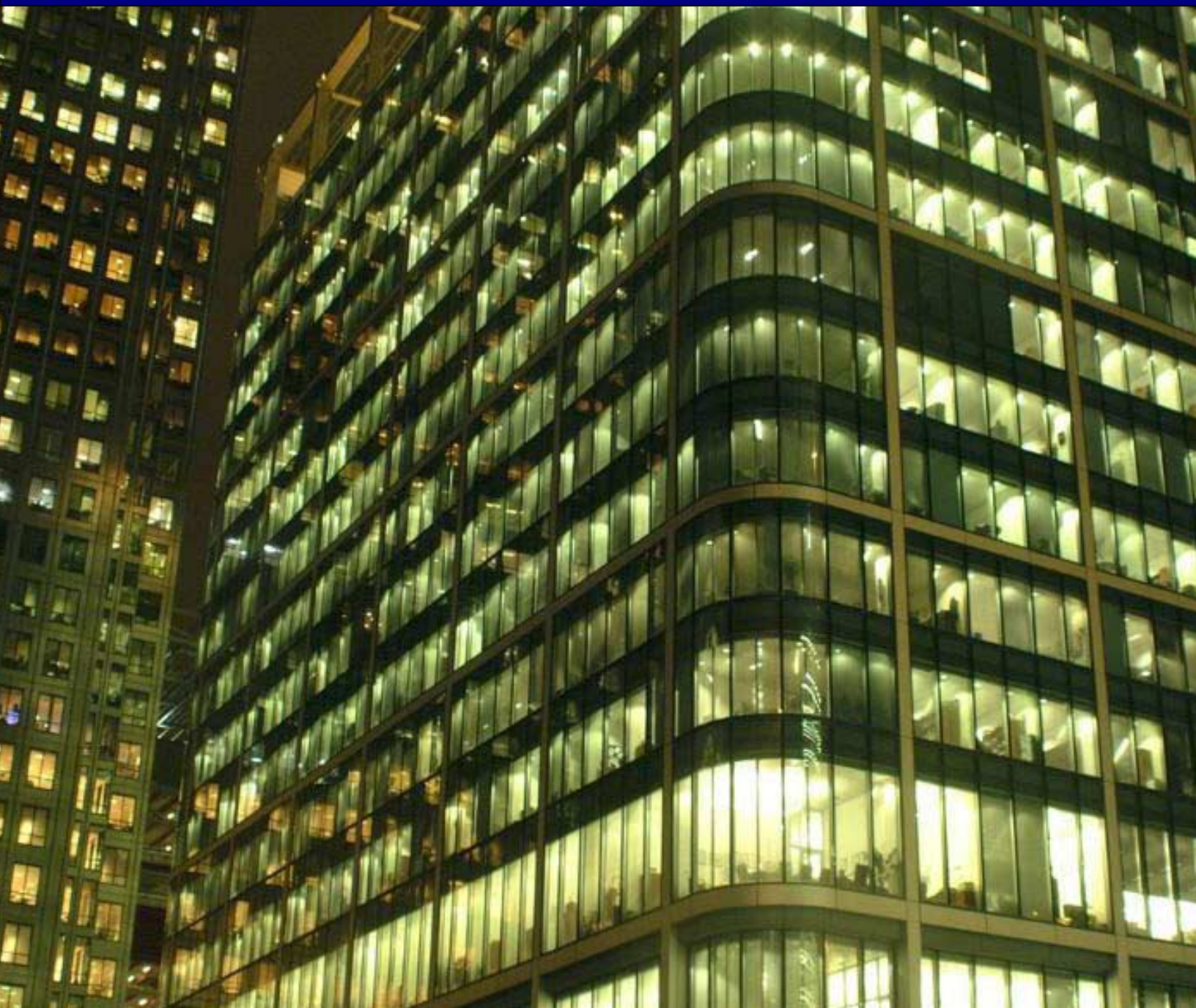


The “Best of Breed” Agile ALM Solution

By Ben Weatherall



We are almost through January of 2010 and with the release of AgileCycle, it seems to be the year Agile ALM will finally be a reality!

Toward the end of 2009, there were several rumors of ALM solutions for those practicing the Agile Methodology and many of these were from single-source vendors. This made sense since vendors that provide a broad solution base have access to the entire codebase and can make whatever changes they want to implement a solution. Unfortunately, as we have seen over the years, the various tools within such a product suite often do not integrate any better than tools from multiple vendors and that if customizations are required the effort by the user tends to be non-trivial. Other solutions, such as the Eclipse Foundation’s ALF project, are not catching on like it was hoped – primarily due to the difficulty in integrating with the various release levels of the different components in place within the user’s environment.

There are three options to single-source vendors: (1) Free/Open-Source Systems (FOSS), (2) Third-party integrators and (3) vendor supported integrations. The FOSS option will eventually catch up, but Agile-specific (or tailored) tools and plug-ins are still in their infancy. Until they mature and settle into a few leading tools, this will remain more of a roll-your-own solution for most users. Third-party integrators are out there and in many cases do a good job of gluing both FOSS and COTS (Commercial Off the Shelf) components together, but selecting a specific integrator can be risky and often, like single vendor solutions, lock a user into a specific provider. This, along with the roll-your-own approach, is pretty much where we are today.

The final option, that of a set of cooperative COTS vendors providing an out-of-the-box Agile solution is one most CEOs and CIOs would prefer to have available. This is where the hopes for those organizations implementing Agile in 2010 lie, and this is where AgileCycle will shine. There will be a minimum user base where this type of solution is viableⁱ and it will probably be somewhere between 25 and 50 users. If payback is not expected as quickly, the number of users could be significantly less.

So what are the minimum requirements for an ALM solution for Agile? There needs to be an Agile-specific management system, a workflow engine, a version control system, a defect and enhancement tracking system and a build management system. And each of these components need to be supplied by vendors (or the FOSS equivalent) that have good reputations and “corporate” stability, be responsive to user requests for change and defect resolution and be considered to be among the “Best of Breed” in their market niche.

The “Best of Breed” Agile ALM Solution

In the realm of Agile-specific management systems, the current leader is Rally. They are typically a hosted solution, but one that provides decent security and encrypted transfers. Rally provides the normal User Story catalogs, backlogs, Sprint management, metrics, and other features deemed necessary to successfully perform in an Agile development environment. They currently report over 2,000 customers with over 54,000 projects so they are definitely a presence in the industry.



Figure 1: The Rally Home Screen

Version control is something that most users already have in place, but the Agile Methodology places certain constraints on the tools that often require replacing existing tools. The ability to support multiple parallel development activities with quick “branching” and efficient (and intelligent) merging is essential. It is also necessary to keep track of the changes to file names and locations to facilitate merging changes into/from refactored codebases in order to keep long term control over the quality of the code. There are several good vendors of these tools, but many of them are parts of tools suites or have APIs that do not fully implement the features available through their native UIs and plug-ins. While not the only one in this category, one of the ones considered “Best of Breed” is AccuRev. More later on why I specifically selected it for mention.

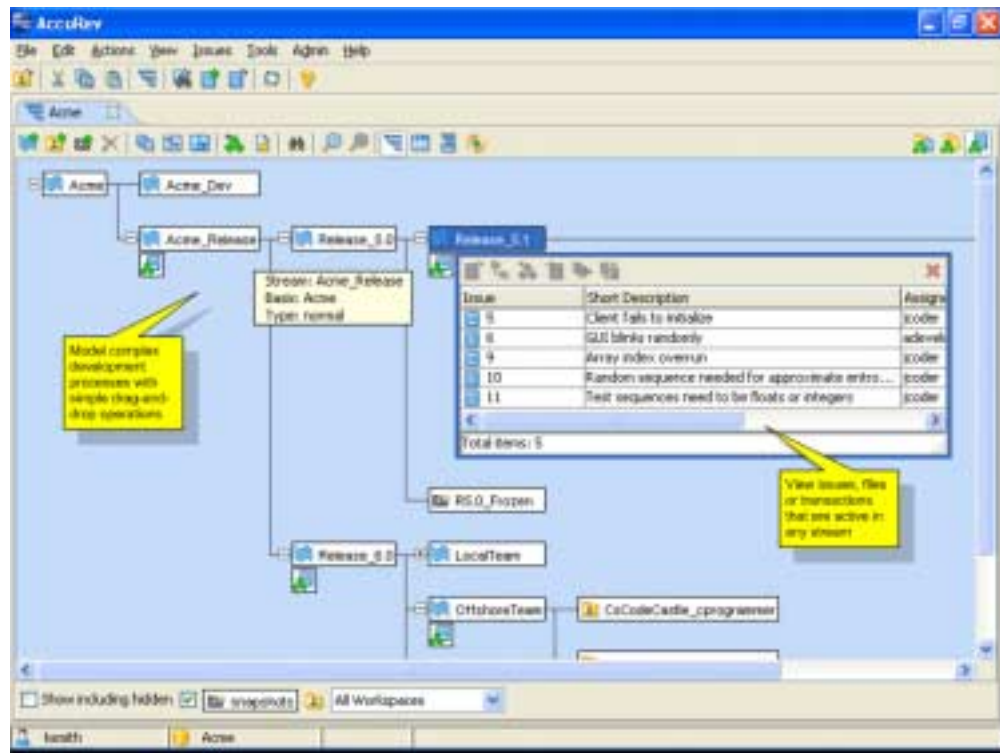


Figure 2: The AccuRev Stream Browser

Defect and enhancement tracking (DIETⁱⁱ) capability is something that just about everyone is including in their tools these days. Sometimes it is even implemented so that it is customizable without breaking other external integrations, but sometimes customizing it makes it difficult to use with other tools and/or plug-ins. Many organizations have already invested significant time and effort into getting their preferred DIET solution working the way they need. Migrating to another solution, even if that solution is “free,” is often so costly that the past history is abandoned and only active issues are migrated. Of course, if a user is starting Agile from scratch on a new product, then they can use a new DIET solution without too much trouble – if one ignores the cost of retraining and maintenance of multiple solutions. This is where bridging comes in handy. Being able to bridge your existing DIET solution to the integrated one solves multiple problems for a minimum cost. This also allows the vendors that have an integrated DIET solution to use it with minimal changes. One of the reasons I pointed out AccuRev above is that it has AccuWork integrated with AccuRev and it also has bridges already available for a number of external DIET products, both COTS and FOSS and the ability to add new ones fairly easily.

Build management systems come in multiple flavors, but there are only a few that stand out as “Best of Breed” ALM components. Of those, each has their strong points, but all of them allow for use of external version control and DIET tools.

Additionally, they often provide their own form of workflow management system, but it is more along the lines of ensuring that the proper steps have been completed prior to doing steps in code/change promotion and/or release/deployment. At this time, *most* Agile projects are not so large as to need massively parallel builds, but the ability to manage a build farm, execute Continuous Integration (CI) builds as well as production/release candidate builds, generate Change Logs, capture build logs, run tests and allow/require electronic signatures prior to doing gated activities are reasonable expectations.

As far as workflow engines go, most of the requirements for this are met by the current “Best of Breed” build management systems. Some, like AnthillPro from UrbanCode, even provide ALM capabilities out of the box. The Eclipse ALF project was basically a workflow engine, but one perceived as tied mostly to the editor and one that was resource intensive. Those like AnthillPro’s are server based and the UIs only provide access to it.

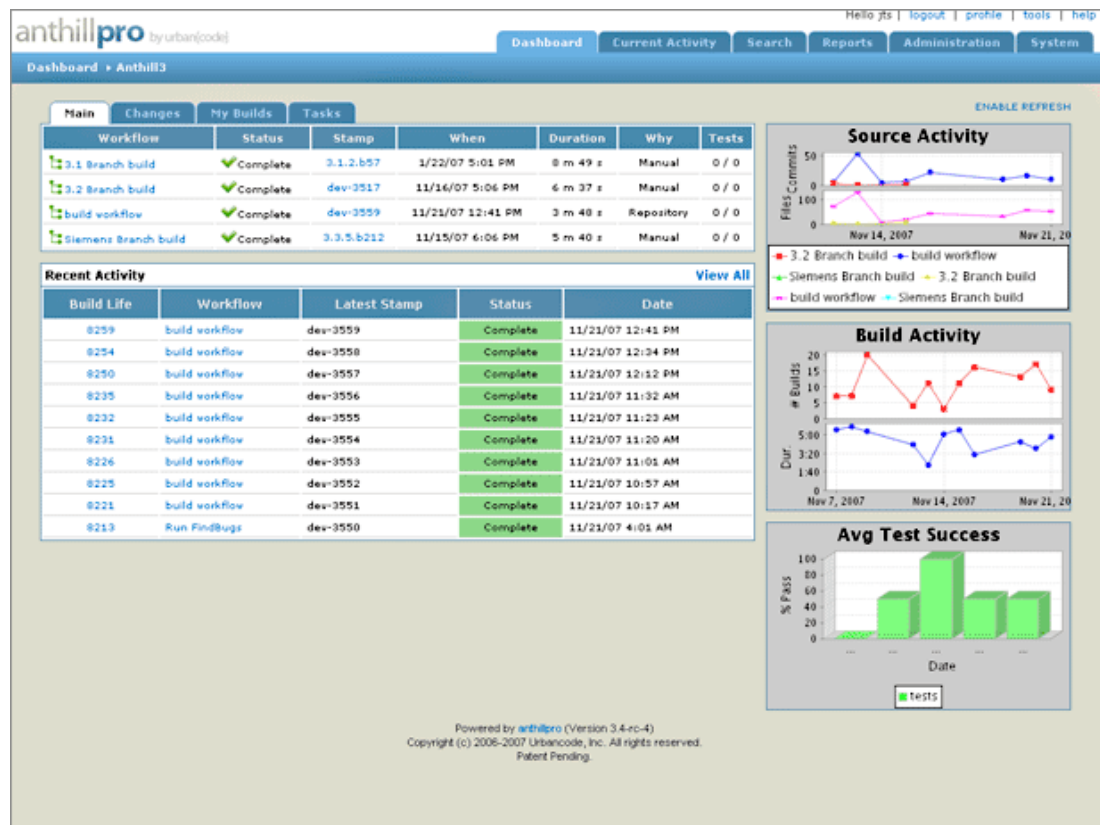


Figure 3: The AnthillPro Project Dashboard

So, forming a “Best of Breed” ALM solution from multiple vendor components would be something like having AnthillPro manage the overall ALM framework and the builds while AccuRev manages the version control, change tracking and DIET functions and Rally handles the Agile-specific management stuff and helped control the content and organization of the reported defects and enhancements. It might look something like this:

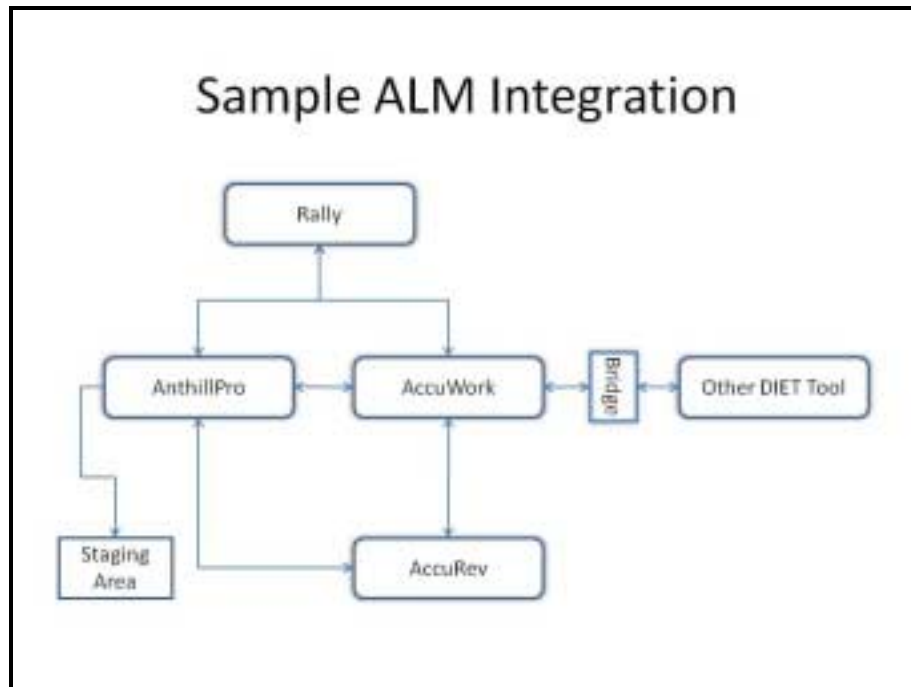


Figure 4: Integrated ALM System

With the advent of AgileCycle, we finally have a solution where the various vendors cooperate in creating an out-of-the-box integration tailored to Agile development. All of them supporting it is a dream come true.

ⁱ In this case, viable means a ROI payback within one year.

ⁱⁱ DIET actually stands for Defect, Issue and Enhancement Tracking. In integrated solutions such as AgileCycle, which tool manages each of these areas is not as important as the fact that they are all managed in a coordinated fashion that helps people do their jobs easier.