



## **Getting your requirements right: Collaborate with stakeholders to work smarter.**

*Using IBM Rational Requirements Composer software*

*Brianna Smith, delivery engagement manager,  
Rational software, IBM Software Group*

*Lisa Garrity, technical professional,  
Rational software, IBM Software Group*

*Theresa Kratschmer, senior software engineer,  
Rational software, IBM Software Group*

---

**Contents**

---

- 2 *Introduction***
- 3 *Requirements management versus requirements definition 101***
- 4 *Requirements management***
- 4 *Defining requirements using improvised tools***
- 6 *Good tools used for the wrong purpose are no longer good***
- 8 *Solve requirements definition challenges with a proper solution***
- 11 *Prioritizing requirements definition by choosing the right tools***

**Introduction**

Given the economic downturn, “cheaper, better, faster” seems to be a universal mantra in business. To stay competitive, organizations must continually strive to be more agile and develop higher-quality solutions more quickly—despite obstacles such as geographically distributed teams, limited budgets and resources, quick delivery times, language barriers and government regulations. These challenges require teams to consider new ways of doing business so they can be more responsive to frequent business changes.

One area that businesses can optimize is their software development processes. If they want to be competitive, companies don’t have the luxury of long development lifecycles. To keep timeframes short, organizations must foster a collaborative environment by making tasks and responsibilities transparent and breaking down silos across the development lifecycle.

At the heart of the collaborative development environment are software requirements. Business analysts, product managers, developers, testers and architects leverage requirements as part of their daily activities. By ensuring that these players have access to the content they need at the right level of detail, organizations can be more nimble and better respond to change within the development environment.

Requirements definition is the foundation of effective software delivery. Requirements should be developed—or defined—through consensus among stakeholders to address their needs, business problems and the vision of the software. Business demands require that solutions meet stakeholders’ goals and objectives—no more and no less.

In this white paper, you will learn about the changing focus of the requirements space and how requirements definition differs from requirements management. You will also read about a business analyst’s struggles with defining requirements using common office software and tools. This paper addresses how IBM Rational® Requirements Composer software can streamline definition processes and help you achieve a faster ROI by reducing costly rework and speeding time to market.

---

Highlights

---

**Though many project teams tend to focus on managing requirements versus defining them, organizations are beginning to realize that defining requirements is key to effective software delivery.**

### Requirements management versus requirements definition 101

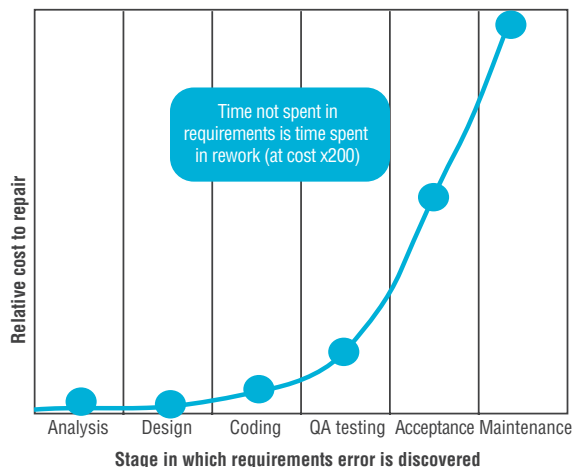
In the requirements space, people often focus on *managing* requirements. But few focus on *defining* requirements. Although developers partake in requirements definition on some level, it is often informal, and few consider requirements definition as an area for improvement. But defining requirements is key to effective software delivery. Organizations are beginning to realize the importance of understanding their stakeholders' business problems and gaining agreement on the vision of the end product. By investing time in requirements activities — either early in the software development process or later as change occurs — you can save time, effort and money in the short and long terms.

#### Requirements can impact time and effort

- Errors cost 1–200 times the budget cost\*
- More than 40% of IT development budget will be consumed by poor requirements†
- Typically 20–40% lines of unused code‡

#### Requirements can impact project success

- 28% of projects on time and in budget‡
- 49% of projects overrun original estimates‡
  - Time overrun averaged 63%
  - Cost overrun averaged 45%
- 23% of projects canceled before completion‡
- Driving forces: Low executive support, low user involvement, unclear business objectives, no scope control, inability to capture firm basic requirements‡



\* IBM research † IAG consulting ‡ Standish group

Figure 1: Realizing that software requirements are misaligned with stakeholder needs and goals later in the development lifecycle is much more costly and time consuming to rectify.

---

**Highlights**

---

***If stakeholders do not reach consensus during the requirements definition stage, project teams will have difficulty building the solution without rework.***

***Requirements management governs requirements throughout the development lifecycle.***

#### Requirements definition

Requirements definition includes eliciting, specifying, analyzing and validating requirements. It is an iterative process in which one or many stakeholders define and refine problem statements and then conceptualize potential solutions and their impacts. By involving all pertinent stakeholders, you can understand their business goals and help ensure that the defined software requirements align to them. If stakeholders do not reach consensus during the definition stage, developers will have difficulty building the solution without rework — resulting in longer development lifecycles and higher costs.

#### Requirements management

Requirements management governs requirements throughout the development lifecycle. Product and software teams gather, store, track, prioritize and implement requirements captured during definition phases. It is the process of communicating and controlling software development scope while understanding and incorporating changes simultaneously.

#### Defining requirements using improvised tools

Whether it is acknowledged as a formal activity or not, developers must define requirements in some way before creating software. The following scenario describes the experience of Anna, a business analyst for a large technology company, in eliciting, defining and elaborating requirements.

---

### Highlights

---

***Common office software and flip charts become time consuming and challenging to use, especially when the requirements process becomes more complex and dynamic.***

Anna had been using the Microsoft® Office suite as well as office supplies such as whiteboards, flip charts and notepads to document requirements definition meetings with her software project clients. During meetings, Anna used flip charts and whiteboards to capture a client's project expectations and goals. Later she would transfer the information into Microsoft Word and Microsoft Excel to store and share text records, and then use Microsoft Visio and Microsoft PowerPoint to create visuals and use-case diagrams. Keeping track of the information contained in of all these tools was time consuming, but they served their purpose. It was when Anna needed the tools to go *beyond* their intended purpose that she struggled.

When the entire client team was unable to meet at the same time, Anna's team had to host several use-case sessions. This not only drew out the process but also caused problems when the stakeholders in different sessions did not agree on requirements or priorities. To achieve a consensus, Anna had to consolidate all of the information from all of the meetings into a single document that could be distributed among all the stakeholders. In one particular instance, she spent days creating a storyboard in Microsoft PowerPoint for the software requirements and then distributed the document to 14 client stakeholders. After adding all of their changes, the client stakeholders were presented with a document printout that was three inches thick, requiring weeks of review before requirements could finally be defined.

---

### Highlights

---

***Defining requirements using different rudimentary tools can result in a loss of time, effort and money.***

***Using tools not intended for requirements definition usually means time will be wasted, collaboration will be stifled and requirements updates will be missed.***

In this same episode, the client needed to demonstrate the software to its investors just three months after agreeing on the requirements. But because the client had engaged a third party to write an application that implemented some of the defined requirements, Anna's team needed to identify the impacted use cases so the client could properly present the software to the investors. Anna had to manually map the use cases to the requirements to find out which were affected by the third-party application. This activity took a great deal of time and ultimately delayed the project.

Defining requirements using different rudimentary tools can result in a loss of time, effort and money. Anna had to duplicate work by copying relevant information from one tool to another. She had to prepare for and host several meetings to gather information. And she had to navigate through a tangled web of communication — a major problem when trying to gain consensus on a solution. Not only did these challenges affect Anna's team, but they also impacted customer satisfaction.

#### **Good tools used for the wrong purpose are no longer good**

The scenario illustrated above is representative of the way many business analysts and other professionals use common office tools and processes to develop requirements. But like Anna experienced, there are three major issues with using common office tools:

- *Using disparate tools is labor intensive for both the business analyst and the stakeholders.*
- *It's challenging to use office tools and processes to obtain stakeholder consensus on business objectives and associated requirements.*
- *It's difficult to use disparate tools to manage changes to the requirements definition artifacts.*

---

**Highlights**

---

***When you use unintegrated tools, you can waste a lot of time cutting, pasting, dictating or transcribing information from many places into one.***

***Gaining consensus among stakeholders is difficult when only some of them have the right tools and only others of them can attend meetings.***

Time consuming and labor intensive

Using several different tools and various techniques for capturing the information related to the problem and potential solution can be exhausting. If you capture text and diagrams on whiteboards and flip charts, you'll need to transfer that information into something you can preserve and easily share. Depending on the amount of information you've gathered and how many different tools you've used, this could take hours, days or even weeks.

When you use unintegrated tools, you need to find a way to relate data across them. You can waste a lot of time cutting, pasting, dictating or transcribing needed information into a single document, and it can be difficult to visualize the relationships among various pieces of information. Further, you need to choose a single tool that your clients and other stakeholders can easily review—but if you choose the wrong one, you'll need to transfer all the information all over again.

Difficulty in gaining consensus

Gaining consensus among stakeholders can be challenging. If you want the stakeholders to review the requirements documents using the tools in which you created them, then all stakeholders need access to those tools, and they need to know how to use them. In addition, they need a method of commenting on the documents or individual artifacts.

If stakeholders don't have all the tools in which you created the documents, then you must consolidate everything into one large artifact. Stakeholders often don't have the time or inclination to read through long documents. Consequently, you receive a small percentage of the feedback you need to deliver a solution that will bring value to your client.

If you have to hold multiple meetings to accommodate different schedules, you also struggle to gain consensus. At least a few stakeholders will always be at a disadvantage because they're missing important insight and feedback from other stakeholders in other meetings. In that situation, it can be very challenging for everyone to agree on and prioritize the requirements, thereby lessening the group's knowledge and understanding of the proposed solution.

---

## Highlights

---

***When you need to make changes to requirements that have related information stored in multiple tools, it's nearly impossible to find all the related data to update—leading to rework and overrun budgets.***

***Requirements definition software such as Rational Requirements Composer helps organizations enhance and simplify their requirements processes and address the challenges outlined in this paper.***

Inability to manage changes to goals, objectives and requirements

When you have documents or artifacts in many different tools, it's difficult to identify relationships between them. So when you make changes to requirements, it's nearly impossible to update every related document or artifact that needs to be changed.

Whether you're defining or changing requirements, the facts surrounding projects and requirements in particular are staggering:

- *Thirty percent of all project costs are associated with rework.<sup>1</sup> Requirements errors account for 75 to 80 percent of this cost.<sup>2</sup>*
- *Requirements activities account for 35 percent of a project's effort. Wait time and redundant activities can consume 10 percent of the total budget.<sup>3</sup>*
- *A six-month delay can cost companies 33 percent of ROI in a five-year business case.<sup>4</sup>*

IBM can help you address these challenges head-on with the IBM Rational Requirements Composer solution.

### **Solve requirements definition challenges with a proper solution**

Requirements definition software such as Rational Requirements Composer helps organizations enhance and simplify their requirements processes with user-friendly requirements elicitation, elaboration and validation capabilities. It helps evolve captured and refined business needs into clearly defined software requirements that improve quality, accelerate time to market and align business and IT across the development lifecycle. The software provides users with the ability to visualize results before committing resources, helping users avoid costly requirements failures and project rework.

To address the problems detailed in this paper, Rational Requirements Composer offers enhanced requirements development features.



---

## Highlights

---

***Rational Requirements Composer enables users to quickly and easily create rich text documents, business process diagrams, use-case models, sketches and storyboards, which help companies define the right solution faster.***

***Rational Requirements Composer helps teams collaborate by enabling them to more easily share information about requirements artifacts.***

### Visual and textual techniques

Rational Requirements Composer can help greatly reduce time spent defining requirements and ensuring that they align with an organization's business goals and objectives. Using multiple definition techniques, the software enables users to quickly and easily create rich text documents, business process diagrams, use-case models, sketches and storyboards. Users can create these assets by simply dragging and dropping widgets onto the editor workspace. Additionally, using Rational Requirements Composer, you can link a business process diagram, use-case model, sketch, storyboard, requirement or any artifact—or element in that artifact—directly to any other artifact or element in that artifact to quickly establish relationships.

### Stakeholder collaboration

Rational Requirements Composer is built on the IBM Jazz™ collaboration platform, which helps enable distributed teams to closely work together.

In the software framework, a user can add input on any artifact or any element of an artifact by clicking on an icon and typing in comments. So the author can tell stakeholders what he or she specifically wants them to review, and stakeholders can create and view comments no matter the artifact's format: text in a rich text format document, an actor in a use case, a task in a business process, or an image in a sketch.

Users can access comment threads from the specific artifact they address or view them as discussion threads in the sidebar of a user's Rational Requirements Composer project home page. In addition, comments can be prioritized and set to "resolved" once they've been addressed. This capability helps make communication transparent by enabling stakeholders to review existing and resolved comments and questions so the same issues are not resurrected time and again.

---

## Highlights

---

***Rational Requirements Composer has a centralized repository with tagging, filtering, and advanced searching functionalities, helping to nearly eliminate time wasted finding documents, comments, associated images and more.***

***Using integrated features, Rational Requirements Composer enables teams to select adaptable lightweight frameworks.***

These features simplify gaining consensus among stakeholders because virtually all project stakeholders can easily access requirements data and discussions, often eliminating the need to hold formal meetings. Using Rational Requirements Composer, stakeholders can view their project home page to read their colleagues' opinions and input, and they can add their own responses to discussion threads. Commenting on artifacts within the software is a much simpler and faster method than cutting and pasting information into a separate document.

### Centralized requirements information repository

One of the most time-consuming activities in a software development project is finding information: documents you or others have created, comments, associated images and more. Rational Requirements Composer has a centralized repository so all requirements information is stored in one place. By using folders, tags, filters and advanced searches to customize their project home page, stakeholders can categorize and find information quickly and easily, thereby greatly facilitating reuse of requirements.

### Best-practice framework options

Many organizations are seeking ways to adopt requirements processes in a less prescriptive manner. Teams don't want heavyweight process techniques; instead, they prefer lightweight frameworks that can adapt to any requirements process. Rational Requirements Composer offers such frameworks via its integrated IBM Rational Process Advisor feature, which enables teams to select techniques that can help in their requirements definition and management processes.

---

Highlights

---

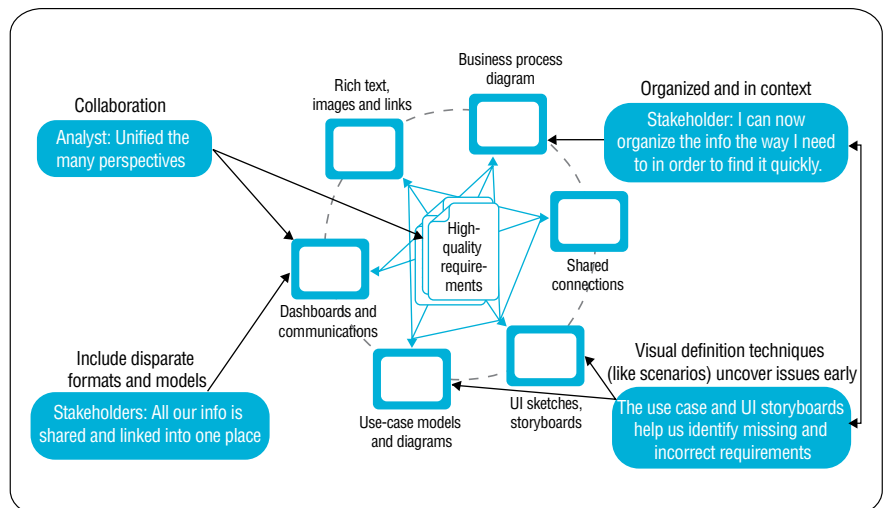


Figure 2: Connecting pieces to the bigger picture reduces costly mistakes and omissions. You can use requirements definition tools to capture, connect and organize the web of requirements information to help simplify information and make better decisions.

**A good requirements definition solution helps ensure that you are creating, testing and tracking requirements that meet your stakeholders' business goals and objectives.**

### Prioritizing requirements definition by choosing the right tools

Since requirements are the foundation for effective software delivery, relying on multiple rudimentary tools with no integration, collaboration or process capabilities is not going to facilitate a streamlined development process. In fact, it will likely decrease productivity, slow time to market and ultimately negatively impact customer satisfaction. A good requirements definition solution helps ensure that you are creating, testing and tracking requirements that meet your stakeholders' business goals and objectives.

When Anna discovered Rational Requirements Composer, it couldn't have been at a better time. Because of the economic downturn, her company travel became severely restricted. Rather than hosting face-to-face meetings with clients and stakeholders, she leveraged the collaboration capabilities and central repository of the software to share information and achieve consensus on requirements definitions. Moreover, she used user interface sketches, storyboards and use-case models to represent the requirements, significantly improving the quality of client feedback. And having integrated definition techniques on a collaborative platform meant Anna no longer spent time duplicating the requirements information across disparate tools.



When you properly define and build requirements — with collaboration, stakeholder consensus and requirements verification — you can reduce rework, increase reuse and productivity, and deliver your product faster. Teams may drive an understanding of high-level requirements sooner through visual prototypes; gain consensus about requirement artifacts in a collaborative fashion; and increase productivity and reduce rework with better ways to organize and retrieve information — all leading to a faster time to market and value. And as a result of this, you can increase customer satisfaction and loyalty, helping your organization get a larger slice of marketshare.

#### For more information

Rational Requirements Composer was specifically designed to address the problems discussed in this paper. By solving these issues, you may realize a near-immediate ROI. If you are ready to explore the capabilities and benefits of Rational Requirements Composer, download the software demo at:

[ibm.com/developerworks/offers/lp/demos/summary/r-rrcoverview.html](http://ibm.com/developerworks/offers/lp/demos/summary/r-rrcoverview.html)

You can also download the free trial evaluation at:

[ibm.com/developerworks/downloads/r/rrc/?S\\_TACT=105AGX28&S\\_CMP=TRIALS](http://ibm.com/developerworks/downloads/r/rrc/?S_TACT=105AGX28&S_CMP=TRIALS)

For a candid look at the software and its users, visit and join the Rational Requirements Composer fan page on Facebook at:

[www.facebook.com/home.php?#/pages/Rational-Requirements-Composer/47378244431?ref=ts](http://www.facebook.com/home.php?#/pages/Rational-Requirements-Composer/47378244431?ref=ts)

And in Q3 2009, be on the lookout for part two of this white paper: “How to uncover missing requirements: ways to work smarter with stakeholders.” It will describe in more detail how you can leverage Rational Requirements Composer to drive out an understanding of requirements.

To learn more about IBM Rational Requirements Composer, contact your IBM representative or IBM Business Partner, or visit:

[ibm.com/software/awdtools/rrc/](http://ibm.com/software/awdtools/rrc/)

© Copyright IBM Corporation 2009

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
May 2009  
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

- 1 Dean Leffingwell and Don Widrig, *Managing Software Requirements: A Use Case Approach*, Addison-Wesley Professional, May 2003.
- 2 Tom King and Joe Marasco, “What Is the Cost of a Requirement Error?” StickyMinds.com, June 2007.
- 3 Noah B. Kindler *et al.*, “Applying Lean to Application Development and Maintenance,” McKinsey & Company, March 2007.
- 4 Donald Waters, *Operations Management*, Kogan Page, in association with PricewaterhouseCoopers, February 1999.