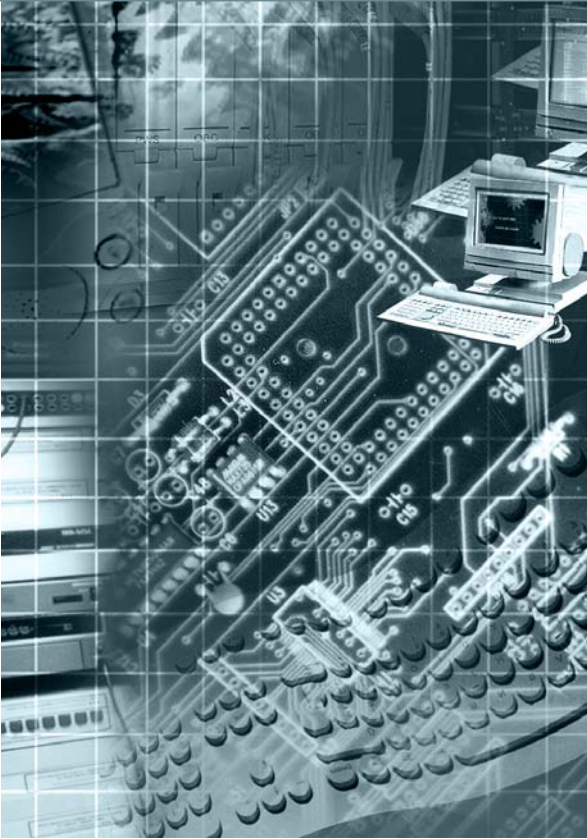


## APPLYING TEST AUTOMATION IN PRODUCT DEVELOPMENT



### Situation

Celstream was the co-development partner for a new software framework product being released by a leading broadcast equipment manufacturer. As the hardware to test the product before release was diverse, it was not practical and financially viable to make them available in Celstream; nor were they easily accessible over the Net. To enable successful delivery, within the time and resource constraints, an alternative test strategy was required. The product rollout plan, with incremental features being released every month, lent itself to Test Automation.

### Challenge

The product design did not facilitate easy Test Automation. Off-the-shelf Test Automation tools were too generic and not easily customizable for this particular situation. Further, they were also expensive. There was also an urgent need to develop the necessary hardware simulators to facilitate product testing, and to automate reliability testing for reducing cycle time.

- ❑ Reduced cycle-time by automating test execution
- ❑ Saved development cost with hardware simulators
- ❑ Tools developed were leveraged by the customer for other projects

## APPLYING TEST AUTOMATION IN PRODUCT DEVELOPMENT



### Celstream Response

In order to fulfill the test requirements of this project, Celstream set up a team of five engineers with development experience to design an automated test tool.

The team first assessed the customer's product qualification strategy. Leveraging their background in product development and test automation, the team explored a different approach to testing and developed prototypes of end-user hardware simulators. These prototypes were implemented using state-of-the-art technologies such as C#, .NET, and XML in such a manner that they could be used across various Operating Systems making them very interoperable.

The team developed a custom test automation tool utilizing these simulators in less than three months. The Celstream team also designed a tool for sustainability. The effort was so successful in saving the customer time, effort and eventual costs that the customer began using the tools developed for their internal use.

### Technology

In developing this solution, Celstream used the following key software platforms and technologies:

- XML™
- Java™
- XSL
- Rational Rose®
- C++
- KQML
- Rational Purify®

### Significant Gains From 'Design For Testability'

The hardware needed to test the customer product's reliability over time was unavailable. Manual testing would have taken a month and the execution time would have taken twenty days for one cycle. Celstream's engineers wrote software simulators to behave as hardware by employing a technology called 'Design for Testability', and successfully finished the testing in six days with six hours of execution effort. This allowed for another round of testing if required.

Celstream's ability to develop test automation helped in reducing the testing cycle time, testing errors and time-to-market cycles, thereby improving the product quality, saving on valuable human, hardware & software resources and improving profitability.

[www.celstream.com](http://www.celstream.com)

Copyright© 2005, Celstream Technologies Private Limited. All rights reserved. Printed in India. The information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. For the most up-to-date information please visit our website at [www.celstream.com](http://www.celstream.com).

Celstream is a trademark of Celstream Technologies Private Limited. XML is a trademark of MIT, INRIA or Keio on behalf of the World Wide Web. Java is a trademark of Sun Microsystems Inc. Rational Rose and Rational Purify are trademarks of Rational Software Corporation.

All prices, specifications and characteristics set forth in this publication are subject to change without notice.



*Extending Your Engineering Team*

