

Successful projects start with high quality requirements

Defining requirements clearly and accurately at the outset speeds software development processes and delivers dramatic savings.

EXECUTIVE SUMMARY

Software development projects suffer most when changes in requirements set off a cascade of delays, revisions and rework. Existing processes for establishing requirements are often ad-hoc and inefficient, leading to miscommunication and insufficiently defined requirements. By ensuring effective requirements definition (RD) at the outset – involving elicitation, analysis, specification and validation – enterprises can reduce rework, speed up development and deliver dramatic time and cost savings.

THE BUSINESS CHALLENGE OF POOR REQUIREMENTS PRACTICE

The successful enterprise is agile and responsive to changing business conditions and customer needs. It applies best practice to everything from data center management to security to CRM, aiming to maximize efficiency and optimize resources. It also invests heavily in application development, creating customized software built by programmers who work closely with business analysts, matching functionality with the company’s needs.

In reality, when it comes to best practices and efficiency, development projects frequently fail to live up to the ideal. The Standish Group’s Chaos Reports regularly point to projects which are plagued by a combination of delays, reduced feature sets, budget overruns and cancellations – and ultimately failure.

In the majority of cases, poor requirements definition and management practices remain a root cause of project failure and waste.

Of course, planning and launching a development project is complex. A wide range of stakeholders – business analysts, coders, quality engineers, legal and managerial staff, end users and IT administrators – have to collaborate on the initial set of requirements, and they must all evaluate and participate as the application evolves. At various points in the process, any of these stakeholders can introduce change into the workflow. This complexity can lead to miscommunication which, in turn, leads almost inevitably to the problems mentioned above, involving costly and time consuming rework.

A survey by Cutter Consortium found that, ‘E-projects are time compressed, intensive, and mission-critical efforts with poorly defined requirements.’

Respondents cited several problems that torpedoed the timely delivery of software projects:

- Unstable, constantly changing requirements (66%)
- Poor requirements specification (55%)
- Client behavior, such as approval delays, requirements changes and poor communication (42%)¹

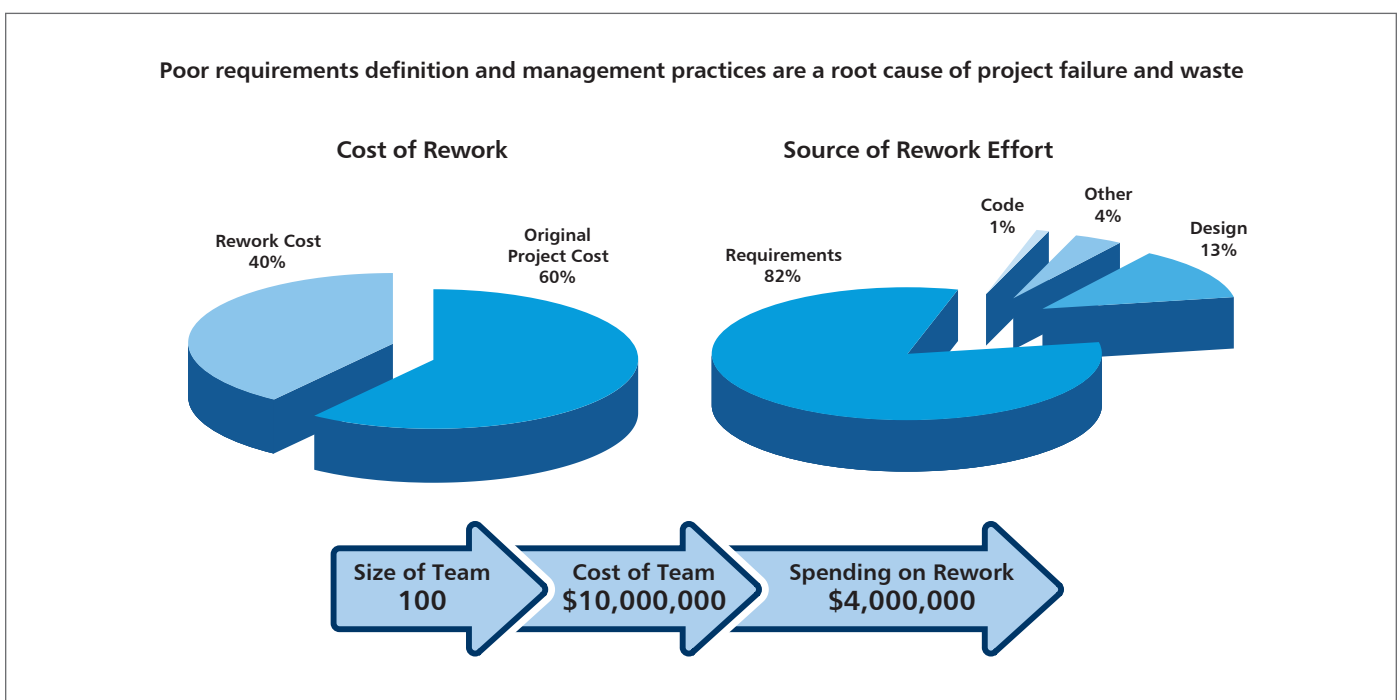


Figure 1: Software rework and waste

CHANGE REQUIREMENTS AND LOSE MONEY

Many factors affect the development process including unexpected budget cuts, time pressure from senior management, personnel turnover and bugs in the code. However, it is changes in requirements that are often the most disruptive, problematic and costly.

Take, for example, a scenario where a project is already well underway when a principal indicates that several initial specifications are inaccurate. A company could be updating its sales force applications to align with an upgraded, more versatile database structure. Development work is in progress when the CIO points out that as well as interfacing appropriately with the back-end, the new application must also ensure secure transmission of customer data and feed into data-mining and customer relationship management tools. At this point, work that has already been completed must be modified and reworked to reflect the changed requirements. Changes introduced later in the process are very disruptive and require significant time and effort to accommodate.

Catching and fixing problems at the requirements definition stage has a far greater effect on improving the final software product than changes made later in the process.

The cost of correcting defects rises exponentially the later in the development lifecycle they are identified. Research carried out by HP² highlighted that a hypothetical \$1 to fix a defect found in the requirements phase becomes \$2-3 when the defect is located in the design phase, rising to \$5-6 in coding, \$18-20 in testing, and a staggering \$100-110 if the change is made once the application is released.

In addition to the monetary cost, several studies have shown that 40% or more of the effort within a software project is taken up with rework. It follows therefore, that defining requirements correctly at the outset is perhaps the single most effective way to reduce overall development time and resources.

THE REQUIREMENTS PROBLEM IS A COMMUNICATION PROBLEM

A poor track record of defining requirements is often down to an inability to communicate effectively. Getting concepts across to different stakeholders who, in turn, filter information according to their own agendas can leave intentions open to misinterpretation, as ideas pass from person to person, team to team.

In addition, today's adult learners – or 'net-geners' – have different needs in terms of the most effective way to absorb concepts and ideas, compared to even a decade ago:

Net-geners learn differently³

By 21 years of age, the internet generation will have spent:

- under 5,000 hours reading
- 10,000 hours on cell phones
- 10,000 hours playing video games
- 20,000 hours on e-mail, chat, blog
- 20,000 hours watching TV

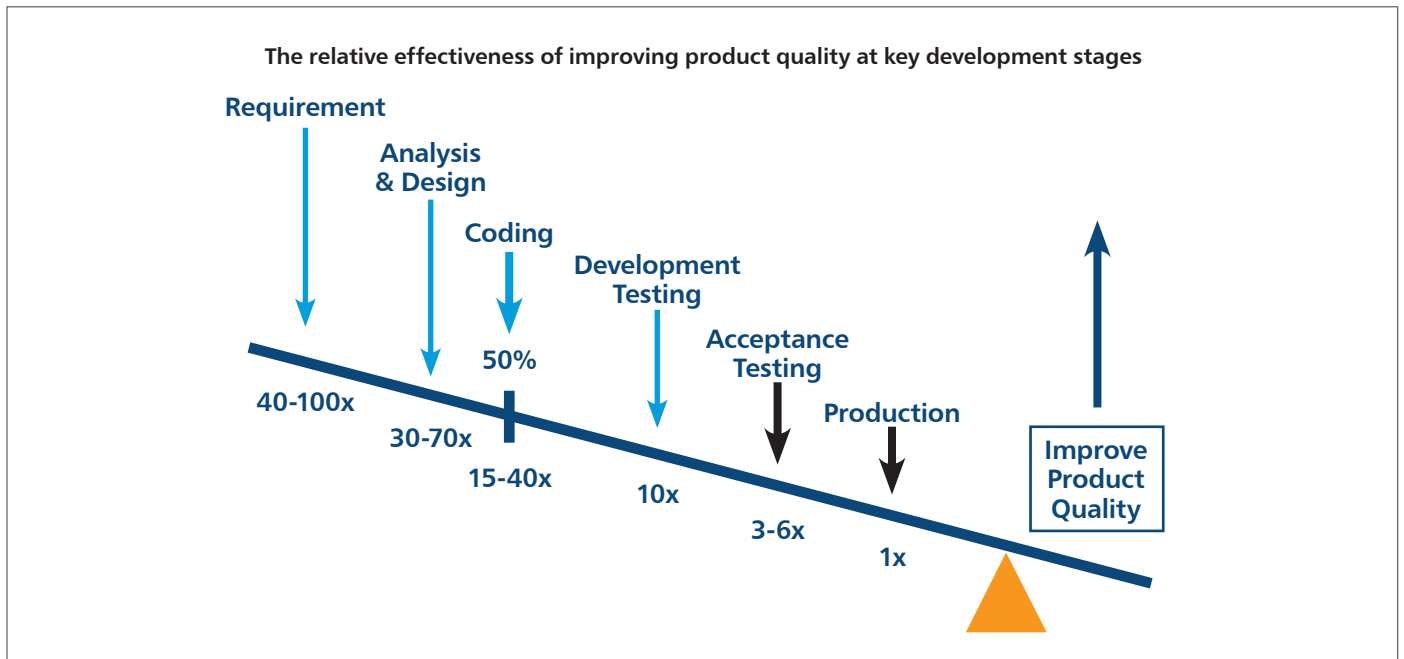


Figure 2: Leveraging the power to improve

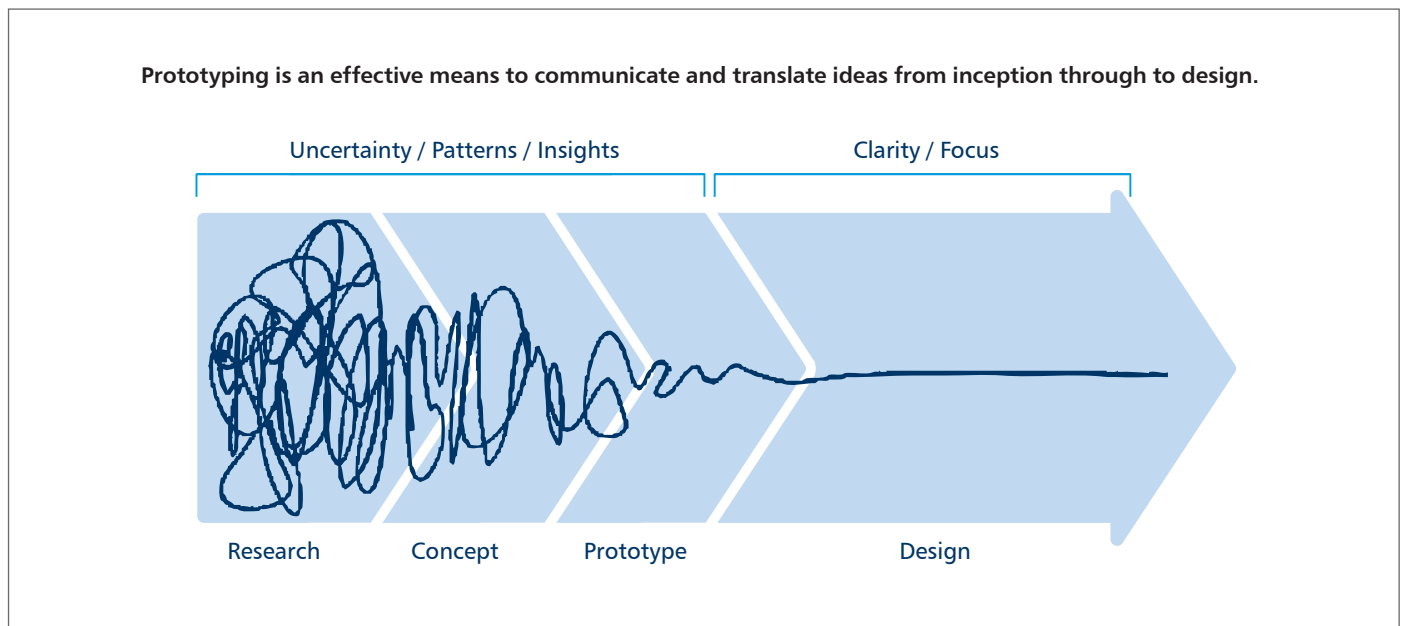


Figure 3: From uncertainty to focus – how prototyping can bridge the gap

To learn effectively, they need:

- Fast paced, highly stimulating presentations
- Increased interactivity with content and each other
- Information that relates to the learners' world
- Multiple options for obtaining knowledge

These net-geners form the nucleus of people building software now and for the next decade or two – the very software on which business relies today, and which will determine development and growth in the future. Harnessing the abilities of this generation of adult learners and information workers for project success demands a rethink of how requirement information is conveyed. A strictly document-centric approach may have been the de facto standard for many decades, but a more interactive and dynamic approach is required going forward.

This has driven a shift in emphasis towards Requirements Definition and not just Requirements Management in software development. Modern requirements definition techniques include the use of a wider variety of tools and approaches than ever before, and clearly any method that involves more collaborative, visual and interactive ways to convey concepts will be attractive to current and next generation information workers. This is where prototyping has special relevance as a technique to help organizations transition smoothly from business concept to eventual design – driving focus on the execution and delivery of a software project.

While it has long been recognized that requirements need careful ongoing management once in production, organizations are increasingly realizing that higher quality requirements need to go into the software development process in the first place.

GETTING IT RIGHT THE FIRST TIME

Much of the delay, rework and waste associated with a development project can be avoided when requirements are defined clearly and collaboratively from the outset. This is achieved by focusing on the requirements definition process. Many companies overlook this crucial step and let business analysts drive the requirements process; defining and communicating their vision of the application’s inputs and outputs using text documents, spreadsheets, presentation slides and e-mail. There is no central monitoring or file repository for the vetting process, and everyone is expected to read, comment and revise cumbersome documents until, eventually, a set of requirements emerges from the chaos. To further complicate the problem, most requirement management software tools don’t provide Requirement Definition (RD) functionality but simply track changes once a project is underway and help traffic the code through design, testing, quality assurance and deployment.

A thorough, easy- to-use, RD process and technology that integrates smoothly with a company’s existing application lifecycle management (ALM) resources is needed. By implementing a reliable and consistent RD process workgroups can collaborate on activities that ensure requirements are established and defined by the right parties and approved by all the stakeholders.

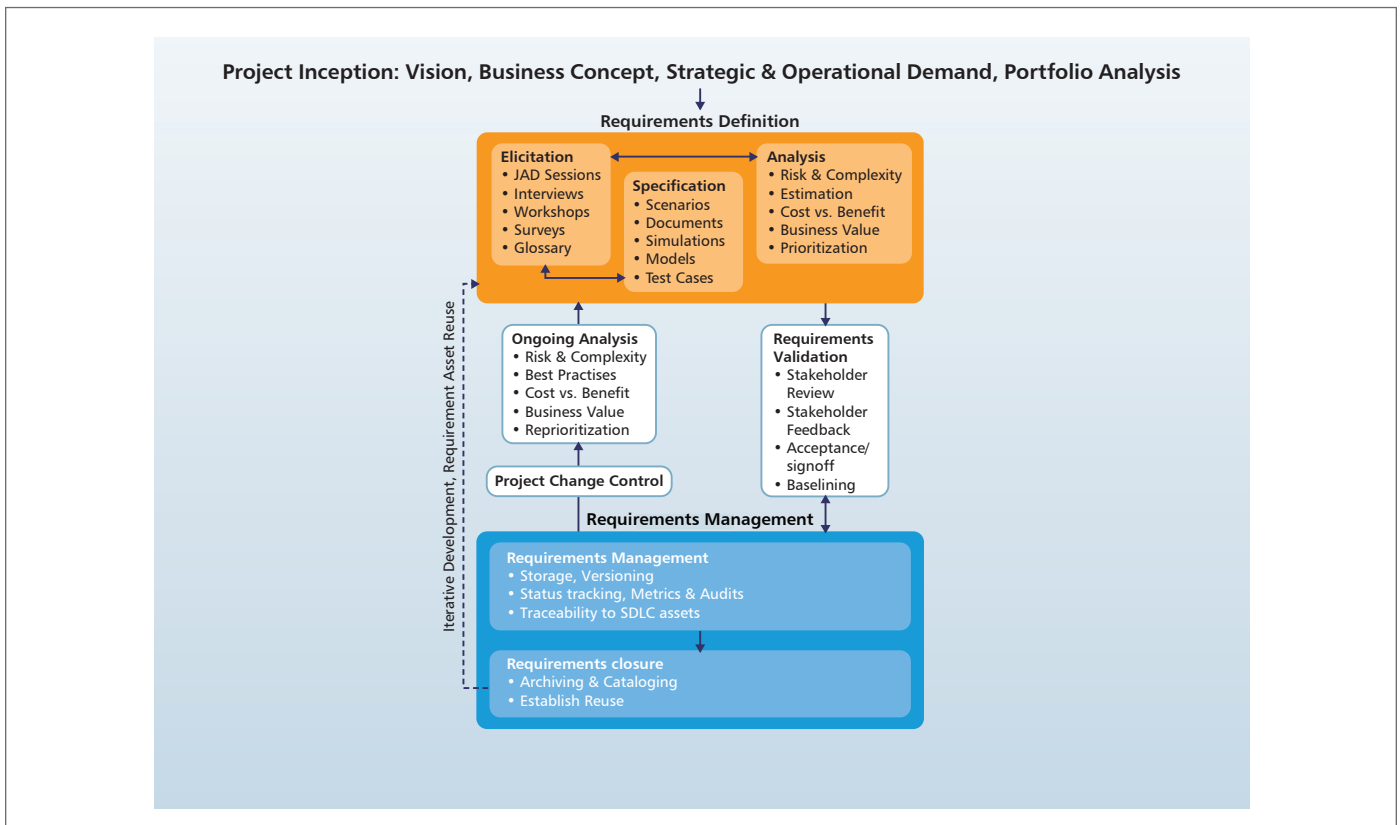


Figure 4: The requirements definition & management process flow

The four-step process for Requirements Definition:

➤ **Elicitation**

As the first step, all the individuals involved in defining the requirements collaborate to outline their needs visually. The parties determine what business flows the application will deliver, who the users will be and how they will utilize the software.

➤ **Analysis**

Once the business flows are collected, the development and IT teams perform a reality check to ensure that they understand what is needed and can deliver it based on the resources at their disposal. This helps verify the feasibility of the plan and catches any serious problems or inconsistencies early on.

Analysis includes prioritization, a step that saves time and enables smoother adjustments later in the cycle by ranking the various functions and features of the application in order of importance. This way, the high-priority tasks are built first, with mid-tier and lower-priority features added as time and funds allow. If something changes and the full list of requirements can't be met immediately, the application will still carry out its core functions and serve the prioritized end user needs.

➤ **Specification**

As the requirements take shape, stakeholders are able to add detail through expanded use cases, business rules, business models, and prototypes. This step involves documenting the requirements, establishing the management protocol to be followed and determining the way this project integrates with existing applications and processes.

➤ **Validation**

Once all the requirements are specified, all the stakeholders validate that their initial vision is reflected in the flows, and that the details are accurate and complete. At this stage business analysts interactively review the scenarios to ensure they deliver the desired results, and end users ensure that the flows will be most effective in meeting their needs and integrating with their work environments. Test cases and release criteria are established based on these requirements.

By following this four-step process, enterprise groups collaborate simply and effectively, bridging the gap between business workers and technologists and creating a common platform for their vision to take shape (see Figure 4). For elicitation and analysis, stakeholders should share a visual, business process-style modeling tool to create business scenarios, steps, decisions, and identify actors. Each party involved in the RD process should be able to access and view the requirements in the format they understand best-text, flowcharts, or code-to facilitate review and approvals.

Visual depictions help all stakeholders collaborate on software requirements.



Figure 5: Software simulation creates clarity and stakeholder buy-in

An effective RD solution that is aware of the ALM dependencies will generate business scenario diagrams to enable fine-tuning and documentation, and should support attachments of other resources, such as documents or files. Prioritizing requirements saves significant time and effort throughout the development cycle. Good requirements definition practice ensures that the most-needed features are placed high on the list of developer priorities.

In addition, the test cases generated by an RD tool should mirror the business process flows created by the users and analysts, so testing and Quality Assurance (QA) personnel are not guessing or duplicating effort in running the application through usage scenarios and models that don't relate directly to the original driving requirements.

CHAIN REACTION OF BENEFITS

When requirements are defined accurately and ratified by all stakeholders before the development team begins work, efficiencies accrue throughout the project lifecycle:

- The design and coding tasks follow the models already agreed upon by business analysts, users and IT staff, dramatically reducing misunderstanding and miscommunications
- As fewer clarifications and adjustments are required later on, rework is reduced
- By developing features with respect to priority, unnecessary or rarely used elements are less likely to hinder the project's progress as the development team focuses on the features that deliver the most return
- Testing and QA can be carried out directly against the requirements, removing the risk of quality teams focusing on the wrong test cases and thereby increasing user acceptance
- Performing testing and QA is faster and more efficient when carried out earlier in the lifecycle and as development progresses. This saves time and money over the traditional practice of testing the finished code against requirements
- End user satisfaction rises as the finished product matches initial expectations and requests

BUT DOES IT WORK?

Requirements definition and management tools don't just increase efficiency in theory, major development organizations implementing RD have seen dramatic results in practice.

For example, global business consulting firm CapGemini builds Oracle applications for some 200 client projects annually. Before implementing RD, its application design period from first distributing process flows to stakeholders, to conducting workshops, capturing and correcting future process flows and creating and validating test scripts, took about 20% of the overall project time.

Process flows were depicted on PowerPoint slides and Word documents outlining procedures and step-by-step requirements. The text in the Word document wasn't linked in any meaningful way to the PowerPoint graphics, and the size and cumbersome nature of the files rendered real-time discussion and collaboration impossible.

After launching an RD tool and incorporating it into the design phase, CapGemini⁴ reported a 40-66% reduction in the time needed to capture, define and validate requirements. Design teams use the software to document business process flows in real-time, while users describe the processes they employ and expect for future tools.

The 2009 CHAOS report from the Standish Group points out that a staggering 50% of software features are either never used or wasted effort, while a significant percentage of other features are simply missing. Over- and under-building applications represents the biggest form of software development waste. Using simulation as a requirements definition technique identifies unused or infrequently used features in a proposed system, prior to committing to the expense of developing and testing such features. Simply put, prototyping software through simulation helps ensure you deliver more of what's needed and less of what will go unused.

THE BEST TOOLS FOR THE JOB

When looking to implement a comprehensive requirements strategy there are a number of factors that a requirements tool should deliver.

Requirements need to be expressed by way of both business scenario diagrams as well as rich text requirements descriptions. By enabling different views of the project, business analysts, end users and developers can each review and verify the information in the 'language' best suited to their work style, whether they require more traditional text or visual models.

Requirements Visualization

- Visual clarity reduces process ambiguity
- Identifies missing or wrong underlying assumptions
- Stakeholder feedback improves quality of requirements
- Establish common vision and shared responsibility

Visual business scenarios are an effective means of communicating business process flows and concepts. Visualization helps reduce ambiguity and, for most stakeholders, presents an almost instant way to recognize issues of sequencing, broken decision points and even missing process steps or assumptions – all elements which are easily missed when describing business concepts and processes is limited to a text based only approach.

To ensure that business processes are accurately described and improve the overall quality of all derived requirements within the development lifecycle, a RD tool has to reach many stakeholders for process validation. The most appropriate way to ensure a common vision and shared responsibility for a project's outcome is by simulating business scenarios.

Simulations help cement implementation consensus. They simulate the pro-active validation of an interpreted set of business process flows. By effectively letting stakeholders try before they buy, the likelihood of delivering applications exactly to expectation is significantly increased. Simulations also provide the key learning experience by which stakeholders get to grips with the limitations as well as the possibilities of an envisioned system. This creates opportunities to explore innovative solutions to problems that otherwise only present themselves deep into the actual development process where it is much more expensive to address these issues.

Pervasive feedback mechanisms built into the simulation ensure that stakeholders are made part of the process and therefore buy in to the end result. This allows for very quick turnaround on stakeholder feedback, so acceptance can be carried out quickly.

Requirements Simulation

- Solidify implementation consensus
- Pro-active validation, try before you buy
- Platform for innovation
- Stakeholder buy-in

Managing textual requirements in direct alignment with the visualization of the same requirements in a dynamic and interactive simulation is of immense value to those concerned with managing system specifications and contractual software delivery obligations. Authors of textual requirements should be kept in direct alignment with business needs as well as the technical realities of their implementation. And, because every project is unique, being able to customize the requirement types and containment with unique properties is required in order to support agreed ways of working. As requirements should never live in isolation, it's vital to extend the project's relevance by mapping them to supporting documentation.

Requirements management

- Rich text editing of requirements
- Aligns business needs with requirements
- Flexibility to support your process needs
- Traceability provides context

The days of leaving testing to the last step in the development process are long gone. By using a requirements tool to generate test cases derived from requirements in the inception stages of a project, quality assurance is involved much earlier in the development process. With earlier QA participation comes better alignment and shared responsibility for the end result.

Jump start testing

- Early test generation – not an afterthought
- Shared responsibility for project outcome
- Measure risk and complexity
- Build in quality

It's worth noting that simply creating test cases for every possibility has limited value unless you are able to identify exactly where the testing emphasis should lie. It is something of a luxury for software developing organizations to test each and every possible use case, so testing effort has to be focused. Tools which identify an application's high risk areas by calculating the magnitude of complexity in proposed systems will help QA teams establish meaningful testing priorities.

Once a project's requirements are defined and validated they must be tracked and managed throughout the development process, this ensures that quality assurance is embedded throughout the process.

EMBEDDING QUALITY THROUGHOUT DEVELOPMENT

Micro Focus champions embedding quality assurance practices throughout the lifecycle of projects, bringing the business value of delivered software in line with other goals and practices to maximize productivity and minimize waste. The Micro Focus requirements definition and management and testing solutions create a framework where tools work together flexibly to support any methodology or process already in place and integrate seamlessly with an organization's existing management solutions - either commercial or open-source.

Micro Focus offers solutions that support the three critical ALM processes involved in successful quality assurance: requirements definition and management, software change and configuration management and automated software quality solutions. These tools work together and with third-party software to automate and improve software development, with continuous techniques for defining, developing and validating software to increase time to market and deliver at far less cost.

CaliberRDM™ helps drive efficiencies in every subsequent phase of the lifecycle by enabling easy and accurate requirements definition in addition to powerful ongoing requirements management capabilities. CaliberRDM ensures teams stay true to the initial vision and keep tracking changes throughout the process, while making requirements accessible to unsurpassed numbers of stakeholders via its web based client.

StarTeam® provides a system for managing software configurations and automated build readiness checks. Teams are able to track, measure and improve release quality throughout the development process, while staying accountable and transparent to stakeholders at all times.

In addition, Micro Focus SilkCentral® Test Manager provides a full solution for planning, managing, executing and reporting software test activities, whilst being tightly integrated to the underlying requirements of a project. Traceability and test coverage of requirements provides insight into the actual health of a project.

SUMMARY

Accurate requirements definition and ongoing management means that organizations save time and money in the design phase, during development and throughout the testing and quality assurance processes. Micro Focus' complete portfolio of flexible and powerful requirement tools ensure that application development projects break free from traditional bottlenecks and delays, become streamlined and efficient, and ultimately drive growth and success for your business.

To learn more about the Micro Focus requirements definition and management solutions, visit www.microfocus.com/rdm

References

- 1 Rodrigues, Alexandre, "Project Goals, Business Performance, and Risk." Cutter Consortium e-Project Management Advisory Service Executive Update 2(7), 2001
- 2 Hewlett-Packard, "Applications of Software Measurement Conference," 1999
- 3 Susan El-Shamy. Training for the new and emerging generations. Quoted in <http://scope.lidc.sfu.ca/mod/forum/discuss.php?d=521> - 2010
- 4 CapGemini, "Accelerate Your Oracle Application Implementations," October 2006

About Micro Focus

Micro Focus, a member of the FTSE 250, provides innovative software that allows companies to dramatically improve the business value of their enterprise applications. Micro Focus Enterprise Application Modernization and Management software enables customers' business applications to respond rapidly to market changes and embrace modern architectures with reduced cost and risk.

For additional information please visit: www.microfocus.com

© 2011 Micro Focus IP Development Limited. All rights reserved. MICRO FOCUS, the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. All other marks are the property of their respective owners. WPSPQR0211