A Comparison of Requirements Analysis Methods

Kim Ellison, Dawn Levy, Mark Nelson, Charles Popelka

Introduction

Over the last several years, The Mercator Group (TMG) Process Design Studio (PDS), in support of its Government clients, has relied on several different approaches or methods for eliciting and documenting business requirements; ranging from more traditional functional requirements elicitation to leading-edge Agile techniques. This Article presents our evaluation of these methods and recommendations for their usage:

1. Describes the requirements analysis methods
2. Compares and contrasts each method
3. Recommends when to use each method

Our findings and recommendations focus on business applications with a high-level of end-user interaction, workflow, and collaboration. Our recommendations may not be as applicable to other types of applications, such as batch processing, system-to-system, and rules-centric applications.

Description of Requirements Analysis Methods

At the PDS, we have successfully executed a wide range of requirements analysis efforts, which can be categorized into three industry standards:

1. Requirements Enumeration
2. Model-Driven Requirements
3. Piloting/Prototyping

Requirements Enumeration is a more traditional approach to requirements elicitation and documentation. Typically, requirements are functionally organized and documented.   Some level of static modeling, such as use cases and data modeling, may be used as part of the methodology. This approach is closely associated with Waterfall methodologies, where there is a focus on fully completing requirements gathering before any development is initiated.

Model-Driven Requirements are organized around process models and include more detailed models, such as screen mock-ups, process-informed data diagrams and business rules. A better understanding of requirements is reached by modeling in the context of the actual business process. Data diagrams and business rules are also strengthened through discovery of those models as part of a process, and screen mock-ups provide the required level of system usage detail.

Piloting/Prototyping encompasses the development of what we refer to as a **Working Model**, which is a significantly complete pilot or prototype of system functionality. This method does not necessarily include the operational and integration of an actual system, but is focused on demonstrating the required functionality, user experience and interactions with the eventual system.

The Working Model is typically developed on an application platform that allows for a non-programmer to implement a significant level of end-user facing functionality without resorting to programming.   The Working Model may be used as a basis for deploying the new functionality, or may be subject to re-platforming based on architectural considerations.

In all cases, we apply the best practices associated with each method, such as:

- Requirements Traceability Matrices (RTMs)
- Integrated Project Teams (IPTs)
- Business Process Management (BPM)
- Agile techniques, such as Scrum

Additionally, industry-grade tools are used in all efforts.

Requirements Analysis Methods Comparison

When the Telegraph was first introduced in the United States, it was difficult for people to visualize its utility. In one instance, a woman went to her local telegraph office with a plate of sauerkraut, and asked that it be sent to her son. In another, a man brought a written message to be sent, and was confused when the paper itself was not actually sent. [1]

While the above examples are a bit dramatic, they illustrate the difficulty people have visualizing change. This has long been a well-known challenge in software development.   GAO reports that 48 percent of the federal government's major IT projects have been rebaselined for several reasons, including changes in project goals and changes in funding. Of those rebaselined projects, 51 percent were rebaselined at least twice and about 11 percent were rebaselined four times or more (www.**gao**.gov/products/**GAO**-08-925).   While all issues with large government projects cannot be traced back to difficulties with the requirements analysis phase, it is clear from the preponderance of evidence it is often a key part of the problem. The common industry term "scope creep" is indicative of the inability to fully capture requirements early on, and gain consensus. Our analysis reflects that visualization is vital to capturing requirements and obtaining consensus. Additionally, the linear nature of Requirements Enumeration methods, often results in additional work to ensure consensus across requirements to development and acceptance testing. The collaborative nature of Model-Driven and Prototyping/Piloting methods, often substantially reduce the efforts required in those later steps.

1. James Gleick The Information (Vintage, 2011) Page 150.

Method	Visualization Rating	Comment
Requirements Enumeration	Low	Uses static models, such as use cases and scenarios, but offers a limited level of visualization
Model-Driven Requirements	Medium	Provides some level of visualization by virtue of the business context provided by process modeling and the mock up of screens
Piloting/Prototyping	High	Comes close to presenting the end user experience of the targeted system

The following table lists the advantages and disadvantages of each of the evaluated methods:

Method	Advantages	Disadvantages
Requirements Enumeration	▪ The most straight-forward approach ▪ Requires the least amount of specialized skills on the part of the practitioners ▪ Potentially the lowest cost for initial requirements phase ▪ Easiest to manage, with clear delineations of responsibilities ▪ Potentially the least amount of upfront time commitment from end users ▪ Provides clear traceability	▪ Allows for a lack of clarity and specificity ▪ May be difficult for end users to accurately conceive requirements ▪ Often necessitates unplanned for requirements analysis work during design phase ▪ Can be difficult to discover inconsistencies and disconnects among separately delineated requirements ▪ Can result in higher backend cost and unrealized business savings ▪ Produces a much higher failure rate for complex systems

Method	Advantages	Disadvantages
Model-Driven Requirements	▪ Provides more comprehensive discovery of requirements and identification of improvement opportunities ▪ Produces requirements business users intuitively understand ▪ Includes process models that can drive a better understanding of data, and eliminate duplicate data entry ▪ Usually promotes more of a sense of ownership by the end users ▪ More in line with Agile iterative development and SOA	▪ Requires a higher skilled analyst ▪ Requires more extensive upfront commitment from the end users ▪ Requires a higher level of collaboration with less distinct lines of responsibility ▪ May face resistance to the required adjustment to the SDLC ▪ Does not provide a working model that users can visualize and test their requirements against

Method	Advantages	Disadvantages
Piloting / Prototyping	▪ Enables end users to experience the targeted environment and test/validate assumptions ▪ Produces a system that more accurately aligns with the actual business needs of the end users ▪ For smaller projects, reduces the time to gather and validate known and uncover unknown requirements ▪ Facilitates customer buy-in, thus typically shortening overall development cycle ▪ Provides the potential to reuse the working model as a basis for the final system, eliminating duplicative and unnecessary development efforts ▪ Provides the potential for a shorter development cycles ▪ Identifies risk early, preventing failure after spend scenarios ▪ Enables iteration and Agile development ▪ Can be performed directly in a BPMS or other tools, with no or little programming required	▪ Requires very highly skilled business analyst who can: (1) use Business Process Management Software (BPMS) to create working models, (2) understanding system fundamentals enough to ensure the piloted models can be implemented ▪ Requires a skill base that is in high demand, more expensive, and difficult to acquire ▪ Usually requires a higher degree of collaboration between the business analyst group and the IT group ▪ Delineation of responsibilities between the business analyst and IT group usually are not as clear ▪ User expectations must be even more carefully managed to control potential false perceptions that the working model is actually a system ▪ Produces higher upfront costs

Recommendation on Usage of the Methods

Requirements Enumeration

Both Model-Driven Requirements and Piloting/Prototyping could be overkill for mature processes, or incremental and maintenance system changes. **Requirements Enumeration**, with some screen mock-ups and data models, should be sufficient to achieve the required level of visualization at an acceptable cost.

Model-Driven Requirements

Model-Driven Requirements generally provide the required level of visualization for major upgrades to well known processes, where Piloting/Prototyping may be overkill. Model-Driven Requirements typically is the most cost effective in scenarios requiring a business case or risk analysis, and Piloting/Prototyping would be premature. Overall, Model-Driven Requirements results in refined requirements, at a lower total cost than Piloting/Prototyping. However, the confidence gained by a Working Model is not realized.

Piloting/Prototyping of Requirements as a Strategic Advantage

Piloting/Prototyping is the most effective method when there are significant process changes and/or a new process is being implemented. This method provides the additional benefit of delivering superior quality results in an overall shorter timeframe, and at lower cost. This is especially true when the Working Models do not need to be re-platformed.

New-to-market collaboration tools and BPMSs enable system savvy business analysts to quickly re-design and deploy critical business processes and supporting IT capabilities. Organizations with a heavy reliance on empowering their staff with IT would be remiss at not taking advantage of this opportunity.

Authors:

Kim Ellison, Dawn Levy, Mark Nelson, Charles Popelka

BPTrends Linkedin Discussion Group

We created a BPTrends Discussion Group on Linkedin to allow our members, readers and friends to freely exchange ideas on a wide variety of BPM related topics. We encourage you to initiate a new discussion on this publication, or on other BPM related topics of interest to you, or to contribute to existing discussions. Go to Linkedin and join the **BPTrends Discussion Group.**