) believe their organizations have a high degree of business strategy –IT project alignment. The research also found that, compared to prior years, fewer projects are strategic initiatives. In 2016, only 48 percent of their projects were designed to "achieve formulated strategy," down from 54 percent the previous year.

## Measuring benefits realized.

PMI emphasizes the importance of this best practice: measuring benefits is what helps organizations determine the return on their investments, as well as identifying the intangible benefits such as improved customer satisfaction. However, the study found that only 17 percent of survey respondents reported high degrees of maturity in realizing benefits, down slightly from the prior year.

## Organizational agility.

Companies are increasingly leveraging the strengths of Agile methodology. Nevertheless, surveys are finding that companies are still not making measurable improvements in the ability to respond and adapt quickly in response to market changes or other external influences. 43 percent of survey respondents characterize their organizational agility as low – up from 39 percent in 2015. Without the ability to quickly respond to threats and opportunities, organizations can't keep up with the competition, much less get ahead of it.

# Align. Measure. Pivot.

The trend toward failure may be discouraging, but there is a way forward. Specifically, organizations must find ways to improve business-IT alignment, not just by addressing the common causes of project failure, but by finding ways to:

- **Align projects with business strategy;**
- **Measure the business benefits delivered by technology solutions; and**
- **Pivot with agility as the ecosystem changes.**

Enterprises have increasingly turned to Agile and DevOps practices to reduce the failure rates of large scale projects. Many organizations have found that while Agile and DevOps is a practical solution for smaller scale projects, the complexity of scaling larger enterprise class projects can test the limits of these frameworks.

Without a sophisticated process for defining and managing requirements, Agile and DevOps can contribute to increased failure rates in large enterprises.

# The Importance of **Requirements**

Why do we see increased rates of failure?

It always comes back to one fundamental thing: ***the understanding and alignment of requirements.***

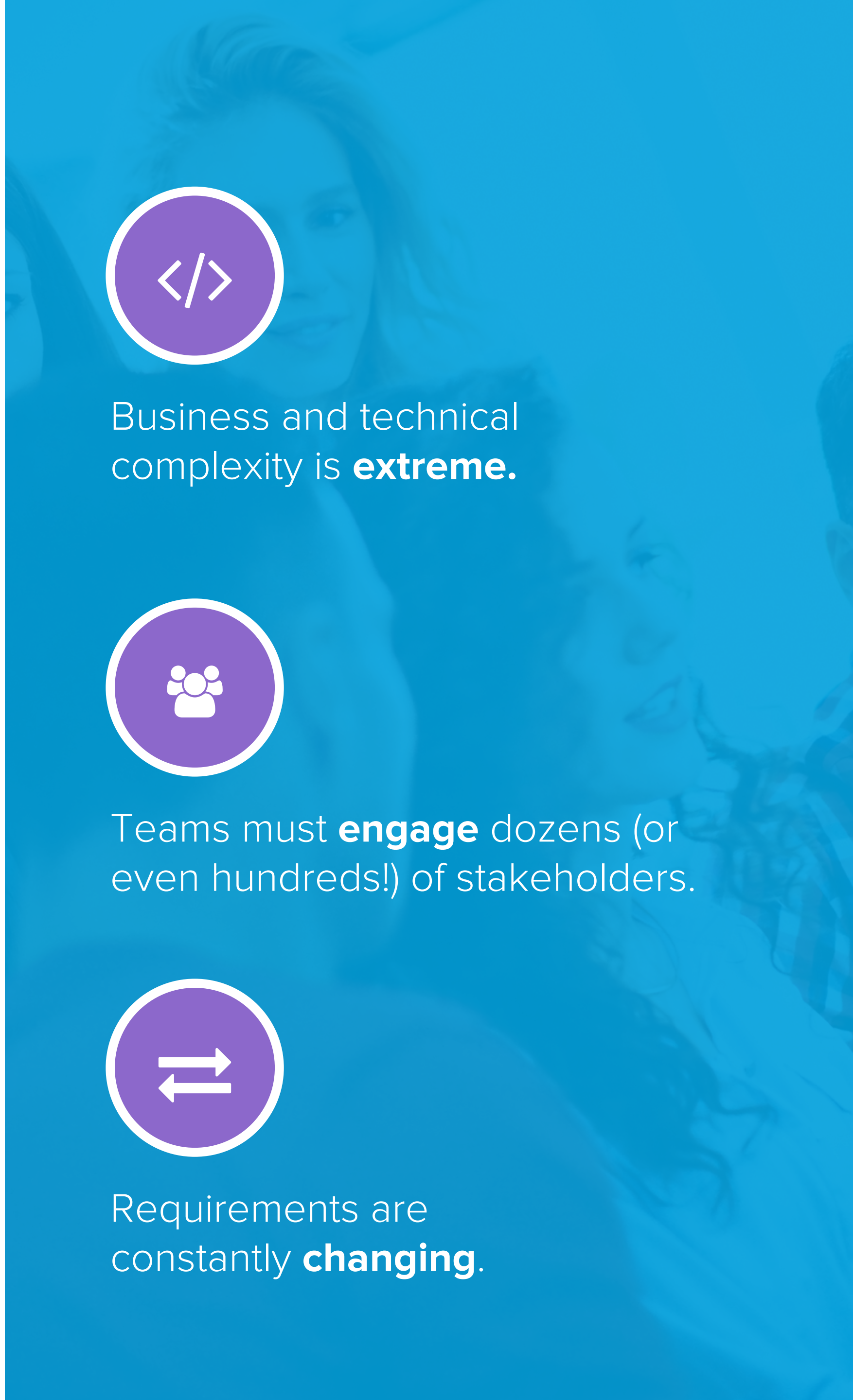
**Failure to effectively define and manage software requirements is the biggest driver of project failure.**

Indeed, research shows that project failure is often due to an inability to define high-quality requirements and manage them effectively as priorities and objectives change.

For experienced professionals, this is no surprise. For example, a recent Bloor Research report stated: “Requirements are the bedrock upon which all software development and testing is founded.”

When we look at smaller scale Agile and DevOps projects, those toolchains lack solutions for easily capturing, sharing and automating critical business requirements.

If requirements are bad, the rest of the process is doomed. However, getting them right is complicated, especially in large organizations due to the following realities:



Business and technical complexity is **extreme.**

Teams must **engage** dozens (or even hundreds!) of stakeholders.

Requirements are constantly **changing**.

# The Importance of **Requirements**

**Business and technical complexity is extreme.** Businesses operate under the stresses of heightened competition, mergers and acquisitions, and compliance demands. IT has to support large portfolios of new and legacy technology solutions, without fail, while keeping up with business change. A lot of research and analysis is required to understand what's needed to support the business and how to deliver it effectively.

**Teams must engage dozens (or even hundreds!) of stakeholders.** Consider this list: Legal, Finance, Human Resources, Marketing, Sales, Operations, Compliance, and Security. And let's not forget the C-Suite with its demands on outcomes and timelines. Good requirements demand input into these and other groups, but they're often scattered across the globe, and busy schedules make them hard to get to at the right time. Requirements suffer when their participation is absent.

**Requirements are constantly changing.** Companies need to accommodate change coming from many directions, including changes in strategy, technology and regulation. It is difficult to define good requirements upfront, but it is even more challenging to keep them up-to-date as business and technology evolve. Agile and DevOps approaches have promised relief, but neither has fully delivered.

These challenges are well understood by software development and support professionals. So why have we not found ways to overcome them?

> *"Requirements are the bedrock upon which all software development and testing is founded."*
>
> *-Bloor Research*

# Effective **requirements management** demands technology to support it.

Bloor analysts Philip Howard and Daniel Howard have stated that a collaborative requirements tool is “obligatory” to keep the business and IT in sync and improve requirements quality, particularly in large, complex environments.

The consulting firm recently published a Bloor InDetail Report describing the top capabilities companies should look for in a requirements solution.

Specifically, they highlight the need for:

- Cross-company collaboration
- Traceability and change management
- Process automation
- Integrations with other Agile ALM and DevOps tools

# Effective **requirements management** demands technology to support it. *(cont'd.)*

**Cross-company collaboration.** Collaboration is critical but difficult to support. Bloor's analysts write that capturing feedback from reviews can get "very fiddly" without the right tools, and "it becomes completely impossible to keep track of not only whether a user story was approved (or not), but why it was approved (or not)." (If you've ever sat in a room filled with business stakeholders asking "Why did we decide that?", you know what they mean).

Collaborators should be able to view information in context and provide input in an easily accessible way. And the history of that collaboration is valuable corporate knowledge that should be preserved in a centralized repository.

**Traceability and change management.** Given complexity and the need for speed, detailed traceability from business objectives to requirements, along with tests and the test results, is mandatory to analyze and manage the impact of changes.

According to Bloor: "A small change in one requirement could break something, say, seven layers removed from it." If your company operates in a highly-regulated environment, this is an important consideration for achieving, maintaining and proving compliance.

**Process automation.** Requirements are information-rich documentation of business needs and constraints that directly inform the downstream work of the development, testing and operations teams. Tools that draw on requirements to accurately auto-generate downstream work items – like user stories, test cases, or end user documentation – dramatically improve both productivity and quality. They also build traceability and impact analysis into the Agile and DevOps lifecycle – something governance, risk and compliance specialists depend on.

**Integrations with other Agile ALM and DevOps tools.** We are in a best-of-breed ALM/DevOps world where each team should be able to collaborate with other teams from the tools they use every day. However, organizations cannot afford to create those integrations manually. Information needs to flow freely in both directions across the toolchain to gain speed without losing quality. To enable this, teams need a requirements solution that easily and flexibly integrates with the breadth of productivity and SDLC solutions on the market today.

# **Enterprise DevOps** Will Never Work Like a Software Delivery Assembly Line

DevOps has become a high priority for all CIOs, but there is a lot of confusion about what it is and how to make it work in enterprise IT.

Some industry professionals focus on the cultural aspects of the DevOps movement: **While development and operations teams traditionally worked in siloes, with DevOps they should be in constant alignment and all concerned about the holistic quality of a product from creation to end-of-life.**

Others focus on the technology needed to enable DevOps, which relies on integrating the various people and processes involved in software development and operations. You hear the words “continuous” and “automated” frequently, which conjures up images of a software “factory” – a long assembly line of business analysts, developers and testers who are constantly producing and delivering version releases. This analogy doesn’t work for enterprise IT.

Why? Because an assembly line is a linear progression of activities creating something piece-by-piece until it is complete. Each person has a unique specialty – a task or set of tasks they must accomplish, before the product is handed off to the next in line. Their work is orchestrated by an integrated set of tools that controls the overall process. Quality assurance is often automated to ensure defects are discovered early and reduce risk.

The end-to-end development, delivery and support of software in enterprise IT will never work like a software delivery assembly line.

First, it lacks predictability and uniformity. Every project and product is different.

Second, numerous stakeholders are involved, and business needs change mid-stream. While people have specialties, their collaboration is critical to solving complex problems. Agile approaches, in particular, rely on this cross-functional collaboration to speed delivery and react to change.

# **Enterprise DevOps** Will Never Work Like a Software Delivery Assembly Line *(cont'd.)*

Thirdly, an assembly line depends on automation. But in today's large organizations, the many people involved in software development and operations work in disparate toolsets that don't talk to each other. Business people work in e-mail and Microsoft Office, developers work in Agile lifecycle management (ALM) tools, and testers work in test management systems. Business analysts may have the advantage of working in a purpose-built requirements management solution, like Blueprint and Storyteller, but often, they are still stuck in Microsoft Office. And operations teams have their own set of tools to manage the work they do.

The fact is that to achieve DevOps success and realize its benefits, organizations must find a way to integrate the people, processes, and tools across a diverse, distributed, complex ecosystem. This is difficult to achieve. Both cultural and technology change must be managed. And as with any other large productivity and quality improvement initiative, CIOs have to take an incremental, pragmatic approach to accomplishing their DevOps goals.

## STEPS TO SUCCESS

Among other steps you can take to enable these best practices, we see many of our customers finding success by:

- Establishing traceability from strategy to software requirements and further downstream, so the investment of efforts aligns with the most important business priorities.
- Designing, collecting and analyzing business value metrics from production to calculate ROI and inform subsequent planning cycles.
- Investing in best-of-breed tools to create a continuous, automated BizDevOps toolchain. This is a "shift left" from DevOps to embed the business firmly into a continuous chain of innovation.
- Automating business-IT collaboration and up front business-driven release planning activities.

**Blueprint's** leading application lifecycle management solutions are key enablers of continuous delivery and DevOps success. The addition of Blueprint to their toolchains helps organizations improve their **Agile, Lean and DevOps practices.**

Blueprint provides industry-leading solutions that accelerate and de-risk the digital transformation of large organizations. With our products - Blueprint Storyteller for Agile, Blueprint Automate for DevOps and Blueprint RegTech for Compliance - organizations receive greater business value from IT, faster and more frequently, while dramatically increasing the efficiency and confidence of compliance.