

## Requirements and Bridging the Silos

By: Andrew Hayward  
Part 1 of a 3-part series

### Introduction

Sixty to eighty percent of software projects fail or are delivered long after original deadlines and over budget. These statistics are common and cited frequently by industry analysts (most famously by the Standish Group) and industry thought leaders. These figures are primarily drawn from the experience of large organizations,<sup>1</sup> and communication, accuracy of requirements, execution and management of the projects are cited as the main contributing factors to these failures.

From my experience, the factors cited are in a large part due to the fact that large organizations, and the industry as a whole, promote and actually foster a work environment that segregates the different groups responsible for software project delivery.

These segregations, or silos, limit the visibility and effective communication and cooperation between the participants in the development effort. Communication between these groups becomes very formalized, for instance relying on PowerPoint presentations, discouraging collaboration or involvement and are often focused only on what is important from the presenter's perspective. Email, a communications method very low on the scale of effectiveness<sup>2</sup>, is used to send specifications for review and prioritization, resulting in more analysis and often misunderstandings of the sometimes contradictory changes requested. Teams rely on daily or weekly 15-minute status meetings which are focused on their progress, accomplishments or challenges, but don't foster understanding or participation in the decisions team members make.

In part one of this series, I intend to demonstrate why requirements for a project must be communicated and understood across all of these silos. In part two, I will look at how to remedy some of the challenges silos create through effective communication, improved visibility and complete traceability. Finally, in part three, I will provide recommended practices to improve collaboration and communication and, therefore, increase project success.

### The Silos

Within a large organization there are many forms of silos, from organizational to cultural to geographic. Here I will focus on the functional silos that exist within the IT organization.

The people who work on development projects are usually grouped based on the functions they perform during the lifecycle of the effort. Each organization, and even sometimes projects within an organization, will define these groups differently but generally there is the customer, internal to the organization or external, the IT business analysts, the design and/or development team, quality assurance or testing, the group that does the implementation, and the operations or maintenance group. It is essential for each of these groups to have visibility and understanding of the requirements defined for the system or application being developed as those requirements represent the scope of the project itself. Each group needs this visibility for different reasons and each needs to be informed of the changes to the requirements as the project is executed.

---

<sup>1</sup> Interview with Jim Johnson of the Standish Group. <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>.

<sup>2</sup> Scott Ambler, Agile Modeling: Effective Practices for Extreme Programming and the Unified Process, John Wiley & Sons, 2001.

## Analysis and Design

Most organizations recognize the importance of the communication between the first three groups mentioned above, each organization may, however, manage this communication in a different way. Generally, customers and IT business analysts work together to come to an agreement on the initial requirements for a system. From this point the IT business analyst and the development team may work together to refine the requirements from a product or business level of abstraction to a system level of abstraction.

This sounds efficient and the appropriate way to begin a project. However, the silos created around the roles involved complicate communications and start the project off on the wrong foot in two key areas.

First, the development group does not always participate in the initial requirements elicitation. This is particularly true when an interview approach is used for gathering the requirements. More often than not the result is that the development team has to rely entirely on the documentation produced from the interview sessions, a 'Pitching it Over the Wall' approach, as it is often described, and the IT business analyst becomes the subject matter expert for the requirements. Gaps in the documentation, interpretation of the documentation and assumptions that are made when reading it may lead the development group to incorrect conclusions about the design of the system and how it meets the customer's needs.

One remedy to this outcome is to hold workshops for the initial requirements gathering and ensure that a representative from the development team, and likewise someone from the testing group, is present for the sessions with the customer. These individuals should be there as observers and not to try to design the solution but rather to witness the discussions about the problem or opportunities that the business is trying to address. This participation will lead to improved understanding of the business case and enable the development and testing teams to think about the challenges the project will have in the correct customer context in addition to the technical or implementation context.

The second aspect of the problem is that when the requirements are refined through analysis, for example when developing system use cases from the business use cases, the customer is often not involved. The IT business analyst will either work alone or with the development team and then present the result to the customer. This can result in wasted effort in developing the lower level of abstraction and/or proof of concept, presenting it to the customer, then having to rework it repeatedly.

Having customers and key subject matter experts involved in this process can reduce the amount of rework cycles, build trust between the groups and clearly define ownership of the results with the customers.

This may seem burdensome to include customers in some of the initial downstream activities but consider the following: No requirement specification is complete. What's more, even if it were possible to have a complete specification, it would likely be unusable due to its size and complexity. In fact, many system development methodologies, including the Unified Process and Agile, recommend that requirements specifications not focus on getting all of the details written down at the outset of the project because throughout the execution phase the requirements will change. This inevitable churn means that the effort spent on getting it right would be wasted and that trying to go into too much detail will most often lead to "analysis paralysis." Having a

customer involved in this process, even if only as an observer, will help with the gaps and questions the team discovers, and ensure that the requirements are understood. Holding Joint Application Design (JAD) sessions, a methodology designed by Chuck Morris of IBM Raleigh and Tony Crawford of IBM Toronto in the late 1970s, is one way to ensure this type of cross-team collaboration is effectively practiced.

In addition to enhancing your methodology to involve additional roles, it is also very valuable to provide visibility into the relationships between the artifacts being generated at each phase of the project. For example, the business requirements and the technical or system requirements derived from them should be clearly related themselves and to the downstream functional designs and tests which satisfy and validate them. Sometimes how a lower level of abstraction is related to the higher level of abstraction is not immediately apparent. This is particularly true if you start by using a customer friendly model, such as activity diagrams, and then move to a more development friendly model, such as class, object or sequence diagrams, during the analysis phase. You can address this visibility by maximizing the communication capabilities of your requirements management tool by enabling sponsors and other customer representatives access to view in progress and completed work and its relationships to previous work.

## **Concluding Remarks**

The first installment of our three part series has described how the silos, in which resources in a project are divided, often affect the communication and understanding of requirements. This installment started with the analysis and design groups, those who work directly with the customer and those who design and build the solution, and showed how bridging the silos by using collaborative analysis and design is an effective way to reduce misunderstandings and gaps in understanding. It was also recommended to use a requirements management tool that enables all three groups to view and communicate the evolution of the requirements throughout the development effort.

The second installment will continue with two equally important silos, testing and implementation, and will describe how to increase understanding of project requirements within and between these groups.

-----  
Andrew Hayward has five years of experience working as an independent consultant, gathering and managing requirements and providing expertise in Requirements Management best practices for organizations across North America and Europe. Currently, as a National Application Engineer with MKS Inc., he works primarily in implementing MKS Requirement Management processes and solutions.