Lowering business costs: Mitigating risk in the software delivery lifecycle

Roberto Argento
IBM Rational Business Development Executive

Valerie Hamilton
IBM Rational Solution Marketing Manager and Certified Senior IT Specialist

Contents

Introduction

In today's challenging economy, businesses are under increased pressures to reduce overall costs while mitigating and controlling risks. The reality is that many businesses continue to struggle to successfully deliver quality software on time and on budget that meets their market needs. One study states that, on average, 32% of projects do not align with company goals and 39% of projects are over budget.[1] Such surprising statistics only touch the surface of the software delivery problems faced by businesses today.

Other studies confirm that success rates are not improving; software delivery challenges continue to adversely affect business costs. According to the Standish Group, businesses continue to struggle with project success rates and failures. Their latest published Chaos Report[2] states that "this year's results show a marked decrease in project success rates, with 32% of all projects succeeding which are delivered on time, on budget, with required features and functions". Additionally, "44% were challenged which are late, over budget, and/or with less than the required features and functions and 24% failed which are cancelled prior to completion or delivered and never used."[3] The numbers are staggering.

Why is it so difficult to deliver quality software on time and on budget that aligns to the business goals? Imagine if "wasted" resources and dollars lost on software delivery failures could be used to innovate – to create new products or to pursue new market opportunities. Think of the positive impact to the business's bottom-line. There are plenty of businesses that, by mitigating risks and having a systematic (but not bureaucratic) approach to the software delivery process, have achieved improvements in their success rates leading to a decrease in their development costs and a better and greater impact to their business outcome. In this paper, you will learn how requirements and performance management can help to reduce risks and lower business costs by:

- Achieving consensus on project requirements and delivering quality products that meet stakeholder's business needs
- Understanding the impact of changes to your deliverables and having the ability to accurately track and communicate these changes
- Providing the ability to analyze, monitor, and trend project status across the entire software lifecycle for objective and informed decision making
- Assessing and measuring process and project performance to drive continual process improvements

Defining software delivery risks

In everyday life a risk is an exposure to loss or injury: A factor, thing, element, or course involving some degree of danger. Similarly, in software development a risk is something that can compromise the success of a project. Examples of potential sources of risk in software delivery are listed below:

- Missing, incomplete or changing requirements and specifications
- Manual or undefined development processes and practices
- Incomplete testing and validation
- Lack of stakeolder input
- Disconnected distributed teams
- Project interdependencies and lack of team collaboration
- Changes to regulatory compliance
- Insufficient time or schedule expectations
- Unclear business goals and objectives

Highlights

Making informed decisions by consciously assessing what can go wrong, as well as the likelihood and severity of the impact, is at the heart of defining software delivery risks.

Making informed decisions by consciously assessing what can go wrong, as well as the likelihood and severity of the impact, is at the heart of defining software delivery risks. Risks are exposed and can be addressed through the identification, analysis, planning, tracking, controlling, and communication of the software delivery lifecycle.

Balancing risks with value

Risk is a potential manifestation of the likelihood of unexpected things happening to the project, and risk stands in the way of value creation. Every project carries an amount of risk. If project risks are not identified and addressed, there is a very real chance that they will arise at some point in the project lifespan. The longer a risk goes unnoticed, the more costly it is to correct. Risk is directly proportional to uncertainty in estimates, and stakeholders typically want to know sooner rather than later what value the project can deliver in the stipulated time. In many cases, you reduce risk when you create value by implementing and testing the most critical capabilities.

Risk-driven development focuses on identifying high risk areas and addressing those items early in the project lifespan. The goal is to flush these issues out early before you spend a considerable amount of time on project development and any re-work is required.

Highlights

Value-driven development focuses on identifying those features most important to the customer and business, and completing those items early so the stakeholders can see progress as well as provide feedback on those features early and on-going.

Risk-driven development recommends focusing on those high risks areas that are critical to the success of the project, while value-driven development focuses on those functional requirements perceived to be of high value to the stakeholders.

The risk response requires careful balancing of competing priorities to maximize stakeholder value.

Risk response should be in proportion to the significance of the risk. However, there are situations where risks and business value are at odds with each other. Thus, the risk response requires careful balancing of these competing priorities to maximize stakeholder value. A risk-driven iterative approach involves iterations that strike a balance between high-risk issues crucial to the project and features that bring value (see Figure 1).

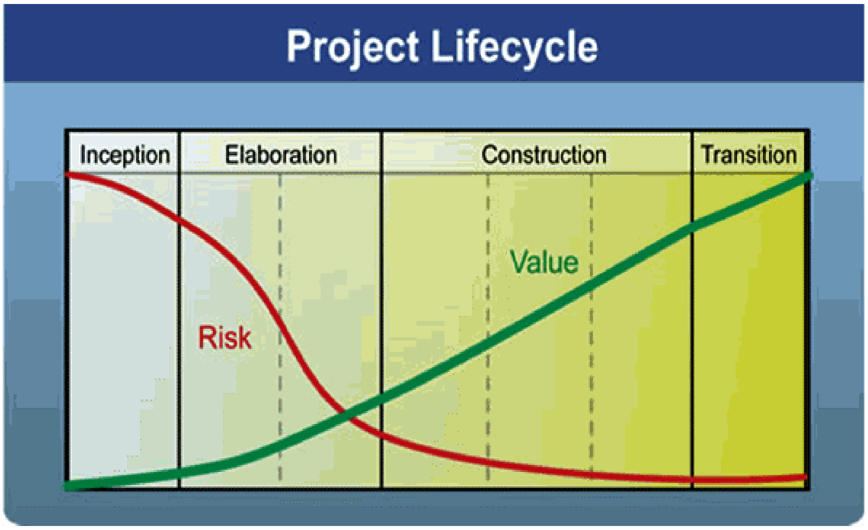

Figure 1: A risk-driven iterative approach balances risks with value

In reviewing the project lifecycle phases (Inception, Elaboration, Construction and Transition) shown in Figure 1, risks are normally higher in the Inception phase, and they continue to decline over subsequent project lifecycle phases as risks are identified and addressed. In the Inception phase the initial risk assessment is performed and core project requirements and key features are defined. In the Elaboration phase, the primary objective is to mitigate the key risk items identified by analysis. The Elaboration phase is where the project starts to take shape: the problem domain analysis is made and the architecture of the project gets its basic form. This is the phase where there is a balance between risk and features that are perceived to be of high value to the stakeholders. Creating value by reducing uncertainty is the best way to reduce costs and improve project success.

The cost of failure

Software delivery is a team-based activity. In order to avoid the high cost of failure teams must work together to mitigate and manage risks for successful outcomes to the business. To avoid the high costs of failure and reduce risks, there must be repeatable, predictable processes and best practices in place throughout the software delivery lifecycle that improve productivity and quality.

Typical software delivery activities include requirements elicitation, planning, design, implementation, testing, documentation, deployment, and maintenance. With so many activities of the software delivery lifecycle, it is no wonder that failures frequently occur.

"Requirements elicitation" refers to those activities focused on understanding your stakeholders' business goals and objectives, which serve as the starting point for gathering project requirements. Only when you understand the business goals and objectives can you begin defining a solution that will solve stakeholders' business needs.

The planning phase within the software delivery lifecycle involves the important task of defining the requirements or requirements analysis. Stakeholders typically have an abstract idea of what they want as an end result, but not what the software should do. Project requirements need to be accessible so that all team members understand their part of the whole project and are informed as changes occur.

Highlights

Stakeholders typically have an abstract idea of what they want as an end result, but not what the software should do. Project requirements need to be accessible so that all team members understand their part of the whole project.

Highlights

Missing requirements is one of the biggest contributors to schedule delays and budget overruns; there's nothing worse than being half way done with a project and then realizing a major stakeholder need has not even been addressed. Incomplete, ambiguous, or even contradictory requirements need to be analyzed and extracted prior to defining the scope of a project.

Making changes to a deployed system is much more costly than getting it right the first time. The immense cost associated with reworking existing code is one of the key areas of risk in the software delivery lifecycle.

Requirements analysis and management is a high-leverage activity because making changes to a deployed system is much more costly than getting it right the first time. The immense cost associated with reworking existing code and tests when requirement errors are identified is one of the key areas of risk in the software delivery lifecycle.

Measuring project and process outcomes allows for continual process improvements across the software delivery lifecycle, which reduces project failures and lowers business costs.

Organizations must have the ability to review, assess, and improve current processes within each phase of the software delivery lifecycle. Having accurate insight into enterprise data, processes, and practices is a key component of success. However, many organizations suffer from a lack of visibility into the software delivery lifecycle. Additionally, they struggle with data gaps and manually created metrics, which creates data blind spots that put an organization's success at risk. Measuring project and process outcomes allows for continual process improvements across the software delivery lifecycle, which reduces project failures and lowers business costs.

Building the right thing: Requirements management

Requirements management is the process of eliciting, defining, elaborating, documenting, analyzing, prioritizing, and agreeing on requirements and then controlling changes over the course of a project to ensure that deliverables remain aligned to the stakeholders' goals. Requirements management thus helps ensure that risks are minimized, stakeholder problems are solved, and business costs are lowered.

What is so hard about requirements management? Words can be ambiguous. It is difficult to clearly communicate a vision to all team members. It is complicated to show relationships or dependencies between requirements. Requirements come in a variety of shapes and sizes. Some may be more complex, more risky, more important, or less stable than others, and the sheer number of requirements can be grueling to manage. But most of all, requirements frequently change.

Understanding the impact of changes to other requirements and to your project, then tracking and communicating those changes to all team members, is one of the greatest challenges in the requirements management discipline.

It is clear that doing things right can take us down a path of risk mitigation and value creation. But how do we know when we are doing the right things?

When evaluating fourteen software development and delivery projects in six countries of Latin America (Mexico, Brazil, Chile, Argentina, Peru, and Colombia), in different sectors of the economy (finance, insurance, telecom, ISVs, government agencies, and energy) and covering several technology platforms (mainframe, Java development, creation of new products, and ERP maintenance), we discovered a common problem: a pervasive lack of effective requirements management.

Each of the projects presented a variety of requirements issues, including the inability to properly test or validate functionality, rework and churn due to constant change of features and functionality, and high development costs. The fact was that all projects invested very little time in planning and managing their requirements.

The conclusion of this project evaluation is that most organizations are very good at proposing new ideas and solutions. But they are commonly ineffective in planning and managing requirements and executing software delivery with reduced risks and costs.

Highlights

Most organizations are very good at proposing new ideas and solutions. But they are commonly ineffective in planning and managing requirements.

CIO Magazine reported that "as many as 71% of software projects that fail do so because of poor requirements management, making it the single biggest reason for project failure."[4] Domain experts such as Rikki Kirzner cite the need for requirements management for project success. He was quoted in a recent article:

> *The successful outcome of any project starts with defining, creating, and distributing project requirements in a form where all stakeholders can easily access, collaborate, and contribute to creating requirements. While this seems simple enough to explain, time and again, good intentions are undermined by the capabilities of the tools you are using and the environment in which you are working. This situation only exacerbates the problems encountered when all the stakeholders don't understand the original business intent, goals, or cannot connect those goals with what the team is doing. Geographic dispersion, communication difficulties and conflicting priorities can also cause problems.*[5]

Highlights

One of the core principles of requirements management is promoting team collaboration. It is imperative to make the requirements accessible to everyone on the team, thus facilitating team collaboration and providing visibility to the impact of change.

IBM has found that the investment in a requirements management solution, such as IBM Rational Requirements Composer, IBM Rational RequisitePro, or IBM Rational Doors facilitates team collaboration and helps implement requirements management best practices. This in turn reduces costs and helps mitigate risks. Rational Requirements Composer is a collaborative requirements definition and management solution, perfect for teams that want to focus on definition and need "just enough" requirements management. It provides textual and visual techniques to elicit, elaborate, and validate requirements collaboratively with your stakeholders, and it helps deliver quality solutions that meet their business and product requirements. Both Rational RequisitePro and Rational Doors provide traceability and impact analysis that keeps analysts, architects, developers, and testers aligned to business needs and requirements throughout the software delivery lifecycle.

Visibility into project and process outcomes: Performance measurement and management

Performance measurement and management helps software delivery teams make the right decisions by providing deeper insight into an organization. It enables organizations to measure, monitor, and analyze project and process performance against business objectives for trending, to identify issues, and to drive continuous process improvement.

When data is gathered manually, when it comes from different sources that may not align, or where there is an inability to gather information from the various teams (thus creating data silos), it is nearly impossible to make the right decisions for an organization. It becomes hard to react to organizational and market changes, and to measure the effectiveness of processes and practices to improve them. And of course, this ineffectiveness can directly impact costs and business risks.

Highlights

You cannot improve what you cannot measure. Real-time information in a single view across various projects, products, and geographical boundaries helps project teams to review, assess, and improve current processes.

Measuring organizational effectiveness improves processes and practices, which not only increases efficiency and speed, but allows quick detection of problems.

You cannot improve what you cannot measure. Raising the visibility into an organization by having real-time information in a single view across the various projects, products, and geographical boundaries helps project teams to review, assess, and improve current processes within each phase of the software delivery lifecycle.

Having access to the right data at the right time helps organizations make objective and informed decisions. They are able to measure organizational effectiveness to improve processes and practices, which not only increases efficiency and speed, but allows for quick detection of problems and continual status monitoring for reduced costs and risks.

IBM Rational Insight, a performance measure and management solution, allows organizations to measure, monitor, analyze, and trend project and process performance as well as continually improve organizational outcomes through process improvements. It eliminates inefficient time wasted on manual data collection through automatic collection of data presented in a single view. Rational Insight also helps deliver the IBM Measured Capability Improvement Framework (known as MCIF) for process improvement and best practice guidance. This aligns guidance with performance measurement for faster improvements to an organization, thus lowering costs while raising team capabilities.

Highlights

Conclusion

Lowering business costs by mitigating risks in the software delivery lifecycle is not simply an aspiration among businesses today. During these challenging economic times, it is imperative to improve the success rate of projects with reduced costs. Improving the success rate of projects requires that teams focus on business outcomes, become as productive as possible, and mitigate risks with proven tools and techniques. This requires the right mix of information, collaboration, and automation — all at the right time.

Successful leaders will leverage innovation to help them seize business opportunities, manage to objectives, and execute with reduced cost and risk.

Successful leaders will leverage innovation to help them seize business opportunities, manage to objectives, and execute with reduced cost and risk. To drive business transformation, companies need greater agility and responsiveness from their teams. They demand greater differentiation in their products and services in order to surpass the competition and increase profits.

Requirements management helps achieve consensus with your stakeholders. It focuses on the ability to deliver quality solutions that align project deliverables to stakeholder business needs.

Performance measurement and management helps to monitor, analyze and trend process, resources, and project outcomes to drive execution of projects and processes with consistent and predictable results.

Doing the right thing means to be able to:

- Reduce risks across the software delivery lifecycle through team collaboration to clarify actual business issues, increase individual and team productivity, compress development cycles, and rapidly deliver high-quality software
- Make better decisions based on real-time, accurate information and consensus across stakeholders
- Drive faster project delivery by providing requirements collaboration, history and context that team members need to reduce rework, and allow for efficient reuse of requirements artifacts for future projects
- Iterate quickly by analyzing and managing change for continual alignment with business priorities.

Endnotes

1. IDC Whitepaper: *How Project and Portfolio Management Solutions Are Delivering Value to Organizations*, September 2008
2. Standish Group Chaos Report, CHAOS Summary 2009
3. “The Standish Group” - http://www.standishgroup.com/newsroom/chaos_2009.php
4. Christopher Lindquist, “Fixing the Requirements Mess,” in CIO Magazine, Nov. 15, 2005
5. Rikki Kirzner, “*Requirements Management: A Core Part of the Development Lifecycle*,” DevX.com, Aug 8, 2008, at http://www.devx.com/ibm/Article/38819/4047

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
August 2009
All Rights Reserved

The information contained in this document is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind, express or implied.