THANKS TO CLOUD COMPUTING, mobile devices and social-media-like features, teams can collaborate from anywhere on the planet. This issue will look at how Agile ALM offers businesses techniques to make that collaboration possible. It also faces down SCM challenges and examines the pantheon of ALM tools and processes.
DEVELOPING COLLABORATION, AUTOMATION STRATEGIES THROUGHOUT THE SOFTWARE LIFECYCLE

Enterprise software development and management can involve coordination of hundreds, maybe even thousands, of people and processes. Is it any wonder communication and collaboration is so difficult? According to a recent Gartner report, “ALM improves collaboration among clients, development organizations and project management, allowing enterprises’ Agile concepts to be successfully propagated.”

Thanks to trends such as cloud computing, mobile devices and features adopted from social media, we live in a world where collaboration from anywhere is possible. We can now connect with our customers and each other. Getting teams out of silos and in close touch with business, project and operations management, and most important, users and customers can improve business agility.

ALM is the glue that holds the various software development practices together. By integrating the data and the voices of the stakeholders into a global repository, Agile ALM provides transparency and modern-day communication techniques aimed at creating teams without physical boundaries.

In our feature story, “Realize Full Potential of Agile ALM With Customer Collaboration,” we learn how requirements management has progressed from a cumbersome review process of lengthy documentation to one which allows for continual collaboration with all stakeholders throughout the lifecycle.

This issue also addresses the pain points in communication between people and systems. “Updating Tools, Processes Key to Overcoming SCM Challenges In ALM” covers software configuration management strategies, providing advice to senior managers on automation, cross-group coordination and maintaining consistent procedures to reap the full benefits of the latest SCM tools. Our final article, “Piece Together a Distributed World With ALM Communication Methods,” describes collaborative tools for geographically dispersed teams.

What collaborative features is your team using in ALM? Let me know.
A strong customer collaboration strategy can turn your high-potential ALM team into a high-value delivery team. These Agile ALM collaboration tactics can help managers work with their customers and stakeholders to define solid requirements and to be responsive to change. **By Yvette Francino**

Thanks to collaborative features in ALM tools, software teams can now connect with customers as well as communicate more effectively internally. Let’s take a look at how requirements management in Agile ALM environments is allowing for strong collaboration throughout the lifecycle.

In traditional Waterfall development environments, requirements are documented in lengthy tomes that must be complete and approved before a line of code is written. Just when the team thinks the requirements have all been approved, a lengthy change is suggested, requiring a re-review from everyone. And despite the attempt at freezing requirements once the document has finally been approved, changes are invariably needed throughout the lifecycle, causing scope creep and schedule delays.

Those who have been through this tedious process recognize that unless there is communication and collaboration throughout the entire lifecycle, like a game of “gossip,” the end product is going to look very different from what was expected. But any changes that are requested during user acceptance testing are much more difficult to implement than if they were caught early. The later code changes are requested, the more complex they
become because of the dependencies in the application. The authors of the *Manifesto for Agile Software Development* recognized that business and IT collaboration throughout the lifecycle would allow for a partnership and flexibility that would lead to a higher-quality product. ALM vendors have followed suit by including collaborative features in their tools, which allow for strong teaming and the ability to respond quickly to changing needs, regardless of where team members or customers are. In this Agile scenario, updating requirements throughout the cycle becomes a collaborative process and business users and customers end up with the quality products they expect.

**COLLABORATIVE ALM**

ALM tool suites are designed to break down the former silos that existed between groups and instead provide an integrated and collaborative approach to software development. Technology trends such as cloud computing, social media and mobile computing have facilitated a work-from-anywhere world where collaboration and communication is possible without the physical boundaries imposed by an office. Such technology advances mean that distributed teams will continue to grow in number but will require strong collaborative ALM for successful teaming.

ALM tools are incorporating features both for facilitating stronger teaming internally as well as creating avenues for interaction with external customers.

Collaborative ALM features include things like the following:

- Any kind of communication feature (IM, forums, status updates)
- Linking artifacts
- Traceability
- Dashboards that combine data from various parts of the lifecycle
- Sharing of documentation or other artifacts
- The ability to “follow” discussions
- Notification of changes
- Capture collective feedback
- Polls and voting

Anything that allows for stronger communication and group teaming is considered a collaborative feature.

**INTERACTING WITH CUSTOMERS**

Though many ALM collaborative features are aimed at stronger internal communication and collaboration, some features are allowing organizations to communicate directly with customers. Many of these features are borrowed from the popular Web 2.0 or social media movement which allows for communication with anyone who has access to the Web.

Forrester analyst Tom Grant says that collaborative features in ALM tools are either actionable or contex-
tual. In using these features to gather information from customers, he says:

“I think that’s where the distinction between the actionable and contextual information is important. There’s information that’s coming in all the time in requests for new functionality that have to be processed and that’s actionable information. And with social media tools, that ramps up the amount of that information that’s coming in. There is also the contextual information so as a business analyst or product manager, I want to know how representative are these individual requests so I can go back and do some sort of aggregate analysis. How many people have ever asked for something like this? Why are they asking for it? That’s contextual information. So absolutely, there’s a lot

Why Requirements Change

Requirements are atomic but interdependent and exist in the context of “higher level” requirements. For example, a functional requirement exists to enable a use case, which exists to help a user achieve a particular goal (which the user has, given the user’s approach to solve a problem worth paying to solve).

Your team can experience a “change in requirements” in one of four ways:

1. The actual requirement changes—markets are dynamic, and users’ expectations (and approaches to solving problems) change.

2. The product manager (or owner or business analyst) changes her understanding of the actual requirements—we operate based on imperfect information. When we get smarter about our markets, it gets reflected as changes to the (documented) requirements.

3. The product manager changes how she communicates the requirement—just as developers can create bugs, so too can product managers, in the way they document what they understand.

4. The product manager has to adapt to the realities of feasibility. There are times when an approach to solving a problem is impractical (or impossible) for the team, given existing operating constraints. Going back to the original problems to identify alternate strategies for solving them, leads to different user cases and functional requirements. ■ By Scott Sehlhorst
of information that needs to flow in. Some of its very deliberate information as part of an ongoing conversation with the user; part of it is archival so I can go back and test hypotheses that I have. But this is another area I’ll certainly say I strongly believe there is no science yet to it.”

RESPONDING QUICKLY TO CHANGES
Collaborative features in ALM requirements management is more than communicating with customers, though. The internal business and development teams need to stay current on which features are being included in the product and know when clarifications or changes are made to requirements.

SearchSoftwareQuality.com’s requirements expert Scott Sehlhorst notes:

“There are two challenges in communicating changes in requirements: making sure the people consuming the requirements realize what has changed and making sure the requirements are still referentially consistent (that changes that propagate or ripple through the dependencies) are identified, documented and communicated.”

From a requirements-management point of view, collaborative features in an ALM would include things such as the following:

- The ability for all stakeholders (customers, business, development, quality assurance, product managers, business analysts, executives) to review and comment on the requirements in-line
- The ability to flag and discuss issues
- The ability to propose changes and resolve issues as a team
- Version control of changes to the requirements
- The ability to vote or take polls
- The ability to capture decisions and approvals from stakeholders
- The ability for stakeholders to be notified of changes
- The ability to create output documents tailored to different audiences

Sehlhorst explains how using a tool with such collaborative features, in which data is stored in a common repository, is much more efficient than the old-fashioned approach of a requirements document that is passed around for reviews and approvals.

“When you use a monolithic document (like an MRD or PRD or FRS), you can track changes to see where the document has changed. On large projects, this can become a significant burden, with high risk that changes will be missed by the consumers. There is also a high cost of repeatedly re-reviewing the things that didn’t change in order to discover the things
With today’s collaborative ALM tools, software teams are able to gather data from customers, stay notified of changes and communicate effectively throughout the lifecycle.

Requirements management has come a long way since the early days of passing around lengthy documents for approval. With today’s collaborative ALM tools, software teams are able to gather data from customers, stay notified of changes and communicate effectively throughout the lifecycle. □

Yvette Francino, the site editor for SearchSoftwareQuality.com, has more than 20 years’ experience in all phases of the software development lifecycle, having worked at IBM and Sun Microsystems. She has held management positions in software development, quality assurance and customer operations, managing diverse workgroups. She has a master’s degree in management and project management Regis University, and a bachelor’s in electrical engineering and computer science from the University of California, Davis.

that did. Most monolithic documents also fail to provide tools to discover the inter-dependencies between requirements.

When you manage your requirements as atomic entities in a repository (database), you can provide better tools for discovering when one requirement has changed (eliminating the need for consumers of requirements) to re-review those items that haven’t changed. You can also explicitly manage the network of inter-dependencies through traceability, enabling the product manager to make sure that they are discovering all of the ripples of change, assuring requirements consistency and completeness.”
UPDATING TOOLS, PROCESSES SEEN AS KEY TO OVERCOMING SCM CHALLENGES

The use of outdated software configuration management tools and processes is creating unnecessary challenges for ALM managers. So it’s worth it to compare organizational processes and tool sets that help ALM managers struggling to reap the benefits of SCM. BY CRYSTAL BEDELL

Software change and configuration management is an accepted part of the application lifecycle and is well established in most organizations. But outdated tools and processes are creating unnecessary challenges for many. If your organization is struggling to reap the benefits of SCM, it may be time to look closely at the tools and processes you’re using to see if there’s room for improvement.

According to Whatis.com, software configuration management (SCM) enables developers to “keep track of the source code, documentation, problems, changes requested and changes made” to software.

When SCM is done well, organizations benefit from repeatability, traceability and auditability, said Stephen Berczuk, an engineer at clinical informatics company Humedica. “Code is easier to reproduce and you have a more reproducible process. You have fewer people scratching their heads, wondering why something isn’t working,” he said. “People can move more quickly and be more predictable.”

CHALLENGES SURROUNDING SCM
If the results of our 2011 SearchSoftwareQuality.com Readership Survey are any indication, fundamental challenges with SCM prevent a majority
of readers from achieving these benefits. The top three challenges readers report with change and configuration management are a lack of standard and consistent procedures, lack of coordination with other groups and the need for additional automation to reduce human error.

These challenges likely stem from the use of outdated tools or inadequate processes, according to Sean Kenefick, a research director at Gartner Inc. “In so many cases, some of the older tools have weaknesses and issues due to their age. They were low-cost or free solutions. There wasn’t a lot of time spent trying to improve them, and they had major problems. Folks still using those tools are going to run into problems,” says Kenefick. He cites Microsoft Visual SourceSafe and CVS as examples.

In other cases, organizations are using older tools that are simply difficult to use. “They still perform their duties well, but they’re not efficient,” Kenefick said. He cited ClearCase, which is still a popular option among developers but well-known for its administrative overhead.

It also isn’t unusual for folks to use a lot of different tools within an organization, which can make coordination difficult. “One group might be using an old tool that doesn’t work well, while another group is using a new tool,” Kenefick said.

But Kenefick said the use of outdated tools is less of a problem than the processes that surround the tools. “Mostly folks need to improve processes. The tools are strong; they do what they are meant to do for the most part. It’s how you use the tools that is the real problem.”

If your organization takes a continuous improvement approach to SCM processes and still struggles with automation, cross-group coordination or maintaining consistent procedures, perhaps it’s time to standardize on
a new tool. “A lot of developers are switching to open source SCM tools,” says Jeffrey Hammond, principal analyst at Forrester.

The rising popularity of open source SCM tools is the result of the tools’ natural progression, Kenefick said. In the 1990s, developers had to choose between free or very expensive tools. There weren’t any quality midrange tools that were also cost-efficient. Subversion came and filled that void, paving the way for other open source tools. “Because people trusted Subversion, they started trusting others—Git and Mercurial—in ways they wouldn’t have if that history hadn’t taken place,” he said.

Several commercial tools are also popular. Hammond cited IBM Rational ClearCase, Microsoft Team Foundation Server, Serena Dimensions and AccuRev.

Must-have SCM features are relatively common among both open source and commercial tools. The experts we spoke to commonly cited integration, and branching and merging as core functions. While most tools have these capabilities, organizations should look for tools that are particularly strong in their areas of interest. “If your process requires a lot of branching and merging, then this is an important consideration,” Kenefick said.

The latest generation of tools emphasizes integration capabilities. “You need some level of integration between SCM and [continuous integration] tools,” Hammond said. “It makes the process of changing features and tracking their evolution less painful.”

Berczuk agreed. The most important aspect of choosing an SCM tool is finding one that works well in the organization’s environment and can be made transparent. “If you have a tool that doesn’t integrate well, then people will check in less often,” Berczuk said. Ideally, the tool integrates into your work transparently so that people don’t have to think too much about what they’re doing, he said.

Crystal Bedell is an award-winning writer specializing in technology. She writes articles, tips and guides to help IT professionals evaluate technology, secure and modernize their IT infrastructure, solve business problems and prepare for IT certifications. Email her at cbedell@bedellcommunications.com.
Today, a variety of ALM tools and techniques allow your team to communicate effectively through the development lifecycle, regardless of where they reside. Continuous integration, build servers, collaboration suites, Agile planning tools, version control can all help virtual ALM teams work well together. **BY STEFFAN SURDEK**

Technology for distributed collaboration for application lifecycle management has come a long way in the last few years. Email and conference calls used to be the main ways to allow team members to work together across great distances, but that is no longer true. Here project managers will learn about some modern-day uses of ALM tools that will have your virtual team members communicating as effectively as if they were sitting face to face.

**ONLINE COLLABORATION SUITES**
Many online collaboration tools allow team members to work together on the same document in real time. You can find either full featured suites with word processing, spreadsheet and presentation software or other tools for more specific tasks such as collaborative mind mapping.

Before using any of these tools in a corporate environment, verify company policy before storing and sharing confidential company information in the cloud.

**AGILE PLANNING TOOLS**
When working with distributed team members, it is more difficult to use a physical task board to keep track of
the progress the team is making during a sprint. Instead, having the team use a planning tool in a centralized location will help everyone collaborate better. You can find either commercial or cloud-based solutions on the market.

These tools should provide the team with the ability to create and update a product backlog, projects, sprints and sprint plans. They should also provide burndown charts and various other reports to help the team keep track of their progress.

Most tools have functionality such as attachments, links and discussions that can help distributed teams centralize information relating to a backlog item. More advanced tools provide Web-based dashboards the team can use to help the team track their progress easily.

SCREEN RECORDING TOOLS
I quickly introduced screen recording capability when discussing screen sharing tools but here are some other interesting ways to use screen recordings with distributed teams:

- Document designs by recording a presentation, talking through it and putting it in a shared location.
- Record snippets of working functionality for work from remote team members in a sprint review.

Any time when making recordings, you need to make sure to let others know before starting to record so they can voice their objections if they have any.

VERSION CONTROL SYSTEM
In ALM, virtual teams must have a centralized version control system and before using it, teams need to identify and document their project branching and integration strategy to make it clear to all team members.

A common centralized source code repository ensures everyone is working on the right version of the code and enables the team to use other practices, such as having an automated build server or continuous integration.

BUILD SERVERS, CONTINUOUS INTEGRATION
For a distributed team, having a common build server that can extract the project code from the common repository, compile and package it is an absolute minimum. Figure 1 illustrates...
four different levels of build server maturity and how the server evolves from one level to the next.

Once you have the build server in place, even at the first maturity level, you can reuse your build scripts as part of a continuous integration build. A continuous integration server will regularly check the code repository for changes and will automatically launch a build once it detects a change.

To keep the builds fast, some teams will regularly run a lighter build process that does not do any packaging but extracts, compiles the code and runs all the unit test suites. These same teams will run the full build process at least once a day at a predetermined time to create the deployment packages and carry out more extensive build verification testing.

For distributed teams, continuous integrations allows teams working in different locations a way to make sure the code they delivered in the common source code repository does not break the build.

SOCIAL NETWORKING

Social networks are no longer only for individuals keeping touch with friends and family. Companies can now also create their own private social networks on hosted sites such as Yammer and Facebook, which
make them easily accessible to employees from anywhere. These collaborative sites allow employees to easily share information, links and pictures with their colleagues.

Distributed teams can create separate groups on these social networks where team members can discuss and share information about their projects in real-time. Employees taking part in special committees at work (such as fundraising or green initiatives) could potentially join other groups specific to those activities.

Virtual teams have so many alternatives available now compared with only four or five years ago that it is difficult not to be successful working remotely. The various tools mentioned in this article provide a canvas to work together limited only by the imagination of the people using them. Take the time to put some of them together and find which combination works best for you.

For a comprehensive resource on social media in ALM, see Social media: A guide to enhancing ALM with collaborative tools.

Steffan Surdek is an Agile coach in Montreal and co-author of the book A Practical Guide to Distributed Scrum. He coaches software development teams to help them adopt Agile practices and speaks at various conferences and user groups about using Agile practices with distributed teams. He regularly blogs on his website, www.surdek.ca.