Borland®
THE OPEN ALM COMPANY

Harnessing Change

How to anticipate, accommodate, and leverage change to provide competitive advantage through software development

White Paper
February 2006

Contents

Executive Summary

To excel in today's business climate, organizations must continually deliver differentiated products and services that provide high business value. Even the most successful enterprises can experience failure because they unsuccessfully manage change in software development. With more demand than ever for IT to quickly and accurately achieve and support corporate objectives, IT organizations must cope with a continuous barrage of changes—changes in product requirements, technology, development processes, and deployment environments. As a result, IT can only successfully deliver software assets and applications through proactive management of change.

Yet market demand, corporate executives, and IT organizations are often at odds. Executive staff is frustrated because development projects fail to meet expectations, which affects business objectives. With equal frustration, IT team members point to unrealistic schedules, business and technology demands, complexity, and uncontrolled requirements changes as factors that impair development projects.

The Standish Group's CHAOS report shows that common causes of project failure are a lack of both skilled project management and executive support. These deficiencies reflect an inability throughout the organization to proactively plan, adapt, and manage changes in requirements, technology, and methodology. The temptation is to restrict and control change, thereby suppressing innovation. To maintain a competitive edge, businesses must end the blame game and learn to leverage change as a competitive opportunity.

Change always comes at a price, but change also offers opportunity to organizations prepared to handle it. Rather than expecting to freeze requirements and stifle changes, forward-looking IT organizations anticipate and accommodate change. They define software development processes, undertake sound change management practices, and provide tool support for the entire development organization to manage tasks and software configuration, as well as optimize workflows.

This paper discusses how organizations that develop software can take steps to manage change proactively in order to reach business objectives. It also describes how organizations can leverage change as a tool by applying the Borland® Software Delivery Optimization™ strategy—a combination of process and measurement services from Borland, effective project and requirements management, and software change and configuration management technologies that help turn change into advantage.

The Reality of Change, It Happens

The single, most constant factor in software development today is change. IT organizations and project teams must deal with rapidly changing business and product requirements, development methodologies, delivery technologies, and application architectures. The pressure on software development teams is compounded by internal demands to integrate more business applications and automate manual business processes, as well as external demands for more feature-rich and software-intensive products. These demands, coupled with increased constraints for compliance, failover, worldwide availability, performance, and security, impose additional stress and lead to increased software complexity.

Enterprises cannot avoid or ignore change. Instead, proactive IT organizations acknowledge change and the business opportunities that change offers to those prepared to exploit it efficiently. For software development organizations, "change control" shouldn't be about preventing change, but rather about managing change. Effective change management

"Extensive processes help everyone throughout the IT organization understand the implications of making a single change. As a result, Verizon Wireless is among elite web properties maintaining high service-level agreements.

The Borland tools coupled with our processes help me to do my job, so that I can focus on strategy instead of tactical issues." – Verizon Wireless

provides key stakeholders with the information that allows them to make optimal and timely business decisions, and to accept the right set of proposed changes in order to achieve both tactical project and strategic business objectives.

Effective change management is a balancing act driven by choosing the right projects, well-defined product requirements, accurate estimation and planning, and ongoing prioritization. The ability to respond to changes effectively can provide a competitive advantage over organizations that cannot accommodate change.

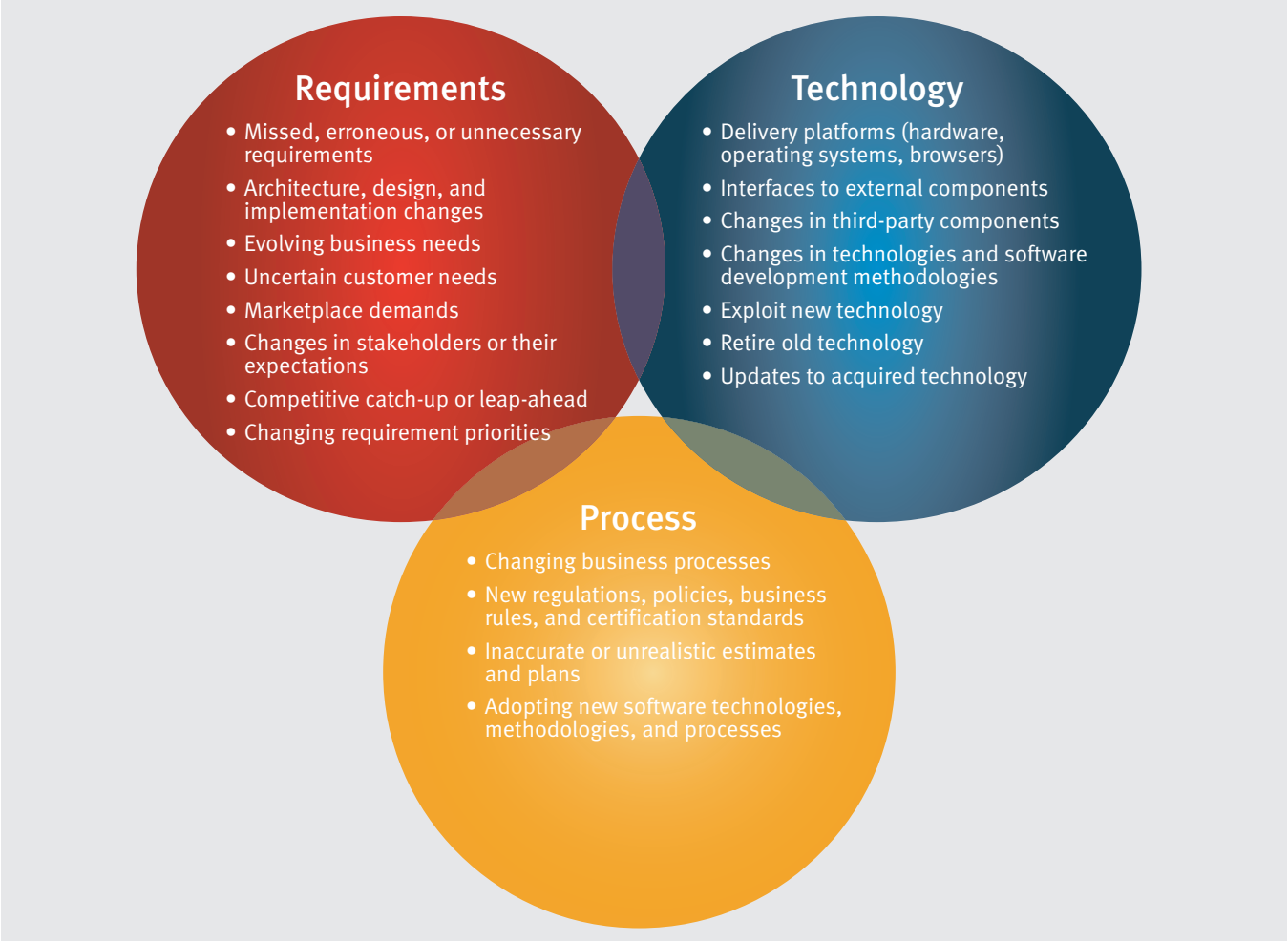
There are three elements to proactive change management:
Anticipate change (know the market, its business drivers, and likely sources of change)
Accommodate change (understand your team's capabilities and capacity to adapt to change)
Manage change (establish processes and tools to assess the impacts of change and to absorb changes into the organization)

Project teams that spend all of their time chasing changes never release products. Therefore, software development organizations must create an environment that embraces the reality of change and accepts it as an unavoidable and often value-added aspect of software development. This means putting structures in place to help development teams exploit the changes they encounter as opportunities to make the delivered product closer to what customers really need.

Sources of Change

Software organizations and project teams are bombarded with changes from many directions, including changes to product requirements, the technologies being applied, and the business and engineering processes being followed. The following figure identifies some of the changes an organization is likely to encounter in each of these categories.

Types and Sources of Change



Software organizations must anticipate and accomodate many types of change.

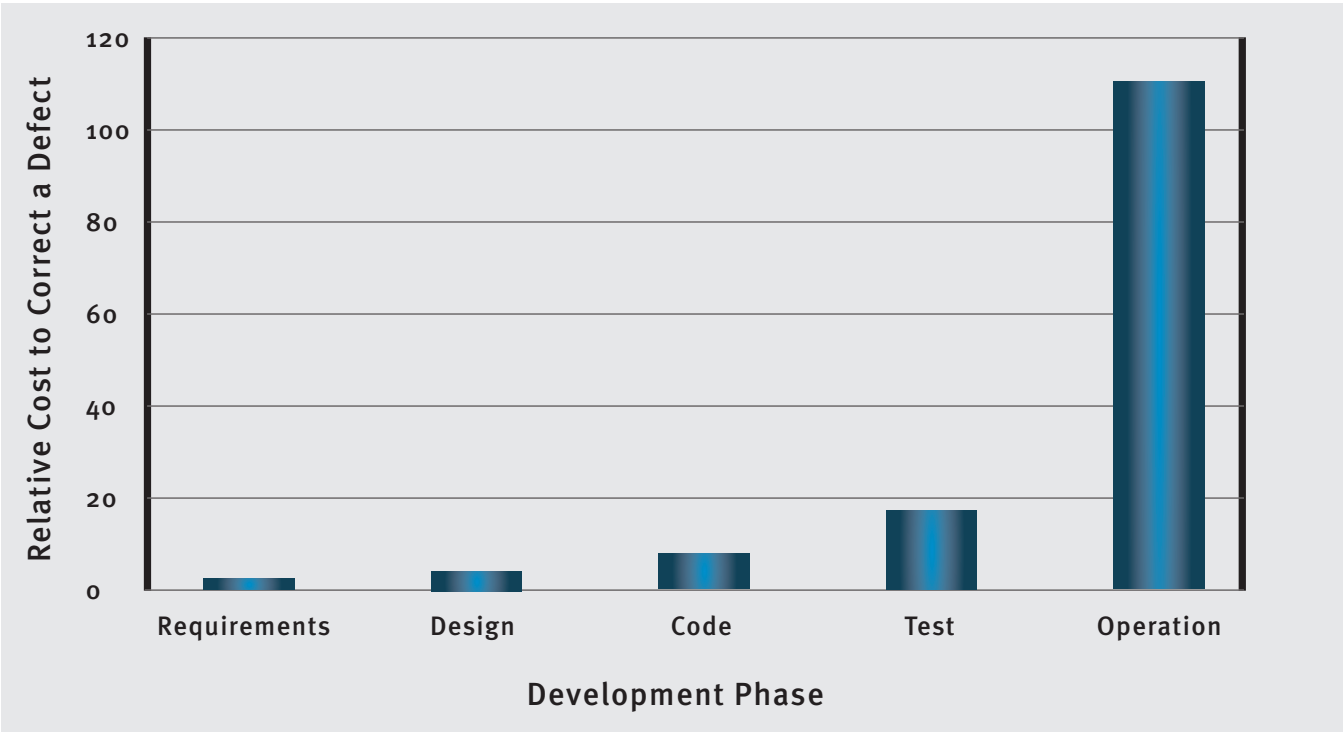
Consequences of Change

Handling changes effectively can provide organizations with the ability to quickly adapt projects to unanticipated needs and deliver superior business value. It also gives teams the ability to continually upgrade applications to take advantage of new technologies and the best development methodologies.

While change is often beneficial, it always has a price. For example, changes to requirements can cause a significant amount of rework. Software projects typically devote 30 to 50 percent or more of their total effort to rework. This siphons off staff to deal with constant changes and corrections, making those team members unavailable to develop new and innovative products. Additionally, the further along in the project lifecycle that changes are introduced, the more they cost. As the chart below illus-

trates, correcting an erroneous requirement after the product has been released can cost 100 times more than if that same error had been discovered during requirements development.[1]

Relative Cost to Correct a Requirement Defect



The later in the development lifecycle that a defect is identified, the more it costs to correct it.

Organizations should be aware of both the wide variety of adverse change impacts and how they reduce an organization's ability to meet business goals. For example, underestimating the impact of change can lead to poor business decisions when it comes to knowing which proposed changes to accept. Functionality changes that are implemented directly in the code, rather than flowing through requirements, architecture, and design, can lead to brittle infrastructures and undocumented spaghetti code that cannot easily accommodate changes in the future. The application can exhibit degraded performance as more and more functionality is added over time to a stressed architecture and inadequate design, and maintenance costs can soar.

Excessive change can cause significant schedule and budget overruns, and failing to anticipate the likelihood of change can lead to unrealistically low estimates for project effort, schedule, and cost. A constant stream of change requests obscures management visibility into project status and makes it difficult to estimate when a product can be released. Furthermore, poorly communicated and managed change requests inhibit effective teamwork as project team members work at cross purposes.

From the end user perspective, handling change ineffectively can lead to poor quality software that doesn't satisfy expectations. For this reason, appropriate changes must be incorporated during development so a product does not lack the requested functionality or contain obsolete, inadequate, and unnecessary functionality.

[1] Grady, Robert B. 1999. "An Economic Release Decision Model: Insights into Software Project Management." In *Proceedings of the Applications of Software Measurement Conference*, pp. 227–239. Orange Park, Fla.: Software Quality Engineering.

"Using a hybrid agile development process, our team delivered a series of seven different releases with added functionality in each release. The integrated Borland solution allowed us to keep track of our requirements, our configuration management and our defects as we launched a reliable, accessible system three weeks early and $65,000 under budget." – US Army

Finally, an organization that does not effectively adopt process changes misses opportunities to fully exploit contemporary methodologies for strategic advantage. The process of adopting new technologies is always more difficult for companies that are not set up to quickly climb the learning curve.

Symptoms of Inadequate Change Management

Several indicators suggest that an organization's change management practices are ineffective. One strong indication is that change requesters bypass the "official" process because they do not trust it to work quickly enough to make appropriate business decisions. The following highlight some additional symptoms of change problems.

Scope Ambiguity. Although some growth in requirements is to be expected on every project, scope creep is the uncontrolled growth of requirements with no end in sight. This usually indicates that a project's scope boundary is not well defined. Another clue is that proposed requirements and changes are in scope one week, out of scope the next, and back in scope the following week. In this case, it is nearly impossible to make appropriate decisions about change requests because it is not clear whether each change lies within the project scope or not.

Communication Shortcomings. Sometimes team members do not hear about a proposed change or a decision to accept or reject a specific change. If it is not clear who receives change requests, evaluates them, makes decisions, and communicates those decisions, the change management process is deficient. Without consistent communication, stakeholders who might disagree with the intent or implication of a change may overrule the decision. As a result, developers may implement changes that were never approved, and new and unexpected functionality becomes evident during testing.

Unanticipated Impacts. Sometimes changes are approved that are not technically feasible. Changes that take much longer than anticipated to implement suggest a lack of adequate change impact analysis. For example, two developers at an insurance company hastily estimated that it would take four weeks to add a customer-requested enhancement to an information system. The customer approved the estimate and the developers set to work. After two months, when the enhancement was only about half done, the customer lost patience and canceled the request, thereby discarding hundreds of hours of effort already invested in implementing the change.

Resistance to Change. When people resist change the business loses opportunities to exploit change for strategic benefit. A project must become more resistant to change as the planned release date approaches. However, the threat of change earlier in the development lifecycle should not inhibit an organization's ability to make decisions, innovate, or deliver.

Strategies for Dealing with Change

Through effective change management, organizations can incorporate appropriate changes into projects in a way that incurs minimum adverse impacts and provides maximum business value. For every project, dealing with change begins with choosing the right project and then a solid foundation of well-defined requirements. Processes and tools for project planning, estimation, and configuration management also play a critical role. The objective is to establish a culture within the enterprise that accepts the reality of change and has the tools and methods in place to accommodate it. The recommendations below can help any organization deal with the variety of changes it is likely to encounter.

1. Establish a Project Initiation Practice

- Set up processes and technologies to gather requests for IT services and investments. This will make it easy to submit project requests, enforce policies about how requests are handled, and collect information in support of the evaluation of requests.
- Put in place processes and technologies used to analyze the costs, benefits, and risks of proposed projects and other investments collectively, in the context of existing in-progress investments, so that the best decisions can be made on what to approve.

2. Adopt Requirements Developement and Management Best Practices

- Establish best practices for requirements elicitation, development, prioritization, and management to better manage the requirements of all projects. Multiple studies have shown that investing more time in requirements development can accelerate project schedules by reducing unprofitable rework. For example, a large insurance company discovered that increasing the fraction of project effort devoted to definition and requirements from 19 to 33.6 percent reduced its average total project effort by four percent. Similarly, several European companies found that devoting more of a project's schedule and effort to requirements activities resulted in more rapid product development:[2]

	Schedule Devoted to Requirements	Effort Devoted to Requirements
Slower Projects	9%	7%
Faster Projects	17%	14%

- Clearly define and document the scope of each project. The scope describes a specific subset of the ultimate product that will be implemented in a particular release.
- Work directly with stakeholders to capture and verify specific accurate requirements.
- Establish baselines of agreed-upon requirements and make changes against them.
- Use tools to store and communicate requirements and changes.
- Deploy requirements definition and management tools, such as Borland® Analyst,™ to capture accurate requirements, retain version histories of all incorporated changes, and allow teams to define the requirements allocated to each baseline. Also use the tools to define traceability links between requirements and other pieces of project information, which will facilitate effective change impact analysis.

3. Implement a Change Control System

- Define a process for submitting, evaluating, and prioritizing change requests.
- Adopt a configuration management tool to collect, organize, communicate, and track change requests. The Internet development group at Eastman Kodak Company found that a robust change control process and tool was invaluable to managing its huge backlog of change requests.[3] The team used it to handle requirements changes for systems under

[2] Blackburn, Joseph D., Gary D. Scudder, and Luk N. Van Wassenhove. 1996. "Improving Speed and Productivity of Software Development: A Global Survey of Software Developers." *IEEE Transactions on Software Engineering* 22(12):875-885.

development, problem reports and suggested enhancements in production systems, as well as updates to Web site content and architecture, and new project requests.
- Identify and empower the change decision makers, typically called the change (or configuration) control board (CCB).
- Analyze the likely impacts of changes, with the help of requirements traceability data, to accurately estimate the individual and aggregate cost of each.
- Ensure that change decisions align with project and organizational business objectives.
- Record and communicate the approved changes to team members and stakeholders.

4. Track All Changes to Assets
- Put in place a software configuration management (SCM) system to manage version control, branching, and labeling of all the assets of software development (source code, requirements documents, design documents, test plans, etc.).
- Establish policies to control who can change which assets at each phase of project development.
- Link all changes in assets to activities that caused the changes, so that the reasons for making changes are clear and auditable, for history and policy compliance.

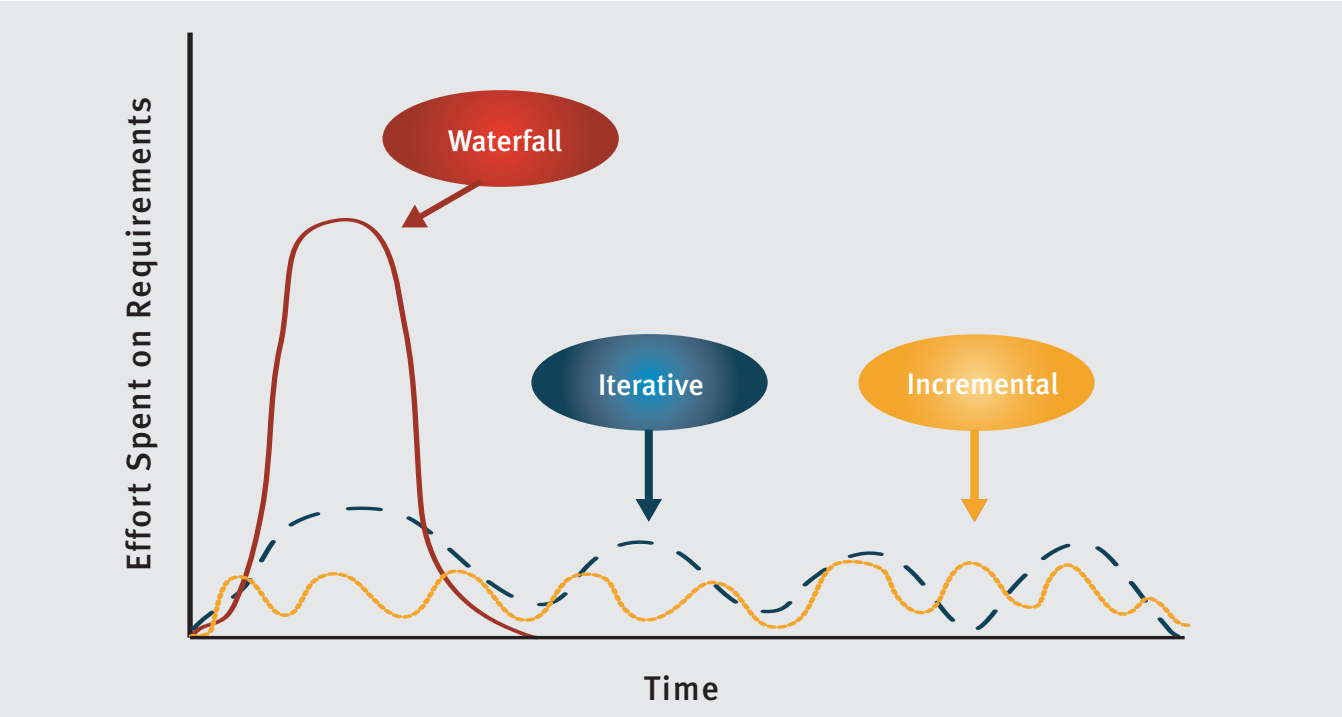
5. Adopt Effective Project Management Strategies
- Incorporate contingency buffers (management reserve) into all estimates so that excessive requirements growth or technology changes do not destroy project schedules. According to software expert Capers Jones, requirements can grow between one and three percent per month. This normal and expected growth can have a huge cumulative impact on a long-term project. As Change Management principals and processes are adopted project schedule predicability will increase making it s that these buffers can be reduced, but not eliminated.
- Adopt software development lifecycles, workflows, and development methodologies that are appropriate for the expected levels of change on a particular project. If requirements are uncertain and business processes are rapidly changing, use an incremental software development lifecycle. Requirements efforts can be distributed in different ways during the project schedule for waterfall, iterative, and incremental development lifecycles, as the figure below illustrates.4
- Establish a process improvement program. Perform root cause analysis to identify opportunities for better change management. Select specific process changes that could lead to better results. Define new processes, pilot them, and instill them into the organization's culture as routine practice.
- Train team members so they can take advantage of new processes, methodologies, and tools.

Combining People with Process and Technology

Successful change management is the result of a broad organizational effort that fundamentally shifts how an organization anticipates, plans, and accommodates change. This effort enables the organization to adapt to requirements, technology, and methodology changes in a planned and well organized approach. It is a shift that must be realized both at the software development team level—where change management is an essential part of project management across the entire application portfolio—and at the organizational level—where executives set expectations and constraints regarding opportunities, cost, schedules, quality, and the delivery of desired applications.

[3] Wiegers, Karl E. 1999. "Software Process Improvement in Web Time." *IEEE Software* 15(4):78-86.
[4] Wiegers, Karl E. 2003. *Software Requirements, 2nd Edition*. Redmond, Wash.: Microsoft Press.

Requirements Effort as a Function of Time



The distribution of requirements effort varies over time for projects that follow different development lifecycles.

To capitalize on the opportunities presented by a proactive change management strategy, businesses must be ready to make three fundamental, disciplined shifts.

First, change management must be considered from a broad, organizational perspective. Decisions occurring in the line of business and IT organizations ripple across sales, IT, and operations. Change management planning that encompasses a comprehensive view of the organization helps planners anticipate how to accommodate changes into the organization most effectively with the least disruption. This expansive view enables enterprises to balance the risks from change that could negatively affect the business against the benefits provided by superior innovation.

Second, enterprises must look to solutions and strategies that control change without unduly restricting the productivity and creativity that fuel advancement and growth. Many organizations look only at the goal of eliminating the negative possibilities related to change. Unnecessarily restrictive approaches can compromise an enterprise's ability to compete effectively. To capitalize on change as an opportunity that offers competitive advantage, organizations must identify change strategies and tool solutions that encourage proactive planning, positive risk taking for new gains, supported by sound change management workflows.

Third, enterprises must address change holistically. This includes examining how the skills of the teams in the organization, the processes that guide teams, and the products and infrastructure employed by the IT organization all contribute to effective change management. This intersection of people, process, and technology creates the foundation for proactive change management which advances the goals of the entire enterprise:

"The Borland tools are primarily oriented to specific roles in the development organization—architect, analyst, developer, tester—so for the managers who report to me and who also manage the process, Work is reduced. Our standards are being enforced not by people, but by the tools or through the tools." - Conformia

- **Process** provides the coordination and control to allow teams to work collaboratively.
- **Technology** enables teams to collaborate with higher productivity and quality, eliminating error-prone, inefficient manual activities.
- **People** provide the skills to move an organization forward.

As a strategic partner, Borland can help your organization advance its change management strategy using the right combination of people, process, and technology. Borland can assist you in developing a plan for change management through the following four phases that address change types, rates, processes, and control mechanisms:

* Consulting services, not packaged yet

Phase 1: Define goals

The first step on the way to improving change management is to define the specific objectives the organization wants to achieve. For example, are the goals strategic business opportunities or tactical performance improvements? Borland works closely with an enterprise's executive team to clarify and prioritize objectives and translate them into specific milestones against which the initiative's success can be measured. With these goals in mind, Borland assesses the organization's current change responsiveness

to surface potential improvement levers in process, technology, and people skills. This assessment may be focused on specific functional capabilities, such as requirements engineering, or applied to the complete software development lifecycle.

Some of the issues examined include:

Requirements	Change and Configuration Management	Project Management
• Customer involvement • Selection of requirements elicitation techniques • Requirements development process • Cross-functional roles and responsibilities in requirements engineering • Use of requirements baselines • Requirements development skills • Requirements management tools	• Change request process • Change approval process • Decision-making authority • Impact assessment • Traceability • Change management workflow • Process for ensuring change completeness • Change quality assurance • Asset management policies	• Project planning approach • Project request management • Use of contingency buffers • Estimation process and infrastructure • Project management process and infrastructure • Project manager roles and responsibilities

At the end of Phase 1, the enterprise will have clearly defined goals for the initiative that enable the organization to improve its change responsiveness.

Phase 2: Architect the approach

The second phase is focused on structuring and executing performance improvement initiatives. Borland organizes Phase 2 around several Action Planning Workshops that bring together members of the enterprise's development organization in cross-functional teams. Borland experts work with these teams to identify and prioritize the specific issues that first need to be addressed to rapidly improve change responsiveness. They then brainstorm potential solutions, evaluate the ideas and recommend actions. The Action Planning Workshops complement a measurement workshop in which Borland professionals, together with development teams, review and refine the measures and metrics used to monitor change responsiveness and the overall performance of the development organization. Effective use of these measures is critical to ensuring that the proposed actions result in the desired improvement and that the improvements are lasting. Where necessary, enterprise development teams are also trained in basic functional concepts, (e.g. the fundamentals of requirements elicitation) to help facilitate more effective discussions.

At the end of Phase 2, organizations have a prioritized action plan for validation and translation into enterprise rollout.

Phase 3: Develop and deploy the solution

This phase represents the bulk of the effort and cost associated with the initiative, since it involves implementing the process changes, supporting those changes with appropriate technology solutions, and building the appropriate skills in the development team to leverage the process and tools. In Phase 3, Borland's market leading products, targeted packaged services, and cutting-edge education offerings deliver accelerated time-to-value advantages. Often before being rolled out enterprise-wide, these ideas are validated in a pilot project to ensure the changes to process, technology, and people skills are effectively complementary.

Phase 4: Validate results

As a final step, Borland and the organization measure and compare the solution's impact against the goals established in Phase 1. Given the complex interactions between the changes to process, technology, and people skills, this phase often reveals the next cycle of improvements needed to continue to upgrade the organization's change management capabilities.

Leveraging Change to Achieve Success

Software development success is largely dependent upon the ability of an organization to ascertain and commit the necessary time, human and technology resources, as well as budget, for project completion. Once a project is underway, however, failure to accurately manage change can derail potential success and instead deliver unpredictable results, incomplete or inaccurate applications, and missed business opportunity.

By incorporating and proactively managing change, software development executives gain unexpected opportunities to increase effectiveness, encourage innovation and make better use of technology. Accepting, even embracing, change also helps software development managers ensure that delivered products will satisfy current customer needs.

Without the right information to effectively manage change, software development professionals continue to put project plans at risk. With the right information, these same individuals can turn change into a competitive advantage. Borland provides software development teams with powerful solutions that combine education, process improvement, and technology to manage both change and risk, allowing corporations to build a collaborative environment that optimizes the planning, execution, and management of software delivery across roles, teams, and technology platforms.

Borland is the leading vendor of Open Application Lifecycle Management (ALM) solutions - open to customers' processes, tools and platforms – providing the flexibility to manage, measure and improve the software delivery process.

Borland®
THE OPEN ALM COMPANY

www.borland.com