MICRO FOCUS | Borland

WHITEPAPER

IMPROVING BUSINESS AGILITY

How stakeholders and developers can talk and think the same

17092015

Copyright© 2015 Micro Focus. All Rights Reserved. Portions Copyright © 1994-2009 Borland Software Corporation (a Micro Focus company).

INTRODUCTION: GIVING STAKEHOLDERS INSIGHT INTO THE DEVELOPERS' WORLD

Agile development has been a phenomenal success when it comes to execution: development organizations can evaluate and embrace change faster, and deliver business value more often. But the adoption of Agile practices has developed or exposed some interesting symptoms such as the struggle to communicate how business needs are interpreted and represented by development; a lack of insight into the development process; no evidence for why key decisions were initially made; and a disconnect between business and development teams.

A key challenge is that business and technical teams often appear to talk different languages when collaborating on Agile projects, rather than sharing a common objective and understanding of the business goal.

> *People know two languages: their native language and gibberish.*
>
> *Maribel C. Pagan*

This White Paper examines how organizations can align their business objectives more closely with the software development process and ensure all stakeholders involved in the project collaborate effectively to capture business needs, manage change and track progress. All three are essential if organizations are to react quickly to shifts in business priorities and improve business agility.

NEW TEAMWORK CHALLENGES

Agile practices have typically been driven through bottom-up adoption. Individual development teams have embraced practices and built solutions that make sense for the organization and its technologies, as they see it, via user stories. Requirements, on the other hand, approach the issue from the other direction, being written to satisfy specific business needs.

Because of this, all too often the execution – the 'what is being built' – will be disconnected from the original request. For example, physical card-based implementations of Agile do not generate a history of why decisions are made, nor do they serve as documentation. Instead, the original cards are torn up and thrown away. The system that emerges is what it is; changes and why they were made are not captured or tracked.

And although some web-based systems track history on a card, few have a view of the project history – when cards were introduced, changed and deleted, and in what order.

Coping with different needs

Questions like "Should we use requirements or stories?" imply that the two concepts conflict – that teams should pick a single way to plan and develop software. The reality is that different methods are needed to meet different needs.

Business customers tend to think and plan in terms of business needs, while the delivery team thinks and plans in terms of development needs. The two units of measure rarely align. The business needs are high-level goals that are increasingly refined until they are fully-formed requirements. The development team then takes those requirements and breaks them down into actionable units of work – stories and tasks. Connecting the two, and deriving a business status from development progress, is often challenging – because the ongoing work of the development team is difficult to translate into a business context.

Mark Kulak, Director of Development for Borland at Micro Focus, explains the problem this way:

"We see this in the Agile world during demos, when a sponsor or core customer is watching the software in action. During the demo they'll see a small piece of functionality that doesn't clearly tie back to the original requests, or isn't complete. The customer ends up doing one of two things: either trying to connect the dots in their own mind, or asking questions no one can answer, such as: "When will the recalculate feature be available?". You know you're in trouble when you ask the question and get an answer that includes terms like 'front end', 'JS-Unit' or 'PL/SQL' that are nothing to do with the original business need."

> *The customer ends up trying to connect the dots in their own mind, or asking questions no one can answer.*

Copyright© 2015 Micro Focus. All Rights Reserved. Portions Copyright © 1994-2009 Borland Software Corporation (a Micro Focus company).

CONNECTING LEFT SIDE AND RIGHT SIDE

Connecting business needs and delivery goes a long way to solving the disconnect between business and delivery. The answer is to add versioning, traceability and baselining across all assets, from business needs to stories and tasks, whether already actioned or sitting on a backlog (see panel). This means that the organization can not only see the business and delivery relationship, it can understand how different needs have evolved over time, and answer one all-important question: "How did we get here from there?".

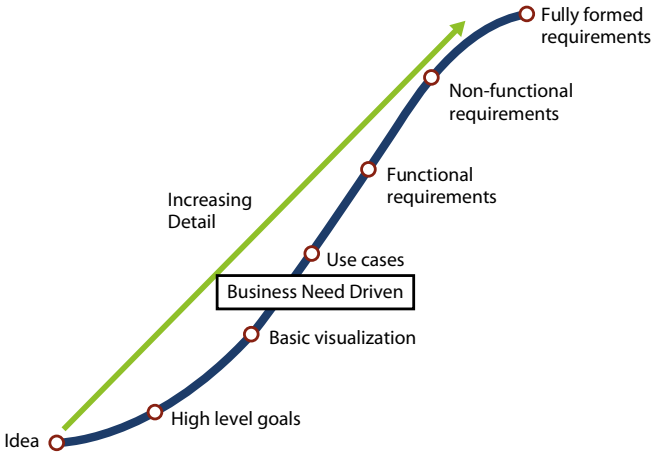

Figure 1: Defining business needs: 'requirements'

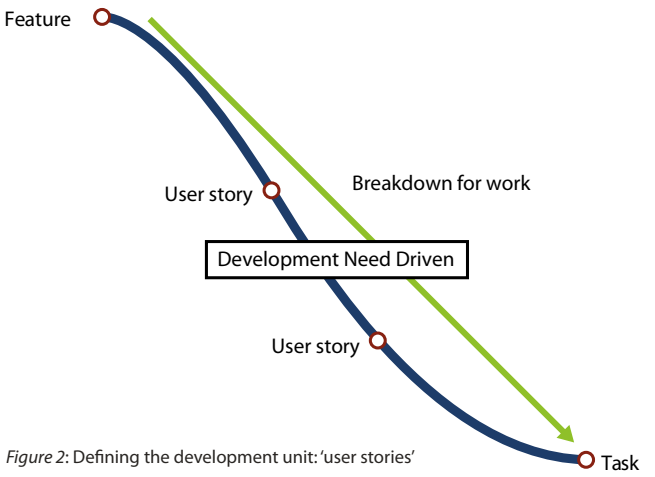

Figure 2: Defining the development unit: 'user stories'

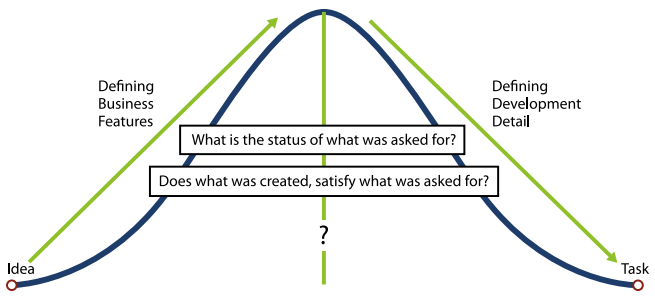

Figure 3: The requirements bell curve

Making the connection between business and delivery gives you an Agile requirements bell curve as in Figure 3. The left-hand side of the software project (Figure 1) is the 'what' to build, while the right-hand side (Figure 2) represents development tasks and technologies (which the Agile movement has historically emphasized, or often, over-emphasized). An integrated approach includes both sides – and connects them.

When business users and developers both see the same systems, your technical teams can recognize opportunities they might not have otherwise seen, such as potential shortcuts or connections. Crucially, by breaking the work down in business terms, the stories and tasks also make sense to the business, which enables informed decisions about priorities and scope. This adaptive approach solves all of the problems discussed earlier, preventing the team from making a radical change from the baseline unless overseen by business users.

Avoiding 'hidden processes'

During the development lifecycle, a process may be completed accurately, but not consistently. For example, a development team might work outside of their Agile tool – in, say, a word processor – and simply update the tool with the results. Organizations with real business agility avoid this lack of visibility with three crucial components of requirements management: versioning, traceability and baselining.

Versioning – it's standard practice to maintain different versions of code, but when it comes to other artifacts like requirements and tests, there's often no history of change. Each change should be versioned, along with who made the change and appropriate comments.

Traceability – this can apply to development artifacts (code, designs, unit tests, user interfaces, deployment processes), which should trace back to the business need they address. However, traceability is also a way of understanding which requirements have already been delivered (and how successfully in relation to the artifacts deployed), how much is currently under development and what is still at the planning stage.

In all this, what matters is not just the story, but the business capability the story will enable. A single item may enable multiple business needs or have several 'owners' with complementary, though differing, needs.

Baselining – the original vision for the project and any subsequent changes or iterations should be stored so that the version implemented can be compared to that original vision, and how it was arrived at, stage by stage. Having a baseline allows the team to understand the downstream impacts of change; while business users can 're-wind' the current state of a plan and its related delivery assets to any point in time during the plan duration to understand the impact of change.

Copyright© 2015 Micro Focus. All Rights Reserved. Portions Copyright © 1994-2009 Borland Software Corporation (a Micro Focus company).

THE THREE ESSENTIALS OF BUSINESS AGILITY

Capturing business needs

Beginning at the bottom left of the bell curve in Figure 2 we have the idea: the point from which everything – literally – develops. Unless ideas around business needs are accurately captured, shared and turned into requirements, the end result of a project won't meet stakeholders' needs and rework will be inevitable.

It follows that requirements must be captured and organized as flexibly as possible so that all stakeholders find it easy to collaborate and share essential business information. Results should not be restricted to any one file format – since ideas can come from anywhere, there must be the ability to organize and share pictures, documents, reports or even quick scribbled ideas in the most expedient way.

Effectively manage changes to business needs and priorities

Requirements inevitably evolve as the needs and demands of stakeholders change to support business demands and expectations. While Agile development teams are very responsive in supporting these changes, the challenge of capturing and communicating how they impact Agile backlogs and delivery over time is still a time-intensive and often neglected task. It needn't be. Agility means being able to change plans between iterations and still have everyone understand the impact of these changes, and how the product is evolving throughout the lifecycle.

Choosing the right software to manage change helps greatly: the ideal tool will tie the original business need to the right requirement and to the right build. This effectively combines requirements management with change management. Without these two elements, your evolution of work will be uncontrolled and less well managed – with the development team building one piece at a time that may, or may not, track back to the original goals of the project.

Tracking progress

Here we hit the apex of the bell curve: the point at which business users have defined their goals and now need to see that what was requested is being built. The transparency of linking stories with requirements makes this possible when using the right tool. In effect, the business user (or any non-technical stakeholder) is able to 'see' into the developer's world, through information captured directly from the development team's Agile tools of choice. This offers immediate and continuous tracking of the evolution and changes made to ideas while they are under development – a true form of business transparency that accelerates progress.

THE FINAL ANALYSIS

Keeping Agile projects tethered to business goals is a challenge. The solution, however, is exactly that – tethering business and development teams in a loose-tight coupling. This sets the development team free to build the right solutions, while the business team ensures they are being built through a responsible governance framework. Requirements – versioned, traced and baselined – offer the insight into the why and when of business decisions.

With stakeholders increasingly spread across the organization and outside of it, and business priorities and expectations continually shifting, the ability to bring business and delivery teams together in this way is the key to reacting faster and improving business agility.

Find out more about connecting business and development teams at www.borland.com/businessagility

Copyright© 2015 Micro Focus. All Rights Reserved. Portions Copyright © 1994-2009 Borland Software Corporation (a Micro Focus company).