Borland®
THE OPEN ALM COMPANY

# Managing Change For Competitive Advantage

Effective Change Management for Software Delivery

White Paper
July 2007

## Contents

## Executive Summary

***"It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to change."***
***- Charles Darwin***

While this statement was made more than a century ago and pertained to a different subject, its key message – being able to adapt to change – is highly relevant to software development teams today. You could read it as, "It is not the strongest software delivery team that succeeds, nor the most intelligent, but the one most responsive to change."

What makes it so relevant today? Many software development teams assume that they can make a plan and execute it without needing to make adjustments mid-course – a misconception affecting nearly aspect of the development process. They may cling to this notion despite years of painful and expensive lessons learned from trying to execute on an original plan in an environment dominated by change.

While much has been written in recent years about the concept of managing change, few organizations today truly have an effective, repeatable, project-level change management process. And yet it is one of the most critical success factors for software development teams today. Those companies able to cope successfully with change are more agile on both the strategic and tactical levels – a significant competitive advantage.

This paper takes a fresh look at how you can solve many of the most common issues companies struggle with concerning change management. It is intended for the following two audiences:

- Software development or delivery organizations that recognize the need for change management and have already implemented process and tools to address it but think they could do better.
- Software development or delivery organizations that recognize the need for change management and desire to implement a solution to address it.

## The Challenges of Managing Change

Change can be a very daunting obstacle for IT organizations – particularly those engaged in software development.  Most see change as an enemy they must combat. And they define their weapon – change management – far too narrowly to be effective. Many development teams view change management as simple defect tracking – and in some cases, management of enhancement requests. But few really understand what successful change management involves.

As a result, only a minority of businesses today have found a way to turn change management challenges into a source of competitive advantage. When your company masters the art of change management, you can increase agility, respond more quickly to customer needs, get to market faster, and develop projects on time and within budget. In addition, you can ensure better product quality, increase customer satisfaction, and lower development costs.

Achieving these results requires that your organization view change management as an integrated approach to managing any change from any credible internal or external source. It also requires that you have integrated processes, supporting tools, and automation to drive the change management process.

But how do you move your company from its current state usually some combination of effectiveness and chaos – to a desired state of controlled flexibility that allows you to cope with and utilize the opportunities resulting from constant change?

## Partial Solutions Are Not Enough

Software development teams first need to understand what are they doing right today, where the gaps are relative to best practices, and what they need to do to improve. But this can be more difficult than you think, especially when you have narrowly defined change management. Consider the following companies – each representing common scenarios – and guess which one has the best approach to "fixing" their change management issues:

- **Company A:** In this case, the IT organization had what at first appeared to be a perplexing problem with managing the constant stream of defects, issues, and enhancement requests on their software delivery projects. They had a standard change tracking process used across their project portfolio, which they automated using an industry-leading change tracking and service management product and trained the entire organization to use. This company had clearly invested considerable human and financial resources to solve their change management issues, and the solution should yield significant, measurable results.
- **Company B:** This organization chose a different approach. They recognized that much of their pain in addressing the rapid rate of change was related to uncontrolled modifications to the file assets on their projects. As a result, they decided to implement tight controls on changes to file assets. They employed a straight-forward process whereby only one person at a time could change any file by checking out the file, making the changes, and checking the file back in. They automated this process using an open source but proven version tracking tool, which provided an audit trail of all changes to files, including information regarding who performed each change and when it was made. They even required users to enter comments describing the changes made to a file as they check in.
- **Company C:** After considerable study, Company C concluded that they needed to address governance by implementing organizational demand and portfolio management along with the same rigorous asset management approach implemented by Company B.

Which company do you think correctly identified the root cause of their change management issues and implemented a successful solution? Let's take a closer look:

- **Company A:** Despite their best efforts, they continued to experience significant cost and schedule overruns with little advance warning. Project managers continued to complain that they were unable to understand the impact of change when assessing actual performance against plan and revising plans. Project staff continued to assert that they could not effectively address work they had to perform in implementing accepted changes.
- **Company B:** Although they allayed some of the issues facing Company A, their approach created another set of issues that continued to plague the organization, as management did not have a clear picture as to why files were changing.
- **Company C:** Despite investing heavily in tooling and implementation, they found that they were still highly vulnerable to the effects of uncontrolled change across their software development projects.

## Identifying the Root Cause

To understand the root cause of these failures, you need to step back and examine the overall approach that these companies used to manage software delivery. In all three cases, the three critical processes involved in change management – project and portfolio management, asset management, and change management – were operating independently rather than as part of a single, closed-loop process. These disconnects played out differently in the three company scenarios:

- **Company A:** This company had a good handle on change management on a project by project basis, but it lacked the connection to the assets being changed and maybe more importantly to the project and portfolio management processes for managing change and task execution.
- **Company B:** In this case, the company managed project assets very well, but from a management perspective, it did not have the visibility needed to manage both individual projects and the overall portfolio.
- **Company C:** This company had effective, high-level change management at the project and portfolio level, as well as low-level change management at the asset level, but there was nothing connecting them at a project level.

As these examples suggest, the root of the problem is really twofold. First, a considerable amount of the work done on projects, which often affects project assets, has little or no connection to the processes focused on maintaining project control and visibility. Project management is about planning, managing, and overseeing the work required to complete the project. Change management is about managing the impact of change to project assets and thus outcomes. These two processes are rarely linked so that the work (i.e., the activities) performed as a result of a change request is visible and controlled as part of the project plan.

And second, changes to assets need to be integrally linked to the activities and requests driving the change. Otherwise, once again, there is a loss of control and visibility from an overall management perspective.

## Solving the Problem by Integrating Project, Change, and Asset Management

The only way you can effectively solve this problem is to integrate asset, change, and project and portfolio management processes so that all project work is managed in an integrated fashion. This includes work that was part of the original plan and work that is the result of re-planning due to change. This integrated approach represents a significant deviation from the traditional view of project, change, and asset management.

Key indicators that your organization has successfully made the transition to comprehensive, integrated change management include the following:

- All work performed on project assets is linked with planned activities, which are in turn linked to tasks in the project work breakdown structure.
- Change management is viewed internally as more than just the identification, assessment, and resolution of defects, issues, and enhancement requests. Rather, change management is viewed as a process for managing all change to the project – including its assets, and its outcomes – regardless of the source of or reason for the requested change.

In this new paradigm, all requested changes are triaged and assessed as part of an iterative re-planning process that accounts for the impact of each requested change on the overall work plan. Once accepted, the tasks and activities required to implement the change are planned in the context of all other planned and ongoing work. At the same time, the project plan is updated accordingly, and the progress made on change requests is measured as part of the regular assessments of progress against plan.

## Integrating Change Management with Project and Portfolio Management

The notion of tightly integrating project and portfolio management processes with change management processes is certainly not a new one. It has been written about in the context of standard project management methods and software delivery processes across a variety of books and industry press. What follows is a summary of some key concepts and best practices that will help you integrate these processes.

### Implement Real Project Planning – A Continuous and Iterative Process

Waterfall-style project management rarely works in modern-day business environments where teams are shrinking, timelines are shortening, complexity is increasing, and change is ubiquitous. To move toward an integrated change management approach, you need to create a culture within your organization – and especially your software development team – that embraces change.  At the same time, make a commitment to invest in change management automation solutions – such as project planning and software change & configuration management tools – that support and enable them. The automation, best practices, analytical functions, and visibility provided by change management software will help drive the adoption of a more agile software development approach.

### Recognize that Change Management Is More than Tracking Defects and Issues

As stated previously, change management should be viewed as an integrated approach to managing any change from any credible internal or external source. So if your company still views change management as merely defect tracking – possibly combined with enhancement request tracking – you need to re-educate your team so that they understand that the need to manage the entire gamut of change. Equally important, you need to give your team a process and supporting tool automation that makes this possible.

### Assume That Not All Change Requests Are Requirements

A change request can come from almost any source and in various forms, such as email, phone calls, and handwritten notes to a passing conversation in the hallway. The information provided may be incomplete, inaccurate, or a duplicate, as well as ambiguous. In contrast, a requirement should be complete, accurate, unique, unambiguous, testable, and managed for a start. That being said, change requests many times drive the creation, modification, and deletion (yes it can happen…theoretically) of requirements. And in almost all instances, change requests should be directly linked to one or more requirements.

### Manage the Extremes of Requirements Management

Software delivery organizations know that requirements – including business and functional requirements – are a major source of change. Business requirements tend to dramatically impact strategic plans, while functional requirements tend to impact day-to-day tactical planning. A well-integrated change management approach to requirements management should create sufficient organizational flexibility to cope with both extremes.

### Integrate Your Processes

As we saw in the earlier example, most organizations think of change management as separate from their project management process, which leads to loss of control and visibility. Specifically:

- The tasks and activities being performed to implement changes are not included in the work breakdown structure.
- Project managers have no accurate way to assess the impact of new, change-related work on remaining work or deadlines to meet the project objectives and delivery dates.
- Status on the completion of change management activities is not reported to management effectively.

In addition, they may also see change management as unrelated to requirements management.  This is another incorrect assumption that leads to scope creep, significant project re-work, and an ever-widening gap between the business's expectations and IT delivery.

Effective change management approaches integrate these processes, along with the following:

- Lifecycle quality management processes and tools
- Defect tracking processes (which should be part of the overall CM solution)
- Test case, test script, and test log data, which should be logged and linked automatically because every minute wasted in the process of trying to recreate, fix, and re-verify bugs is a minute lost on other tasks that keep the project moving forward.

## Integrate Portfolio Demand with Project Change Requests

Change requests can come in from all levels of your organization and  affect project outcomes. What's crucial is that you clearly define how and where all this demand for change will be collected, organized, prioritized, completed, and managed. Managing this change at only one level - such as the portfolio/organization level or the project level - is not good enough.  The reality is that requests and demands for project changes come from many sources, both internal and external.  Some affect the organization and some are specific to a project.  Some come from within a project and some come from outside a project. To effectively deal with this change, you need two collection points, or funnels, that form task queues:

- **Portfolio Demand Funnel** – The portfolio funnel addresses demand coming from inside or outside your organization that may impact your portfolio (i.e., across projects) or a single project. Demand management or enterprise change management often addresses this collection point. Your change management processes must allow you to quickly assess, prioritize, and trickle changes down to specific project funnels where action can be taken. At the same time, it is important that you maintain a link to the original request for status and reporting purposes.
- **Project Change Request Funnel** – This collection point is for changes that come from within a project or are trickled down from your project and portfolio management process because they relate to a specific project.  The project manager handles these change requests.  If a request at this level is deemed to have a wider impact (i.e., beyond a single project), it must be promoted to the portfolio-level funnel.

By clearly establishing these collection points and the paths that link them, your organization can support more agile, responsive change management - and ultimately achieve better visibility and control.

Organizations that make the link between these processes – demand management and change request management – often forget that the link must go both ways. They often flow information from the portfolio level down to the project level, but neglect to pass valuable project status and metrics data back to the portfolio level.  There are actually several useful points of interaction that are initiated from the project level, including the following:

- Activities added or changed in the project plan need to be pushed to the portfolio level project management (PMO) process.
- Resource assignments at the project level often need to interface with portfolio-level resource management processes.
- Work estimations and actual progress updates should be sent to the portfolio level process on a regular basis.
- Many metrics – including defect rates, state aging, completion percentage, and code churn – are very helpful in managing a portfolio of projects.

### Leverage Automation

A surprising number of companies today still rely on manual processes to manage change. For example, you may be using spreadsheets for tracking change requests and assignments. But the best practices discussed previously require a high degree of automation in order to be truly effective and efficient, in part because modern software development projects tend to be large and complex. The sheer number of individual changes during the software development lifecycle, the increasingly distributed nature of stakeholders, and the need to ensure regulatory compliance mean that fewer and fewer projects can be adequately managed using manual processes. In fact, doing so almost certainly puts your company at a competitive disadvantage. In order to avoid stagnation, software delivery organizations must rely on automation as a key to process enablement, not just focus on process enforcement.

## Manage Activities, Not Just Assets

The traditional view of project management is that activity, change, and asset management are three separate processes that should be addressed independently. But in reality, no real change can occur in a project without altering one or more of the project assets as part of some task or activity. They are entirely interdependent. As a result, activities, changes, and assets must be treated as separate and distinct but inextricably linked as part of a comprehensive, integrated activity and asset management (AAM) process – often referred to as software configuration and change management (SCCM). The following are concepts and best practices that will help you successfully implement such a process.

### Recognize that Change Requests Are Not Activities

One of the most common flaws in most change management processes is the management of change requests as if they were activities or tasks. In reality, they are vastly different. A change request simply documents the required change. The activities needed to actually carry out a change request could take one person an hour to complete or fifty people working hundreds of hours. For this reason, you need a way to manage activities resulting from change requests that allows you to estimate, prioritize, and manage the work associated with a diverse set of activities.

As a best practice, change requests should be broken into a set of activities according to the following guidelines:

- Each activity should be assigned to only one person.
- The different activities required to execute on a change request may require different types of work performed by people in various roles, such as the following:
  - Requirements
  - Design
  - Coding
  - Testing
  - Documentation
  - Project Management
- Each activity should be sized in such a way that an individual can complete at least one activity per week.
- An individual may be assigned one or more activities to complete a change request, if needed.
- An activity should be linked back to the parent change request for reporting and measurement purposes.

## Link Activities to the Project Plan

As you look at how activities have been defined above, notice that they resemble the tasks that comprise a typical project. As a best practice for managing change, you should link the activities that stem from change requests to tasks in the project plan. Doing so will ensure that all work – both planned work and re-work – is managed as part of the overall project plan.

Creating and maintaining these linkages manually can be a very tedious task – which is one reason why many organizations automate this process. Manually managing these links, or relationships, can hinder your organization's ability to respond to change. This is also true with regards to management of all traceability relationships within a project's activities, tasks, and assets. As a best practice, you should manage these linkages using automated tool support. Ideally, you should associate change requests with tasks and/or requirements so that you have a complete, contextual overview of all changes impacting a project. This overview is immensely useful for anyone who has to deal with changes to any of the impacted assets. Think of these associations as providing the "context needed for problem solving." Armed with this information, your development team is empowered to better solve problems because they can analyze the contextual relationships within which these activities and assets interoperate. This context allows information workers to make more informed decisions than those equipped with just lists.

## Integrate Activity and Asset Management

Integrated approaches to SCCM take asset and activity management to the next level. They give development teams complete visibility into and control over activities, assets, and the relationships between the two – the ultimate goal. Managing activities and assets as one cohesive unit is essential to overall success because it is what gives you real-time visibility into the status of development projects. In addition, your organization benefits in the following ways:

- Development, QA practitioners, and project leads gain reliable control of the digital assets used in the development process and can make them available to all members of your team – complete with a full history of changes so that work is repeatable and auditable.
- Development, QA practitioners, and project leads have greater control of the project activities – especially if the integrated SCCM solution being used can be customized to track important information using the workflow and forms that your development team is already used to.
- Managers benefit because they can manage assets and activities in a single system, which ideally should support distributed development with fully automated administration. At all times, there will always be one version of corporate truth, no matter what the project involves or where development teams are located.
- Common SCCM approaches result in simplified, unified forms and workflow for all team members. This reduces the need to train them on different tools and makes it easier to achieve corporate standards.

For more detail on how Borland can help you with activity and asset management, please see the Borland white paper entitled, "Integrated Activity and Asset Management: Increasing Visibility and Control through a Unified Software Configuration and Change Management Solution."

## Implementing Best Practices as Part of a Comprehensive Change Management Framework

The concepts and best practices introduced thus far are extremely important to the success of any software development organization. If you apply them consistently, you will significantly improve your organization's ability to manage change. But to leverage your approach to change management for even greater competitive advantage, you should apply them as part of a comprehensive change management framework. In addition to supporting and integrating the processes described previously, this framework should optimize how you do the following:

- Plan for change
- Manage requests
- Manage activities
- Manage assets
- Manage releases

The following sections summarize some of the key best practices and guidelines that can help you optimize your change management framework.

### Optimize Planning

The old adage, "Failing to plan is planning to fail" is true for software development organizations that need to respond to change.  The following best practices can help you plan more effectively for change:

- **Identify stakeholders:** Start by clearly defining the roles and individuals in your organization that have a stake in the change management process.
- **Specify the change management requirements:** Identify all sources of demand so that you can control the flow and consistency of change requests. It's also beneficial to identify and define the types of change requests you receive, as this further prepares your organization for change.
- **Design a change management process:** Clearly and simply defining change request data, forms, and workflow – an essential step to ensuring the usefulness of request information for developers, as well as the timeliness of your software development team's response. When you define reports and metrics relating to aging reports, defect rates, and resource assignments, you gain useful insight into how your change management process is working and where improvements can be made.
- **Create a change control board:** Change control boards act as the gate through which change requests are reviewed and channeled in the right direction within your organization so that they can be addressed or tabled in a timely manner. They can be immensely useful in assuring that changes are prioritized and then executed upon using a predefined and well understood process. Change control boards serve a dual purpose in that they not only prioritize and channel changes in the right direction, but also highlight any deficiencies in existing change processes, thus acting as one of the best means of identifying opportunities to improve existing change control processes.
- **Implement the automation framework:** To make your comprehensive change management solution truly effective, invest in software-based tools that automate the key processes involved, wherever possible.

### Optimize Request Management

Many of the change requests that come through your project change request funnel (i.e., your request management process) are not related to strategic demands or business requirements. Rather, they are low-level change requests tied to specific projects. Typical examples would include defects, changes in architectural constraints, or simple user interface changes. Make no mistake, these are as important to manage as the high-level requests. And if they are left unmanaged, they will throw your day-to-day tactical planning into chaos that will quickly be reflected in your high-level project status reports.

To avoid the chaos resulting from a barrage of low-level change requests, implement the following guidelines for managing requests:

- **Submit:** Start by defining a simple and consistent submission process for change requests.
- **Notify:** Notify appropriate individuals as requests are submitted so they can take action swiftly.
- **Triage:** All requests should be accepted, rejected, or deferred quickly and efficiently.
- **Review:** Requests should be reviewed by the change control board and prioritized based on some combination of metrics relating to impact, risk, complexity, and business value.

### Optimize Activity Management

Managing activities associated with change is critical to your organization's ability to execute. Implementing the following mandatory steps in activity management can help you optimize this process:

- **Analyze:** Once a request has been deemed a priority by your change control board, the lead or project manager should create a set of activities and estimate the overall effort.
- **Assign:** Each activity needs to be assigned to a resource that has the availability and skills to fulfill the request.
- **Execute:** Once activities are assigned, owners should be notified and review, accept, and complete the work.
- **Monitor:** During the life of a change request, its status should be tracked, measured, and reported on.
- **Verify:** All work completed must go through validation and verification before closing the change request.
- **Close:** Mark change requests as closed when all work has been verified and delivered.

### Optimize Asset Management

Project assets include the requirements, design, code, test, and documentation files used by the software team. They need to be controlled and managed very carefully. The following are critical best practices for optimizing asset management:

- **Store:** Determine an efficient way to store and provide easy access to project assets. Centralized storage of assets provides many benefits to your organization, especially as it grows. You can also consider using other approaches, such as federated and distributed repositories.
- **Version:** For effective enterprise asset management, you need a way to check in, check out, compare, view histories, and trace and link versions.
- **Baseline:** It is critical that you have a way to snapshot and label sets of files to configure, promote, test, and release code and other assets.
- **Branch:** Parallel development, patch management, and customer/product variants are all practices that drive the need for powerful and elegant branching. What gets branched must be merged, preferably with visual differencing.
- **Secure:** Deploy flexible and reliable security to protect the business-critical assets related to software development.

- **Unify:** All asset types, together with their associated activities, should be stored using a unified approach that makes it simple to find, access, modify, and reuse. If we assume that re-use is an organizational objective, then unifying your assets and activities will both promote re-use and make re-use more feasible for developers.
- **Traceability:** Use software to automatically create and manage links from assets to other assets or work items. These links allow you to perform real-time impact analysis and send change notifications.

### Optimize Release Management

Releasing software for production use is the ultimate purpose of software development and delivery. And as a result, having an effective release management process is an integral component of any comprehensive change management solution. What follows is a brief list of the most important aspects of release management that should be a part of a comprehensive change management framework:

- **Build:** Your software development team should have a way to consistently and frequently generate builds and assess the state of those builds. Automation, parallelism, distribution, and dependency analysis are key aspects of an effective build process.
- **Promote:** Developers should have a seamless, secure way to promote builds from development to QA and on to production without any opportunity for error.
- **Deploy:** Enable developers to physically copy or move files to the right servers and directories at the appropriate time.
- **Feedback:** When production applications go down, linking production components back to the right versions of source files is critical for fault finding and reducing down-time.

## Integrate Across the Life Cycle

How does change management fit into the rest of application lifecycle management (ALM)? The following sections summarize how the change management best practices described previously tie into the key elements of ALM, including the following:

- Project and portfolio management
- Requirements definition and management
- Lifecycle quality management

### Project and Portfolio Management (PPM)

As discussed extensively in this paper, change management must integrate the way in which change is controlled and reported on from the portfolio level down to the project level.

### Requirements Definition and Management (RDM)

Change affects all artifacts of the application lifecycle, including the requirements. Requirements represent the needs and desires of the business and are critical for project success. In fact, the one set of project assets most impacted by change may well be the requirements. Furthermore, you need a way to manage the different approaches required to cope with high level changes (i.e., business requirements) versus low-level changes (i.e., functional requirements), which typically are identified while developers are actually working on a given project. It is absolutely critical that both types of changes to requirements are tracked and controlled. In addition, you need to know how changes impact requirements through traceability tools, which guide teams to make better decisions and help them communicate changes quickly to other team members affected by a change.

### Lifecycle Quality Management (LQM)

Test artifacts are also affected by changes that typically flow down from changing requirements. The most prominent role of LQM in the area of change management is, however, the discovery and submission of defects. It is very important to capture defects efficiently and with sufficient detail so that your engineers can understand how the error occurs and how to fix it.

## For More Information

This paper has discussed many of the ways that your organization can improve its ability to respond to change. For more information about how Borland can help you integrate your asset, change, and activity management processes for competitive advantage, please visit Borland at www.borland.com to request that a sales associate contact you.

Borland®
THE OPEN ALM COMPANY

www.borland.com