[redacted]

Project Summary Report

Build and Deployment Automation
29XS218 TO 07

December 2009 to January 2011

Rob Daly
Rob.daly@stelligent.com

January 7, 2011

Project Summary Report
Build and Deployment Automation
December 2009 to January 2011

1. Personnel Roster

Name	Title/Role	Contact Information
Duvall, Paul	Automation Architect	Paul.duvall@stelligent.com
Daly, Robert	Automation Architect	Rob.daly@stelligent.com
Saksa, Steve	Automation Engineer	Steven.saksa@stelligent.com
Bassett, Tim	BDA Engineer	Tim.bassett@stelligent.com
Nyika, Michael	BDA Engineer	Michael.nyika@stelligent.com
Narra, Mahi	BDA Engineer	Mahi.nara@stelligent.com

Insert additional rows as needed.

2. Technical Status Overview

Insert a narrative describing the current status of the project. This section goes beyond the month's activities to include all of the activities for the entire effort. Include high-level accomplishments, future directions, and major risks and issues.

2.1 Description of Work Completed During the Contract Period

Table 2.1: Technical Progress/Accomplishments

Activity	Description
Completed and refined the BDA 1.0 Framework.	The BDA framework provides consistency and certainty to development teams that their software is built in as automated a fashion, as they desire. IT is possible, with the framework, to build, test, document, and promote JADS-based software.
Completed, maintained, but eventually retired, a BDA Dashboard	The BDA Dashboard provided development management a single view of the state of each of the projects that were employing the BDA framework, or as we termed them, were BDA-ified. Sadly, the development management did not find sufficient value in having this knowledge on an automated and accurate tool, so it was retired.
BDA-ified over 40 projects, including ones mentioned in the statement of work, but also several others.	The BDA team helped each project team to ready their work products for BDA-ification (BDA compliance). The BDA was overwhelmed with the effort, and was forced to move to a self-serve model of BDA-ification.

Describe each discrete activity on which work was performed during the month. When the activity encompasses work on a discrete task (see below), explicitly identify the task and task number.

3. Issue Management

Table 3.1: Business/Process/Technical Issues Encountered, Resolved and open

ID	Issue	Type	Status	Resolution/Proposed Resolution
	See NOTE below			

Type = B (Business)/ P (Process)/ T (Technical)
Status=R (Resolved)/ O (Open)

NOTE: Please refer to the living Jira Project tracker for BDA issues. There are some 648 issues maintained in this toolset, and it represents all tasks, feature requests, and bugs.

https://tracker.nci.nih.gov/browse/BDA

4. Success stories

Table 4.1: Items that went well

ID	Items that Succeeded
001	Cancer Center installs were easier. • GUI was easier for users to use. • Troubleshooting was easier. Since scripts installed and configured container, given the properties a user used we could easily reproduce install/configure problems locally.
002	CI servers improve code quality. Automated deployments are required for automated functional testing.
003	Consistent deployment process for the three types of deployment (workstation, tier and external) helped teams find integration and deployment issues earlier.
004	Complex deployments were improved with Disposable Application Containers DAC, the two teams that went live to production with DAC, CTRP and NBIA, were very pleased with the results. They felt they got more consistent builds and faster deployments because of DAC.
005	Using the BDA templates speeds up integration in fully automated or self-service environments like caBIG Clinical Information Suite.
006	Matrixing or moving people across the projects is easier with BDA. Since the build/deploy processes and naming standards (properties, targets, etc.) are consistent users are up to speed quicker. Also it was easier for systems to support multiple projects because of the consistent processes and standards.
007	Good documentation – The team received many complements from end users about the amount of information available on the NCI BDA wiki. Our documentation included NCI specific processes and policies in addition to general BDA related activities. We have actually received

	emails from users outside of our organization asking about some of our content.
008	Good support – The team received many complements from teams about how quick and thorough our support of the project was.
009	Training – We offered online and in person training on many topics both in person and online.

5. Lessons learned/Future Improvements

Table 5.1: Lessons Learned/Suggestions for Future Enhancements/Improvements

ID	Lessons Learned/Enhancements/Improvements
001	There absolutely needs to be a dedicated and engaged product owner for any project.
002	As BDA transcends both Dev (app dev teams) and Ops (systems team), it is important that the product owner of the BDA (or CDO) have authority over both organizations to ensure execution of the vision and minimal finger pointing.
003	We needed to partner with the systems team on this solution – The systems team had no resources assigned to work with us on making the solution, only to support us when there were problems. We often offered to do in depth reviews of what we came up with but were never taken up on our suggestion.
004	With Automation the complexity of the container is moved from that actual container to the scripts and properties – Since the systems team manage the environment they need to own the environment related properties (and scripts). It never felt like we got this ownership established. The dev teams, which did not have the expertise or the access to the infrastructure, were considered the owners of the properties (and scripts).
005	BDA team should have been more involved with the deployments to tiers (If the systems team owned the properties we would not need to be involved, but that was not the case.) – This was the only way to ensure that deployments were more successful and that the project's usefulness was viewed better. We were not adequately staffed or prioritized on this task to get better results.
006	Properties are complex so we needed better property checking – Most of the issues we got from users were in getting the property values right for a tier. Though we had some property checks we did not have comprehensive set of syntax and relationship validation routines. We implemented this in BDA-1.7 but were not able to pilot with a project before we started to spin down the team.
007	Template model is too expensive to implement and maintain – Initially it appeared that templates provided users with a quick way to implement a consistent set of processes and naming standards but allowed them to custom as needed. However it was fairly expensive to integrate changes in the template back into the modified build scripts. As a solution the BDA team was planning to go to a "binary" build tool. Each project would manage configuration and property files that the binary tool would use to deploy their code. The BDA Team contract ran out before we could work on this project.
008	Build failures are not a big deal – With the BDA model build/deploy failures are not big deal. The whole process is "inexpensive" so if one gets a failure, fix the property and build again. There is

	a stigma at NCI that any failure is a huge deal. We needed to prove our "build failures are not a big deal" and get buy-in from key people at NCI. We would occasionally see multi-email message chains distributed to hundreds of users about a simple deployment failure, the resolution of which would be something as simple as a change to a couple of properties and then redeploy.
009	Participants should plan that the first build to a new tier is going to have lots of failures – As with above, since build failures are not a big deal it should not be a big deal that one might fail many times before getting a successful build on a new tier. Properties are complex and one is not going to get them all right for a new tier on the first pass. As one gets more familiar with the BDA process, errors on new tiers will be less frequent, but should still be anticipated. Once there is a working set of properties, the targets should be able to build forever without issue. As code changes and new properties are possibly added, failures may occur again on the first build to a tier with the new properties.
010	Customization of the templates was difficult – The templates include man features that a given project my not use and it was up to the dev team to prune that code from the templates. To address this the BDA team wrote a project start helper script that would automate this for them. They would set a couple of properties, run the script and get a customized set of templates to begin on that was near completion for them.
011	GUI Installer code (IzPack) is hard to maintain and takes to long in the install to fail –The BDA team was in the process of analyzing an alternative the tool we were using IZPack, but the contract is not being renewed and replacing IzPack was halted.

State here possible future implementation strategy, if different from the one used in the project

Enhancements – Software changes or additions; Improvements – Process/Communication/Structure changes.

6. Deliverables

Table 6.1: Status of Deliverables

Deliverable/Milestone	Delivery Date	Status
Project Management Plan	February 22, 2010	Complete
Monthly Status Report	February 1, 2010	Complete (to date)
Initial Project Meeting	February 2, 2010	Complete
Requirements Traceability Matrix	February 28, 2010	Complete https://ncisvn.nci.nih.gov/svn/automation/trunk/docs/requirements_traceability/bda_requirements_traceability_matrix.xlsx
Lessons Learned Report	December 27, 2010	This document. Complete https://ncisvn.nci.nih.gov/svn/automation/trunk/docs/project_summaries
Design Review	March 31, 2010	Complete
Individual Project(s) Deployment Review	March 31, 2010	Complete

Deliverables Readiness	September 30, 2010	Complete
System and Software Design	June 30, 2010	Complete https://ncisvn.nci.nih.gov/svn/automation/trunk/docs/system_and_software_design/bda_software_architecture_document.doc
BDA Software Code Base and API Documentation	Ongoing	Complete; Source can be found at https://ncisvn.nci.nih.gov/svn/automation/trunk/software
System and Software Test Report	End of each development iteration	Complete (and ongoing – framework is continuously tested on a Hudson server); best place to start: https://ncisvn.nci.nih.gov/svn/automation/trunk/docs/test/system_and_software_test_report.doc
Project Documentation	End of each development iteration	Complete https://ncisvn.nci.nih.gov/svn/automation/trunk/docs
BDA Install Guide	Ongoing	Complete; best place to start: https://wiki.nci.nih.gov/x/Ghmy
BDA User Guide	Ongoing	Complete; best place to start: https://wiki.nci.nih.gov/x/Ghmy
CONOPS Document	December 28, 2009	Complete: https://wiki.nci.nih.gov/x/Mg6y

Final status of each deliverable identified in the contract