

Requirements and Bridging the Silos

By: Andrew Hayward

Part 3 of a 3-Part Series

Introduction

This installment of our three part series on *Requirements and Bridging the Silos* continues with some examples of how silos have hindered communication and, therefore, understanding of requirements, and the solutions these organizations applied to resolve these problems. Following this is a list of eight best practices to bridge the silos in your organization.

Recommended Practices

Barriers to communication are barriers to software project success. Software development should not be seen as a set of discrete functions in an assembly line, where one group puts together the requirements, the next group takes those requirements and builds the solution, and another group then tests the solution. Today's software development projects demand collaboration between all of the groups, where each group has its responsibilities and expertise, but also understands and is involved in the contributions made by the other groups.

The silos that restrict people and systems too often limit effective communication. For example, a few years ago I was asked to assist in the requirements gathering for a large mortgage servicing firm which was attempting to develop an automated mortgage adjudication system, a project which was identified as crucial for the company's continued success in its market. The company did not have a formal requirements gathering process or requirements management tool, and relied on interviews conducted by business analysts associated with specific business areas such as broker relations, adjudication, mortgage administration, etc. These business analysts worked with their assigned business units and then presented the requirements to the development team. This resulted in numerous requirements that appeared to contradict each other, requirements that could not be met for technical reasons outside of this company's control, with a disjointed vision of what the system was expected to do, and the project never moved past the analysis phase.

In order to solve this, we set up several days of facilitated requirements meetings with representatives from the different stakeholder groups, as well as a technical architect for the organization. It was a considerable investment in time and resources for all parties, and some very difficult discussions, but the result was a team that completely understood what the system was expected to do, what it was not going to do, and why. The resulting understanding allowed the development team to prioritize the development work and complete the first phase two months earlier than planned.

Another, more recent example, was with a company that provides information services for large financial and telecommunication companies across North America. The silos at this company were different than the example described above. This company had well defined requirements management processes, but different departments managing different services for their customers used different requirements methodologies. Specifically, the majority of the company

used a waterfall approach, but some had adopted an Agile methodology. In projects that affected multiple areas, the analysts using the waterfall approach would record the requirements in documents while the Agile teams had a customized tool for managing their user stories. Also, the waterfall teams would often be in development before the Agile teams even started working on the project. In the course of their analysis, the Agile teams would update or even change requirements, which would impact the work the other teams were doing, but with no central repository of requirements, determining whether there were impacts on other groups was very difficult.

To remedy this, the company has started to implement a requirements management tool. They needed a tool flexible enough to manage the process needs of the different groups but also ensure that all of the teams are working from the same user stories. Also, the tool needed to provide updates to the business analysts in other groups, and the development teams, when someone updates a user story or adds a new user story. This tool will enable the different groups, many of which are geographically disparate, to work in the manner they find most effective and still communicate productively.

Some recommended practices to improve collaboration and communication and, therefore, increase project success, are:

- 1) Use facilitated workshops whenever possible. Workshops are effective not just for eliciting requirements, but also validating and prioritizing the requirements, designing the solution and making key decisions. They get more people involved, increasing the level of understanding and ownership;
- 2) In your requirements management plan or project plan, assign members from each group to act as representatives for their area, and ensure they attend the workshops or meetings either as participants or observers;
- 3) Train operations people in how the system supports the end users and the value that the system provides for the organization;
- 4) Provide a simple and accessible means to keep requirements and design documentation up to date, and assign people to be responsible for maintaining it. The right CASE tool can make this process more manageable;
- 5) Leverage traceability throughout the entire application lifecycle, including operation and maintenance. Use the relationships you set up to understand the impact of prospective changes and the root causes of change requests;
- 6) Ensure that traceability spans across time and across all projects within your organization when utilizing a parallel development methodology and where reuse is important;
- 7) Ensure your ALM tool can support collaborative, cross functional processes and can provide visibility into progress and efficiency; and
- 8) Use email as a last resort for communication. Even a quick phone call will be more productive and informative than a series of emails.

Concluding Remarks

The silos that exist in all organizations are expected and necessary to some extent. They enable resources to work in the subject areas that have the most expertise, the location that is the most convenient, and culture that is the most comfortable. It is important, however, to take steps to reduce the barriers to communication that these silos create.



Using workshops with key representatives from each group and involving those representatives in the decision making processes, rather than informing them of the outcome of these processes, improves communication and encourages trust and ownership. Approaching system development with an application lifecycle perspective, supported by a unified platform and process across all of the silos, expands communication opportunities, enhances visibility for all involved and will encourage collaboration. These approaches, taken together, will increase communication about and understanding of requirements, which will in turn result in more effective teams and more satisfactory results in development efforts.

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.