The Requirements Management Challenge

Exposing the Limitations of Traditional Techniques
Executive Summary

According to a recent study of 590 CIOs, 98% of all completed software projects required rework. In today’s demanding market, with pressure to release software faster than ever, this is an alarming statistic. Considering the average project requires 31% requirements rework, why are highly experienced teams with continually improving processes unable to perfect requirements in the first instance?

The main reason for these statistics is that organizations use documents and spreadsheets to manage requirements manually. According to a survey conducted by Vanson Bourne, 73% of organizations use documents and 76% use spreadsheets to manage the requirements of software delivery projects.

This white paper explores specific challenges caused by using documents and spreadsheets while recommending industry proven best practices to overcome the trend of failed and mismanaged projects.

The Importance of Requirements Management

As business processes evolve to address market trends and technologically advanced consumers’ demands increase, organizations must be more flexible. The need to deliver the right requirements more efficiently with higher levels of quality has never been more important. Test cases, designs and development deliverables rely on requirements being as complete and accurate as possible. The cost of fixing these related rework increases exponentially, so it is essential to identify any shortcomings as early in the process as possible.

Specific Challenges

When it comes to requirements management, documents and spreadsheets have basic flaws. They are essentially collections of static data relating to the status of a project at a point in time, making it almost impossible to conduct real-time impact analysis (traceability), which is extremely important for making informed decisions throughout a project. They are also ineffective when it comes to reuse for core requirements such as security, performance, usability, and other non-functional requirements. Furthermore, documents and spreadsheets include everything, not just ‘items of interest’. This amount of information makes it difficult to find the right information, which quickly inhibits collaboration, especially during feedback and approval cycles.

The value of good and the cost of bad requirements are difficult to quantify. Many organizations will not spend valuable resources fixing problems related to requirements or investigate the root causes of poor ones until it is too late. At that point, they must remedy problems after a project fails, or even worse, they are forced to mitigate security vulnerabilities or address non-compliance findings.

Teams that struggle to collaborate efficiently fail to communicate requirements to key stakeholders, during elicitation, definition, socialization and approval lifecycles. The following sections include key tasks that many organizations are failing to manage because they are capturing content in documents and spreadsheets, which can cause several problems:

Elicitation

Despite repeatable processes and techniques, elicitation seldom happens the same way twice. Business analysts will use many approaches to gather as much information as possible in the limited time they have with stakeholders who are considered to be the domain experts.

“Mistakes made in elicitation have been shown many times to be major causes of systems failure or abandonment and this has a very large cost either in the complete loss or the expense of fixing mistakes.”—Morgan Masters, Business Analyst

---

1 Borland Vanson Bourne Survey 2013, slide 20. www.microfocus.com/_ex/borland/assets/reports/Vanson-Bourne-testing-research-results.pdf
2 Borland Vanson Bourne Survey 2013, slide 16. www.microfocus.com/_ex/borland/assets/reports/Vanson-Bourne-testing-research-results.pdf
3 An overview of requirements elicitation 2010. www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1427/An-Overview-of-Requirements-Elicitation.aspx#sthash.GuxpzKA5.dpuf
Problem 1: Failure to Reuse and Understand Relationships
Projects often have a business case or ‘concept of operations’ document, listing high-level business objectives and goals. A business analyst must understand the business need and ensure elicitation activities are within project scope. In theory, high-level business needs should be reused and refined to become more detailed requirements.

These documents often serve as an initial starting point, but the content of the documents is never truly reconciled with the resulting requirements, to ensure the original high-level goals and content are complete or in agreement. Once the project gains momentum, the earlier documentation is often left behind while new requirements become the project benchmark. Therefore, incongruences between the initially funded scope of work and plans going forward are often unaddressed.

Solution: Real-Time Traceability
Traceability is an excellent technique for creating relationships and dependencies while linking downstream items to a business requirement. Reports that show untraced items often indicate project additions, which may have been made outside the business requirements context. This could indicate out of scope or low priority work that will not add business value.

Problem 2: Generating a Complete Set of Requirements
Common questions include, “How many requirements should a project of this size have?” and “How do I know when I’m done gathering requirements?” It is impossible to answer these questions without conducting sufficient analysis. If you fully understand the business need, the domain space and the scope of the project, this task will be more quickly accomplished.

If you are working on an existing system, you may have documentation (which may be incomplete) that must be analyzed, but if the system is new you could start by listing all requirements, which a variety of elicitation techniques derives. As the number of requirements increases, the ability to understand them and how they relate to each other becomes increasingly difficult.

Scenarios or use cases can be used to understand the functional flow of a system and they are usually captured in use case templates or diagrammed in Unified Modeling Language (UML) notation. Providing an excellent way to gain an understanding of the various activities taking place in a system, the actors that are participating in the system and the various paths through the system. This becomes the thread that links several requirements together in a meaningful way and helps to define the business rules that govern the process flows. It also helps to understand what roles and permissions might be necessary.

Without exploring all the scenarios, actors involved, business rules associated with key activities, system integrations and all the related constraints, it is impossible to know whether you have a complete set of requirements. Using a combination of text and diagrams without specific traceability makes managing large projects challenging, especially when trying to gauge whether what you gathered is on track or correctly capturing business needs.

Solution: Use a Visual Approach with Scenarios (Storyboards)
Depicting Relationships to Requirements, Business Rules, Actors and Other Supporting Information at the Activity Level
You can tie simulations to various activities to ensure user interface (UI) functionality is correct before development begins. Domain experts should maintain non-functional requirements in a reusable repository and share them across multiple projects. When requirement scenarios and simulations are properly modeled, you can automatically generate the corresponding test cases for all unique paths through the system.

Traditional approaches with documents and spreadsheets rely on testing the deployed system before valuable feedback is received, making rework much more costly.

Definition (Word-Like Capabilities)
Following elicitation, you are likely to end up with use cases, UML models, notes, pre-existing documentation, lists of roles and permissions in the system, and so forth. It is now necessary to normalize this information into

Miller’s Law states that the number of objects that can be held in working memory is 7 (plus or minus 2). As requirement lists in documents or spreadsheets quickly exceed this number, we need to develop categorization schemes to simplify them.

---

requirement specifications that describe system behavior and establish the basis for agreement between the customer and the contractor or supplier as to what the system will do. This should also provide a realistic basis for estimating product costs, risks and schedules.\(^5\)

Typically, templates in document format are used to ensure requirements are captured in a standard way.

**Problem: It Is Difficult to Provide the Optimal Level of Detail to Communicate the Business Need**

Too much detail results in waste because the development and test teams have to break down excessive detail into meaningful tasks. On the other hand, too little detail results in rework because multiple clarifications are required to ensure the requirement is fully understood. Failure to identify key requirements is considered an omission, posing the greatest risk to security, performance and compliance and usually results in severe repercussions.

**Solution: Practical Fidelity—Providing Only the Necessary Detail Required for Business Clarity**

A picture says a thousand words: simple illustrations can provide details for processes and designs that will minimize time needed for refinement with stakeholders while reducing the time needed for teams to break business requirements down into development or testing tasks.

Visualizations are supplemented with actual requirements and business rules relating to appropriate activities and decision steps in scenarios. In addition, a familiar interface for entering and editing these requirements is essential to ensure adoption and productivity.

**Management (Excel-Like Capabilities)**

Once requirements are captured, they should be organized. Normally, categories are used and traceability matrices are created with requirement IDs, corresponding test cases, IDs for each object type and design element IDs.

**Problem: Managing Requirements through the Lifecycle with Change History**

Typically, organizations use spreadsheets to categorize requirements for ease of management. Lifecycle statuses such as “New”, “Draft”, “In Review” and “Approved” might be used to help understand where a requirement is during its evolution through the feedback and approval process. This approach can be cumbersome, requiring complicated filters to make bulk updates. The history of these status changes is not captured unless the document is controlled by a document management system, which requires additional overhead. The ability to work efficiently in this manner is almost impossible with large teams.

**Solution: Use a Grid View to Make Bulk Updates Simultaneously Based on Filtered Criteria**

A grid view allows you to create custom real-time filters that will always provide the latest statuses and other data attributes of individual requirements.

---

\(^5\) An overview of requirements elicitation 2010. [www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1427/An-Overview-of-Requirements-Elicitation.aspx#sthash.GuxpzKA5.dpuf]
Updates that are ready to move to the next state in the process can be easily updated with bulk updates for individually selected items. This approach is ideal for large teams that are working on requirements. An individual change history is captured for each one as updates occur.

Socialization (Review) Feedback
Refinement is an ongoing process. From elicitation to approval, requirements will have commentary from a variety of sources such as emails, edits to documents or spreadsheets, phone calls, messages, and meeting notes. Finalizing and completing them with precision is necessary to reach the approval phase.

Problem: Often Documents or Spreadsheets Are Sent by Email as Attachments or Links to Large Teams Requesting Feedback
This feedback comes in many forms, and for longer documents there tend to be fewer collective inputs towards the end of the document. Once all feedback is consolidated and requirements are defined, the process is repeated to keep stakeholders informed. This can go on for multiple cycles and is highly inefficient. Furthermore, once feedback has been incorporated into the document or spreadsheet, the history of previous versions, including who requested the changes and why, becomes disassociated from the document. If you have ever saved a document and included your initials, the date, version number and so forth, you are contributing to this problem.

Solution: Use a Web-Based Centralized Repository with Visual Storyboards and Simulations to Conduct Initial Stakeholder Reviews
When feedback is captured at the requirements level, those who missed the initial review or wish to provide additional feedback can always see the latest version of the requirement with related feedback. This prevents duplication of effort and eliminates the need to resend documents to stakeholders after any substantial feedback has been received.

Approval (Review) Baseline
A major challenge for business analysts is obtaining buy-in for approval from key stakeholders. With a good socialization process in place, the approval phase can be significantly reduced. In many organizations, the requirements review is a formal process for gaining approval so the baseline can be sent to the development team for implementation.

Problem: In Theory, Requirements Should Be in a Ready-For-Approval State During the Review
However, it is not uncommon for several key stakeholders to have concerns and offer ‘conditional’ approvals. This is indicative of incomplete requirements, failure to get consensus from stakeholders on the scope of the work or failure to clarify them in advance.

Solution: By Looking at Each Requirement Individually, Along with Visualizations and Dependent Requirements, It Is Much Easier to Advance the Majority of the Requirements to an Approved State
Instead of sending out entire documents, meetings can focus on more challenging requirements on an ongoing basis. Objections can be identified earlier in the process instead of during actual reviews. Reviews become a formality with no surprises and all approving stakeholders can approve the baseline with an electronic signature to satisfy auditing and compliance.

Delivery (Word-Like Capabilities)
It is important to manage requirements from traditional Waterfall teams as they relate to user stories in Agile development teams.

Problem: Business Analysts and Development Teams Often Struggle to Maintain Real-Time Collaboration between Business Requirements and Agile User Stories
Agile development teams require the means to demonstrate compliance. Teams using documents and spreadsheets are not able to keep business analysts and developers in sync as business requirements and product backlogs frequently change.

Solution: Create a Relationship from Business Requirements to Agile User Stories
This provides the business analyst with progress visibility of user stories supporting requirements for better impact analysis. It also ensures communication and collaboration between business analysts and development teams are in cadence with Agile delivery. Most importantly, this approach ensures constant visibility, enabling high-priority requirements to be delivered by development teams.

Change and Impact Analysis (Traceability)
From the time the first requirements elicitation takes place and even after the product is in production, changes are inevitable and ongoing. To understand change impact, it is essential that all requirement dependencies are established and constantly maintained.

Problem: Documents Are Not Inherently Effective for Traceability Purposes
Typically, spreadsheets include the ID numbers for the requirements and test cases, as well as any other related requests or objects in each
This tedious approach is prone to error and it is nearly impossible to keep changes to documents and spreadsheets up to date.

Solution: Manage Requirements at Individual Requirement Level and Create Trace Relationships as They Are Created
Inspect relationships regularly to ensure all items have been traced and that these items have not changed since the initial trace relationship was established. Tracing from a business requirement ensures you are working within the scope that was intended and creates a systematic approach for breaking high-level requirements down into more detailed functional versions.

Conclusion
Organizations adopting a centralized requirements management approach can significantly improve collaboration to produce better quality requirements earlier in the lifecycle, which can significantly reduce rework. In addition, a visual approach will enable teams to communicate the appropriate level of scope in the most efficient way, creating a more Agile approach to conveying business requirements to development teams.

The next time you receive a review in the form of a document or spreadsheet, think twice before you contribute to the endless cycle of rework. Once organizations embrace the practice of centralized management, they will be on their way to ‘getting requirements right first time,’ being more Agile and ensuring project success.

A Centralized Requirements Management Approach
Whether your organization adopts a Waterfall, Agile or hybrid approach, Micro Focus® Caliber can deliver immediate gains in productivity and precision in your requirements management processes. Caliber can immediately align with your current business practices and has the flexibility to allow continuous improvements in quality, compliance, reuse and collaboration to ensure development teams meet business requirements.

To find out more visit: www.borland.com/caliber