The Rapid Project Platform

Speed without Compromise

For Development Teams

Presented By

ProjectLocker™

Software Quality Without Pain

http://www.projectlocker.com

Introduction

Building software is hard enough with the right tools, but many teams settle for sub-par development platforms, thinking that they do not have the time to bother with setting up source control for their code, issue tracking for their bugs, or an automated build process. In fact, the opposite is true. When teams do not invest in a solid development platform, not only does it make their work harder and more frustrating, but it also ends up costing the team *more* money over time, in the form of lost work and delays. Thanks to hosted software development platforms, teams no longer have to choose between working fast and using a robust platform.

In this paper, we are going to discuss some of the problems frequently faced by software development teams. We will review some of the known solutions to these problems and learn why more teams do not use these solutions. Finally, we will examine hosted options that solve the same problems without the drawbacks of conventional approaches.

SCENARIOS

Consider the following common scenarios.

Onsite Development

A team of consultants is assigned to an onsite custom software development project. Like many IT projects, this one has an aggressive timeline with demanding goals. The project manager knows that not using best practices will jeopardize the team's success, so he presses for access to the customer's existing source control and issue tracking systems. The customer is a large firm, so it takes many weeks and a substantial amount of paperwork before the appropriate clearances are given. Later in the project, when the long hours and weekend work requires remote access via VPN, the process begins all over again. *How can the team adhere to robust development processes without losing weeks of time to bureaucratic procedures?*

Small Company

A small company is preparing for its initial launch to the public. Although they are not building a software firm, they plan on building software to manage their internal operations and customer-facing website. Their technical team is savvy, and they avail themselves of open-source software for bug tracking and source control. However, after a few months they find that they are spending about 25% of one employee's time managing these servers. The CEO wants to know why the company is dedicating so much time to these activities when important deadlines are approaching. *How can the firm benefit from a complete development toolset and still make its deadlines?*

Offshore Contractors

The IT group within a large industrial firm has endured ongoing pressure to reduce costs and headcount. As a result, the group has reconfigured around a core of permanent employees and makes extensive use of outside contractors, many of whom are overseas. The contractors' work is fantastic, but coordinating them and getting new contractors rapidly up to speed is increasingly difficult. Sharing source code is difficult because the team uses Microsoft SourceSafe™, which does not work well over the Internet. Even worse, the project manager spends hours each day updating and distributing the latest spreadsheet of all the issues that everyone is working on, and team members spend more time reviewing and determining which spreadsheet is most current. *How can the distributed team share source code and process bugs without being in the same office?*

OBSERVATIONS

Do any of these sound familiar? Unfortunately, scenes like these play out all too frequently. As these scenarios illustrate, building software in today's environment is about more than just writing code. Attention to the details of process and infrastructure is crucial to controlling costs and reducing risks. These details are often in conflict with the aggressive time constraints that govern most IT projects. The tradeoff between Doing It Right and Doing It Fast are at the heart of why many teams abandon their planned development processes.

Moreover, since many organizations ignore these aspects of software projects, utilizing a few simple best practices allow forward-thinking teams the opportunity to gain competitive advantage. In the following sections, we will outline some of these practices and present suggestions for introducing them to your project quickly.

The above scenarios also demonstrate that even when teams know what they should do, external factors sometimes get in the way of adherence to best practices. The most frequent culprits include:

- **Schedule**: The team is under pressure to complete the project soon, so they believe they do not have time for best practices. In fact, use of best practices in software development leads to shorter release cycles with fewer defects.
- **Bureaucracy:** Interminable delays to gain access to internal systems or to allow access to external resources
- **"Not Invented Here" Syndrome**: A team member insists on managing process tools, creating their own tools, or sticking with an inefficient set of tools, even though the organization does not have the resources to do so efficiently
- **Connectivity:** Working with distributed teams is difficult if the team uses tools that are not designed to work over the Internet.
- **Documentation:** There are no standardized documents. There is no easy way to collect ad-hoc project documents, so as a result there is no documentation about basic processes, data, etc.

There Is A Better Way

Fortunately, teams are now able to reap the benefits of best practices on their projects without falling prey to the typical external roadblocks. There is an emerging group of hosted online software process tools that address most aspects of the basic software development life cycle. With their incredibly short start-up times and zero maintenance effort, they are the model for the future. In much the same way that sales teams are migrating to online tools like salesforce.com, software teams will begin to reap the benefits of connected infrastructure tools.

These tools enable users to develop on a Rapid Project Platform (RPP). The basic ideas behind RPP are simple: 1) you understand and want to follow best practices and 2) you want or need to start using them immediately. RPP, achieved via hosted process tools, is the fastest way to accomplish these goals. You can establish a RPP in less than a day, letting your team reap the benefits of adherence to best practices without the lead times that have historically made adoption of such practices difficult.

Hosted tools come with varying levels of guarantees around availability of the tools, compatibility with users' operating platforms, backup, and recovery planning. Some providers are focused on helping individual developers or informal teams reap the immediate benefits of best practices without necessarily being concerned about security (as the projects may not be particularly important to protect or may be open source). Other providers focus on solving the concerns for corporate teams and organizations around accessibility, security of the data, intellectual property protection, and disaster recovery.

Hosted source control providers typically use CVS or Subversion, both of which are open source and supported by many popular IDEs. CVS is the older of the two products, and many developers are familiar with it. Subversion is newer, but it already includes many improvements over CVS. In many cases, developers will not have to install any additional software to switch from an internal repository to a hosted source control repository.

Web-based issue tracking is probably the most mature of the online software process tools. These providers do not typically require any software other than a Web browser to be installed. Providers offer a number of solutions, ranging from hosting variants of popular open source issue tracking systems such as Bugzilla, Scarab, or Trac to offering custom solutions.

Web-based document-sharing services can provide your team with a quick and easy way to begin building a knowledge repository for your project. If you prefer a structured folder-based approach, there are document management services that will allow you to organize your team's documentation according to your style. For quick file sharing, there are unstructured document sharing services that allow file uploads to a server or allow you to map drives using the WebDAV protocol. A third option, Wiki, allows your team to organically build a database of documents and notes for the project. Many teams

prefer Wiki for project documentation because it allows teams to quickly capture information before it can be lost from the collective knowledge base of the team.

Why Hosted?

There are two main reasons for choosing hosted solutions for Rapid Project Platform: speed and cost.

Hosted solutions are the hands-down winner for speed of implementation. It is not uncommon for a team to set-up its project infrastructure and begin working in a single day using hosted solutions. Unless your team can piggyback on an existing infrastructure at your organization, setup times for in-house systems will be longer. Consider that in addition to securing the necessary approvals, you may have to procure and configure hardware; procure and install software; have the IT team open the firewall for remote access; define a back-up and recovery strategy; and perform regular OS and software upgrades. In short, properly managing a software development infrastructure is a lot of work. Hosted solutions let you outsource the grunt work to experts and focus on delivering software.

Hosted solutions are also less expensive than in-house solutions. This surprising outcome is a result of the fact that there are tremendous hidden costs associated with any in-house software implementation. Your team's labor is by far the most expensive asset you have, and managing infrastructure is a sure way to spend it quickly. Routine tasks that take only a few hours a week quickly add up, robbing your team of productive time.

Table 1 illustrates the hidden costs in managing an in-house source control system. These scenarios are by no means extreme. In fact, many in-house implementations will cost significantly more due to lack of in-house expertise, lack of hosting facilities, etc. Hosted solutions allow you to sidestep these costs. Before deciding to manage your systems in-house, make sure you understand all of your costs and how they compare to hosted options.

	In-House Commercial	In-House Open Source	Hosted
Software License	$ 1,000.00	$ -	$ -
Software Maintenance	$ -	$ -	$ 1,800.00
Hardware - Rackmount Server w/RAID	$ 2,000.00	$ 2,000.00	$ -
Software Setup - 20 hours	$ 750.00	$ 750.00	$ -
Ongoing Maintenance & Support - 3 hours/week	$ 11,250.00	$ 11,250.00	$ -
Total	$ 15,000.00	$ 14,000.00	$ 1,800.00

Assumptions:
1) 5 developers
2) Company already has a hosting facility, firewall, etc. for the server.
3) Setup and maintenance is performed by staffer with annual salary of $75K.

Table 1: 2-year Source Control Costs

Getting Started with Rapid Project Platform

Once you decide that a Rapid Project Platform approach to project infrastructure could benefit your team, how do you begin? While the specifics of how you utilize each tool are dependent on the process your team uses, the basic steps will be the same.

1) Determine which of CVS or Subversion your team will be more comfortable with. If there is no clear winner, go with Subversion. Then choose the hosted source control provider that best fits your needs. Factors that go into your decision might include pricing, support availability, whether you want to see advertisements, etc. Also keep in mind that data storage quotas imposed by the vendor may have unintended consequences on your team's work style and may make your billing unpredictable.
2) Choose a hosted issue tracking vendor. There is a tremendous variety in this space, so ask for demonstrations and evaluate the tool first. Support for attachments and customizable issue types are typically important features to look for. Make sure there are no hidden fees triggered by the number of issues or the amount of data stored in the system. These metrics inhibit usage, which is precisely the opposite of what you want to accomplish.
3) If you have a distributed team, determine how you will share documents among team members. The simplest option is a hosted Wiki, which will allow your team to build an ad-hoc knowledge portal for your project. If the Wiki allows document attachments, it can also serve ad-hoc file-sharing needs.
4) If you do not have standard documents to capture project artifacts, download a template set such as the open source ReadySET (see "Further Reading" below). You can then choose to use the pieces that make sense for your project.

With these three simple steps, you can achieve a fairly complete project infrastructure without issuing an RFP or installing any software. Best of all, it can be done in a hurry – if you are flexible, you can have all of this ready for your team to use in under a day.

Conclusion

Software development best practices can save you money and make your team more productive. If you find that the traditional avenues to setting up project infrastructure take too long or cost more than you had hoped, hosted alternatives may be a better choice for your team. There are no longer any obstacles preventing teams from using a best-of-breed development platform. Hosted platforms provide a level of convenience and savings that make them the obvious choice for today's IT development, IT management, and consulting organizations.

Further Reading

CMMI Product Team. 2002. *Capability Maturity Model Integration, Version 1.1*. Pittsburgh: Carnegie Mellon University.

CVS Development Team. CVS Home Page. http://www.nongnu.org/cvs

Wells, Don. Extreme Programming: A Gentle Introduction. http://www.extremeprogramming.org.

McConnell, Steve. 1996. *Rapid Development*. Redmond: Microsoft Press.

ReadySET Development Team. ReadySET Home Page. http://readyset.tigris.org/

Rothman, Johanna. Achieving a Repeatable Process. http://www.jrothman.com/Papers/AchievingRepeatable.html

Subversion Development Team. Subversion Home Page. http://subversion.tigris.org/

