JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Agile SCM January 2007 - Looking Back to Move Forward

ARTICLE

By Steve Berczuk , Brad Appleton Robert Cowham | January 16, 2007

SUMMARY

The previous two January Agile SCM columns made some short-term (and a few longer-term) predictions about the impact of current trends on the practice of Agile development as it pertains to SCM. This January we revisit those predictions to see where we were right, where we were wrong, and recalibrate our ideas of what's in store for Agile SCM in 2007-2008.

The previous two January Agile SCM columns made some short-term (and a few longer-term) predictions about the impact of current trends on the practice of Agile development as it pertains to SCM. This January we revisit those predictions to see where we were right, where we were wrong, and recalibrate our ideas of what's in store for Agile SCM in 2007-2008.

Trends from 2006 SCM Predictions

There was a great deal of synergy between many (if not most) of the predictions of the various contributors to the January 2006 CM Journal. Brad summarized those as follows in an April 2006 ACME blog-entry:

Globally Distributed development is the new "normal"
Globalization is kicking into high gear in the "flattened world" (see Thomas Friedman's The World is Flat and Barry Lynn's End of the Line : The Rise and Coming Fall of the Global Corporation). The trend of globally distributed development is increasing at a rapid rate and will be a more important focus (and more frequent "buzzword", along with "collaboration").

[Note: One of the authors has been working with a large financial institution looking to replace/augment their existing (very significant) investment in SCM tooling because it

We use cookies to ensure you get the best experience on our website.
Read More
Accept

solution just wasn't fast enough. They found that working on the same repository (with one side accessing it remotely -- performance was sufficient), is a better solution.]

ALM is the "new" SCM!
Application Lifecycle Management (or ALM) is becoming the new/trendy term for the full-spectrum of Software CM. Vendors have spent the last 6 years or so buying-up individual tools to provide "full-lifecycle" suites to enhance an IDE: modeling tools (that do their own versioning), requirements tools, version control, test management, change-request tracking, and others are all being offered as singly-packaged suites of integrated tools. And the vendors are using the term "ALM" for the result.

TeamSystem, and Eclipse ALF & Corona are the new "Vision"
Microsoft's new Visual Studio Team System (and Team Server) will make a huge splash! Look's like the splash already started, with TeamSystem winning a "Jolt" award (though not in the SCM category). The fact that it's Microsoft and integrated with VisualStudio is enough, by itself, to warrant other vendors "taking notice." With a single integrated tool-set running off the same server and integrated with the IDE, you can do all sorts of amazing things with automated logging and traceability (especially if it's web-services enabled, or the .NET equivalent).

In order to compete, other vendors are aligning with the Eclipse ALF project, which just recently had its "proof of concept demo." ALF is the *Application Lifecycle Framework*, an under-development set of web-service and interoperability standards for vendor-independent integration of tools in the ALM space (version control, change/defect tracking, requirements mgmt, test management, project tracking, build/release mgmt, and even IDEs and modeling tools). And at **EclipseCon 2006**the Eclipse Corona project was announced as a recent "spinoff" of ALF. According to a March 20 InfoWorld article:

> ALF addresses the issue of integration and communication between developer tools across the lifecycle; Corona enables Eclipse-based tools to integrate with ALF, according to Eclipse. Also known as the Tools Services Framework, Corona provides frameworks for collaboration among Eclipse clients."

Other related Eclipse projects are **BIRT** (business-intelligence & reporting tools), the **Data Tools Platform**, the **Test & Performance Tools Platform** and the SOA Tools Platform.

Integration of CM with ITIL is the new "Enterprise CM"
More and more SCM services departments are having to deal with many of the same issues of IT (and even being lumped together with IT) as they support not just the functions for CM, but also provide deployment, training and support for the corresponding ALM tools, repositories and servers. The integrated CMDB is also useful for both traceabilty and accounting/accountability (think Sarbanes-Oxley).

Also, more and more folks are needing to extend CM into their deployment sites at their customer's operations to help them handle field-issues and upgrades, even monitoring

We use cookies to ensure you get the best experience on our website.

whole shmeer" are having to learn and understand CM, ALM and ITIL. Word has it that integration of CMMI with ITIL is coming soon.

SOA, BPM and TCO are emerging priorities on the horizon
There wasn't quite as much mention about these, but it seems clear that they will be growing more important this year. Service-Oriented Architecture (SOA) will be important to the extent that TeamSystem and ALF are important, and some vendors are already using SOA-based integration for the tools within their own suites.
As the integrations between ALM tools and with the IDE become more seamless, Business Process Management (BPM) and Workflow will become a bigger concern as folks try to more readily define and automate their processes, particularly for deployment to multiple distributed sites.

And Total Cost of Ownership (TCO) is becoming an increasingly more prevalent factor in selection of CM/ALM tools. The spiffy features and functionality are nice, but just as much weight (if not more) is being given to business issues of global licensing and upgrade/support, administration/maintenance, along with total cost of ownership.

I would say that all of the above are still "spot on" and we can expect to see more of the same this year - with one notable exception ...

ALF and Corona? or Open-Sourced Jazz?
While there has been activity on the websites for the ALF and Corona projects, we havent been hearing so much about them in the media outside of Eclipse-centric conferences, and press-release announcements from yet another vendor with an Eclipse "compatible" release of their software. While this certainly doesn't shatter the prediction, there has been more attention (perhaps due to more marketing?) about the upcoming ALM collaboration framework from IBM Rational called "***Jazz***." Based on the **RDSC 2006 Day 2 Keynote presentation** (and also at JavaOne),

Jazz is touted as the next generation of the Eclipse platform and the team based software development platform. Rational will of course provide a tool-set to go with it, but says it plans to release the overall framework (and code/APIs) under Open-Source. The claim is that the result is an ALM collaboration architecture which is inherently scalable to provide true support for geographicall dispersed development projects in a true collaborative fashion. Well known names from Rational have been presenting Jazz at various conferences throughout 2006 and generating lots of "buzz," as evidenced by articles and blogs like the following:

- IBM is Revamping Rational
- IBM Jazzes Collaboration

We use cookies to ensure you get the best experience on our website.

- Jazz is the news
- IBM Jazzing up Rational Offerings, from InfoWorld
- IBM to employ open-source development style for tools, from eWeek
- Big Jazz News on IBM developerWorks blogs
- **Agile Requirements Engineering in the Jazz Collaborative Development Environment**, from Rutgers 2006 Software Engineering day
- **Hot Jazz vs. Cold Beans**, blog-entry

Interestingly enough, with all that marketing buzz, Rational was also very clear about that fact that Jazz would not be released until *after* 2006. So look for even more "hype" and "hope" in 2007 when the first releases of Jazz become available, and especially when the core Jazz infrastructure is released as open-source under the auspices of Eclipse project. It will be interesting to see the impact this has on the Eclipse **ALF** and Corona projects (which Jazz doesn't currently use, althought it could conceivably make use of Corona in the future) and the Eclipse **BIRT** project (which Jazz does make use of). Current marketing would suggest that Jazz is the new vision, or more aptly, more likely, the new open-source hope to slay (or at least compete with) MS VSTS

Trends in ALM Tool Functionality
With the MS VSTS and Rational Jazz marketing, collaboration and collaborative development environments (CDEs). And with "Agile" becoming mainstream, and getting used more and more as a marketing term, "Agile" and "Collaboration" may very well be the most frequently used marketing terms (with the possible exception of "Governance" and "Compliance"). When it comes down to the details of specific functionality, we've been forecasting the following since 2005 to happen during 2005-2008:

Component-Based Versions (Baselines)
Many version control tools were already supporting the notion of versions as things that encompass more than just one file or one directory. More tools are starting to provide automatic support for capturing versions of the entire project or repository. There still seem to be relatively few that implement the notion of a "component" (or "version group") separately from the notion of a project or repository. In some cases, the notin of a "project" is equivalent to a component, but in most it is not. And more often than note, most tools today consider a "project" to be either "the whole repository" or else a top-level directory in the repository. With a select few exceptions, there is still little automated support for the notion of a "component" as a versioned group of files that evolve together, and that don't comprise the entire repository.

Composite Streaming & Baselining
Here there has been very little activity or evolution. I'm aware of at most one or two tools that support the notion of a "composite baseline", and none that support the notion of a "component of components" or of a "stream of streams." It seems we'll have to wait a bit longer to see this one. Perhaps we won't see this until product-line-oriented practices and

We use cookies to ensure you get the best experience on our website.

Task/Activity-Based Development
Task-based Development (TBD) does seem to be on the rise, in practice. There doesn't seem to be a lot more tools adding it that didn't already have it. There does seem to be more open-source offerings and integrations with tracking tools making this easier to do for one's existing tool. So the trend itself is on the rise, there seems to be less demand for tool vendors to add it to systems that didn't already have it (other than via integration with something else).

Improved Refactoring Support
We have been seeing this, and the trend has definitely risen in the past two years. It may have already peaked. The capability to version directories and track renaming certainly does seem to be on the rise.

Continuous Integration "Architecture" and Enterprise Continuous Integration/Staging
This is perhaps one of the areas where the most strides have been made. With newcomers and/or new acquisitions and the marketing for the likes of Buildforge, Electric-cloud, and the increase in books on systems such as Maven, CruiseControl, and many other tools (both commercial and open-source), systems, servers, and architectures for continuous (or very frequent) integration seem to be getting much more attention both from tools vendors and from projects.

Testing Framework Integration
Perhaps we simply haven't been hanging out in the right circles, but this one doesn't seem to be getting as much attention as we originally suspected. It certainly seems to be on the rise, but some of us had expected this to keep pace with the rising spread of continuous integration practices and automated build architectures.

Private Branches/Streams and Private Versions
As project-oriented (stream-based) branching has gained wider use, so has the use of private branches/streams and private versions as a practice. **AccuRev** seems to be one of the tools leading the way in this area, with help from old standbys like **Perforce** and **Clearcase**, and relative newcomers like **PureCM**, **Seapine**, **OurayCM**, **SpectrumSCM**, Hansky Firefly, SnapshotCM, Plastic SCM and more. (And let's not forget about **CM+**, which was one of the earliest tools to implement the notion of streams not merely as project-oriented branches, but also as first class objects with additional useful properties.)

There are even more tools that take a Bitkeeper or GNU-Arch style of approach to decentralized repositories where each workspace is a first-class repository (thus

We use cookies to ensure you get the best experience on our website.

Continuous Coordination
This one is hard to tell. The emphasis on distributed development and of collaborative environments and their corresponding collaboration architectures seems to indicate this is coming more into the forefront. Architectures such as those from Collabnet, Rational Jazz, and MS VSTS seem to indicate we'll be seeing more of this to come.

Workflow and Promotion Hierarchy
Very slow progress on this one ... there seems to be more emphasis on this in tracking systems than in version control tools (or their integrations). You would think that Architectures and frameworks such as MS VSTS, ALF & Corona, and Jazz would bring this more into the forefront. The Scarab project reached version 1.0 in 2006 with very little hoopla. Atlassian's Jira seems to be the leader in this area among Agile shops.

More/better integration with IDEs ...
and with agile-friendly Project Management & Test tools, and even Instant Messaging, Collaborative editing environments (e.g., Wiki-Webs), and Content Management Systems (CMS). Here again the emphasis on collaboration architectures from MS VSTS and IBM Rational Jazz appear to be setting the bar, with ALF still serving as a decent incentive for many other vendors. Integration of version-control tools with content-management seems to have taken place only in the form of CMS' with versioning capabilities. The notion of integrating the two together doesn't seem to have taken off yet.

Convergence of CM tools and Content Management Systems (CMS) capabilities
Here, Brad seems to have been way off (or maybe it's an idea whose time hasn't come -- yet). With more and more Web-based software and the rise of XML usage, it would seem natural that source-code and active/dynamic content increasingly need to be managed together. Yet there seems to be very little work going on in this area.

Real Tacking Tool
This has definitely been happening in the Agile world, and is on the rise. Despite the call from many of the most prominent vocal voices in the Agile community to "not go gently" into that good tracking database, more and more are resorting to doing just that, but with an emphasis on Agile-friendly tools that are light, fast, and minimally invasive. The most well known/liked ones in the Agile community seem to be Jira, VersionOne, Scrum Works, RallyDev, and TargetProcess (Microsoft also has MSF for Agile as well as MSF for Scrum).

Event-Based Traceability
Collaboration Frameworks and their integrations with IDEs, such as those provided by

We use cookies to ensure you get the best experience on our website.

of traceability is becoming more popular. It is sometimes called discovery-based, query-based, or "Just Google it" traceability. It is a form of traceless traceability, where smart probabilistic search engines with some syntactic and semantic "knowledge" can search thru a project based on terms and keywords. It does require using a consistent (and documented) vocabulary throughout the project. Still, it can be greatly enhanced with event-based mechanisms. So I would still say look for this trend to start picking up due to VSTS and Jazz.

SCM and Requirements Repository Integration/Unification

This is another area where there have been some rumblings, but not as much as expected (similar to integration with content management). Some next generation requirements tools are sporting Eclipse integration, but we've seen little in the area of tighter integration between requirements and code in a single (possibly virtually unified) repository. The use of Wikis/Mashups, and Wiki-like content management systems for managing requirements (and project/process) knowledge does seem to be on the rise.

Lean Traceability

There have been a few rumblings and strides in this area, but there's a lot of talk/desire, and occasional presentations of techniques used, very little in the way of automated support for this.

Full Application Lifecycle Management

This one is almost unavoidable. Certainly if you believe all the marketing hype, it has happened. In reality, we probably still have a ways to go, but the attention certainly seems to be there.

Enterprise CM Integration Architecture & Enterprise Data Management/Warehouses

This does seem to be happening to the extent that it is spurred on by the likes of ITIL and CobIT. The tool-sets for this seem very different from those dealing with SCM and there doesn't seem to be much integration between them yet. Despite some vendors claiming to be truly enterprise class CMDBs, a true CMDB seems to be a "holy grail" quest of sorts. Systems like VSTS and Rational Jazz do seem to have some nice data warehousing & reporting capabilities, so we do expect to see more in this area.

The Rise of COBIT

Control Objectives for Information and related Technology (COBIT®) provides good practices across a domain and process framework and presents activities in a manageable and logical structure. COBIT's good practices represent the consensus of experts. They are strongly focused on control and less on execution. The business

We use cookies to ensure you get the best experience on our website.

against business requirements, management should put an internal control system or framework in place. The COBIT control framework contributes to these needs by:

- Making a link to the business requirements
- Organising IT activities into a generally accepted process model
- Identifying the major IT resources to be leveraged
- Defining the management control objectives to be considered

Companies are successfully putting together frameworks with COBIT at the top using ITIL and CMMI for lower level best practices and execution.

What did we miss?
So that seems to be the wrap-up on our predictions for the last two years and how they are faring! We'd be interested in hearing from our readers if you think we've missed anything else. What do you think?

Topics: configuration management, lifecycle models,

We use cookies to ensure you get the best experience on our website.

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

Brad Appleton

Brad Appleton is an Enterprise Agile+DevOps leader, coach & manager, and seasoned DevOps/ALM/CM solution architect at a large Fortune 100 company. Currently he helps organizations and teams scale, adopt and apply lean/agile development methods and DevOps/ALM/CM practices and tools. He is co-author of *Software Configuration Management Patterns*, the Streamed-Lines branching patterns, and a former section editor for *The C++ Report*. You can read Brad's blog at blog.bradapp.net.

Robert Cowham

Robert Cowham has long been interested in software configuration management while retaining the attitude of a generalist with experience and skills in many aspects of software development. A regular presenter at conferences, he authored the Agile SCM column within the CM Journal together with Brad Appleton and Steve Berczuk. His day job is as Services Director for Square Mile Systems whose main focus is on skills and techniques for infrastructure configuration management and DCIM (Data Center Infrastructure Management) - applying configuration management principles to hardware documentation and implementation as well as mapping ITIL services to the underlying layers.

UPCOMING WEBINAR

We use cookies to ensure you get the best experience on our website.

LETS HANG!

FEATURED RESOURCES

SECURITY

Shifting Security Left: DevSecOps eGuide

White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

0 comments

YOU MAY ALSO LIKE

SECURITY

Agentic AI Opens Up New Security Flaws

Nahla Davies | August 27, 2025

We use cookies to ensure you get the best experience on our website.

SECURITY

Countering the Threat of Ransomware-as-a-Service (RaaS)

Nahla Davies | August 22, 2025

SECURITY

Building Trust in Blockchain: The Role of Smart Contract Testing

Ekaterina Egorova | November 18, 2024

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.