Successful Projects Begin with Well-Defined Requirements

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

Executive Summary

Software development projects suffer most when changes in requirements touch off a cascade of delays, revisions, and rework. Existing processes for establishing requirements are ad-hoc and inefficient, leading to miscommunication and insufficiently defined requirements. Effective requirements definition (RD) at the outset, involving elicitation, analysis, specification, and validation, will help to reduce rework, speed development, and lead to dramatic time and cost savings. Borland® Caliber® DefineIT™ provides a powerful and easy-to-use platform for requirements definition and integrates smoothly with Borland's family of open application lifecycle management tools.

Introduction

Successful enterprises are agile and responsive to changing business conditions and customer needs. They've applied best practices to everything from data center management to security to CRM, aiming to maximize efficiency and resource utilization.

They've also invested heavily in application development, because customized software, created by programmers who work closely with business analysts, is the best way to match functionality with the company's needs.

But when it comes to best practices and efficiency, development projects are often less successful. According to research from the Standish Group, many projects are plagued by some combination of delays, reduced feature sets, budget overruns, and cancellations.

The fact is, planning and launching a development project can be complicated. All the 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 those parties can introduce change into the workflow, and this uncertainty leads almost inevitably to the problems we mentioned above.

In fact, a survey by Cutter Consortium analyst Alexandre Rodrigues[1] found that, "E-projects are time-compressed, intensive, and mission-critical efforts with poorly defined requirements." Respondents cited several problems that torpedo 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%)

Change requirements, lose money

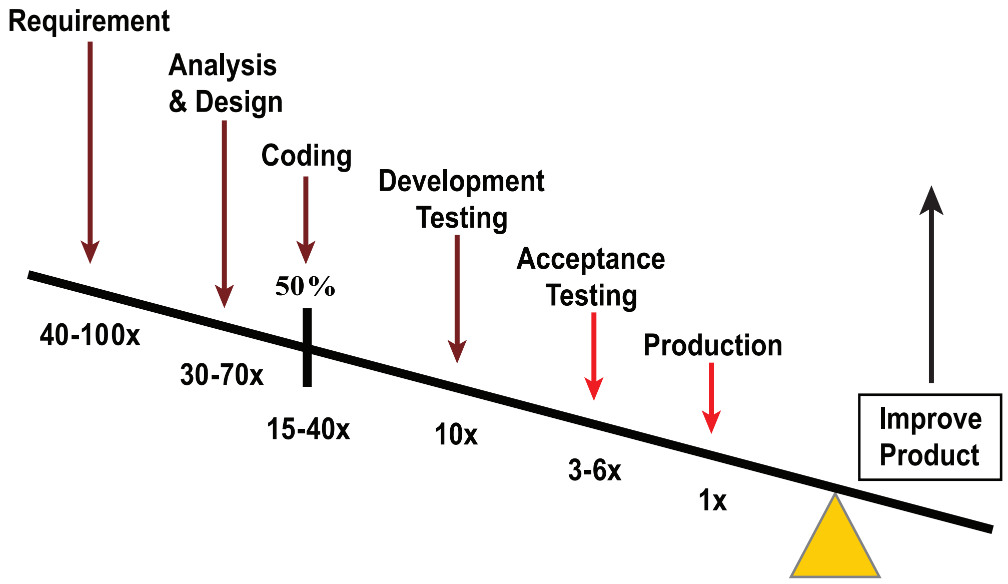
Many factors can affect the development process, including unexpected budget cuts, time pressure from upper management, personnel turnover, and bugs in the code. But changes in requirements are often the most disruptive, problematic, and costly.

Imagine that a project is already well underway when a principal indicates that several initial specifications are inaccurate. For example, a company may be updating its sales force applications to align with an upgraded, more versatile database structure, and the dev team starts work. Later, the CIO points out that in addition to 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 as well. At this point, work that's already been completed must be modified and reworked to reflect the changed requirements. It follows logically that changes introduced later in the process will cause a correspondingly heavier disruption and require more time and effort to accommodate. (see Figure 1)

[1]Source: Rodrigues, Alexandre, "Project Goals, Business Performance, and Risk." Cutter Consortium e-Project Management Advisory Service Executive Update 2(7), 2001.

Figure 1: Leveraging the power to improve

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



According to research conducted by HP,[2] a theoretical cost of $1 to fix a defect found in the requirements phase rises to $2-3 in the design phase, $5-6 in coding, $18-20 in testing, and $100-110 if the change is made once the application is released.

Multiple studies[3] estimate that software projects typically spend 40% or more of their total effort on rework, which means that requirements defined correctly at the outset can go a long way toward reducing overall development time and resources.

Getting it right the first time

Many of the delays, rework, and waste associated with a development project can be avoided when requirements are defined clearly and collaboratively at the outset, by focusing on the requirements definition process. Unfortunately, most companies overlook this crucial step and let the requirements process be driven mainly by the business analysts, who define and communicate their vision of the application's inputs and outputs using text documents, spreadsheets, presentation slides, or even e-mail messages. There is no central monitoring or file repository for the vetting process; everyone is expected to read, comment, and revise the cumbersome documents, such that a set of requirements emerges from the chaos. To further complicate the problem, most requirement management software tools don't provide RD functionality; they simply track changes once a project is underway and help traffic the code through design, testing, quality assurance, and deployment.

What's needed is a thorough and easy to use RD process and technology that integrates smoothly with a company's existing application lifecycle management (ALM) resources. 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.

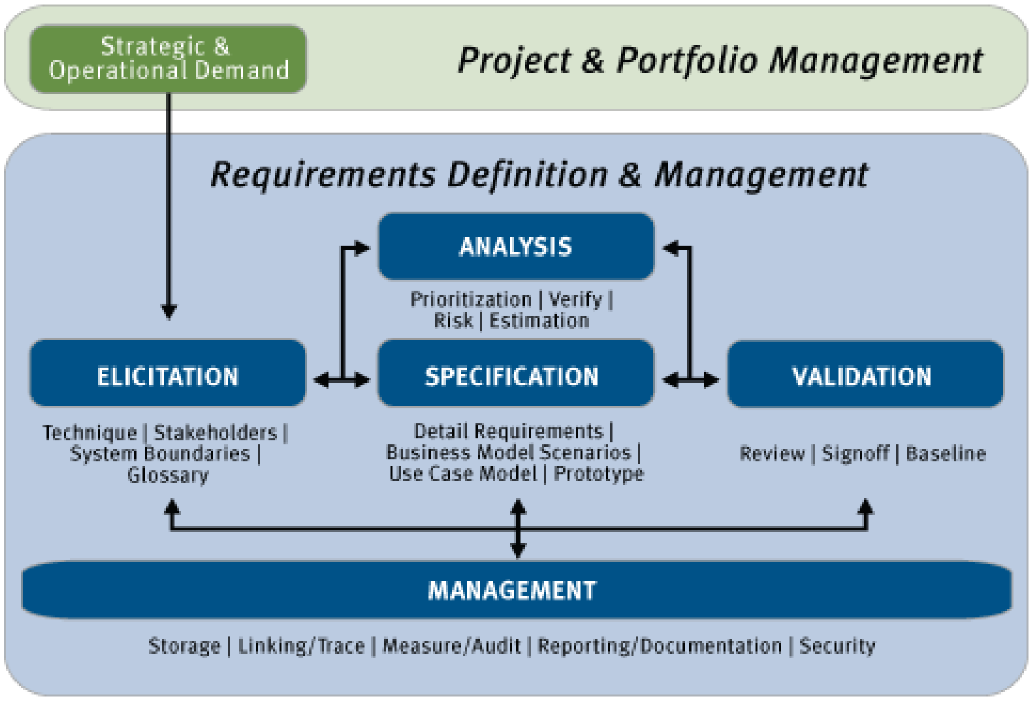
- **Elicitation:** As the first step, all the individuals involved in defining the requirements collaborate to outline their needs visually. The parties determine what business flows the application will deliver, who the users will be, and how they will utilize the software.

- **Analysis:** Once the business flows are collected, the development and IT teams must perform a reality check, ensuring that they understand what's 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.

[2] Source: Hewlett-Packard, "Applications of Software Measurement Conference," 1999
[3] Source: IEEE Spectrum, "Why Software Fails," Sept. 2005.

Figure 2: The requirements definition process flow

Effective RD enables smooth software design and requirements management.


Strategic & Operational Demand
Project & Portfolio Management
Requirements Definition & Management
ANALYSIS
Prioritization | Verify | Risk | Estimation
ELICITATION
Technique | Stakeholders | System Boundaries | Glossary
SPECIFICATION
Detail Requirements | Business Model Scenarios | Use Case Model | Prototype
VALIDATION
Review | Signoff | Baseline
MANAGEMENT
Storage | Linking/Trace | Measure/Audit | Reporting/Documentation | Security

- 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 app will still carry out its core functions and serve most of the end users' needs.

- **Specification:** As the requirements take shape, this step enables stakeholders to add detail through expanded use cases, business rules, business models, and prototypes. It also 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 in fact 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.

Following this four-step process for RD, 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 2)

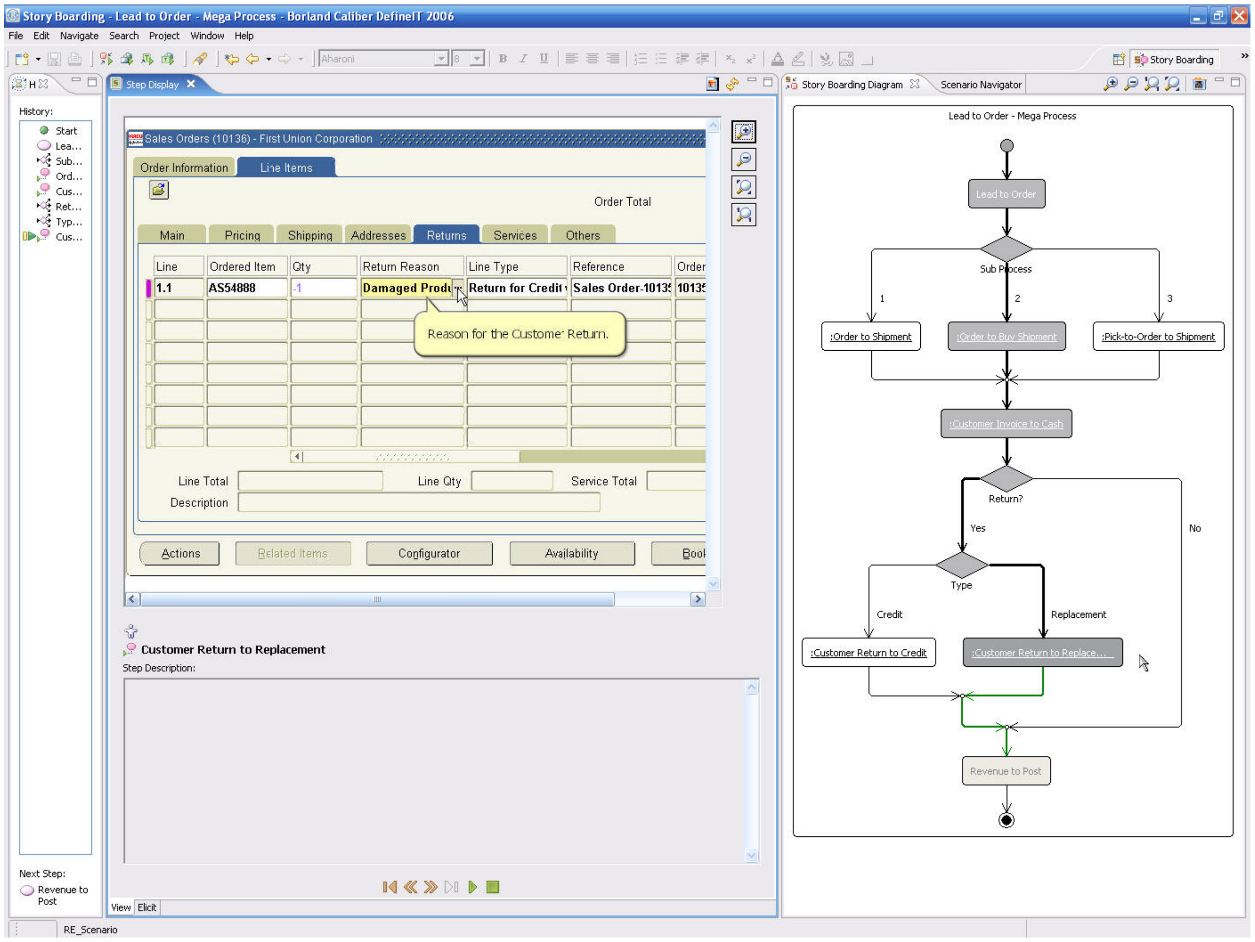
For elicitation and analysis, stakeholders should share a visual, business process-style modeling tool, with which to create scenarios, steps, decisions, and variables. Each party involved in the RD process should be able to access and view the requirements in the format they understand best—be it text, flowcharts, or code—to facilitate review and approvals.

An effective RD solution, that is aware of the ALM dependencies, will generate use cases to enable fine-tuning and documentation, and it should support attachments of other resources, such as documents or files. As we mentioned, prioritizing requirements saves significant time and headache throughout the development cycle, and a 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

Figure 3: Storyboarding in action

Visual depictions help all stakeholders collaborate on software requirements.



Line	Ordered Item	Qty	Return Reason	Line Type	Reference	Order
1.1	AS54888	-1	Damaged Produ	Return for Credit	Sales Order-1013	1013

analysts, so testing and QA personnel are not guessing or duplicating effort in running the application through usage scenarios and models that aren't related directly to the original 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's lifecycle:

- The design and coding tasks can follow the models already agreed-upon by the business analysts, users, and IT staff, so misunderstanding and miscommunications are dramatically reduced.
- Because there will be fewer clarifications and adjustments later on, rework is reduced.
- Features are developed with respect to priority, so unnecessary or superfluous elements are less likely to bog down the project's progress.
- Testing and QA can be carried out directly against the requirements; there's no opportunity for the quality teams to be focused on the wrong test cases.
- Testing and QA can be performed more quickly and efficiently—earlier in the lifecycle and as development progresses—saving time and money over the traditional practice of testing the finished code against requirements.
- End user satisfaction rises, because the finished product matches their initial expectations and requests.

But does it work?

Requirements definition and management tools aren't just increasing efficiency in theory. Major development organizations have implemented RD and seen dramatic results.

Global business consulting firm CapGemini, for example, builds Oracle applications for some 200 client projects annually. Before implementing RD, its application design period would take about 20 percent of the overall project time, from the first distribution of process flows to stake-holders, to conducting workshops, capturing and correcting future process flows, and creating and validating test scripts.

What's more, these process flows were depicted on PowerPoint slides, with an accompanying Word document 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 sheer 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, CapGemini[4] 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 best tools for the job

When your company is ready to implement a smart and comprehensive RD strategy, Borland Caliber DefineIT is the right choice.

DefineIT diagrams requirements steps automatically as they're entered, creating different views of the project so that business analysts, end users, and developers can each review and verify the information in the 'language' best suited to their work style: natural text for the non-technical parties and visual models for the developers and IT staff. (see Figure 3)

When the requirements are ready for validation, DefineIT's visual storyboard capability renders all the steps of the application into an understandable and easily reviewed narrative, so the vetting can be carried out quickly.

Once a project's requirements are defined and validated through DefineIT, they must be tracked and managed throughout the development process. Borland's family of management and testing tools integrates seamlessly with DefineIT, and it is ideally suited for directing every phase of an application's lifecycle.

Borland's mission is to impose a business process-like structure on application lifecycle of projects, bringing them into line with other goals and practices to maximize productivity and minimize waste. With its Open Application Lifecycle management (ALM) initiative, Borland creates a framework where tools work together flexibly to support any workflow style or process already in place and integrate seamlessly with a company's existing management solutions—either commercial or open-source.

Open ALM enables customers to streamline and improve existing processes by integrating with lifecycle tools and adding a management layer that aggregates information from all the components and presents a unified platform for reporting and asset tracking.

In fact, Borland offers solutions that support the four critical ALM processes involved in successful software delivery:project and portfolio management, requirements definition and management, change management, and lifecycle quality management (LQM). These tools work together and with third-party software to automate and improve software development, testing, and deployment for customer organizations.

Borland Tempo™ represents a complete portfolio and IT project management solution that tracks resources, assets, finances, demand, and technology resources. Caliber DefineIT helps smooth out every subsequent phase of the lifecycle by enabling easy and accurate requirements definition.CaliberRM™ provides intuitive and powerful requirements management functions, helping teams stay true to the initial vision and tracking changes throughout the process.

The newly announced Borland Gauntlet™ provides an automated system for code building and testing, defect detection, and development reporting, enabling teams to track, measure and improve software quality throughout the lifecycle, to meet milestones on schedule, and to stay accountable and transparent to stakeholders at all times.

In addition, Borland SilkCentral® Test Manager suite provides a full solution for planning, managing, executing, and reporting software test activities, with a Web-based interface for distributed workgroups.

Summary

With accurate requirements definition using Borland Caliber DefineIT, companies save time and money in the design phase, during development, and throughout the testing and QA processes. Borland's complete portfolio of flexible and powerful project management, testing, and deployment tools ensure that application development projects break free from traditional bottlenecks and delays, becoming streamlined and efficient, and ultimately driving growth and success for your business.

To learn more about Borland's requirements definition and management solutions, visit www.borland.com. ■

[4]Source: CapGemini, "Accelerate Your Oracle Application Implementations," October 2006