

Chapter 2

Anatomy of an Enterprise Software Delivery Project

Chapter Summary

I present an example of a typical enterprise software delivery project. I examine its key characteristics and analyze specific areas for improvement. I explore both the enterprise-level and project-level issues within this specific example, and conclude the following:

- A systematic approach to enterprise-level improvement is essential to achieve the benefits over short-, medium-, and long-term periods.
- Substantial project-level improvements can be made by focusing on several critical application life-cycle management areas.
- Optimizing the global work force requires a balance of resources and a focus on collaboration across people, assets, and processes.

2.1 Introduction

To provide the context for this book, it is important to have a clear view of the current challenges faced in enterprise software delivery. To enable this view, I take a specific enterprise software delivery organization and describe how it executed a particular enterprise software delivery project. After discussing the project's critical elements, I analyze where and how improvements to that project are possible.

While many facets of this real-world enterprise software delivery project are of interest, I choose to highlight four areas that amplify key themes I will address in detail throughout the book:

- Collaboration across distributed teams, particularly when those teams span geographic, organizational, and company boundaries. In such projects, we frequently find that inefficiencies and misunderstandings are a major source of errors, frustration, and waste. Clear approaches to communication and coordination greatly improve the team's performance.
- Agility in the enterprise delivery team's means of organization and operation, which enables effective delivery, and in its interaction with the business. A project's progress and delivery rhythm is based on how it reacts to changes in its goals, context, and delivery environment.
- Continuous quality assurance, which provides superior solutions using efficient, predictable techniques. Late software breakages are often a result of misunderstood requirements, unstable architectures, and poor communication across the team. They prove very costly and highly disruptive to the project's success. Early and continuous attention to quality is essential.
- Governance and metrics to provide insight into a project's status and ensure the project stays on track. Lack of governance can lead to chaotic behaviors. Furthermore, good governance approaches based on poorly defined metrics schemes can result in suboptimal decision making. Visibility and traceability across the project are essential to manage the project complexity.

In this chapter, I establish the context for this enterprise software delivery project, and I then explore these key themes within this real-world example. I conclude with a number of observations about what we can learn from this study.

2.2 MyCo and the MyProj Enterprise Software Delivery Project

Drawn from a real-world project, this case study is illustrative of a wide variety of situations. A multinational utility company is supported by an enterprise software delivery organization under significant cost and efficiency pressures to deliver IT services to the business. For ease of reference, I'll call this company

MyCo, and I'll name the project we'll examine *MyProj*.¹ Faced with many challenges over the years, MyCo has invested in several different facets of its enterprise software delivery, making the following changes:

- From very siloed, geographically dispersed solutions and technologies to centrally driven technology platforms
- From wholly onshore to a mix of onshore, offshore, and outsourced application development and support
- From globally dispersed in-house-managed to outsourced IT infrastructure services
- From ad hoc vendor engagements to use of preselected vendors for defined domains
- From undefined core software delivery processes to the implementation of common, centralized IT processes
- From company-specific governance of enterprise software delivery to governance and improvement supported by standardized activities, particularly in enterprise architecture with the open group architecture framework (TOGAF) [26] and the control objectives for information and related technology (COBIT) framework for IT governance and control [27]
- From a collection of uncoordinated enterprise software delivery improvement schemes to significant investment in measured improvement against the capability maturity model (CMM), currently assessed at level 2 and on a path to level 3 compliance

Within this broad context, I'll analyze the structure, staffing profile, and delivery approach of MyProj. I'll then examine where improvements to enterprise software delivery are possible, and I'll assess their potential impact on this project.

2.3 Business and Organizational Context

MyCo is a large multinational organization with a presence in more than a dozen different countries around the world and a work force of over 100,000 employees. As with all utility companies, its business includes very high levels

1. Although this example is based on a real-world situation, I have simplified the example for presentation purposes, and I use fictitious names for reasons of privacy.

of innovation, research, and development in areas such as energy production and distribution. However, it also has a large historical investment in infrastructure, systems, and processes that have been in place for decades. Pressures are placed on costs and levels of efficiency to both drive innovation to be more effective and optimize operational costs to be more efficient.

The MyCo enterprise software delivery organization has a key role in addressing these challenges. Its strategy involves four areas:

- *Collaboration across global delivery teams.* The common mode of operation involves multiple suppliers, multiple geographies, and multiple business units.
- *Waste reduction and optimization of resources and assets.* Aligned practices provide a consistent and integrated development approach with standardized tooling across the organization.
- *Optimized reuse of core assets and practices.* An ongoing effort to catalogue, categorize, and assess the value of the current asset inventory aims to make it more accessible across the organization. These asset categories are aligned across all aspects of software development, delivery, and deployment.
- *Business cost management.* Increased cost transparency is enforced in all activities through continual monitoring of project health across the portfolio of projects and across a wide variety of tools and practices. A specific move toward virtualized and cloud-based infrastructures is foreseen, with several pilot efforts already in progress.

2.4 Project Context

Within this strategic context, the MyProj project is particularly critical to MyCo. In this project, a core back-office system for customer management is being redesigned to improve customer-handling activities and provide greater analytical capabilities across several disparate customer data sources. MyProj has the following important characteristics:

- *Worldwide delivery and deployment.* MyProj involves not only the collaboration of development and delivery resources from many locations but also the global deployment and management of the resultant solution.
- *Significant criticality, size, and investment.* MyProj is a 2-million-euro investment and involves almost 2,800 staff days of effort within a broader

10-million-euro system improvement program. The planned elapsed time of the project is 1 year, with some significant business impact if that date is not met.

- *Outsourced delivery model.* The majority of the effort on MyProj is contracted with an external system integrator in a “time and materials” contract.
- *Focus on reusable assets.* MyProj both uses and contributes assets that have been used and can be reused in other projects. These assets include software code, component specifications, and documentation.

To manage costs, MyProj’s system integrator uses delivery centers across the world. Using up to 80 percent of resources in offshore locations, MyProj’s offshore delivery model features the following:

- Offshore teams that are handpicked based on required skill sets
- Offshore project managers and technical leads who spend an initial period colocated with the onshore management team (a so-called landed phase) before returning to their offshore locations
- Regular *high-touch* visits by a core team and the customer to manage progress and create more informal relationships among the extended team
- Explicit expectation management of surface risks as early as possible
- The enactment of a collaborative solution-oriented management approach to analyze and resolve any blocking problems that arise
- High levels of collaboration across the teams to ensure synchronization at all stages of development and delivery

2.5 Project Execution Results

Figure 2.1 illustrates the staffing profile for the execution of MyProj. Here we see the loading of resources over the lifetime of the project and the levels of effort for each development phase. MyProj was completed within a few weeks of schedule (1 year elapsed time) with a peak effort of almost 300 staff days per month and a total investment of almost 2,800 staff days from inception to delivery of the enterprise software.

The profile depicted in Figure 2.1 is typical of many projects (both within MyCo and elsewhere), with a characteristic peak loading of effort late in the

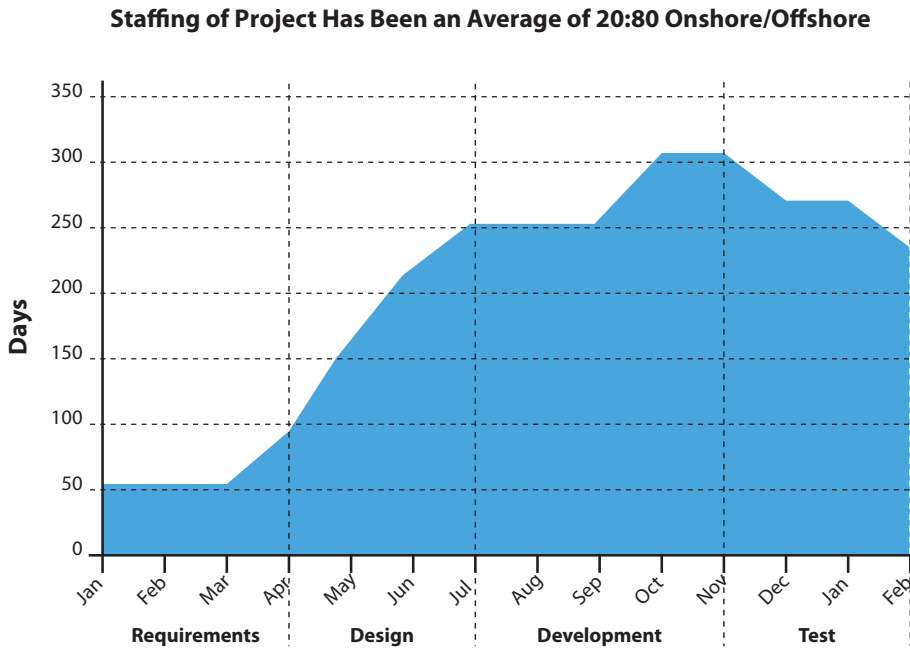


Figure 2.1 *The resource profile for the MyProj project*

development phase as unit test efforts give way to integration and system testing. Often in this phase of the project, the majority of the more complex bugs are found, usually as a consequence of misunderstandings in requirements, architectural weaknesses, and failures in communication across the team. Throughout the testing phase, this high level of staffing is maintained to address these concerns, correct errors, and bring the resulting system to a deliverable state. In this regard, MyProj displays many aspects typical of such projects. For this analysis I do not explicitly address some of the issues concerning postdeployment costs; instead, I simply note that this high level of staffing continues through the first weeks of deployment due to the large number of late fixes and changes in the system.

2.6 Post Hoc Analysis

As a consequence of the MyProj resource profile, MyCo undertook a detailed post hoc analysis of effort across the project phases. This analysis aimed to identify key areas where a more effective delivery approach would provide

measured improvements while maintaining or improving the predictability and quality of the delivered results.

Two levels of analysis were undertaken: one at the level of the enterprise software delivery organization and the other based on the specific details of the MyProj project.

2.6.1 Enterprise Software Delivery Organization Analysis

The first level of analysis considered the areas for improvement across the enterprise software delivery organization. Specifically, there are improvement areas in which efficiencies are possible as a result of addressing systemic issues in the enterprise software delivery approaches currently in use. These would have broad applicability and show value to the organization in the short, medium, and long term.

This analysis was multifaceted. Working with management personnel, we looked at results from several similar enterprise software delivery projects that had taken place over the past two years. We then compared that information with benchmark materials from several industry studies (such as [2,20,28]), examined data from our internal project delivery analyses, and engaged a third-party enterprise software industry analyst organization to offer a critical review of our findings, comparing them with data from their current clients and with their own analytical approaches.

Figure 2.2 shows a summary of the results of this analysis. For simplification purposes, the graph shows major areas for enterprise-level improvement across two dimensions: the long-term impact and the timing of potential benefits. Here, there was focus in seven areas, each represented as circle on the graph identifying how that area was considered to contribute to impact and the likely timing of that impact. In each area, we performed quite detailed analysis of savings and improvements by examining representative project data, typical labor rates, and project organization sizes and structures. Using that information, we then were able to show the potential improvements in terms of productivity savings (labor reduction) and quality improvements (defect rate reduction). For example, in the areas of Requirements Management Linked to Quality Management, we used company-specific project information and industry-wide benchmark data to identify that improved life-cycle management techniques could result in a reduction as large as 10 percent in defect density for delivered software in the first year. With added attention to test case management and test planning, this would also provide up to a 7 percent saving in labor costs in the first year. Similar analyses were performed in all seven areas identified in Figure 2.2.

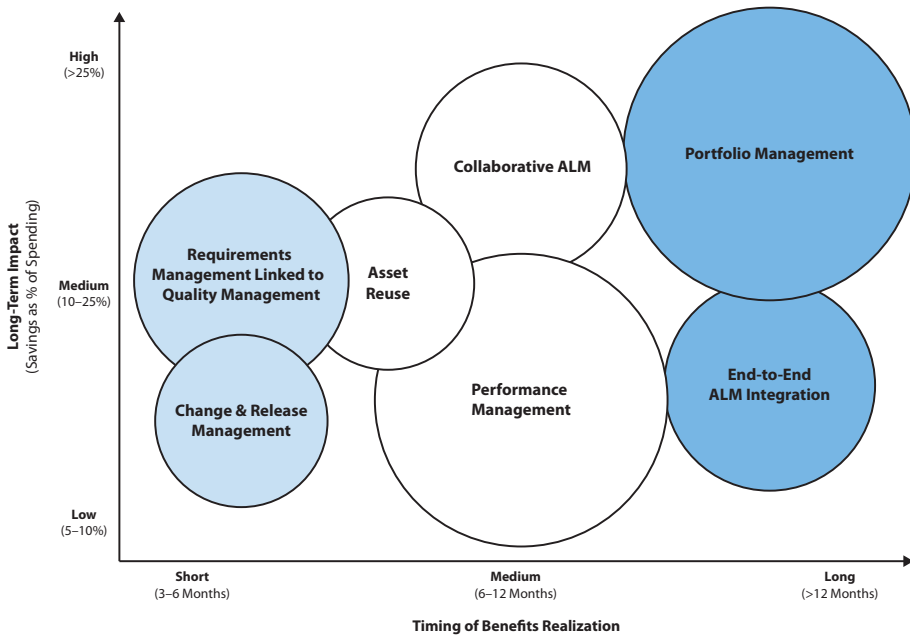


Figure 2.2 A summary of efficiencies across the enterprise software delivery organization

A full description of the analysis approach is not necessary here. Rather, I'll provide a short summary of the key features of the techniques used to highlight its key characteristics.

While a number of analysis approaches are possible, in this case we selected a practice-based approach. Each practice and its associated tools have an underlying benefit for each practitioner. For example, the adoption of an improved change management practice and associated tools will save a developer 1 percent on average. Considering the loaded costs of each resource, we can calculate a benefit, per year, and by multiplying this figure by the number of practitioners, we can calculate an annual saving. While each individual practice improvement is actually quite small (perhaps a saving of one to four hours a week), the accumulation as more practices are adopted results in a significant overall benefit.

As shown in Figure 2.2, our analysis approach, then, assesses a series of selected practices in terms of potential improvement. Each practice is a collection of tools and services aimed at enhancing the development capability of a team or project. It was concluded, for example, that near-term improvements in areas such as change and release management and requirements management could

have delivered labor savings of 5 percent and 7 percent, respectively, with additional reductions in defect density of up to 10 percent in the first year.²

2.6.2 Project-Level Analysis

While the first level of analysis considered the areas for improvement across the enterprise software delivery organization, the second level directly examined the MyProj project. Focused on several areas, the analysis involved detailed discussions with the delivery team,³ analysis of processes executed, inspection of actual defect rates and error-fixing practices, document reviews, and experimentation with real data.

We concluded that improvements could be made in the following areas:

- *Quality management.* Difficulties in project status reporting were commonly experienced. To increase the effectiveness and productivity of the onshore management team, in particular, better decisions could have been made with enhanced real-time quality assessments supported by automated quality data collection tooling.
- *Data management.* Errors in data analysis and models were detected late in the development cycles. Improved integrated tooling would help improve data analysis and modeling, increasing quality of data used to test and reducing development and test timelines.
- *Requirements traceability.* Poor connections between requirements, development, and test activities resulted in several misunderstandings across the team. Greater integration and tracking could have the effect of removing design gaps and misunderstandings. A consequent decrease in the continual need to query requirements to ensure their currency and to understand unclear design implications could result in a significant time savings in development. Similarly, a decrease in the need for such queries results in better estimation for change analysis and reduced critical and major defects in test activities, as build activities are more aligned with the requirements and design status.

2. In practice, many additional factors were also taken into consideration, such as the interactions among the practices, capacity for change across the teams, phasing of changes, and so on. Such factors are part of an internal IBM “business value analysis” tool that was used.

3. The approach was to use a structured risk analysis technique similar in many respects to the approach used by the SEI and based on previous experiences with its application to large software projects and programs [29].

- *End-to-end environment management.* Several administrative activities caused time delays in creating and maintaining appropriate working environments. These delays can be eliminated with faster environment procurement and easier access to predefined developer profiles, thus making necessary resources, such as tools, data, and test equipment, more accessible.
- *Onboarding to access the development and test tooling.* Situations arose where new project members, or existing members switching to new roles, had difficulty getting information or access to the tools for software development and delivery. This time loss can be greatly reduced with faster onboarding, improved documentation, and access to communities for peer-level advice.

The result of this project-level analysis is shown in Figure 2.3. Here we show how MyProj could have been executed as a result of these software factory delivery improvements in the areas stated. In particular, we produced data that shows a one-month reduction in cycle time, with reduced peak effort through

FTE Savings—Major Reduction in Developmental and Test Effort Expected Due to Enhanced Definition and Tracking of Requirements and “Decoupled” Test Cycles

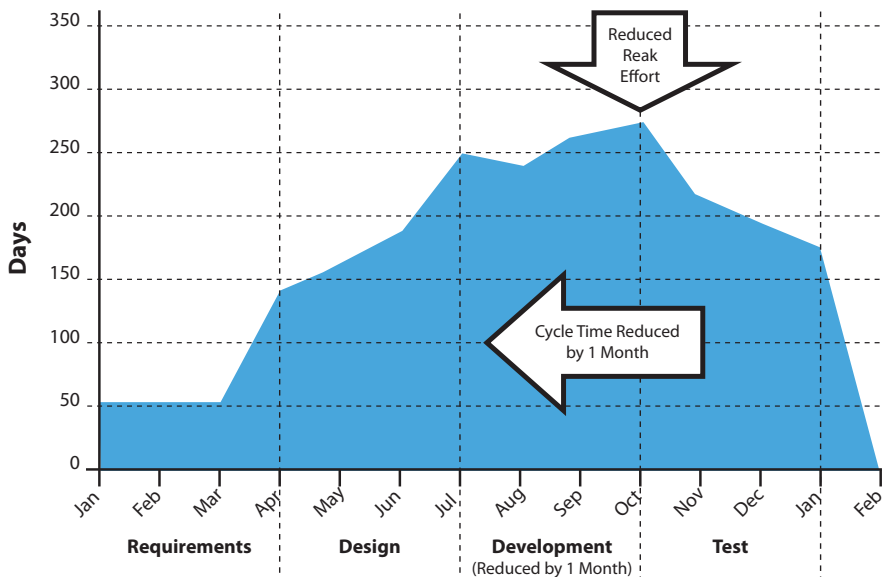


Figure 2.3 Adjusted project profile for the MyProj project

the elimination of certain blockages, reduction in costly misunderstandings, and improved communication channels.

A more detailed view of the efficiencies and cost savings is shown in Figure 2.4. Here we view the change in resource and cost profile over time as a result of these enterprise software delivery improvements.

2.7 Commentary

This review of the MyProj project has enabled us to look in some detail at a typical enterprise software delivery project with respect to the project's delivery context, resource profile, and execution history. In the subsequent analysis, we considered areas that could improve from adopting additional software factory delivery techniques and automations. We can now make a number of important observations.

First, it's useful to summarize the sources of potential improvements we identified from our analysis of both the enterprise and project levels. We can express these as recommendations for improvements in four areas. Each recommendation represents a challenge and an opportunity:

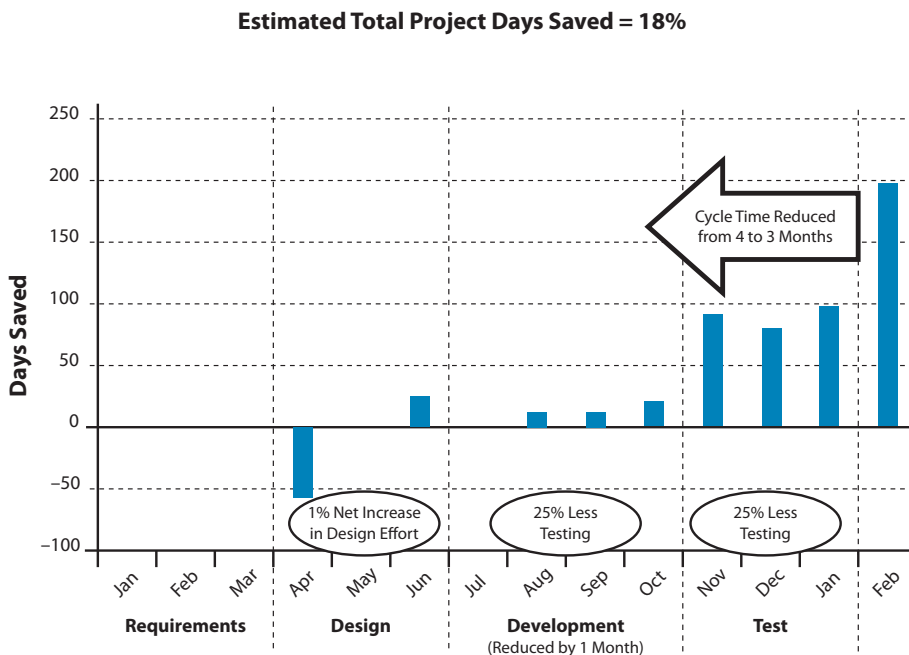


Figure 2.4 A more detailed efficiency analysis of the MyProj project

- *Collaborate globally.* The global approach to delivery deserves particular attention. Many challenges to enterprise software delivery are a result of intercommunication issues. In a globally distributed development team, we must ensure that the value gained by using people in other regions (whether to make use of scarce resources or to optimize costs) is not lost. When cross-organization issues are added to the challenge, investment in processes and technology to ensure smooth collaboration becomes essential and proves to be a determinant factor in the project's success.
- *Deliver with agility.* A great deal of flexibility is needed to manage the continual change inherent in these kinds of projects. Agility is necessary in not only the style of development but also the management of interactions with the business stakeholders; the manner in which project objectives are assessed, understood, and evolved; and the way in which the solution is provided to the operational teams to be moved into production. This broader view of agility in enterprise software delivery is essential. Our analysis takes into account the increased confidence that results from early attention to testing available software and the benefits of guidance from continuous stakeholder feedback.
- *Focus on quality.* We've always known that the peak effort and cost of enterprise software delivery is during testing and delivery phases. During these phases, the real quality issues become more concrete, and many of the more complex problems surface. This project analysis again confirms that a "total quality" view of enterprise software delivery reduces that peak effort substantially and provides the major source of early savings to an organization. The reduction of high-impact late changes is of the utmost importance in any project.
- *Monitor and measure continuously.* Poor decision making was primarily a result of incomplete information and lack of clear insight into the project's real status. The challenge was increased due to the distributed, multiorganizational aspects of the project. Far too much time was spent chasing information, finding relevant data, and making sense of the data gathered from multiple sources. The project would have benefited from an increased focus on automated information gathering, together with a more clearly defined view of the project's most relevant business and technical measures.

Second, we note several additional areas where improvements could be made. For example, we did not spend time in areas of reuse. Many industry studies provide data that supports substantial improvements in productivity and quality through mature, systematic reuse practices. In the case of MyProj, while reuse maturity was low, attention to this area would undoubtedly bring additional rewards.

Another area of importance to MyCo is the move toward cloud-based delivery models. Through deployment of cloud technologies, the company is looking to reduce investments in underused technology infrastructure and increase standard services for areas such as testing. Again, these areas were considered important, but their impact could not yet be sufficiently understood and qualified to include in this study. Further work is ongoing to get more detailed and reliable data in this area, with significant savings expected.

Third, I express a note of caution about broader interpretation and extrapolation from these results. While this analysis has produced very valuable data and provided many interesting insights, we must be careful in understanding the broader applicability of these results to both MyCo and other organizations. An important area of caution concerns the scalability of these approaches to broader enterprise software delivery contexts. It's clear that many of these process and tool optimizations are equally valid for other teams in other situations. However, additional review is needed to understand how broadly and how quickly these results could be applied, say, to the whole of the MyCo enterprise software delivery organization. I believe that many of the same results could be achieved by more broadly adopting such practices in other projects, but further investigation would help validate this assertion.

2.8 Conclusions

This chapter focuses on a concrete example of an enterprise software delivery project exhibiting many properties that characterize today's software delivery challenges: globally distributed delivery of core business capability in a mixed onshore-offshore team. The example has shown us in very real terms how and where efficiencies are possible with the focus on global enterprise software delivery.

As a result of this study, the key factors in enterprise software delivery improvements have been seen in four areas:

- Collaborate globally.
- Deliver with agility.
- Focus on quality.
- Monitor and measure continuously.

I examine each of these areas in more detail in the remaining chapters of this book, focusing specifically on case studies that highlight the results obtained by applying many of the improvement suggestions provided in this study.

