



Accelerating the Delivery of Software Projects

IONA Technologies August 2006



IONA[®]

Making Software Work Together™

Executive Summary

There is a dirty secret in software development: it is a rare thing when software projects are finished on schedule and deliver on all their requirements. Statistics show an alarmingly high percentage of software projects are delivered late, meet only a fraction of their requirements or fail outright. Project managers know they must improve upon this problem and look for ways to build better systems faster with their limited resources.

In our 15+ years of company experience, IONA has found that issues related to system integration are the root cause of many project delays. Interfaces between systems are inadequately specified, poorly shared and the related problems are not discovered until integration testing – usually late in the development cycle when surprises are least welcome and change is most costly.

IONA has developed a Project Accelerator program that combines computer science principles, document management and ideas borrowed from manufacturing. Our program compresses software development schedules and improves the quality and completeness of delivered systems.

Our program focuses on three areas:

- Defining robust, machine-readable, standards-based interface specifications between systems
- Ensuring project information is shared efficiently between all stakeholders
- Providing an automated testing framework to enable integration testing much earlier in the development cycle than is otherwise possible.

Our program has proven itself at several client sites with dramatic cost savings and early delivery of business-critical systems.

Project Management Today

There was a time when system development happened in one place, on one machine and was mostly about crunching numbers. But those mainframe days are long gone. Today's IT shops develop systems with many technologies, spread across wide geographies that address all parts of a business. "You want it when?" is an old adage expressing how the demand for new systems and features always exceeds the ability to supply. So IT managers try to accelerate project delivery through:

- Parallel development – software development tasks can be divvied up across multiple teams to deliver projects sooner.
- Commercial Off-The-Shelf (COTS) solutions - buying rather than building systems in-house lets developers focus on application logic that is unique to or differentiates the business.
- New technologies - scripting languages, integrated development environments (IDEs) and application servers make developers more productive.



- Best practices - adopting appropriate development models (e.g. agile, iterative, spiral, waterfall, RAD, XP) and following architectural design philosophies (e.g. structured, three-tier, object-oriented, model-driven, service-oriented).

These initiatives raise the potential of IT organizations to produce more, but they also make projects more complex. Unless organizations can manage and mitigate this underlying complexity, they may simply take less time to deliver on the wrong requirements.

To truly accelerate project delivery an organization must evolve their process controls and quality assurance practices to match an increasingly complex environment.

Why Projects Run Late

What is clear is that multiple development teams are a given, that systems are developed in a modular fashion and that multiple technologies (both old and new) must work together. Business logic is no longer the sole or even the primary cause of delays; it is the complexity of integrating an architecture's many components that typically delay projects and impact the business.

While the measures described above have the *potential* to improve project delivery, they also have unintended side effects that mitigate their benefits and can even wreak havoc on integration.

- Separate development teams may not share information effectively and their interdependencies can result in serial development (gridlock even), removing the gains expected of parallel development.
- COTS packages are unknown quantities that must be analyzed and tested for compatibility with the deployment environment. They may embed older technologies or versions of technologies that are out of sync with enterprise standards.
- Technological solutions can create situations where systems cannot communicate effectively. Most IT folks recognize that technology never dies - you just get more of it. New systems must work together with incumbent technologies that may be decades old.
- Development models and architectural principles do not adequately address the disciplines needed for orderly system integration.

The issue is not with developing the components of systems, it is with the way those components are brought together. Three aspects of distributed development lead to issues during system integration: specifications, communication and testing.



Specifications: Most IT shops produce design documents that describe how system components will communicate with each other. These documents go by different names within each shop. We refer to them generically as Interface Design Documents (IDDs). IDD's are often very large documents spanning hundreds of pages. They are intended to capture the necessary shared knowledge between two or more organizations. Gaps and ambiguities lead to inaccurate development. Conversely, extraneous content also raises the shared-knowledge burden of the members of both organizations. The net result is lower productivity and delayed project delivery.

Communication: System stakeholders are often isolated from each other by time, distance and organization structure. Yet they must collaborate to determine how their components will interact with each other. Even high quality requirements are of little value if they are not communicated to those who need them. And small version changes in an IDD not communicated to the relevant stakeholders in a clear and timely manner can lead to disastrous development results downstream and major project delays.

Testing: There are scores of books, products and consultants focused on application testing. Yet there are very few resources devoted to *integration* testing. What we see all too often are projects that deliver robust, well-tested applications that still fail to work together. And this unpleasant fact is not discovered until very late in the project lifecycle. Component integration must be tested early and often, even when components may not be available to test against.

To summarize, we have found that ambiguous specifications, poor collaboration tools and late integration testing are primary causes of late system delivery. Addressing these primary causes enables us to accelerate project delivery.

Techniques that Accelerate Projects

To accelerate project delivery we must improve on the system integration process through improved specifications, improved team communication and early, automated integration testing. We do this through applied computer science, document management and techniques borrowed from manufacturing.

Improved Specifications

Through an IONA-developed QA methodology we remove gaps and ambiguity from IDD's that can lead to project delays. We render and decompose them into formats that can be easily and consistently consumed by software engineering tools for validation, certification and testing. Our approach leverages standards like Web Services Description Language (WSDL) and XML Schema Documents (XSDs) to create specifications that act as contractual agreements between components. The enhanced specifications can be consistently consumed and understood by both man and machine throughout the remainder of the lifecycle ensuring a consistently higher level of productivity and far less rework.



Improved Team Communication

Integration by its very nature is a collaborative process between development teams; collaboration that needs support throughout the development lifecycle. The auditing process is iterative. As missing information, ambiguities and other defects are found they must be logged, resolved and re-audited. Two teams developing software components that will be integrated must work from the same version of an integration contract. To support auditing, integration testing and team collaboration we implement tailored bug tracking and document versioning systems. We also provide project management with a dashboard view into the overall process. The integration dashboard makes it easy to spot problems, monitor overall project status and determine what must be done to ensure on-time system delivery.

Early, Automated Integration Testing

Before the 1980's, quality control in manufacturing was based on testing finished goods. W. Edwards Deming revolutionized manufacturing processes by advocating (among other things) testing for quality throughout the production cycle, even up the supply chain. The result was much higher quality, less down time, faster production time and reduced production costs. Our approach protects project timelines by enabling automated integration testing earlier in the system development lifecycle. Integration contracts enable development tools to generate interface certification kits that provide all the support needed to test client or server applications. Certification kits simulate the missing client or server side of an interface, making it possible to test system interfaces well before all the components are ready. Automation accelerates system and end-to-end test phases to reduce inter-dependencies and the manual processes that prevent defects from being discovered earlier.

Case Study

As mentioned before, our approach to project acceleration has been honed through years of experience and application at our customer sites. One recent customer had a lot of experience in distributed system development and had an established SOA development process. Even so their specifications for messaging and application interfaces left a lot of room for interpretation; they did not have a discipline for sharing and updating shared information and they had no way of testing for compliance with their specification documents.

IONA instituted our design document audit process to refine their IDD's with some surprising results. Of the 112 IDD's audited, all had been signed off for development, *yet only 3 were completely clean*. The audit discovered 532 defects, 330 severe enough to result in Integration Failures. Only 11 IDD's were free of *severe* defects.

Once these defects and ambiguities were removed, the IDD's were converted into unambiguous, machine-readable XML Schema Documents (XSDs) and WSDL that acted as integration contracts. The integration contracts drove the generation of Certification Kits hosted in a centrally deployed Universal Test Harness that tests application integration with client and server simulators.



The results exceeded expectations. Our customer was able to reduce the number of full-time employees dedicated to services testing by 75%, enabling those resources to be allocated to more productive use. By establishing a central authority to implement and manage our program, our customer was able to reduce development and QA costs dramatically – about US\$15 million over three years. The 3-year savings for QA alone totaled more than their annual QA budget.

But most importantly, they are delivering systems sooner. They estimate they can exploit significantly more new business opportunities than before. The development lifecycle for major releases was reduced 2-3 weeks. Our customer estimates that getting to market sooner with new services will have a 3-year top-line revenue impact of about US\$50M.

Conclusion

Projects can be accelerated by ensuring that system integration proceeds smoothly. Our Project Accelerator program eliminates integration surprises and expensive rework late in the development cycle. This reduces time and cost, brings new services online sooner and enables developers to put more functionality into those services.

IONA Technologies PLC
The IONA Building
Shelbourne Road
Dublin 4
Ireland
Phone +353 1 637 2000
Fax +353 1 637 2888
Support: support@iona.com
WWW: www.iona.com

IONA Technologies Inc.
200 West Street
Waltham MA 02451
USA
Phone +1 781 902 8000
Fax +1 781 902 8001
Training: training@iona.com

IONA Technologies Japan Ltd
Akasaka Sanhome Building 7/F
3-21-16 Akasaka
Minato-ku Tokyo
Japan
Phone +813 3560 5611
Fax +813 3560 5612
Sales: sales@iona.com

Copyright

Artix, Artix Mainframe, Adaptive Runtime Technology are Trademarks of IONA Technologies PLC. Orbix, Orbix 2000 Notification, and Orbix/E are Registered Trademarks of IONA Technologies PLC.

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

COPYRIGHT NOTICE. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photo-copying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. IONA Technologies PLC assumes no responsibility for errors or omissions contained in this white paper. This publication and features described herein are subject to change without notice.

Copyright © 1999-2005 IONA Technologies PLC. All rights reserved.

All products or services mentioned in this white paper are covered by the trademarks, service marks, or product names as designated by the companies that market those products. 03232KB01



Making Software Work Together™