ECLIPSE review

Forget the Crystal Ball
Manage Dependencies For Better Software Quality

PHONE HOME!
eRCP Puts Eclipse On Mobile Devices

'Hello, World!' Eclipse RCP 101

Best Practices With the Eclipse Process Framework

Announcing The Third Annual

Software Test & Performance CONFERENCE

November 7-9, 2006

The Hyatt Regency

Cambridge, MA

The Software Test & Performance Conference provides technical education for test/QA managers, software developers, software development managers and senior testers.

The wall between test and development has fallen, and the Software Test & Performance Conference brings the whole application life cycle together!

BUT DON'T TAKE <u>OUR</u> WORD FOR IT...

"This is the best conference I have attended. The instructors were extremely knowledgeable and helped me look at testing in a new way."
—*Ann Schwerin, QA Analyst*
Sunrise Senior Living

For Sponsorship and Exhibitor Information Contact David Karp at 631-421-4158 x102 or dkarp@bzmedia.com

Diamond Sponsor
SQS SOFTWARE QUALITY SOLUTIONS

Platinum Sponsors
Google
worksoft
ENERJY

Gold Sponsors
Empirix
LogiGear
McCabe SOFTWARE
OUNCE LABS
PARASOFT
SQA

Silver Sponsor
BEYONDSOFT
COMPUWARE
instantiations
Mindreef
SHUNRA

Media Sponsors
Software Test & Performance
ECLIPSE review
Report
SD Times
Software
Methods & Tools
queue
BETTER SOFTWARE MAGAZINE

ECLIPSE review

Vol. 1 ■ No. 3 ■ SUMMER 2006

Cover Illustration:
"Golden Eye," by José Manuel

FEATURES

"Eclipse is more than a development environment—it's an application life-cycle platform."

—Peter Varhol, "Leaving the Crystal Ball Behind."

DEPARTMENTS

Join Us In 2007!

eclipse WORLD
The Enterprise Development Conference

Hyatt Regency Reston
Reston, VA

November 6-8, 2007

Washington, D.C. Area!

Leveling the Playing Field

The term *integrated development environment* **means many things to many people. Most IDEs target a single** platform, an approach that many developers are averse to. Others target a specific language. Few, however, have the flexibility and features of the Eclipse IDE, an environment that can be "integrated" into nearly any development style and provide the development features that virtually all users agree are important in building applications for just about any platform.

On its own, the Eclipse IDE is a powerful development tool for a single reason: It was and will always be defined by programmers. It truly lives up to the Eclipse Foundation's own definition: The IDE is a "vendor-neutral open development platform and application framework for building software." If a certain feature is deemed necessary by the newsgroups, chances are good that interested companies will gather and put together a proposal to implement it.

The process of moving a project from concept to release is well organized. Anyone within the Eclipse community—including you and I—can present a proposal.

After a proposal is submitted, the Foundation's member companies have an opportunity to bring to light any part of the proposal that may infringe on their intellectual property. Following that phase, the proposal is checked to be sure that it doesn't conflict with the Eclipse organization's purposes and road map. A great deal of care and planning goes into each project from proposal to release to ensure that the membership, from the developers to the reviewers of each project, work together in a cohesive manner— an almost unheard-of occurrence in the world of open-source software.

COMPETITIVE COLLABORATION

The story behind Eclipse's unique and collaborative development approach is just as interesting. Why do software companies, many of whom are direct competitors, work together to build and evolve Eclipse? It's not an act of charity: Nobody believes that any of these companies are driven by pure altruism. The real reason? Eclipse helps companies like IBM, BEA, SAP, Compuware, Intel, HP and Borland bring their own products to market faster.

These and other companies—large and small—devote their time, effort and resources to Eclipse because it saves them from developing similar technology by themselves (which would be expensive) or from licensing it from someone else (who would then control the license). It's cheaper, faster and more efficient to work together on certain common functionality so that each company can spend time and resources working on the unique and more competitive aspects of their own products. Participating in Eclipse shortens their development cycle and focuses competition where it counts the most: individual product value.

Does this competitive collaboration benefit the entire open-source community? Certainly. The free downloads benefit anyone programming in any language that Eclipse supports. However, a large part of the IDE's value goes to those who are selling products. It also opens the door for third parties to develop plug-ins for specific niche markets. Another side effect is that it makes products from individual companies work well together, which leads to a truly integrated group of commercial products.

This month's issue of Eclipse Review offers a microcosm of the type of effort that goes on when large companies and individuals put aside their competitive nature to level the playing field.

IBM's Per Kroll and Kurt Sand of Telelogic worked together on an article about the Eclipse Process Framework ("A Development Library at your Fingertips," p. 25). IBM's Chris Aniszczyk and Mark Rogalski collaborated with Gorkem Ercan of Nokia on an article about the embedded Rich Client Platform ("Eclipse on Cell Phones?" p. 18). Representatives of other companies, as well as our stellar group of writers from the Eclipse community at large, are also represented. All in all, it's easy to see the type of collaboration that's developed as Eclipse has evolved.

When software companies—overseen by other interested parties— work together to lower their bottom line, they can focus on individual efforts. Rather than looking at the commercial aspects of co-development, take advantage of these companies' collective strides. The benefits of this collaboration reach far beyond the individual companies involved; they enrich the entire community— that is, all of us.

—George Walsh, Executive Editor

MyEclipse Tool Suite Focuses On Web Services, Callisto

The latest edition of Genuitec's MyEclipse software—comprising not only the latest "Callisto" version of Eclipse, but a whole slew of integrated packages and components—adds extensive support for Web services. MyEclipse 5.0, which came out the first week in August, has a new Web services project and project creation wizard, customizable library containers and tools for deploying Web services applications. It also includes editing capabilities for XML files, Java code and WSDL, and a Web services explorer. To help developers hit the ground running, Genuitec also includes quick-start documentation and sample projects that show how to hook apps into the MyEclipse database and how to use Hibernate and Spring for rapid application design. MyEclipse 5.0 costs US$29 for a one-year subscription with support. **www.genuitec.com**

WindowBuilder Pro Adds GroupLayout for SWT and Swing

Instantiations has released version 5.0 of WindowBuilder Pro, its Eclipse-based tool for creating graphical user interfaces for Linux, Mac OS X and Windows applications. According to the company, a key new feature offers "Better than Matisse" GroupLayout capabilities for both the SWT and Swing Java GUI frameworks. (Matisse is the Swing-only GUI framework that Sun Microsystems includes with its NetBeans IDE.) By creating a layout technology that works with both Swing and SWT, Instantiations lets developers create forms, position and size widgets, and develop user interfaces using any Java layout manager taking advantage of GroupLayout constraints. The software, which costs US$299, includes both SWT and Swing design tools. The update of WindowBuilder Pro can now also create and edit Eclipse RCP ActionBar Advisors, can configure the UI palette with a new Palette Manager, and has enhanced Swing & SWT absolute layout support. **www.instantiations.com**

SlickEdit Plug-In Lets Editor Work Inside Eclipse-Based IDEs

SlickEdit has released a plug-in that lets developers use its eponymous popular code edits as the default editor within Eclipse or any Eclipse-based IDE. The SlickEdit Plug-in provides a wide range of features for analyzing, working with and improving the readability of code. Seamlessly integrating with the project and workspace management capabilities, the SlickEdit Plug-in complies with the Eclipse project data model, debug and build framework. The SlickEdit editor's features include automatic syntax expansion and indenting to insert entire constructs and place the cursor at the correct line and indent level. A feature called Surround With can select a code block and surround it with language-specific structures. DIFFzilla provides editable dynamic differencing and merging for files, directories and source trees. Other features for rapid coding include aliases, code beautifiers, color coding, clipboard inheritance and smart pasting. Pricing for the SlickEdit Plug-In, which includes the SlickEdit editor, is US$199 for electronic delivery for Linux or Windows, or $229 for physical delivery of a CD-ROM and documentation. **www.slickedit.com**

Wolfram Workbench Brings The Power of Eclipse to Mathematica

Mathematica is a powerful tool for calculation, programming and documentation math—ranging from programmable numeric and symbolic calculations to high-end 2D and 3D graphics to data visualization. The makers of Mathematica, Wolfram Research, have now brought out Wolfram Workbench, an Eclipse-based IDE customized and specialized for Mathematica. It provides project management to group files, code and other Mathematica resources into a single component. It supports Mathematica code and notebooks, Java source and classes for J/Link, DatabaseLink and GUI widgets. It also has code editing and debugging capabilities, so you can study code as it runs, detect and fix any problems, set breakpoints to halt and evaluate variables, and track problems with useful error messages. The IDE supports profiling and version control. Wolfram Workbench is available for download for the company's Premiere Service customers. **www.wolfram.com**

Short Takes

Codign Software has released CoView 2.0, a US$350 Eclipse plug-in that automatically creates JUnit tests for developers and reports relevant path and branch coverage results to managers. Version 2 includes a new user interface, new test manager and JUnit editor, and the ability to measure path and branch coverage. **www.codign.com**

Wirelexsoft unveiled VistaMax Visual Express for S60 2.1, an Eclipse-based rapid application development environment for C++ that targets Symbian OS, Linux and J2ME. The US$50 visual development environment includes about 60 UI elements that can be assembled using drag and drop. The tool has plug-ins for integration with Code Warrior and Nokia's Carbide.C++ Express environments as well as CVS and ClearCase. **www.wirelexsoft.com**

Callisto Arrives On Schedule

The big event for the Eclipse community was the release of 10 different projects on June 30; an event we called the Callisto release. The projects included in Callisto were the Business Intelligence and Reporting Tools (BIRT) Project, C/C++ IDE (CDT), Data Tools Platform (DTP), Eclipse Modeling Framework (EMF), Graphical Editing Framework (GEF), Graphical Modeling Framework (GMF), Eclipse Platform (SDK), Eclipse Test and Performance Tools Platform Project (TPTP), Eclipse Web Tools Platform Project (WTP) and Visual Editor (VE).

This was a remarkable community effort that involved 10 project teams, 23 subprojects, 260 committers located in 12 different countries from 15 different ISVs, and shipping over 7 million lines of code on the same day. Congratulations to all of the individuals involved.

We hope you've enjoyed downloading the new releases and checking out the new features in each of the projects. The planning now starts for the next year's release, tentatively called Europa (another moon of Jupiter).

ECLIPSE ANNOUNCEMENTS
We're pleased that Motorola has joined the Eclipse Foundation as a Strategic Developer and will join the Eclipse Board of Directors. Motorola has also proposed a new Eclipse project to build tools and frameworks for Mobile Linux.

We're also pleased to announce Eclipse Summit Europe, the Foundation's inaugural European conference for the Eclipse community. The two-day event will be held in Esslingen, Germany October 11–12. The entire Eclipse community is invited to participate in an Eclipse membership meeting, Eclipse project talks and technical symposia. For more details, go to www.eclipsecon.org/summiteurope2006.

NEW COMMUNITY PORTAL
The Eclipse Foundation has launched a community portal to promote the adoption of Eclipse-based commercial products, complementary open-source projects and Eclipse-related training, support and services. The portal, Eclipse Plug-in Central (EPIC), makes it easier than ever for developers, software vendors and enterprises to find high-quality open-source and commercial plug-ins, tools and add-on services to enhance their Eclipse development. The EPIC portal will be a key part of the Eclipse Foundation's effort to promote and grow the overall Eclipse ecosystem. For more information about EPIC, go to www.eclipseplugincentral.com.

Here is a summary of project reviews that have arisen so far in 2006. We hope to make this a regular feature of this column. It is based on the RSS feed at www.eclipse.org/projects/reviews-rss.php.

Table 1 | Project Updates

PROJECT	REVIEW DATE
Rich Server Platform Created and Provisions	May 5
The Eclipse on Linux (Linux Distro) Project Proposal Posted	May 5
Modeling Top-Level Project Created and Provisioned	May 10
Dali Move Release Approved	May 24
Callisto Simultaneous Release Pending	May 26
UML2 2.0 Release Approved	June 7
JSF 0.5 Release Pending	June 21
Enterprise Component Framework Waiting for PMC Decision	June 21
Dali 0.5 Release Pending	June 21
Tools for Mobile Linux Proposal Posted	June 22
Aperi Storage Management Project Proposal Posted	June 28
Google Summer of Code Project Approved	June 28
Rich Ajax Platform (RAP) Project Approved	June 28
The Subversive Project Proposal Posted	June 29
Model-to-Model Transformation Proposal Posted	July 7
The Eclipse Orbit Project Proposal Posted	July 10
DTP Enablement Project Approved	July 12
Embedded Rich Client Platform Move Approved	July 19
EMC (Apogee) Creation Review	July 25
ALF 1.0 Release Review	July 25

When the Community Wins, All Eclipse Users Win

In every field of endeavor, there are those who are considered stars and those who work behind the scenes in an effort to help achieve the overall goal at hand. Within the Eclipse community, those stars are the gurus who put immense time and effort into becoming experts on topics that benefit everyone. These few may be thanked by their peers on a daily basis, but isn't it nice to receive real recognition? That's why the Eclipse Foundation launched the Eclipse Community Awards, which kicked off earlier this year at EclipseCon in Santa Clara.

These awards honor individuals who have helped further the reach of the Eclipse community and extend the ecosystem. They recognize the people who have volunteered their time and resources in making Eclipse a vibrant and innovative community. The winners were selected by their peers in an online vote, making the honor yet more personal and, for those who didn't attend the conference, we honor them here as well. The categories this year were Top Contributor, Top Committer and Top Ambassador.

ED BURNETTE, TOP AMBASSADOR

The Top Ambassador award recognizes an individual who best promotes cross-project interactions and building a collaborative Eclipse developer community. The ambassador's work spans the gap between and across Eclipse projects, user groups or plug-in developers. This year's winner was Ed Burnette, a principal systems developer for SAS.

Burnette encourages people to give peer-to-peer support because, even though he's considered a true power user, he understands that there's no way that any single person can answer every question related to Eclipse. He's made an effort to foster the Eclipse community and introduce people to one another for support and advice. Often, he's the one giving the advice.

Some 90 percent of SAS's Java developers use Eclipse in the workplace, estimates Burnette, and his own professional experience with Eclipse includes writing internal plug-ins that adapt the IDE for the company's own internal procedures. For example, using the Rich Client Platform, he wrote an application for testing his group's software products.

"It was a way for us to get a lot of functionality in a little bit of time," he explains. "If I had tried to write it from scratch, it wouldn't have been half as elegant. Internal testing applications are often just one-off. You get them to where they work and then you move on to something else. Generally, they're not all that great, but Eclipse is just a really nice tool that makes creating apps like this much easier."

Burnette was somewhat surprised by the recognition as Top Ambassador. "What I've done is nothing very special and nothing that I can take credit for," he says. "I've been very active on newsgroups and Bugzilla. I think the award was in recognition of that. The Foundation is very much in the mode of trying to enable projects to make progress. They're not trying to make the progress themselves. The progress is typically achieved by the users and developers."

When you look at Burnette's resumé, his qualifications belie his humble nature. He's programmed everything from multi-user servers to compilers and commercial video games. He's also has written several articles and books on Eclipse, most recently the "Eclipse IDE Pocket Guide" (O'Reilly, 2005). In addition, he writes the Dev Connection (blogs.zdnet.com/Burnette) Web site. Burnette is a prime example of the cooperation and collaboration among Eclipse users throughout the world.

ALAIN MAGLOIRE, TOP COMMITTER

The Top Committer award recognizes exemplary support for the community through newsgroups, Bugzilla, white papers, conference presentations, blogs and other formats, expanding beyond a specific project into the larger Eclipse community. Alain Magloire, a systems engineer for QNX Software Systems, was honored in this category.

Magloire believes that his award was primarily in recognition of the amount of work that he did in bring-

ing the QNX version of the IDE up, going back to 2001. "I was nominated simply because I must have the highest number of committed lines of code and as the one that put in the most effort getting the first version of CDT working. I was involved in every component of the CDT," he says.

Back in 2001, QNX was looking for a new IDE to enhance its tool set, and IBM was pushing VisualAge Micro Edition as a base platform on which to build tools. A group of engineers from QNX was dispatched to see how viable it could be to have a Java IDE in the first place. "That's a story in itself," Magloire says. "Imagine a bunch of hardcore embedded C programmers sent to do Java development when even the mention of C++ made them cringe."

Much of this effort led to the C/C++ Development Tools project, and it was Magloire who served as the technical "mentor."

"It took hard work and lots of fighting to get the message across that Eclipse was a viable option," he recalls. "Don't forget that back in 2001, Java was a hard sell to any embedded C shops. We, however, had a great team with open minds. Within QNX, I may symbolize the CDT, but CDT is a collaborative process, where a team is responsible for a component. We have design meetings and brainstorming in conference calls. Somehow, all the teams manage to post a design plan. Then the community sends feedback after the cycle dates are posted, and we work hard to be on time."

In addition to the amount of code that he and his team have produced, Magloire believes that part of the reason that QNX can make an impact is that it's familiar with the approval process that an Eclipse project undergoes. They have a customer agreement that contains the requirements and serves to produce other documents relating to what is possible within a given time frame. Once the design cycle is over, development is started with a team of programmers, the documentation people and the testers. At a different stage, QNX has

milestones and the usual uneasy relations with the testers, which goes on until beta/GA. At a different stage, customers give input and can follow the process.

As with the winners of all of the Eclipse Community awards, Magloire's humility is impressive. He is extremely pleased to see that other companies are joining the CDT effort and providing new directions, new visions and new ideas to the project. "I share the award with the QNX folks and the CDT community for the long nights coding, design and arguing," he says. "However, I will keep the T-shirt and the Nokia N90 phone I received in recognition."

LINDA WATSON, TOP CONTRIBUTOR

The Top Contributor award is bestowed on an individual whose contributions flow into Eclipse projects. The award can also apply to a part-time committer who has made substantial contributions in a non-committer role. The winner in this category was IBM's Linda Watson, visual user-interface design lead for Eclipse, who reports to John Kellerman, IBM's chief Eclipse guru (see Perspectives on page 38).

In the case of Eclipse, the area of visual design goes beyond simply building pretty interfaces. It includes the Workbench, the UI for the Workbench and anywhere a visual element can be linked to usability and ease of task completion. Though, as an open-source IDE, Eclipse is only five years old, Watson has been heavily involved with it for the past 10 years, since its infancy as IBM's internal VisualAge for Java.

"When Eclipse was in its first release, I worked with the developers, and the architecture was fairly set, but I worked with them to get the appearance cleaned up," she says. "At that point, I think our mandate was to be a good Windows citizen, but now we're really looking good as a cross-platform development tool. If you look at the views and perspectives, the color, the way border treat-

ments are placed, the way that tabs are designed and the appearance, our team has probably consulted on it. From the get-go, the appearance of the user interface is where we've contributed, including fixing problems that users have identified."

In addition to the user interface, Watson and her team have worked on developing Eclipse's identity. As most people who use the IDE know, it doesn't look like any other environment. The icons, the splash screens and the things that you see in the task bar are all components that they have developed, paying close attention to the consistency of appearance of the entire set. If there's a particular piece that has priority for a certain release, they make it obvious and appealing to users. One example from the last release was to enhance the Preferences dialog to make it easier to use.

"If you took away the visual elements of Eclipse, all you'd have would be a blank white space and a command line. We're here to think about everything you see," Watson says. "Any design issues that we work on have to be validated cross-platform in terms of design and technical implementation. In addition to the identity aspects and the Workbench UI, we work on the icons and wizard graphics. We also support any of the other plug-in areas like JDT and PDE that require specific visual design support."

While Watson is certainly pleased with what she and her team have accomplished, she too exemplifies the same modesty shown by the other Community Award winners. "The thing I'm most proud of in winning the award is not the honor itself but that it represents the teamwork that I've been part of. It's truly an amazing experience to see the developers and designers and usability people all working together. I've had the rare opportunity to see the evolution and growth of Eclipse from the very beginning. It's become successful through the contributions of so many people. I'm very proud to have been a part of that success."

—*George Walsh, Executive Editor*

The JavaServer Faces Tools Project

JavaServer Faces (JSF) is one of the new standards in the Java EE 5.0 release that has become one of the most prominent tools for building Java EE Web applications. This Eclipse undertaking, known as the JSF Tools Project, provides comprehensive tools to support the development of Web applications.

Extending the Eclipse Web Tools Platform Project with tools that simplify JSF application development, the project aims to bring to JSF application development the same ease of use that Eclipse projects offer. True to the Eclipse charter of providing frameworks that allow tool vendors to extend the IDE, the JSF Tools Project has contributed a core set of frameworks that allow adopters to enable and extend its features.

In concert with the WTP 1.5 release, the JSF Tools Project has released a technology preview that supports the JavaServer Faces 1.1 specification, and support for JSF 1.2 is planned for the upcoming release of WTP 2.0 in June 2007. In this article, I'll provide an overview of the features in this release and walk you through the process of building and deploying a JSF application using the project's tools and frameworks. So let's get started on our guided tour.

THE FACES CONFIGURATION EDITOR

One of the new artifacts introduced by JSF technology is the application

Raghu Srinivasan is a software development manager in the JDeveloper Group at Oracle, where he is responsible for Eclipse projects. He's the project lead of the JSF Tools Project and a member of the WTP PMC.

configuration resource file, typically called faces-config.xml. In this file, you define the managed beans, validators, converters and navigational rules of your Web application. The JSF Tools Project provides a multi-page editor that simplifies creation and maintenance of the resource file.

Figure 1 | Setting the value of the "from" outcome to log-in
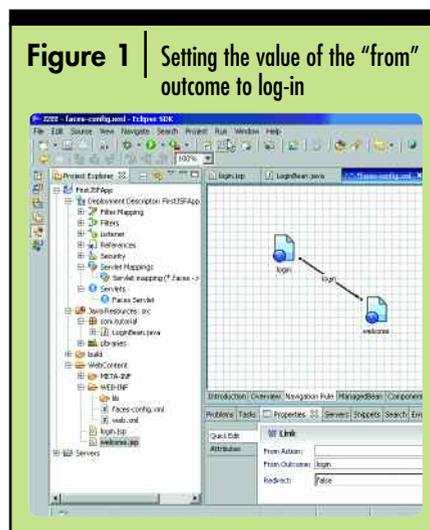
This editor includes the WTP XML Source Editor as well as multiple form-based editors for managing other artifacts in the file. The Source page content is kept synchronized with changes in other pages and vice versa for a seamless editing experience. The editor's Overview page summarizes the information in the

configuration file, and its Navigation page provides a graphical representation of the navigation rules, depicting the linkage of pages and actions in the navigation rule element.

The editor supports drag-and-drop of existing pages from the Project Explorer view to the editor canvas, and features a fly-out tool palette that includes all the operations that can be performed in the Navigation page. The Managed Beans page is a form-based editor that consists of a tree view to display all the managed beans in the file and options to create and manage them. The rest of the pages in the editor provide similar interfaces for other elements in the resource file.

The project has also contributed an EMF model of the application configuration resource file and a framework called the JSF Application Configuration Manager. The framework provides a unified application configuration model through the API, which effectively merges all of the application's faces configuration models. The framework also provides notification services to monitor changes to the EMF objects. These services enable the addition of new functions such as support for cross-model validation of the various artifacts defined in the resource file.

THE JSF–JSP PAGE SOURCE EDITOR

The JSF specification defines JSP as the default page-description language. To be added to a JSP page, JSF components use corresponding JSP tags. The WTP project provides a feature-

rich source editor for editing a JSP page (see "Develop Web Applications Faster With WTP" by Tim Wagner, in our premiere issue). Keeping in mind the features that Eclipse users have come to expect in editors, including code completion and as-you-type syntax and semantic validations, the JSF Tools Project has enhanced the JSP Editor with features applicable to a JSF–JSP page. The project provides these features out-of-the-box for the standard JSF Core and HTML tags. The features are available for both an Expression Language (EL) and a non-EL attribute value. A future release will include a graphical JSF–JSP page designer.

CONTENT ASSIST

In the JSP Editor, content assist or code completion helps you define a JSP tag by providing proposals based on the editor's current insertion point. The WTP JSP Source Editor provides support for adding a tag by offering a list of possible tags, with a list of attributes for each. The project takes this feature further by providing proposals for the values of tag attributes where applicable.

For an example, let's look at an attribute that has a discrete set of values. The messages tag in the JSF HTML taglib has the layout attribute that can be the literal string, list, table or a JSP expression:

```
<h:messages layout="" ></h:messages>
```

The editor will offer these choices in a pop-up when the cursor is positioned after the layout attribute.

The editor takes advantage of the services provided by the framework, JSFAppConfigManager, to provide proposals for attributes whose value can take one of the list of elements defined in the application configuration resource files. For example, the

validator tag in the JSF Core taglib has the attribute validatorId, which can be one of the standard validators defined in the JSF specification or the set of validators registered in all the application configuration resource files in the Web application:

```
<f:validator
 validatorId=""></f:validator>
```

The editor uses the API defined in the framework to provide a pop-up list of all the validators registered for the application.

Content assist is available for values that take an EL expression. One frequent scenario is to bind the value of an attribute of a JSF tag to a property in a managed bean registered in the application configuration resource. The JSF HTML tag

Table 1	Components required for building a two-page JSF application
Java 2 Platform Standard Edition (J2SE) 1.4.2	http://java.sun.com/j2se/1.4.2/download.html
WTP 1.5 and prerequisites	1)www.eclipse.org/webtools/releases/1.5 2)Choose Download WTP 1.5 3)Download and extract WTP all-in-one zip
JavaServer Faces Tools Release 0.5	1)Download and install JSF SDK or non-SDK build from bottom of same page as WTP above
Apache Tomcat 5.5	http://tomcat.apache.org
JavaServer Faces RI v1.1.01	http://java.sun.com/j2ee/javaserverfaces/download.html
JavaServer Pages Standard Tag Library	http://jakarta.apache.org/builds/jakarta-taglibs/releases/standard

inputText has the attribute value, which is typically bound to the property of a managed bean:

```
<h:inputText
 value="#{loginBean.name}">
</h:inputText>
```

For the value attribute, the editor creates a list of proposals that include all the managed beans defined in the Web application. For a specific bean, it lists all the properties defined in that bean. The editor allows you to drill down into successive levels in the case of nested properties and pro-

vides similar support for values of attributes that are bound to methods defined in a managed bean.

VALIDATION

The JSP Source Editor now supports semantic validation for both EL and non-EL values of attributes of the JSF standard core and HTML tags. This significant feature greatly improves a page author's productivity. The editor performs the validation as you type and reports errors, relieving the user of the time-consuming task of deploying and executing the page to find the errors. For non-EL values, the editor validates values of attributes that take a definite list of values, such as the layout attribute of the messages tag. Attributes whose valid values are the set of elements of a particular type defined in the application configuration resource files, like the validatorId attribute of the validator tag, are also checked.

Sophisticated frameworks enable comprehensive validation of the EL values. The editor will flag basic syntax errors such as empty EL expressions ("#{}"), incomplete EL expressions ("#{var >}") and missing closing braces ("#{"}). More advanced validation of EL expressions includes checking for undefined variables. The presence of properties in registered managed beans and messages in resource bundles is also checked. When an attribute value requires a method binding, the method signature is also validated.

JSF LIBRARIES

The JSF Tools Project includes a registry that helps to organize a group of JARs as a named collection called a JSF Library. These JARs typically include the JSF runtime implementation, component tag libraries and utility jars. This named collection can be referenced from a project. The JARs

in the library are added to the build classpath and are optionally deployed to the server. The JSF Library feature is similar to the JDT User Library, but can control its deployment to the server and can be created through extension points.

FRAMEWORKS

These features, implemented using the JSF Tools Project frameworks, are designed to be extensible so that adopters can add or extend them. They can be used to add support for other component libraries, enhance existing tool features or add new features or services. Here's a brief overview:

Content Model Annotation Framework: Allows an adopter to define additional metadata for elements and attributes on any XML-based content model for use at design time. This allows a plug-in developer to provide the metadata to support tooling for other tag libraries such as the MyFaces Tomahawk components. A tutorial accompanying the developer's guide demonstrates how this can be accomplished.

Design-time Metadata Processing Framework: Uses the annotations framework to provide services, such as content assist and validation, to the JSF–JSP Source Editor. It allows for defining new services, such as Quick Fix and Hyperlink, or extending existing services.

Design-time Application Manager: Provides an approximation of certain JSF runtime state information at design time, and allows for a pluggable design-time variable, property and method resolvers.

JSF Application Configuration Manager: Offers a unified application configuration model via the API, effectively merging all faces configuration models into a single model. The framework also includes notification services to monitor changes in the model. Cross-model validations and additional application-wide functionalities will be supported due to the inclusion of

> Frameworks can be used to add support for other component libraries, enhance existing tool features or add new features or services.

this framework.

Let's explore some of these features in action by building a simple two-page JSF application. The first step? Download and install the software list shown in Table 1.

REGISTERING APACHE TOMCAT SERVER

We'll deploy the applications to an Apache Tomcat Server. To register the server, select Windows > Preferences > Server > Installed Runtimes. Click the Add… button and follow the dialogs to register the server. Next, create an instance of the server. Add the Servers view by selecting Window > Show View > Servers. On the Servers view, right-click to select New > Server from the pop-up menu and follow the dialogs to create a server.

CREATING JSF LIBRARIES

Let's create a library called SUN-RI that contains the Sun JSF RI and Apache JSTL jars so that our application can execute against Apache Tomcat. Select Windows > Preferences > Web and XML > JSF Libraries. Click the New... button to add a new library. Browse to the folder that has the JSF Reference Implementation libraries and add the jsf-api.jar and jsf-impl.jar. Check the "is JSF Implementation" check box. Next, add the Apache Commons jars and the JSTL jars. The following jars must be present in the library:

```
jsf-api.jar
jsf-impl.jar
commons-beanutils.jar
commons-collections.jar
commons-digester.jar

commons-logging.jar
jstl.jar
```

```
standard.jar
```

CREATING A WEB APPLICATION FOR TOMCAT

Create a New Dynamic Web Application called FirstJSFApp. Set the target runtime to Apache Tomcat 5.5. On the next wizard page, add the JavaServer Faces facet. Use the preset combo box to choose a version 1.1 project. On the JSF Capabilities page, add the SUN-RI library and select the Deploy JARS to WEB-INF/lib. Click the Finish button to create the JavaServer Faces Application. You may be prompted to accept the license for the Web App DTD from Sun. You may also be asked to choose the J2EE perspective upon completion. In both cases, accept.

Your JSF application has now been created. Note that the web.xml file has been updated with the Faces servlet and servlet mapping. A stub JSF application configuration file (faces-config.xml) has been created, and the build path has been updated with the implementation jars.

CREATING JSF–JSP PAGES

Add the JSF tag outputLabel with an empty EL expression for the value attribute, as shown below, and save. Use the JSP Page wizard to create a page called login.jsp in the Web Content folder of the new application. On the Select Templates page of the wizard, select the New JSP(html) template. Click Finish, and the page will open in the WTP JSP source editor. Add the JSF tag libraries, the view and the form tag. The JSP page should look like Listing 1. Add the JSF tag outputLabel with an empty EL expression for the value attribute as

shown below, and save. The editor will indicate a syntax error.

```
<h:outputLabel value="#{ }">
</h: outputLabel >
```

For the value attribute of the outputLabel tag, enter a value-binding expression referring to an undefined managed bean, and save. The editor will flag an error indicating that the bean is not defined in the application configuration resource file:

```
<h:outputLabel
 value="#{loginBean.prompt }">
</h: outputLabel >
```

Create a Java class, com.tutorial .LoginBean, with the following code, and save:

```
package com.tutorial;

public class LoginBean {
        private String prompt =
           "Name";
        private String name;

        public String getName() {
                return name;
        }

        public void setName(String
           name) {
                this.name = name;
        }

        public String getPrompt() {
                return prompt;
        }

        public void setPrompt(String
           prompt) {
                this.prompt = prompt;
        }
}
```

In the Project Explorer, expand the node JSFTutorial > WebContent. Double-click faces-config.xml to launch the Faces Configuration editor. Select the ManagedBean tab. Click the Add… button. This will launch the New Managed Bean wizard. Select com.tutorial.LoginBean. Click the Next button twice and select the Finish button to register the managed bean. Switch to the login JSP page. Right-click the page and select the validate option from the context menu. Note that the page should now be valid.

Add the JSF tag < h:inputText value = "#{}" > < /h:inputText >. With

Listings

■ **LISTING 1**
SHORT THE JSP PAGE IN THE WTP JSP SOURCE EDITOR

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<f:view>
<h:form>

<h:outputText value="#{mybean.name}"></h:outputText>
</h:form>
</f:view>

</body>
</html>
```

■ **LISTING 2**
THE LOGIN.JSP PAGE CONTENT

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<html>
<f:view>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
        <title>Login</title>
        </head>
        <body>
        <h:form>
                <h:outputLabel value="#{loginBean.prompt}"></h:outputLabel>
                <h:inputText value="#{loginBean.name}"></h:inputText>
                <h:commandButton value="Login"
action="login"></h:commandButton>
        </h:form>
        </body>
</f:view>
</html>
```

■ **LISTING 3**
THE "WELCOME.JSP" JSP PAGE CONTENT

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<html>
<f:view>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
        <title>Welcome</title>
        </head>
        <body>
        <h:form>
                <h:outputLabel value="Welcome!
#{loginBean.name}"></h:outputLabel>
        </h:form>
        </body>
</f:view>
</html>
```

the cursor inside the brackets, hit Ctrl and the space bar simultaneously. You should see a pop-up with a list of all the implicit objects in addition to the managed bean that was defined. Select the managed bean, loginBean. Enter a dot after loginBean and pause. You should see a pop-up listing the properties of the bean. Select Name. The tag should be as follows:

```
<h:inputText value="#{loginBean.name}">
</h:inputText>
```

Next, add a commandButton tag:

```
<h:commandButton action="login"
 value="login" />
</h:commandButton>
```

The final login.jsp is shown in Listing 2. Add another JSP page, called welcome.jsp, with the content shown in Listing 3, and save the files.

DEFINING THE NAVIGATION
Double-click the WEB-INF/lib/faces-config.xml file in the Project Explorer to launch the Faces Configuration editor. Switch to the Navigation tab. Add the login.jsp and welcome.jsp to the Navigation tab. Select the Page control from the Palette and drop it on the Navigation tab. Choose the page in the resulting Select JSP File dialog.

Connect the two pages. Click the Link control in the Palette, select the log-in page and draw a line to the welcome page. Select the line in the Navigation tab and in the property view and set the value of the "from" outcome to log-in, as shown in Figure 1.

RUNNING THE APPLICATION
Now, let's execute the page against the Apache Tomcat server. Select the login.jsp page in the Project Explorer, right-click and select "Choose Run on Server" from the context menu. Choose your Apache Tomcat instance. Click Finish. You should see the Tomcat server start in the console view and the executing log-in page appear in the Web browser. Click the log-in button. You should navigate to the welcome page. Congratulations! You've created and executed your first JavaServer Faces application using the new Eclipse WTP JSF tooling.

MOVING FORWARD
The first release of the JSF Tools Project (v. 0.5) has added features to the WTP project that simplify development of a JSF application. The project team plans to add significant new features in the next release, including a visual page designer. This will truly make the Eclipse WTP Project a compelling environment for building sophisticated JSF applications. Want to participate? Get in touch with the team at wtp-jsf-dev@eclipse.org. ⬧
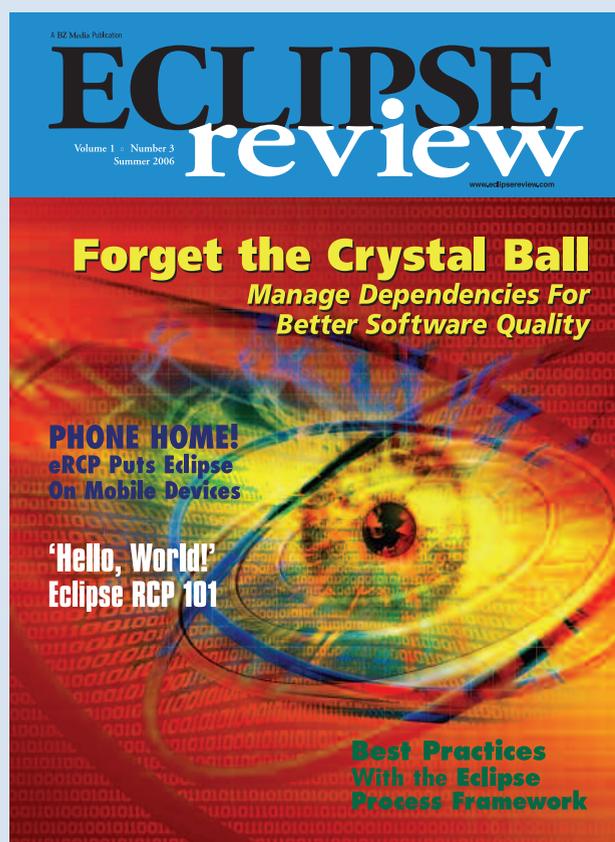
Eclipse on Cell Phones?

Y ou *can* take it with you—literally. Applications that used to run on the desktop can now be accessed through a browser, and apps that needed a browser can today be run anywhere, anytime. That's the power of the embedded Rich Client Platform, an alternative to the traditional big-screen GUI. What's best about building applications using eRCP is that, as with the Eclipse RCP, you can do it all from your favorite workstation, using open-source tools and technologies. So, what are you waiting for? Let's get started!

WHAT'S ERCP ALL ABOUT?

The RCP application model is used widely on desktops, but its architecture is well adapted to embedded devices because multiple applications can run in a single JVM using the Open Service Gateway Initiative (OSGi) platform. OSGi enables devices (including desktops) of any shape or size to conform to a consistent and modular program model on a well-architected set of framework and core services.

This framework provides services and life-cycle management to enable dynamic loading and many other features. The embedded Rich Client Platform (eRCP) project is about bringing the familiar concepts of Eclipse's Rich Client Platform (RCP) to the applications that run on devices such as mobile phones and PDAs. Concepts like plug-ins, views and the Workbench are all retained, but in a much smaller footprint that can fit on today's more capable devices. The prime benefit is that Eclipse skills for writing RCP applications for desktops can be transferred to writing eRCP applications for devices.

eRCP consists of a complete Eclipse runtime based on OSGi R4, several components that are subsets of RCP components and some components specific to small device development. Together, these provide a powerful environment that gives developers Eclipse's capacity to use existing services and features, along with new capabilities that really allow Java applications to run effectively on a multitude of different devices.

A WORD ON J2ME

Java 2 Micro Edition (J2ME) is a Java platform for embedded devices. Similar to the enterprise (J2EE) and desktop (J2SE) platforms, J2ME delivers Java technology's benefits

■ BY CHRIS ANISZCZYK, GORKEM ERCAN, & MARK ROGALSKI

to the embedded space. The important thing to remember about J2ME is the notion of configurations and profiles. J2ME *configurations* comprise a virtual machine (VM) and a minimal set of class libraries that provide the base functionality for particular devices. These devices share similar characteristics like memory footprint and network connectivity. Currently, there are two J2ME configurations: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). *Profiles* are higher-level APIs that, combined with device configurations, provide a complete runtime environment that is targeted for a specific set of devices.

When working with eRCP, it's important to note your target application because it will dictate the devices that can consume your application. (Note: OSGi lets you define execution environments [EE] that mirror profiles.) eRCP currently runs on Windows Mobile 2003/5 and the Nokia Series 60/80 devices—with QTe device support on its way. The eRCP project includes a Windows Desktop runtime so you can run and test eRCP-based applications without going through the hassle of deploying your code to the device. Figure 1 (see page 23) shows Eclipse saying Hello from a Windows Mobile 2005 cell phone.

INSIDE ERCP

The eRCP project provides code patches that make the Eclipse Core Runtime and OSGi plug-ins compatible with the J2ME Foundation profile. It now uses this set of plug-ins directly from Eclipse. For other components,

Chris Aniszczyk is a software engineer at IBM Lotus currently focused on OSGi/Equinox development. He also works on the Gentoo Linux distribution, and is a committer on many Eclipse projects.

Gorkem Ercan is a senior software engineer in Nokia's Java platforms group, and a committer for the Eclipse eRCP and Eclipse webtools projects.

Mark Rogalski is the development lead for an IBM project targeting mobile devices as well as the project lead for the Eclipse eRCP project.

_INFRASTRUCTURE LOG

_DAY 15: This project's out of control. The development team's trying to write apps supporting a service oriented architecture, but it's taking forever. Gil's resorted to giving them all coffee IVs. Now they're on java while using JAVA. Oh, the irony.

_DAY 16: Big crisis—we've just run out of half-and-half!!

_DAY 18: I've found a better way: IBM Rational. It's a modular software development platform based on Eclipse that helps the team model, assemble, deploy and manage service oriented architecture projects. The whole process is simpler and faster, and all our apps are flexible and reusable. The software we write today will be the software we use tomorrow. :)

_The team says it's nice to taste coffee again, but actually drinking it is sooo inefficient!

Rational®

Download the IBM Software Architect Kit at:
IBM.COM/**TAKEBACKCONTROL**/FLEXIBLE

such as the Standard Widget Toolkit (SWT), JFace and Update Manager, it has subsets to reduce their size and provide a concentrated API that's useful for embedded devices. The eSWT Mobile Extensions and MicroXML components were created to solve issues like the memory constraints found in most embedded environments. In the case of Mobile Extensions, giving developers the ability to further abstract device input and output mechanisms to accommodate the wide range of different I/O mechanisms is important. For MicroXML, the need was for a small and fast XML parser. MicroXML definitely fits the bill, providing full SAX and DOM support in only 107 kilobytes. Figure 2 shows Label and Text widgets displayed on a Nokia S60 implementation.

COMPONENTS

The embedded Standard Widget Toolkit (eSWT) is a subset of the well-known Standard Widget Toolkit (SWT). In addition to embedded versions of the desktop SWT components, eRCP provides Mobile Extensions containing new dialogs and controls targeted for embedded devices. Just like desktop SWT, eSWT is designed to be independent from other eRCP components, so it is possible to use eSWT without eRCP as part of a "midlet."

eSWT is divided into three components to provide flexibility for configuring eSWT content according to device capabilities. The Core component is the only mandatory component for a platform and is a subset of basic desktop SWT. The Expanded component contains a subset of more sophisticated desktop SWT widgets that require more resources. The Mobile component includes widgets that are targeted for mobile devices.

> ## ESWT IS DESIGNED TO BE INDEPENDENT FROM OTHER ERCP COMPONENTS, SO YOU CAN USE IT WITHOUT ERCP AS A "MIDLET."

Another important design feature of eSWT is compatibility with its desktop counterpart. eSWT is a subset of the desktop SWT API, but doesn't alter the behavior of widgets or features that are common to both toolkits. A desktop version of the implementation for mobile extensions is available as part of the project so that you can develop eSWT-based applications that are targeted for both mobile and desktop platforms. It also ensures that applications targeted for eRCP can also be run with desktop RCP. For example, the following Hello World example will run in both eRCP and RCP contexts:

```
public class HelloWorld {

 public static void main(
  String[] args) {
   Display display =
    new Display();
   Shell shell =
    new Shell(display);

   Text text =
    new Text(shell, SWT.SINGLE);
   text.setText("Hello World!");

   shell.open();
   while (!shell.isDisposed()) {
    if (!display.readAndDispatch())
     display.sleep();
    }
    display.dispose();
   }
}
```

ESWT CORE

The core library offers the minimal set of functions required for developing embedded applications. These include essential widgets, graphics and events. The API provided is a subset of desktop SWT API, so applications developed using it are binary-compatible with desktop platforms. eSWT core APIs inherit all classes from desktop SWT in org.eclipse.swt, org.eclipse.swt.events, org.eclipse.swt.graphics packages and some of the classes from org.eclipse.swt.widgets and org.eclipse.swt.layout. The included classes from these packages are shown in Table 1.

eSWT ExpandedeSWT Expanded is also a strict subset of desktop SWT. The desktop SWT widgets and layout managers that require more memory and larger screen sizes are provided via this library. All the desktop SWT

Table 1 | Widgets and layouts in eSWT core

org.eclipse.swt.widgets	org.eclipse.swt.layout
Listener	FormAttachment
Button	FormData
Canvas	FormLayout
Combo	
Composite	
Control	
Decorations	
Dialog	
Display	
Event	
FileDialog	
Item	
Label	
Layout	
List	
Menu	
MenuItem	
MessageBox	
ProgressBar	
Scrollable	
ScrollBar	
Shell	
Slider	
Synchronizer	
Text	
TypedListener	
Widget	

classes present in org.eclipse.swt.browser and org.eclipse.swt.dnd are also present in the library. The classes listed in org.eclipse.swt.layout and org.eclipse.swt.widgets packages are also available, as shown in Table 2.

ESWT MOBILE EXTENSIONS

eSWT introduces new widgets and dialogs with this library to better support different input and output mechanisms and better use the native UI capabilities common to mobile devices. The following are the new controls and dialogs contained in the org.eclipse.ercp.swt.mobile package:

- **CaptionedControl:** Displays a caption in front of another control and an optional trailing. Depending on the platform behavior, the caption is highlighted when the contained control is focused.
- **Command:** Represents a general action that is associated with a control. The means of invoking a Command differ according to platform capabilities. While they can be menu items in one platform, they may be associated with a hardware command button or voice commands.
- **ConstrainedText:** A single line text control that limits the characters that can be used by the user. This control is ideal for numeric, decimal and phone number input on devices.
- **DateEditor:** A data entry control that allows users to enter or choose a date or time.
- **HyperLink:** A selectable control similar to a hyperlink on a Web page that launches applications like phone books, messaging or browsers that are common to mobile platforms, as shown in Figure 3.
- **ListBox:** A selectable user interface that displays a collection of ListBoxItems. A ListBoxItem allows the specification of multiple icons and multiple lines of text.
- **ListView:** Displays a collection of options in a multi-column layout to allow you to select certain items. A list item can have an icon and text.
- **MobileShell:** An extension of the desktop Shell that allows the trimmings to be dynamically changed.
- **MultipageDialog:** A tabbed dialog common in the mobile world. Each tab hosts a Composite that leads only to its own contents, as shown in Figure 4.
- **QueryDialog:** A dialog for querying different data types, such as time, date and password.
- **SortedList:** A list in which the elements are displayed in ascending or descending order. This list can be filtered.
- **TaskTip:** A nonmodal window with text and an optional progress bar that is suitable for providing feedback on long-running tasks.

Figure 1

Figure 2

Figure 3
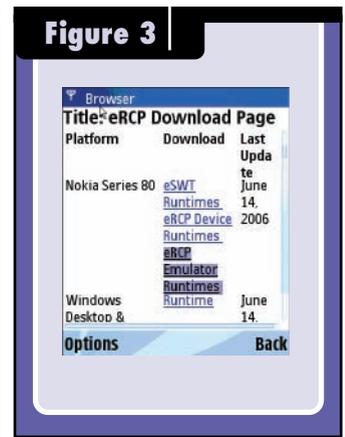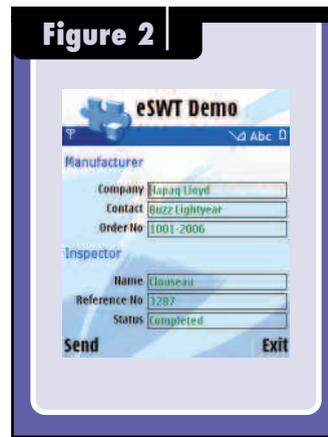
- **TextExtension:** An extension to the Text control that provides the functionality needed for efficient use of keyboard devices that have limited options.
- **TimedMessageBox:** A dialog that closes itself after a certain time to notify the user about a limited amount of information.

In addition to the new controls and dialogs, the mobile library also provides APIs to query and track the changes in certain mobile device features, such as screen and input capacities. The MobileDevice, Screen and Input classes, and their respective events and event listeners, can be used to better adjust eRCP applications to different devices.

Table 2	eSWT Expanded is a strict subset of desktop SWT
org.eclipse.swt.widgets	**org.eclipse.swt.layout**
ColorDialog	FillLayout
DirectoryDialog	GridData
FontDialog	GridLayout
Table	RowData
TableColumn	RowLayout
TableItem	
Tree	
TreeItem	

EJFACE

eJFace, like its RCP cousin JFace, is a platform-independent UI toolkit that provides viewers that simplify the development of eSWT-based applications. eJFace wraps the eSWT widget in the context of the Model-View-Controller (MVC) paradigm. To use it, you supply an eJFace viewer with a model of your choice and controller classes that make the viewer show what you want. The viewer types available in eJFace are:
- CheckBoxTableViewer
- CheckBoxTreeViewer
- ComboViewer
- ListViewer
- TableViewer
- TreeViewer

eJFace provides a resource-handling class that allows

for the efficient manipulation of resources such as fonts and images to save memory and increase performance, as well as a preference page framework. This allows eRCP applications to provide preference pages in a platform-consistent manner. One difference from the desktop implementation of preferences is that it shows preference pages applicable only to the foreground application on a device rather than a conglomeration of pages from every plug-in registered in the framework.

EWORKBENCH

The eWorkbench is similar to the Eclipse IDE. It provides a visual framework for displaying plug-ins. In the case of eRCP, these plug-ins are called eRCP Workbench applications. The eWorkbench recognizes plug-ins that are eRCP Workbench applications and adds them to an application list that it displays. You can select an application, and the eWorkbench then causes its view to be displayed in the Workbench's default perspective.

Figure 4

The eWorkbench is designed to be customized so that it can take advantage of special device capabilities. For example, if a device has two displays, a customized eWorkbench could display a different application on each display or transfer an application's view from one display to another. When a device is opened, an application can move from a small external display to a larger internal one.

Figure 5

Let's go through a simple "Hello World" example for the eWorkbench. The first step in creating an eWorkbench application is to create a view. Here's a simple view that contains only a label that displays "Hello World":
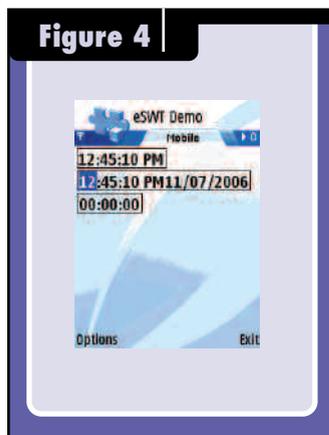
```
public class DefaultView extends ViewPart {

  public void createPartControl(
   Composite parent) {
   Composite composite = new Composite(
    parent,
    SWT.NONE);
   composite.setLayout(
    new FillLayout());

   // create a label
   Label label = new Label(
    composite,SWT.CENTER);
   label.setText(
    "Hello eWorkbench!");
  }
```

```
  public void setFocus() {}
  }
```

The next step is to notify Eclipse that you have a view to offer. This should be familiar if you've ever created a view in Eclipse, because you need only to modify the plugin.xml file:

```
<extension
  point="org.eclipse.ui.views">
   <view allowMultiple="false"
    category="org.eclipse.ercp.eworkbench.viewCategory"
    class="org.eclipse.ercp.sample.views.DefaultView"
    icon="icons/sample.gif"
    id="org.eclipse.ercp.sample.defaultView"
    name="Hello World View"/>
</extension>
```

In the last step, you'll notify the eWorkbench that you have an application to offer. This is done by extending the org.eclipse.ercp.workbench.applications extension point. The following sample takes our previously defined view and lets the eWorkbench consume it:

```
<extension point="org.eclipse.ercp.eworkbench.applications">
  <application
   id="org.eclipse.ercp.sample.application"
   name="Hello World Application"
   singleton="true">
   <views normal="org.eclipse.ercp.sample.defaultView" />
  </application>
</extension>
```

As you can see, creating an eWorkbench application is very similar to creating an application for Eclipse.

EUPDATE MANAGER

Just as Eclipse offers the Update Manager, eRCP offers eUpdate Manager. As shown in Figure 5, eUpdate Manager is a full-function bundle (plug-in) manager, but the user interface has been simplified and optimized for smaller screens.

Some flexibility has been removed to reduce its size, which probably won't be missed because it's overkill for a small device. Bookmarks for update sites can be created and then browsed. Installed bundles can be updated individually or all at once.

BIG THINGS IN SMALL PACKAGES

Looking to the future, eRCP is aimed at supporting more devices and adding the features that they demand. Embedded devices have been around for a long time, but those with GUIs are becoming more and more prevalent.

eRCP offers Eclipse developers a means to work in their environment of choice while allowing them to take advantage of the opportunities available for developing in the embedded space. While cell phones and PDAs may be the first graphics-driven embedded devices that come to mind, we can only imagine the potential for offering such interfaces and capabilities in other devices targeting the consumer and other markets—the possibilities are endless. Find more information at www.eclipse.org/ercp.

A Development Library At Your Fingertips

Best practices can be an elusive term, but any attempt to standardize the development process and give some common guidelines to everyone involved in application creation can't be a bad thing. How would it be to have a flexible library and means of collaboration that helps you get products to market more efficiently? The answer for many may be the Eclipse Process Framework (EPF).

WHAT IS EPF?

EPF is different than other attempts to create best practices in software development. Its open-source aspects provide developers, academia, practice and thought leaders, consulting organizations and technology vendors a forum for collaborating to create reference libraries and processes that work in practice rather than theory. EPF is committed to taking a fresh approach to process development to overcome common challenges such as:

- Practices from different sources captured in different formats, making it hard to combine their content into an integrated process.
- Practices that are difficult to customize to fit the unique needs of a project. For example, it's hard to rewrite a few pages and publish a new book to address the unique style of your project.
- Practices typically expressed in text form without any formal metamodel, making it difficult to use tools to instrument and automate them, resulting in manual, haphazard adoption within your development environment.
- Branding of practices that encourage silos to drive toward divergence rather than convergence in the industry.

Interim builds of EPF are available now. You can access the software by going to www.eclipse.org/epf. You can also download EPF Composer, the tool for configuring and publishing process content (see "Increasing Development Knowledge with EPFC" by Peter Haumer, Spring 2006), along with a process library called

■ **BY PER KROLL & KURT SAND**

OpenUP/Basic. Over time, you can expect the scope of these practices to include a variety of agile development approaches and grow to new software and systems engineering domains.

AN AGILE APPROACH

You may find that the first process, OpenUP/Basic, integrates well with your current practices. It takes an agile approach to development, valuing team collaboration and benefits to the stakeholders versus unproductive deliverables and formality. However, the process does this within a proven, structured iterative life-cycle, so that projects can have enough control and measures to fit within an organization's limits, as illustrated in Figure 1.

The goal of OpenUP/Basic is to deliver an iterative software development process that is minimal, complete and extensible. It was seeded through an IBM contribution of a subset of RUP with many influences from Scrum. It has since been enhanced by committers and contributors from more than a dozen organizations.

OpenUP/Basic is minimal in the sense that it includes only content that is deemed essential. The process is less than 200 pages long and consists of only six roles, 22 tasks and 17 artifacts. This means that it's briefer than your average book on processes, with the benefit of being easier to use, with browser-based navigation, and an index and search engine that allow you to find

Per Kroll *is project leader for the Eclipse Process Framework project, and development manager for RUP/RMC. His most recent book, with Bruce MacIsaac, is "Agility and Discipline Made Easy—Practices from OpenUP and RUP" (Addison-Wesley, 2006).*

Kurt Sand *is a senior manager for Telelogic Solutions. He defines, develops and promotes software development, governance and compliance solutions for Global 1000 companies.*

How to Contribute to EPF

EPF development progresses like any other Eclipse project. Anyone can contribute to the content and tool by reviewing the latest builds and submitting defects, recommendations and improvements via Bugzilla.

If you want to take EPF beyond its current work and create a new content area, you need to submit an initial showcase that demonstrates that you have enough developers interested in evolving the content and a likely target audience.

For example, you can post a note to the EPF newsgroup expressing your interest and use the response to build interest. As your idea gains ground, you'll get many ideas from the community on how to improve the content. By design, it's easy to start working on content within EPF. However, the quality bar to get content included into an official release is set high. You need to have favorable reviews by committers, and you need to have high and consistent quality from a technical and writing perspective. You also need to show that you have a sustainable developer, user and ideally value-add community for your content.

If you've created content that EPF committers don't accept, you can always make it available outside the "official" EPF builds as a plug-in that people can download for a fee or free of charge.

information when you need it.

The process is complete—it covers an entire course of action for building a system. In addition to guidance around roles, tasks and artifacts, it gives you:

- Templates to use as a starting point for producing project plans, iteration plans, test cases and other artifacts
- Examples
- Checklists to get information on what constitutes a good test case or good requirements
- Guidelines to assist you in performing tasks like writing automated tests and producing test suites
- Concepts to educate yourself on topics such as patterns or components

At the same time, OpenUP/Basic doesn't explore topics such as distributed or large-scale development, use of specific tools or technology, or compliance with standards.

It's also extensible, which means it can be used as a foundation on which process content can be added or tailored as needed. This extensibility is aided by EPF separating *method content*, which you can see as a reference library of best practices, from *process*, which describes how to deliver software by using appropriate subsets of the reference library. If you want to extend OpenUP/Basic to address development that needs to comply with a certain standard, you can produce a plug-in that will describe how to add, modify and delete material. A plug-in can bring additional material to the reference library, like how to deal with traceability and audits, or it can help you modify processes, such as inserting formal reviews and audits at appropriate stages of development.

OpenUP/Basic is part of a larger method framework that consists of a base process and, in the future, a number of extensions that will allow you to construct a family of processes addressing a variety of development needs, all united by a set of core principles for agile development. These principles are:

- Balance competing priorities to maximize stakeholder value
- Collaborate to align interests and share understanding
- Evolve to continuously obtain feedback
- Focus on articulating the architecture

GETTING STARTED WITH EPF

Probably the best approach in learning to use EPF is to try to use OpenUP/Basic as-is on a project to gain some experience with it before you customize it. Using their favorite Web browser, team members can find relevant information about doing their job and learn how the artifacts they work on are developed and used collaboratively with other team members. For example, as an analyst, you can see what tasks you typically work on and what artifacts you're responsible for, as shown in Figure 2. If you decide to work on the Define Vision

---

Figure 1 | Coming to agreements. OpenUP/Basic is an iterative approach with four well-defined milestones
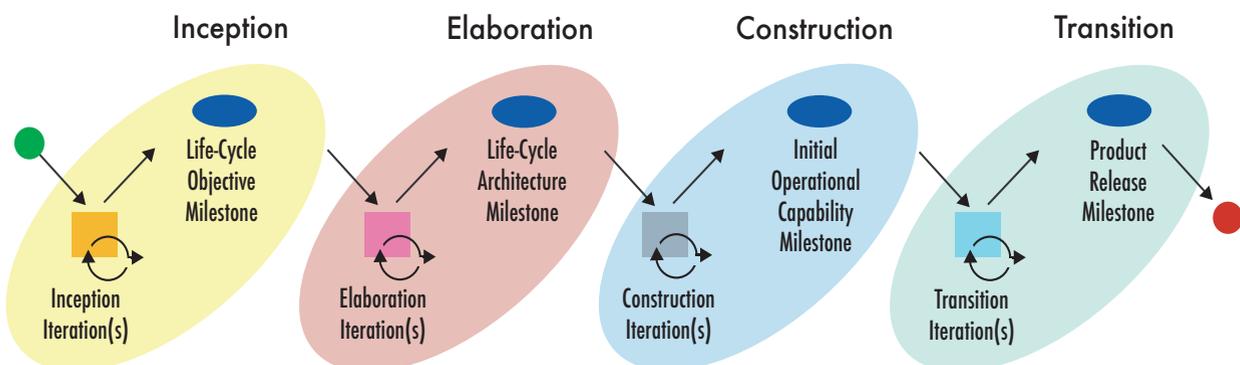
Figure 2 | Browser-based navigation aids with the analyst role
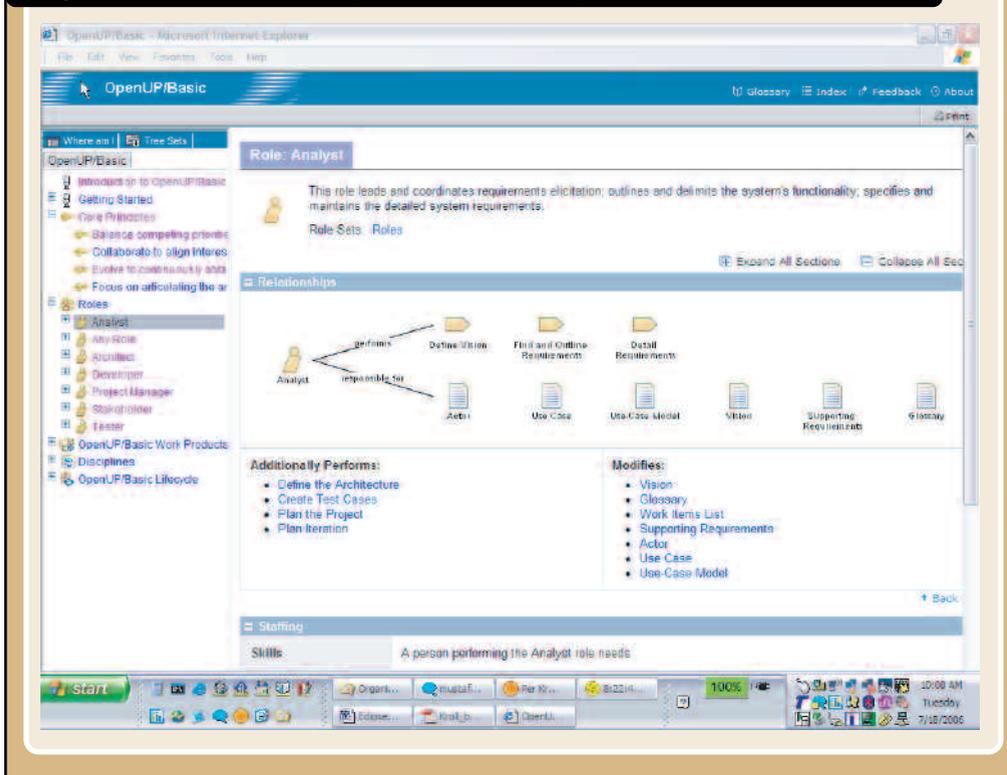
Using OpenUP/Basic doesn't replace the need for the books, articles and coding guidelines you use today. However, it does provide you with an easily accessible knowledge base for specifying process standards for a team.

HOW TO TAILOR EPF TO YOUR NEEDS

As you get familiar with OpenUP/Basic, you may want to change the process, which EPF Composer allows you to do. For example, you may want to add your own examples for artifacts, change templates to match the information you want to capture, or add or rewrite guidelines to reflect your own best practices. All of these changes can be captured in your own separate plug-in that will be expressed as a change set.

task, you receive step-by-step guidance, checklists, guidelines, concepts and other useful information that can assist you in developing a vision, as shown in Figure 3. As you progress through creating a vision, you'll update the Glossary, assuming you've decided to have one.

You can use the template as a vision to get you started and to make sure that the visions for all of your projects are stylistically similar. If you've never written a vision before, you might want to read through some relevant concepts and guidelines for requirements, requirements gathering techniques and achieving concurrence. Once you think you have a decent vision document, you can use the checklist for a vision to help you assess whether it's appropriate. Similar guidance is available for key tasks and artifacts for other people on the project, including developers, testers, architects, project managers and stakeholders.

The advantage of this approach is that you can easily move your changes to the new version as new releases become available. If the EPF project or other content creators produce a plug-in for describing how to use a require-

Figure 3 | Develop a vision. Step-by-step guidance helps both novices and experts
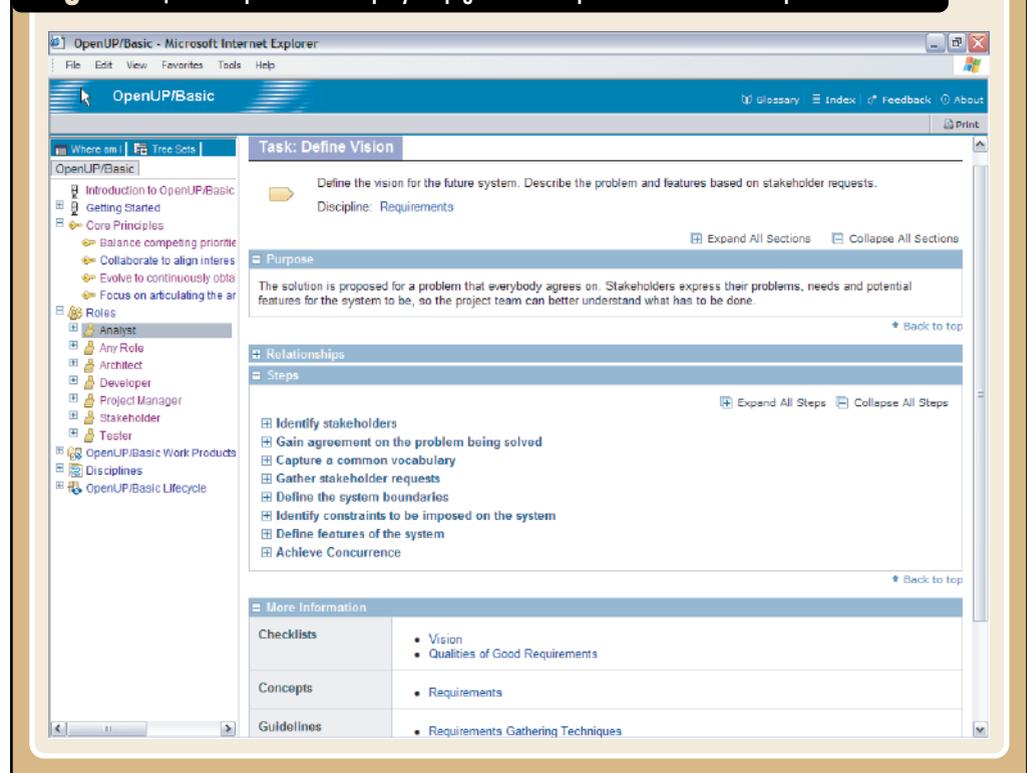
ments management tool to create and manage a vision document, you can pour this bucket of content into your version without having your changes overridden. This lets you integrate OpenUP/Basic into your process while using continuous innovation within the EPF project and from value-add providers. Work is also underway to integrate Wiki technology into EPF, which would make it even easier for projects and individuals to augment and modify the process.

EPF: BEYOND ECLIPSE

EPF Composer uses the Eclipse Platform and technologies to leverage the existing work of other Eclipse projects while focusing tooling efforts on what's needed for process authoring and publishing.

However, a really exciting aspect of EPF is that the content produced can address development processes for any platform and technology—including development using J2EE on Eclipse or .NET using Microsoft Visual Studio. When you want to deploy a process to the team, you can publish the content as a cohesive Web site in HTML format, allowing developers on any platform to access the process regardless of whether that platform can run or is running Eclipse. This extends EPF's value well

> **EPF-PRODUCED CONTENT** CAN ADDRESS DEVELOPMENT PROCESSES **FOR ANY PLATFORM AND TECHNOLOGY.**

beyond Eclipse-based development to any best practice that can be captured, described and published.

COMMUNITY STRENGTH

EPF's true strength lies in the fact that it's an open-source project. Its success is dependent on the strength of the community that supports the effort. Three thriving communities are necessary for it to be successful:

- Committers and Contributors, who will develop the open-source practices and supporting process tools
- Users, who adopt and benefit from the technology
- Value-added Providers, who offer products and services based on the open-source offering

Initially, the EPF project focused on creating a committer and contributor community. Recently, it has become established as one of the most vibrant Eclipse projects, with a growing development team comprising more than 25 committers from over a dozen organizations and many contributors. Close to a hundred participants were present in a total of five face-to-face meetings in Europe, U.S. and Canada, along with many workgroups with weekly e-meetings. The team continues to innovate and make incremental improvements.

A growing number of individuals and organizations have expressed interest in contributing process content for XP, Scrum, Essential Unified Process (EssUP), Agile Unified Process (AUP), DSDM and the OPEN Process Framework. Many of these companies are going through legal procedures associated with contributing content to Eclipse. And several thought leaders in the agile community have expressed an interest in creating a broader collaboration around a consolidated agile framework.

OpenUP/Basic is a tool-agnostic process that can be used with pen and paper, whiteboards or tools. The metamodel does provide for hooks called *tool mentors* that can be added to offer you step-by-step guidance for carrying out tasks using a certain tool. Value-added providers can also produce plug-ins that allow you to add mentors for the tools you're using. You should expect to see a lot of interesting progress in these areas in the future.

EPF MOVES FORWARD

Time after time, open-source projects have shown that they drive unprecedented innovation. We should expect that both EPF content and associated tooling will continue to evolve at an increasingly rapid speed. You can already explore a range of interesting ideas on how to leverage Wiki technology, card game metaphors, visualization and automation technology to make the process easier and more fun to adopt.

So get involved to help build the next generation of software processes. Knowledge, as we all know, is strength, and getting products to market faster by contributing to and using a growing knowledge base makes development easier for everyone. ef

---

The Benefits of Using EPF

An open-source framework for developing, integrating and publishing processes can provide a number of significant benefits to you and your organization:

- The ability to quickly and easily capture and communicate best practices reduces the risk of losing a key team member and provides an effective means of training new team members.

- Using a common framework based on a documented metamodel makes it easier to integrate best practices from multiple sources. It also makes it easier for you to share your practices with others.

- EPF provides an easy way to communicate processes across projects in your organization and with external stakeholders that may include regulatory bodies that require documentation of your development processes.

- The EPF ability to integrate process with tool guidance enables you to first define your practices before choosing tools for automation. This reduces the chance of purchasing tools that don't support the way you develop software and systems. It also guards against tools going unused because people don't know when and how to use them.

- EPF enables you to continue to evolve your practices. Current practices can be documented, baselined and analyzed to identify gaps and areas for improvement. Subsequent process-improvement initiatives are simplified via configuration and change management of the process artifacts. This can also be applied to more formal process improvement initiatives such as CMMI, if you wish.

Leaving the Crystal Ball Behind in App Development

Nothing lasts forever. The sad fact is that application quality degrades during its useful life. Software guru Tom DeMarco has claimed that applications become more "brittle" as enhancements, modifications and bug fixes performed over a period of years degrade their overall quality and their ability to adapt to future modifications and changes in scope. The longer you keep a code base in productive use, the more likely it is to cause problems for users.

What does this have to do with your software project? It helps you to understand the reasons why quality degrades over time. Changing and adding code to an existing application remains largely a manual operation, with a potential for errors and omissions with each change that will ultimately lower its quality.

But application quality also has an architecture component. Brittleness occurs because the original application design may not be ideal, and because changes over time fail to follow good design principles in the rush to address bugs or add new features. The result can be called "accidental architecture," where the architectural model evolved based on a series of tactical decisions rather than a strategic plan.

LINKING THE PIECES

A significant aspect of application architecture has to do with dependencies among components. Code is full of dependencies; that is, most packages call other packages in some capacity and call other code as well. There are also dependencies on the operating system, processor and underlying I/O hardware. Some of these dependencies are direct, while others are indirect.

This is not only desirable, it's inevitable. It's impractical to write code that doesn't contain dependencies. Without them, there would be little more than basic I/O, and the application would have no coherent structure (anyone remember Fortran?). In general, we accept the dependencies on the JVM or on operating system services as vital to writing an application. They provide rich I/O and a fine level of control over application features, with-

■ BY PETER VARHOL

out the need to write and rewrite foundation code for every application.

How we manage the dependencies in our code goes a long way toward creating a sound architecture that helps preserve quality over the application's useful life. It may sound trite to say that there are good dependencies and bad dependencies, but this goes back to best practices for building code. Good dependencies are those that tend to flow in one direction, usually from the least stable packages to the most stable.

> ## UNSTABLE DEPENDENCIES ARE INHERENT IN PACKAGES THAT CHANGE OFTEN THROUGHOUT DEVELOPMENT AND MAINTENANCE

In particular, several different types of dependencies deserve scrutiny. A common one is cyclic dependency, in which cycles exist in the package-dependency graph. Packages that have cyclic dependencies are typically more difficult to unit test, maintain and understand. Cyclic dependencies make it more difficult to predict the effect that changes of a package will have on the rest of the system. A set of packages with cyclic dependencies has issues related to the concept of design layering in that the packages have a disorganized architecture.

Unstable dependencies also have an impact on application architecture robustness. Unstable dependencies are inherent in packages that change often throughout

Peter Varhol is a software consultant and technology writer based in Nashua, New Hampshire. He can be reached at peter@petervarhol.com.

development and maintenance. Due to those frequent changes, the application's structure becomes unstable as the dependencies shift.

We can turn these truisms into design rules for managing package dependencies. Such rules might look something like this:

Minimize dependency cycles between packages: Cyclic dependencies make an application brittle by getting away from a layered architecture. Calling back and forth between packages makes it more likely that

other poor design decisions, then inform designers and even individual developers of rule violations. The actual architecture of the application can then evolve in a deliberate manner instead of being an accident, resulting in significantly better application quality across the entire life cycle.

Dependency analysis provides details on what packages or classes are dependent upon others, so that architects and developers have a clear understanding of the effects that changes and enhancements will have on different parts of the code base. It allows architects to design applications that follow best practices, and developers to implement code that doesn't become brittle over multiple maintenance releases.

Figure 1 | Many dependency analysis tools use visual displays of dependencies between packages

ECLIPSE GETS INTO THE ACT

Fortunately, Eclipse is becoming a popular host for tools that provide an insight into the dependencies within an application. The Eclipse platform provides several ways of performing dependency analysis. Even dependency tools that don't integrate into Eclipse as plug-ins can be used in conjunction with the IDE. For example, some third-party products let you run an analysis through Ant, which, of course, is a standard Eclipse component. In addition, changes to the resulting structure can be applied to the Eclipse project, so that any changes performed can automatically be seamlessly loaded into Eclipse. JDepend, licensed under the BSD license and also executable as an Ant task, provides the ability to delineate all of the dependencies among Java packages.

However, the easiest way to work with dependency analysis from within Eclipse is through plug-in integration. A Stanford University project has taken the JDepend source and turned it into an Eclipse plug-in. Commercial products have also been implemented as plug-ins to offer build, analysis and fix cycles for dependencies without leaving the platform.

Dependency analysis can be approached in several ways. Probably the simplest way to identify and examine dependencies is by listing all the classes and methods along with what they call and what calls them. However, such a list will tell you little about the nature of the dependencies, because our brains don't process large amounts of text information very well.

changes in one package will require corresponding changes in classes, both in that package and the codependent one.

Maintain independence between specific packages: When packages reflect different application aspects but where those aspects are independent from each other, it's crucial to maintain this independence over time. Not only does it minimize complexity, it also allows effective code sharing among various applications.

Maintain a layered architecture: Keep dependencies moving in a single direction only, from the highest-level packages to the lowest. This supports the concepts of layering and stable dependencies. By keeping the dependencies moving downward and to more stable packages, you minimize the chances of disruptive changes and unpredictable results in every change you make.

Structured techniques that fit well with the current emphasis on agile programming are beginning to emerge that scan an application to identify dependencies and

Instead, it's easier and more informative to visualize dependencies. One way of doing this is to draw a diagram of components and the relationships between them. Figure 1 shows such a diagram for part of the Apache JMeter code base and the relationships between Java packages, for example.

These images are visually attractive, contain a lot of information, and can be helpful in determining architecture issues that should be addressed. But in reality, the diagrams themselves contain too much information. You don't necessarily need to know the nature of the dependencies right away to know whether or not you've violated any of the design rules. Until you get to the point of fixing poorly designed dependencies, this may be information overkill.

USING MATRICES AND DIAGRAMS

Alternatively, a natural way of tracking dependencies is with matrices. They present both a visual image of the state of dependencies within an application and avoid providing unnecessary information on the state of the application design. Matrices can show dependencies between packages without the need to delineate each dependency. It's possible to draw out such matrices by hand, but several approaches have been used to automate the process.

One such automated technique is the Design Structure Matrix, or DSM. DSM has grown out of research conducted at MIT on representing and analyzing complex systems. An active community is pursuing research and development into its use (www.dsmweb.org). DSM represents a Java application in a matrix view, with classes listed on both the horizontal and vertical axes. The squares in the matrix represent whether or not dependencies exist between packages.

Figure 2 shows a simple example of a DSM matrix. This application has four packages: A, B, C and D. Each is listed on the horizontal and vertical axes. The diagonal, of course, represents the unity case, where a package has a dependency on itself (which will always be the case).

An X in a block elsewhere in the diagram represents a dependency by a vertical package on a horizontal one. For example, in Figure 2, Package C is used by Package D. Alternatively, you can say that Package D depends on, or has a dependency on, Package C. Likewise, Package A also has a dependency on Package C. It is easy to see what packages are dependent on

Figure 2 | A sample DSM showing dependencies in the wrong direction
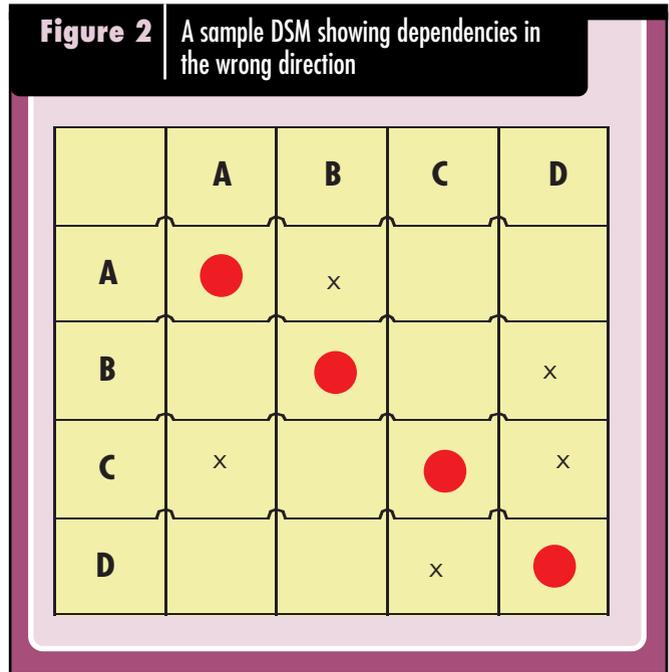
what others just by looking at the Xes. Figure 3 shows just what such a diagram would look like with no undesirable dependencies. In this case, all of the packages farther up in the design are dependent only on lower-level and more stable packages.

This simple diagram uses an X to represent a dependency. You can also take that a step further and use a number to represent the exact number of dependencies between the packages. In practice, the Xes are replaced by the actual number of dependencies between the packages. This provides an even clearer representation of the dependencies' extent between packages. These numbers are, of course, hypothetical, since this diagram doesn't represent the actual packages of a Java application. But it's illustrative of the DSM's ability to immediately point out potential problems with an architecture.
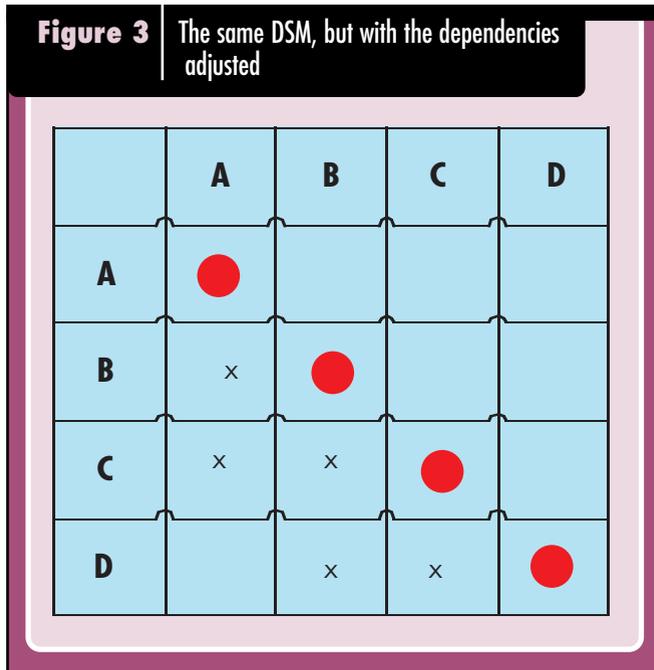
When you rearrange the packages from most stable to least, or from the foundation packages to those that depend on them primarily in a layered manner, all of the numbers are below the diagonal. This means that higher-level packages depend on lower-level ones, which promotes stability and predictability during the life of the application. Dependencies run in the same direction, from top to bottom.

Of course, you may not have designed your application that way. If not, DSM can help you determine the

Figure 3 | The same DSM, but with the dependencies adjusted

layering. But it instills the information needed to practice discipline during the numerous bug-fixes and enhancements that inevitably occur almost continuously while an application is in active use.

The DSM is both infinitely scalable and collapsible. You can make the matrix as large as you need it. If it grows too large, you can collapse it in a couple of different ways. First, you can organize your packages into larger groupings—call them subassemblies or application components. You might organize architecturally, by selecting classes or packages that belong together functionally, and organizing each of those separately using DSM.

Figure 4 shows how dependency analysis can be integrated into Eclipse using the Lattix LDM plug-in as an example. When you build the application, in this case the open-source Ant code base, the Lattix perspective displays the dependency matrix, along with a number of metrics, including the total number of dependencies among packages and rule violations, system stability and the number of package layers between dependent classes. These displays help you assess the architecture so that you can address any deficiencies.

FIXING STRUCTURAL DEPENDENCIES

To make the right decisions as a software architect or developer, you should understand the functionality and requirements of the program. You also need to have some idea about which parts of the program are likely to be extended or reused. However, as will be demonstrated here, package structure can be improved even without this knowledge.

One of the first things to look for is whether any classes are at the top level. A high occurrence of cyclic dependencies associated with the "collapsed" top-level classes, as is the case in this example, is a good indication that top-level classes should be moved to appropriate subpackages.

As shown in the matrix view, the Ant code base has several top-level classes. For improved package structure, it is good practice to move these classes to subpackages. However, care should be taken to

choose suitable packages based on the layer where the top-level classes are located. You can do this within the context of the Lattix LDM tool, for example, or manually.

PLANNING AHEAD

Eclipse is more than a development environment; it's an application life-cycle platform. This was reinforced with the Software Tools and Performance Project, and is being extended with two new projects being defined this year: the Application Lifecycle Framework (ALF), being led by Serena, and the Tools Services Framework, being led by Compuware.

In this light, dependency analysis is a life-cycle tool. Because "bad" dependencies, or broken architecture rules, can degrade an application whenever it's modified, these tools can help you not only in initial development, but as you bring your applications back for bug fixes and enhancements for years to come.

An Eclipse development team can employ dependency analysis as a life-cycle quality tool, enabling both designers and developers to create and maintain a more robust application. It will focus on using the information provided by dependency analysis to make better decisions on architecting the application, and apply analysis of an application's structure whenever code is added in order to reduce testing requirements while delivering enhancements without breaking existing features. **er**

> ## ECLIPSE IS MORE THAN A DEVELOPMENT ENVIRONMENT—IT'S AN APPLICATION LIFE-CYCLE PLATFORM.
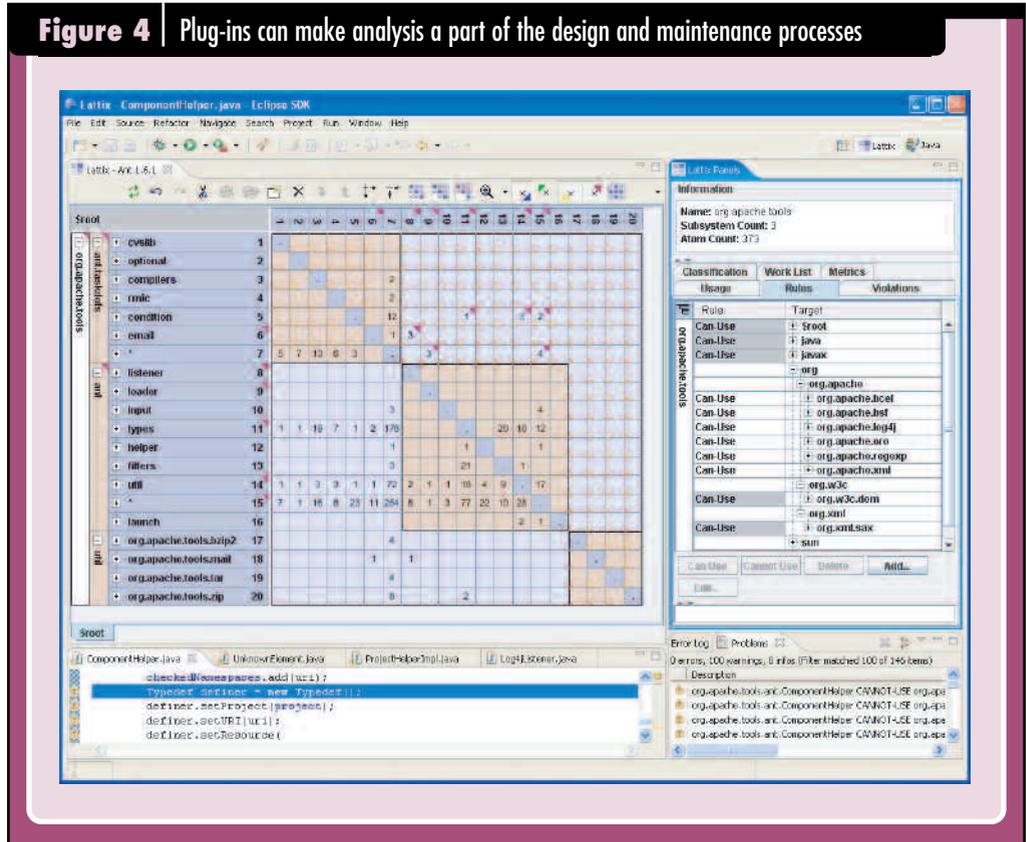
Figure 4 | Plug-ins can make analysis a part of the design and maintenance processes

Creating a 'Hello World' Eclipse Application

T he Eclipse IDE regularly tops the list of open-source tools for developer use and satisfaction. Beneath this IDE is a generic platform called the Eclipse Rich Client Platform (RCP), with which you can build general-purpose cross-platform client applications. To help you learn Eclipse and rich client programming, we'll present a simple example of a "Hello World" executable program that introduces you to some Eclipse RCP concepts, programming techniques and structures. Our example, created using Eclipse 3.2, shows how easy it is to create an Eclipse RCP application, turn it into a product, generate JUnit tests for the UI, and package and deploy the application. You may see some slight display differences if you're using Eclipse 3.1.

CREATING AN RCP PROJECT

The first step in creating a "Hello World" application is to create an RCP project using the Eclipse New Project wizard. Follow these steps to create the project. In the Eclipse IDE, choose the menu item File > New > Project to launch the New Project wizard. Select Plug-in Project from the list. In the next page of the wizard, enter the name of the project. For our example, enter org.eclipse.rcpapp.hello, make sure that Create a Java Project is selected and that Eclipse version 3.2 is selected as the target platform. Then click the Next button. In the next window, select Yes to Create a Rich-client Application, then click Next. Select the Create a Plug-in using one of the Templates options; then choose the Hello RCP template. This is a minimal stand-alone RCP application. Click Finish. The project is created, and the plug-in manifest is opened in the plug-in manifest editor, as shown in Figure 1.

The plug-in manifest editor can be used to manage the plug-in and edit its files: manifest.mf, plugin.xml and build.properties. The Overview page has links that are used to develop, test and deploy a plug-in. The Plug-in Content section can be used to manage the different sections of the plug-in manifest. The plug-in manifest editor has pages to configure and manage the dependencies, extensions and extension points of the plug-in. The Testing section has a Launch an Eclipse Application link that is a one-click launch

■ **BY DAN RUBEL & KEERTI PARTHASARATHY**

for the application. The Exporting Section of the Overview page lists the steps to follow to package the application.

Click the Launch an Eclipse Application in the Testing section of the manifest to run the RCP application. Here's the list of source files that were created:

Activator.java: Controls the life cycle of the plug-in. If you're using Eclipse 3.1, the filename will reflect the plug-in project name.

Application.java: The main program that launches the application and controls all aspects of its execution.

ApplicationActionBarAdvisor.java: Customizes menus, toolbars and status bars.

ApplicationWorkbenchAdvisor.java: Handles the Workbench, provides life-cycle and event-loop hooks for the Workbench and sets the default perspective.

ApplicationWorkbenchWindowAdvisor.java: Used to customize the Workbench window, such as status line, toolbar and title. It also provides hooks for the Windows life cycle.

Perspective.java: Defines the perspective.

The three Advisor classes are used to configure all aspects of the Workbench. The plugin.xml file defines the org.eclipse.core.runtime.applications extension point, which defines the class to run to launch the application; in this case, Application.java, as shown in Listing 1.

Dan Rubel is CTO of Instantiations and has more than 15 years of commercial software development experience, including eight years with Java and four years with Eclipse. He has been the primary architect and product manager for several successful commercial products, including JFactor, jKit/GO and jKit/Grid, and has played key design and leadership roles in other commercial products such as WindowBuilder Pro, VA Assist and CodePro Studio.

Keerti Parthasarathy is a member of the technical staff at Instantiations. She has a master's degree in computer science from Portland State University and more than eight years of experience in software development.

Figure 1 | The plug-in manifest in the manifest editor
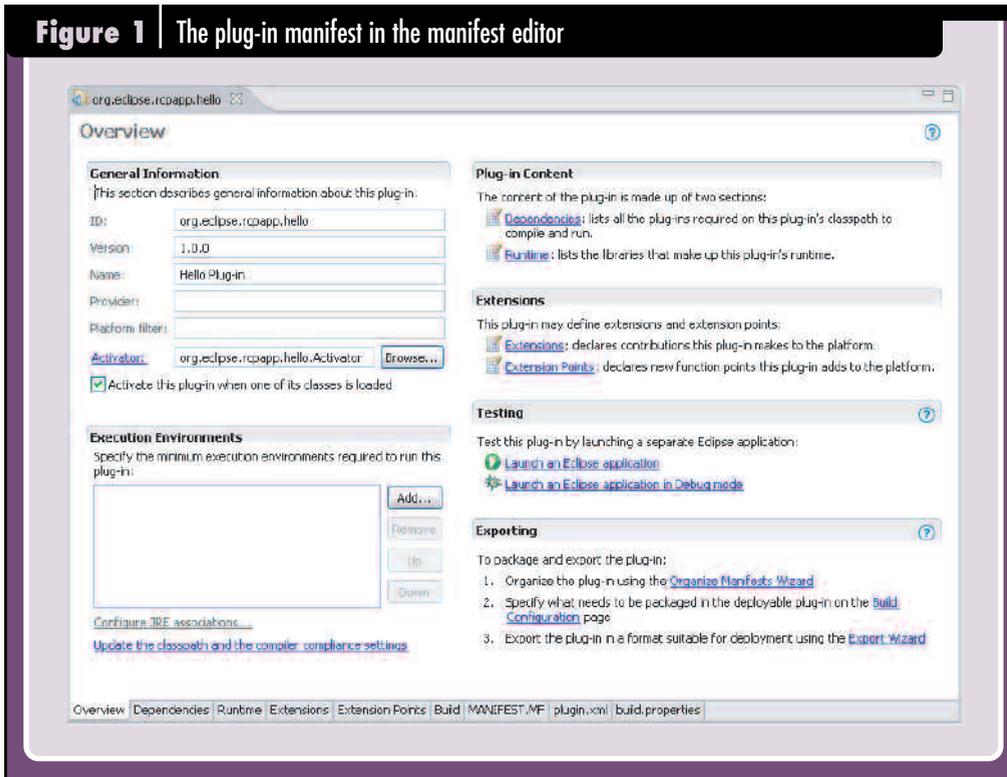
Once an application is created, you often need to turn it into a product within Eclipse. Let's take the "Hello RCP" application we've created and make it into a product.

To do this, we create a product configuration file. The product configuration file defines and manages all aspects of an Eclipse product and has the .product extension.

Some of the product's aspects to be defined are the product launcher name, about dialog and splash screen.

The product configuration file is used at build time to build and deploy the product.

ADDING A MENU AND A MENU ITEM

The next step in creating an application often involves adding menus and menu items. To enhance our basic RCP application, let's add a menu and a menu item, which, when selected, will open a dialog with the "Hello" message. To add the menu, enter the following code in the fillMenuBar method in the ApplicationActionBar Advisor:

```
// create a Hello menu
MenuManager HelloMenu = new MenuManager("&Hello");
// add the menu to the menu bar
menuBar.add(HelloMenu);
```

If you launch the application at this point, no menu will appear, since you haven't added any menu items. To add a menu item to the menu you just created, define an action to open a dialog with the message "Hello!" by creating the action class OpenHello DialogAction.java, as shown in Listing 2.

CREATING AN ACTION

To open a dialog window, you need to create an action. Enter the following code in the makeActions method of ActionBar Advisor:

```
// create the action
helloAction = new OpenHelloDialogAction(window);
```

Add the action to the menu by entering the following code in the fillMenu method of ActionBarAdvisor:

```
// add the menu item, in this case the action
helloMenu.add(helloAction);
```

Now launch the application. Click the Open Info menu selection, and the message dialog is opened.

Select File > New > Product Configuration to launch the Product Configuration wizard. In the Product Configuration file wizard, select org.eclipse.rcpapp.hello as the parent directory, enter helloRCP.product for the file name, and select the basic configuration file. The product configuration file is generated and opened in the editor.

The product configuration file editor is used to manage and edit all the information contained in the product configuration file. Fill in the product name and specify org.eclipse.rcpapp.hello.application as the application to be run.

Click the New button for the Product ID field to create a new product ID, org.eclipse.rcpapp.hello.product. This is defined in the plugin.xml by the org.eclipse.core.runtime.products extension:

```
<extension
    id="product"
    point="org.eclipse.core.runtime.products">
    <product
     application="org.eclipse.rcpapp.hello.application"
     name="HelloRCP"/>
</extension>
```

In the Configuration tab, Click Add to add org.eclipse.rcpapp.hello to the list of plug-ins that constitute the product. Now, click Add Required Plug-ins to add all the other plug-ins that are needed to run the application.

In the Launcher tab, enter hello as the Launcher name. Now, the executable will be named hello.exe.

If the program fails to launch when you click Launch the Product, go to the Run Launch Configuration for helloRCP.product, make sure Run a Product is selected and points to org.eclipse.rcpapp.hello.product.

Figure 2 shows a sample of the Hello World Product Definition window.

TESTING THE UI

We can select a variety of open-source and commercial test frameworks to facilitate GUI testing. Let's use the Abbot open-source framework to write a test to exercise the menu in the "Hello RCP" application.

Download the abbot_1.0-rc1-eclipse-3.0.zip (http://sourceforge.net/projects/abbot) and install it in the plug-ins directory of your Eclipse installation. Create a plug-in test project by selecting File > New > Plug-in Project, then name the project org.eclipse.rcpapp.hello.test. Add these plug-ins to the dependencies, using the manifest editor:

- `org.eclipse.rcpapp.hello`
- `org.junit`
- `abbot`
- `abbot.swt.eclipse`
- `abbot.swt.hierarchy`

Figure 2 | A sample of the Hello World Product Definition window

Create a a test package, then a HelloRCPTest class. The testMenuAndDialog method uses an Abbot BaseOKCancelDialogTester to invoke the message dialog upon selection. Click OK to close the message dialog.
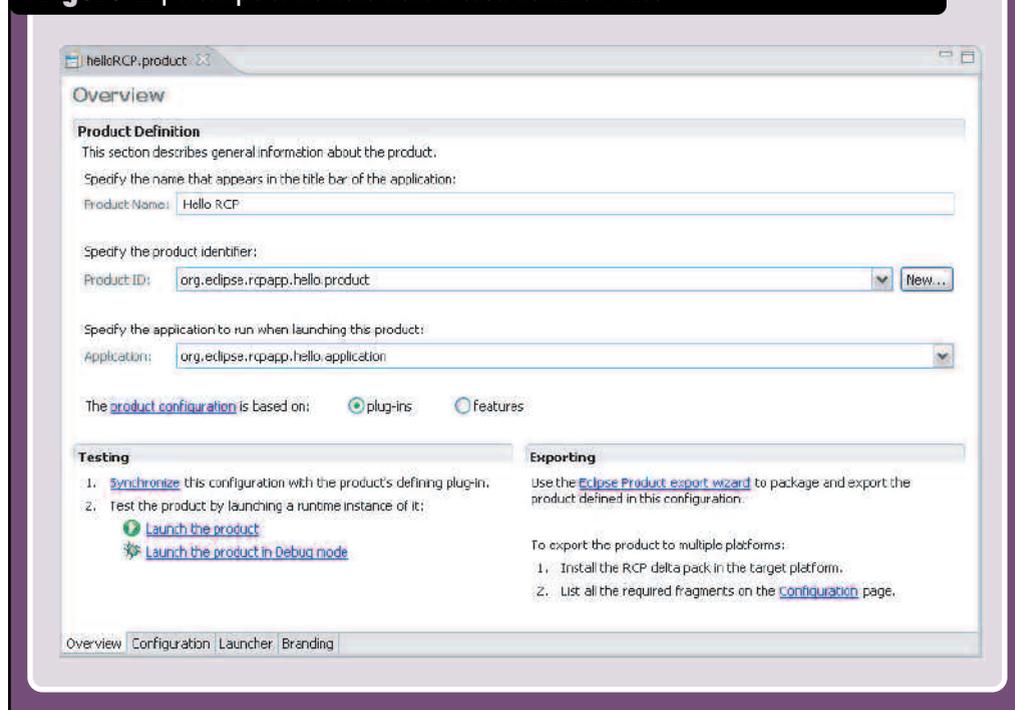
The next step is to create a JUnit Plug-in test launch configuration for the test. To open the launch configura- tion dialog, select Run… from the Run menu. The launch configuration is used to specify the program to be run; the arguments, if any are necessary; and the plug-ins needed in the runtime workspace. Select JUnit Plug-in Test, and click the New button. A new JUnit plug-in test launch configuration is created. Name it testHelloRCP. Select Run a Single Test and enter org.eclipse.rcpapp.hello.test as the project. Then, select test.HelloRCPTest as the test class. Now the launch configuration is as shown in Figure 3.

In the Main tab, select Run a Product and org.eclipse.rcpapp.hello.product. Now run the test. You'll see the "Hello RCP" application window open. When the Open Info menu is selected, the message dialog pops up and is closed by clicking the OK button, and the application exits.

Congratulations! You have now successfully run a GUI test for the "Hello RCP" application!

DEPLOYING THE PRODUCT

Eclipse offers a wide variety of options to package an application for deployment. To export the product, select Export > Plug-in Development > Eclipse Product from the File Menu. Select Export to archive, type in hello.zip as the name of the archive, and click Finish. The "Hello

Figure 3 | Selecting "Run a Product" and org.eclipse.rcpapp.hello.product runs the Abbot test
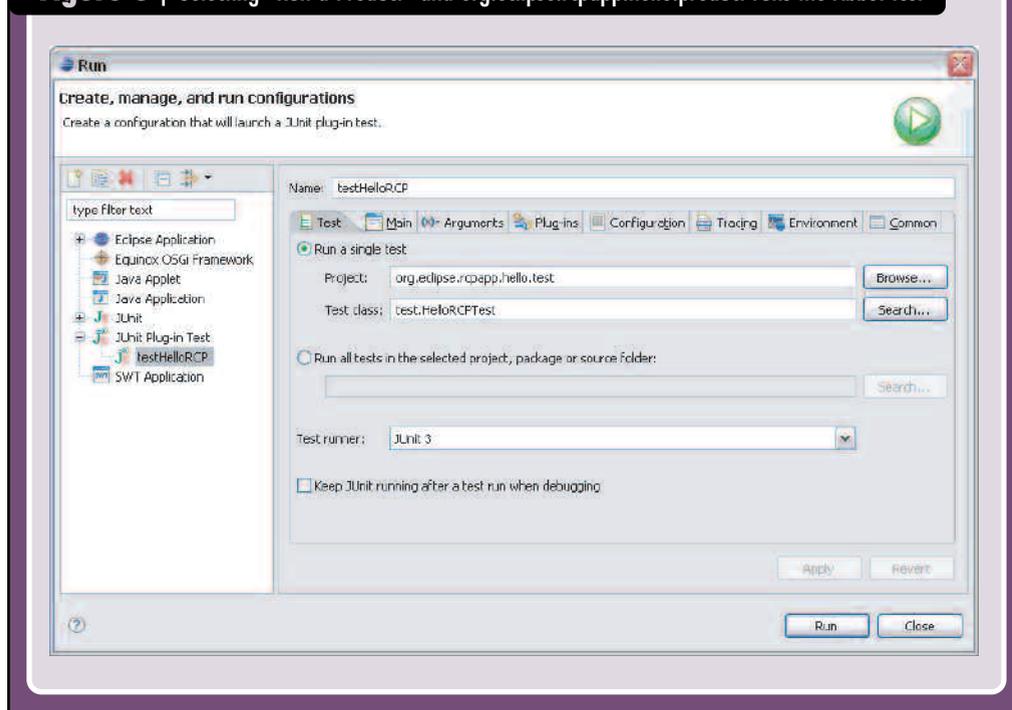
RCP" product is now ready to be deployed. Exporting to a directory can be used in conjunction with a commercial installer to create a professional-looking deployment executable.

INSTALLING AND EXECUTING THE APPLICATION

In the final step in testing the "Hello World" product, we'll install and execute the application from the hello.zip file. This will test what end users would do in installing an Eclipse application that you've created. Unzip the hello.zip file in the destination directory. The "Hello RCP" application is ready to run. Click the hello.exe file to launch the application. If your executable fails, check your entries against the code and screenshots in this article. If it runs, give yourself a hand—you've successfully created your "Hello World" Eclipse application!

REFERENCES

- "Eclipse: Building Commercial Quality Plug-Ins," by Eric Clayberg and Dan Rubel (Addison-Wesley, 2006)
- "Eclipse Rich Client Platform," by Jeff McAffer and Jean-Michel Lemieux (Addison-Wesley, 2006)
- The Abbot Java GUI Test Framework (http://abbot.sourceforge.net/doc/overview.shtml)

Listings

■ LISTING 1: THE EXTENSION POINT THAT DEFINES THE APPLICATION.JAVA CLASS

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.2"?>
<plugin>

   <extension
         id="application"
         point="org.eclipse.core.runtime.applications">
      <application>
         <run
               class="org.eclipse.rcpapp.hello.Application">
         </run>
      </application>
   </extension>
   <extension
         point="org.eclipse.ui.perspectives">
      <perspective
            name="Hello Perspective"
            class="org.eclipse.rcpapp.hello.Perspective"
            id="org.eclipse.rcpapp.hello.perspective">
      </perspective>
   </extension>
   <extension
         id="product"
         point="org.eclipse.core.runtime.products">
      <product
            application="org.eclipse.rcpapp.hello.application"
            name="Hello RCP"/>
   </extension>
 </plugin>
```

■ LISTING 2: OPENHELLODIALOGACTION.JAVA

```java
/***
 *
 * Action to open a dialog with a Hello! message
 *
 */
public class OpenHelloDialogAction extends Action {

        private final IWorkbenchWindow window;

        public OpenHelloDialogAction(IWorkbenchWindow window){
                this.window = window;
                setText("Open Info");
                setId("helloRcp.hello");
        }

        public void run(){
                // open the message dialog
                if(window != null) {
                        try {
                                MessageDialog.openInformation(window.getShell(),
                                        "Hello RCP Application", "Hello!!!");
                        } catch (Exception e) {
                                MessageDialog.openError(window.getShell(), "Error",
                                        "Error opening dialog" + e.getMessage());
                        }
                }

        }
  }
```

- Security services for authenticating user access, managing user credentials, data encryption, digital certificates, secure communication with servers and providing rich client applications with security "sandboxes" to protect the client device from attempts to download information from servers and access local files.

- Desktop integration services include platform-independent access to native editing features like drag-and-drop, clipboard cut, copy and paste, edit undo and redo, and so on. They also include services for application embedding, enabling rich client applications to run native application components (like browsers, spreadsheet, graphing and word processor objects, etc.) within the application window or for the rich client application itself to be embedded within a native application.

- Support for offline computing for cacheing application code and data, extensions for synching cached data with the server after connectivity is reestablished.

I call these *Enterprise Web 2.0* frameworks, which make your RIAs richer, faster and easier to maintain while preserving low-cost deployment and administration.

THE ONE-TWO PUNCH

If you're not already building rich client applications, get ready to start, because that's where Web development innovation is taking place, resulting in easier-to-use applications that deliver more value to users and more market differentiation to vendors of software applications. Invest in both RCP and RIA skill sets and technologies, and always focus on requirements when deciding on an approach. Aim for an applications strategy that features an Enterprise Web 2.0 framework that unites the richness of RCP with RIA ease of development. ■

Table 1	Deciding Between an RCP and an RIA	
Factor	**RCP is best if…**	**RIA is best if…**
Usage Model	Your client application needs to: • Support complex, nonlinear UI navigation. • Mostly operate offline (as in developer tools and office suites). *RCPs provide robust UI frameworks, access to client resources and client/server messaging, etc., making it easy for highly skilled (and expensive) RCP developers to build support for the features above.*	Your client application: • Supports linear UI navigation, repeating a standard sequence of steps, such as in online commerce and data entry. • Will be connected to a server. • Runs as part of an existing Web-based application. *RIA applications are browser-based and require a server connection to run. They feature advanced UI widgets like trees and charts, but enabling nonlinear UI navigation can be complex and hard to maintain.*
Application Lifecycle	Application longevity and maintainability are important, and client updates will occur relatively infrequently. *I find RCP applications easier to maintain due to the robust object modeling, reuse and inheritance enabled by RCP languages (e.g., Java, C#). This can be out weighed by the need to update and synchronize client installations for each change.*	Application updates will occur frequently, and code maintainability is less important. *RIA scripting languages lend themselves well to rapid development, but are harder to maintain than RCP languages. On the positive side, application changes don't require client-side updates, making them easy to deploy.*
Project Scope	Your client application scope is complex, and your development team is larger. *RCP languages have strong facilities for team development and code reuse, making them well suited for large, complex project scopes, but they require more expensive OOP developers.*	Your client application scope is simpler and is developed by a small to medium-sized team. *RIA toolkits and scripting languages make it easy to access server-side components, but aren't well suited for implementing complex client-side logic, nor do they have especially strong support for team development.*
Deployment	• You have control over users' client configurations. • The client application must run as a desktop application, not in a browser. • The client application must interact with local resources—files, other client applications, etc. *A virtual machine (VM) executes the RCP application—as a desktop application or in a browser—and enables access to local resources. However, the client machine must be capable of downloading and running the VM. Due to IT security policies or client incompatibility, this may not always be the case.*	• You're not sure all users will be able to download and run the required VM. • Your application must run in a browser. • The client application doesn't need to access client resources. *RIAs don't require a client VM, making them faster to start and easier to distribute to a broader group of users running on a wider variety of client devices. Because of strict client-side security, they lack an easy and portable way to access client-side resources.*
Application Logic	Your application features lots of client-side logic—especially involving access to client resources or support for offline computing. *RCP programming languages have well-developed capabilities for defining business logic and accessing local resources. In fact, they are the same languages used to implement server-side service oriented architecture (SOA) components.*	You need or want a choice of where application logic lives. *RIA logic can reside mostly on the server or the client (or anywhere in between), but RIA scripting languages make it easier to access server-side business logic than implement client-side logic, which can be hard to maintain over time.*

The Battle for Your Application: RCP vs. RIA

The Web has revolutionized software development by making it easier to create and deploy client applications. Though simple to use, Web user interface (UI) capabilities often limit their utility. They reduce usability by adding click-and-wait page refreshing and limiting usage of keyboard accelerators and shortcuts. They feature just enough UI widgets to make them good at collecting form data and leading users through a series of pages to complete a transaction. And they generate a lot of network traffic transferring Web pages between browser and server, slowing performance.

Rich client platforms, like the Eclipse RCP, address the UI issues with robust frameworks (SWT and JFace), and address deployment issues with update management. With desktop application development seemingly removed, you may believe that the end of HTML browser-based Web applications is near.

But along comes AJAX. Combining JavaScript and XML-based UI definitions, it overcomes the traditional UI constraints of browser-based applications, enabling the development and easy deployment of Rich Internet Applications (RIAs) like GMail and Google Maps. Thus, there are two rich-client technology segments competing for the right to address your client application requirements.

If both platforms offer similar benefits, how do you make the right choice for your particular application? Having architected many RCP and RIA applications, as well as a framework that unites them, here are the guidelines I follow when selecting an approach.

Robert Buffone is principal software architect at Nexaweb Technologies, which sells rich Internet application development tools.

DEALING WITH THE DILEMMA

The best way to cut through the hype is to focus on application requirements. RCP and RIA excel in different scenarios, and I find the following five factors to be helpful when deciding which approach to use:

- *Usage Model:* How will the application be used? Online or offline? Casual or prolonged use? Will the UI drive the user or vice versa?
- *Application Life-cycle:* Will the application be in production for months or years? Will requirements change frequently? Will maintainability be a factor?
- *Project Scope:* Will the development team be large or small? What developer skills are available? Will the client requirements be complex or simple?
- *Deployment:* Will the application be deployed to one client platform or many? Will it be browser-based? Do you have control over users' client configurations?
- *Application Logic:* How complex are the business logic requirements? Will the logic be server-

or client-based? Is access to client-side resources required?

WILL THEY CONVERGE?

I've often wished RCP could be more like RIA, and vice versa. In fact, they can be. Organizations are beginning to implement home-built or commercial Java and advanced AJAX-based frameworks that extend RIAs with powerful RCP-like capabilities, such as:

- *Rich UI services* for generating cross-platform rich user interfaces and binding them to business logic components.
- *Application object life-cycle management services* for implementing more robust, maintainable client-side logic; services often include instantiating, controlling and terminating UI controls and application sessions.
- Persistent storage services enable the application or a group of rich client applications to store and access information on the client device.
- Peripheral device services provide cross-platform access to printers, bar-code readers and other peripheral equipment connected to the client device.
- Messaging services include connection status monitoring and notification, communications (various forms of message sending and receipt: publish/subscribe, message broadcasting, real-time push and alerts, and so on) and message integrity (message queuing with a persistent queue and guaranteed order of delivery).

Sybase WorkSpace:
Do Something More Interesting

A SINGLE ENVIRONMENT OFFERS WHAT YOU NEED

Sybase WorkSpace is a Java™ toolkit offering the five most important design and development tools in an integrated easy-to-use, open source framework:

- database development
- web application development
- services-oriented development
- mobile development
- enterprise modeling

MAKES DEVELOPMENT AND DESIGN EASY TO LEARN AND USE

Using industry-leading integrated model-driven design, visual development and task-based wizards, Sybase WorkSpace automates mundane tasks and cuts the typical development tool learning curve, freeing developers to concentrate on what's important—business logic.

FREEDOM TO CUSTOMIZE BASED ON YOUR BUSINESS NEEDS

WorkSpace's modular packaging allows enterprises the flexibility to decide how to assign business critical tasks to developers, rather than requiring them to fit into ill-fitting "roles" predetermined by a vendor. You buy only the pieces you need, enabling you to customize your environment as you see fit.

With Sybase WorkSpace, you're finally free to do the design and development that's interesting to you. For more information and to download White Papers and an evaluation copy, visit **www.sybase.com/workspace**

SYBASE®

Software Test & Performance CONFERENCE

November 7-9, 2006
The Hyatt Regency Cambridge
Boston, MA

Register Today for Early Bird Discounts! See Back Cover

Course Listing

SUPERB SPEAKERS!

Scott Barber
Rex Black
Clyneice Chaney
Ross Collard
Elfriede Dustin
Jeff Feldstein
Robert L. Galen
Robin F. Goldsmith
Hung Q. Nguyen
Robert Sabourin
Mary Sweeney
And dozens more!

TOTAL IMMERSION!

Choose From 67 Classes

Pick From 8 In-Depth Tutorials

NEW! Birds-of-a-Feather Sessions

Network With Colleagues

Ice Cream Social

Reception in Exhibit Hall

Pose Your Questions to 25+ Exhibitors

Mingle With More Than 40 Speakers

TERRIFIC TOPICS!

Managing Test/QA Teams
Testing SOA and Web Services
C# and ASP.NET Test
Security Testing
Locating Performance Bottlenecks
Just-in-Time Testing
Effective Metrics
Requirements Gathering
Improving Java Performance
Agile Testing
Risk-Based Testing Strategies
Test Automation

Produced by **BZ Media**

SD Times
SOFTWARE DEVELOPMENT
The Industry Newspaper for Software Development Managers

Software Test & Performance

Welcome

Save the Dates! November 7-9 at the Hyatt Regency Cambridge in Boston

From service-oriented architectures to AJAX to application security, change is the one constant in software development. In the face of continuous innovation, experts like you who are trying to improve the quality of your company's software encounter new challenges every day. And with code size and complexity increasing year by year, is it any wonder your information needs continue to grow as well? Now in its third year, BZ Media's Software Test & Performance Conference provides you with the practical, how-to information that will help you meet these challenges and make you successful in your profession.

The technical program for this conference was designed to serve the needs of people just like you: test and QA managers, development managers, test-focused developers and senior testers. The conference addresses such diverse topics as requirements management, security testing and test automation.

You can learn about testing service-oriented architectures and explore the fundamentals of database testing and how to recognize performance bottlenecks. Or you can delve into the intricacies of profiling J2EE applications, learn about performance tuning .NET applications and understand how to use metrics effectively to improve software quality. The three-day conference program packs in eight daylong tutorials plus more than 60 classes ranging from 60 to 90 minutes in length.

The faculty was handpicked for its technical expertise and ability to communicate. You'll meet and learn from industry luminaries like Rex Black, Rob Sabourin, Robin Goldsmith and Bob Galen. The program also features exciting keynote presentations to help give you a sense of where the industry is headed and what challenges you'll likely be facing next year.

While participating in the technical program is important, equally valuable is the opportunity you will have to meet with other software professionals outside the classroom. Conference activities are arranged to maximize your learning experience while leaving you time to compare notes with your classmates and confer with members of the faculty.

As an added bonus, the conference schedule and format will provide time for you to discover the latest products—which will be presented in the exhibit area—and pick the brains of the tool vendors. Read through the class listings and build a custom course of study over three days that will give you and your team tools and techniques that you can take back to the office and put into effect immediately. We look forward to seeing you at the Software Test & Performance Conference.

Register today at www.stpcon.com

Lindsey Vereen
Conference Chairman

NEW!
BIRDS OF A FEATHER SESSIONS

Join your colleagues and our instructors for informal discussions. Pose your toughest questions on the following break-out topics:

- Test Automation
- .NET
- Testing for Security
- Java
- Agile Testing
- Improving Performance

Tuesday, November 7
8:00 pm – 10:00 pm

Wednesday, November 8
8:30 pm – 10:30 pm

Note: Sign-up for these sessions will be on-site.

Monday, November 6

4:00 pm – 7:00 pm	Registration

Tuesday, November 7

7:30 am – 7:00 pm	Registration
8:00 am – 9:00 am	Continental Breakfast
9:00 am – 10:30 am	Full-Day Tutorials
10:30 am – 11:00 pm	Coffee Break
11:00 am – 12:30 pm	Full-Day Tutorials
12:30 pm – 1:45 pm	Lunch Break
1:45 pm – 3:15 pm	Full-Day Tutorials
3:15 pm – 3:45 pm	Coffee Break
3:45 pm – 5:00 pm	Full-Day Tutorials
8:00 pm – 10:00 pm	Birds of a Feather Session

Wednesday, November 8

7:30 am – 7:00 pm	Registration
7:30 am – 8:30 am	Continental Breakfast
8:30 am – 10:00 am	Technical Classes
10:00 am – 10:30 am	Coffee Break
10:30 am – 12:00 pm	Technical Classes
12:00 pm – 1:15 pm	Lunch Break
1:15 pm – 2:30 pm	Technical Classes
2:30 pm – 3:00 pm	Coffee Break
2:30 pm – 7:00 pm	**Exhibit Hall Open**
3:00 pm – 4:15 pm	Technical Classes
4:15 pm – 4:30 pm	Break
4:30 pm – 4:45 pm	Industry Keynote
4:45 pm – 5:30 pm	Keynote (Rex Black)
5:30 pm – 7:00 pm	**Reception in Exhibit Hall**
8:30 pm – 10:30 pm	Birds of a Feather Session

Thursday, November 9

7:30 am – 4:00 pm	Registration
7:30 am – 8:30 am	Continental Breakfast
8:30 am – 10:00 am	Technical Classes
10:00 am – 10:30 am	Coffee Break
10:30 am – 12:00 pm	Technical Classes
12:00 pm – 1:15 pm	Lunch Break
12:00 pm – 4:00 pm	**Exhibit Hall Open**
1:15 pm – 2:00 pm	**Ice Cream Social in Exhibit Hall**
2:00 pm – 3:15 pm	Technical Classes
3:15 pm – 3:45 pm	Coffee Break
3:45 pm – 5:00 pm	Technical Classes

Exhibit Hours:
Wednesday, 2:30 pm – 7:00 pm
Thursday, 12:00 pm – 4:00 pm

Contents

Register at www.stpcon.com

Updated July 26, 2006

"This is the best conference I have attended. The instructors were extremely knowledgeable and helped me look at testing in a new way."
—Ann Schwerin, QA Analyst, Sunrise Senior Living

FULL-DAY TUTORIALS
Tuesday, Nov. 7
9:00 am – 5:00 pm

NEW

T-1 Assessing Your Test Team Effectiveness, Efficiency and More By Rex Black

As a test manager, you're probably looking for ways to do better and demonstrate your team's value and to find ways to improve it. This tutorial delivers. You'll learn techniques to assess your team that are driven by insightful questions and careful data analysis. By applying the ideas in this tutorial to each of the 12 critical testing processes, you'll know where you and your team stand.

This one-day tutorial is very hands-on with lecture primarily used to stimulate discussion. After each process is discussed, attendees will work through exercises that estimate performance metrics for their own test teams. After each exercise, attendees have a chance to discuss their results.

T-2 Testing Techniques: Theory and Application By BJ Rollison

This tutorial presents the formal theory and practical application of functional (behavioral) and structural (coverage) testing techniques. This tutorial will teach functional testing techniques, including exploratory testing, boundary value analysis, equivalence class partitioning and combinatorial analysis. Structural testing techniques covered include statement coverage, decision/branch coverage, condition and basis path coverage.

By attending this tutorial, you'll learn how to use functional testing techniques to establish a solid foundation and minimum baseline of test cases. You'll understand how structural testing techniques can be used to design additional tests from a white box approach to complement the test effort, to ensure that critical paths in the code have been exercised and to achieve higher code coverage results. You will also learn how to apply both black box and white box test design approaches to test more effectively.

T-3 Testing in Highly Iterative, Quasi-Agile Projects—Practical Strategies for Mixed Culture Projects By Timothy D. Korson

In the highly iterative, fast-paced environment of agile development projects, the traditional approaches to testing, quality assurance, requirements gathering and team interactions break down. QA managers trying to encourage best practices recommended by CMMI and SPICE find themselves at odds with developers trying to adopt best practices as recommended by the Agile Manifesto.

In the end, no one wins. Because of the constraints of corporate policies and management edicts, developers can't fully adopt agile practices. Because the developers do adopt as much of the agile process as they can get away with, the QA team finds that traditional approaches to quality management no longer work. Such projects must succeed in a "quasi-agile" development environment.

This tutorial will introduce you to software development processes and practices that affect your world. You will learn practical strategies for effectively integrating testing processes with modern software engineering processes. You will learn how to create effective tests, both component-level and system-level, for modern software systems. Detailed case studies will convey specific techniques for testing both components and entire systems.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

T-4 Twenty-One Ways to Spot—and Fix— Requirements Errors Early By Robin Goldsmith

While many organizations have begun paying closer attention to defining requirements, few fully realize the need to know that their requirements are accurate and complete, and few know how to test requirements effectively. Most rely on one or two weak methods and have little awareness of how many errors they've missed— errors that later turn into expensive feature creep. This interactive class explains why it's so hard to test requirements, and it introduces 21 increasingly powerful methods to help you find frequently overlooked requirements errors when they are easiest and least expensive to fix.

Following the instructor's proven CAT-Scan Approach, participants apply the techniques successively to a real case and discover how each different method reveals additional, otherwise overlooked defects in the requirements. Participants learn ways to find previously overlooked requirements, increase meaningful customer/user involvement, enhance communications and understanding and test the adequacy of requirements definitions.

Special Track!

Development or Test/QA Managers' Classes

Need a higher-level view? You've come to the right place! Here are classes just for you—but feel free to deep-dive into any class or topic you want!

T-1	Assessing Your Test Team Effectiveness, Efficiency and More – **Black** (Full-Day Tutorial)
T-5	Foundations of GUI Test Automation – **Makedonov** (Full-Day Tutorial)
107	Designing for Testability – **Feldstein**
108	Testing the Software Architecture – **Sangal**
109	Effectively Training Your Offshore Test Team – **Hackett**
203	How to Turn Your Testing Team Into a High-Performance Organization – **Hackett**
207	Analyze the Return on Your Testing Investment – **Black**
209	Quality Throughout the Software Life Cycle – **Feldstein**
304	Managing Acceptance Testing Cycles More Efficiently – **Makedonov**
403	Deciding What Not to Test – **Sabourin**
502	Creating and Leading the High-Performance Test Organization, Part 1 – **Galen**
509	Recruiting, Hiring, Motivating and Retaining Top Testing Talent – **Feldstein**
602	Creating and Leading the High-Performance Test Organization, Part 2 – **Galen**
702	S-Curves and the Zero Bug Bounce: Plotting Your Way to More Effective Test Management – **Bradshaw**
705	Coding Standards and Unit Testing—Why Bother? – **Sokolov**
707	Performance Testing for Managers – **Barber**
801	Effective Metrics for Managing a Test Effort – **Bradshaw**
806	Best Practices for Managing Distributed Testing Teams – **Stevens**

NEW **T-5 Foundations of GUI Test Automation**
By Yury Makedonov

Testers and managers find themselves between a rock and a hard place when implementing test automation. From one side they are bombarded by a constant stream of sales pitches promoting the "click, click, click" record-and-replay approach. From the other side they are pressed by test automation gurus promoting their own frameworks. So, it's a challenge to keep their sanity under these conditions and to make sensible test automation decisions on tool and framework selection and test automation management.

In this real-world tutorial, major myths and misconceptions are dispelled, and explanations are provided on how to ensure the efficiency of GUI test automation. The tutorial covers major principles and current industry standards of GUI test automation, how to decide if a specific project should be automated or not, how to define a scope for test automation, how to select a test tool to automate a specific application and much more.

NEW **T-6 Software Endgames: How to Finish What You've Started** *By Robert Galen*

We've all survived more than one software project that ended badly, where either the requirements were misunderstood or were implemented poorly. Or overall quality targets couldn't be met because there were simply too many defects. Or the team simply couldn't decide on priorities and in which direction to steer the project.

Many projects fail during testing. Not because of the testing per se, but because of the massive discovery of defects and functional gaps that indicate the true viability of the project. I call this time the Software Endgame, and I've spent a great deal of time negotiating its challenges through numerous software projects.

This presentation focuses on a set of high-level practices and techniques that will help improve your management and project steering within the endgame, providing guidance that will increase the odds of successfully delivering a project. You'll learn how to create an endgame delivery map that directs your release and testing milestones via entry/exit criteria; the importance of release criteria within the endgame and high-level rules of thumb for defining them; how to manage defect repairs—where to focus your efforts and scheduling rules of thumb—plus the many options you have for "fixing" defects.

T-7 Using Metrics to Improve Software Testing
By Alfred Sorkowitz

Software metrics can aid in improving your organization's testing process by providing insight and early visibility into the "real" status of the testing effort and helping to make assessments as to whether progress, productivity and quality goals are being met. This tutorial presents a practical guide on how to start taking advantage of these new tools/techniques to aid in improving the testing process. These metric-based tools and techniques have successfully been used by software test teams, software developers and SQA/IV&V staffs.

In this class, you will receive an overview of software quality goals, criteria and metrics. You will also learn:

- The cost of inadequate software testing, a set of government/industry "best practices" metrics that can track the real status, quality and productivity of the testing effort, as well as provide an indication of future problems.
- Software complexity metrics: A new structured testing methodology that uses metrics to aid in developing software that is easier to test and maintain and selecting an appropriate set of paths for more thorough testing.
- How to integrate software metrics into the testing process.

NEW **T-8 Creating Agility and Effectiveness in Software Testing** *By Linda Hayes*

Most companies resist test automation until the software is stable, reasoning that any savings from automation will be offset by the maintenance required to keep up with changes. Also, traditional record/script/replay approaches can't be implemented until the code is functional, which is too late in an agile development environment.

This session will present an incremental approach to test automation that supports an agile development environment by allowing automated tests to be written before the code and then be rapidly updated as changes are introduced. You will learn how to improve code development practices through automation, define executable requirements and write self-documenting automated tests before code is developed. You will also learn how to implement a test tool and platform agnostic automation architecture and automate test case maintenance for rapid response to changes.

TECHNICAL CLASSES
Wednesday, Nov. 8
8:30 am – 10:00 am

101 Seven Low-Overhead Software Process Improvement Methods *By Robin Goldsmith*

For many, software process improvement is synonymous with high-overhead, long-term, organizationwide initiatives that often are resisted and fail to produce the desired results. In this interactive presentation, you'll learn seven methods that can help you make software faster, cheaper and more reliable without all the hoopla. Key to meaningful results is recognizing, measuring and then specifically improving high-payback aspects of the instructor's proven REAL software process, which often differs considerably from what we presume we are doing. In truly agile fashion, applying these methods proficiently focuses efforts most efficiently on effectively producing useful software from the start.

102 Just-in-Time Testing Techniques and Tactics, Part 1 *By Robert Sabourin*

As the Boy Scout credo goes, "Be prepared." In this class, you will learn how to be ready for just about anything in a software testing project within the volatile environment of a Web or e-commerce software project. Managers will learn an array of techniques to manage and track software testing in chaotic environments—specifically, projects with continuously changing requirements and shifting priorities. Members of the development and testing teams will learn how, even while working with minimal information, to develop tests and converge the product development effort.

103 Automated Database Testing: Testing and Using Stored Procedures *By Mary Sweeney*

Today's heterogeneous data environments place an increasingly heavy burden on test engineers. Applications, whether Web-based or client/server, must be tested for seamless interface with the back-end databases; this typically goes far beyond what the popular test automation tools can provide. The intricate mix of client/server and Web-enabled database applications are extremely difficult to test productively. As a result, you are increasingly expected to know how to create and use SQL queries, stored pro-

Register at www.stpcon.com

"Good immersion in testing concerns."
—*James Fields, Development Manager, Data-Vision Inc.*

cedures and other relational database objects to effectively test data-driven environments.

In this class, you will learn about testing at the database layer as an important adjunct to current tests. Using demonstrations and code examples, the instructor will present tips and techniques for creating efficient automated tests of the critical database back end using SQL, scripting languages and relational database objects. You'll learn why testing of database objects and stored procedures is necessary; how simple and effective automated tests for the back end can be created using various programming languages, including PERL and VBScript; and how to successfully test database objects, such as stored procedures and views, with many examples and code.

Special Track!

Need To Tune Performance?

Crank up your applications' performance by choosing from these hot classes!

305	How to Optimize Your Web Testing Strategy – **Nguyen**
402	Accelerate Testing Cycles With Collaborative Performance Testing – **Cavallaro**
408	Techniques for Testing Packaged Application Performance – **Feaster**
503	Pinpointing and Exploiting Specific Performance Bottlenecks – **Barber**
504	Performance Tuning ASP.NET 2.0 Applications, Part 1 – **O'Mara**
603	SOA Performance Testing Challenges – **Barber**
604	Performance Tuning ASP.NET 2.0 Applications, Part 2 – **O'Mara**
608	Verifying Software Robustness – **Collard**
707	Performance Testing for Managers – **Barber**
803	Real-World Performance Testing Lab for (Almost) Free – **Flint**
805	Building a Bridge Between Functional Test Automation and Performance Testing – **Sody**

104 Lessons Learned in Test Automation, Part 1 *By Elfriede Dustin*

This class will present and discuss a series of automated testing lessons learned from actual experiences and feedback from real projects. You'll learn how to avoid some typical false starts and roadblocks when you implement your test automation efforts.

Part 1 of this class includes a discussion of better ways to define automation criteria, how to avoid duplicating the development effort when designing automated test cases, how to create reusable automated test cases, the need to verify all vendor claims in your own environment, the pitfalls of delegating the tool selection to a reseller or consultant and how to select the right tool. You'll also learn how to avoid losing sight of the testing efforts because developers or testers are too busy coming up with elaborate scripts to automate their unit and system tests.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

NEW 105 Code Coverage Metrics and How to Use Them *By Rex Black*

More and more testers and programmers are using tools that provide code coverage metrics. These metrics tell the tester or programmer how much of the code has been covered by a given set of tests and, more important, what conditions might not be covered. In addition, some tools can evaluate the coverage of data flows. Some tools can also provide insight into the complexity and, thus, the likely difficulty level of future refactoring of the code.

In this practical class, we'll examine the following code coverage metrics and how you can use them to write better code or tests: statement, branch, condition and loop coverage metrics; McCabe Cyclomatic complexity and basis path coverage; and data-flow coverage via set-use pairs. Each metric will be illustrated using a real program, with tests developed and run to achieve given levels of coverage. The code sample provided can be used, along with the course materials, to evaluate a given tool's ability to provide useful code coverage metrics.

NEW 106 Foundations of GUI Test Automation Using C#, Part 1 *By BJ Rollison*

There is a trend in the software industry for companies to seek candidates whose qualifications include the ability to write automated tests. Additionally, some companies realize the limitations of commercially available tool sets that use proprietary scripted languages. So, modern programming languages are being employed to develop more effective and more robust automated tests. But for testers who lack a programming background, the initial hurdle of learning a programming language can be a bit intimidating. This is further complicated by the fact that most programming courses teach us how to develop applications and not how to use a programming language to write automated tests.

This class is designed as a starting point for testers who lack a programming background, or for those with an understanding of programming concepts but are unfamiliar with automated testing using C# to test a Windows application. This class discusses common testing tasks illustrated with well-commented code examples and developed with free tools available on the Internet. In this session, you will learn how to launch, gather information, synchronize and close an application under test (AUT); manipulate and send test data to an AUT; and generate test data.

NEW 107 Designing for Testability *By Jeff Feldstein*

Many developers believe testing begins when all the features are complete and they hand off their work to the test team. While this may be what actually occurs in many development projects, it is far from the ideal. Software quality assurance begins in the definition phase of the project. One important aspect of software quality to consider is testability of the software architecture. Testability of your application can have a profound effect on its overall quality. This class will explain what to look for in a testable architecture, avoiding common mistakes and pitfalls, how to present your ideas to the development team and how to build test automation systems that take advantage of the testable architecture.

NEW 108 Testing the Software Architecture *By Neeraj Sangal*

Automated testing of the software architecture can keep quality from degrading and help preserve the design intent. In this class, a new lightweight approach to specifying and verifying the architecture is presented. Inter-module dependencies are utilized to represent the architecture of a software system using a dependency structure matrix (DSM). Once the architecture is specified, it can then be verified through automated tests during development. Furthermore, architectural violations can be easily prioritized for remediation. This

"Excellent conference—provided a wide range of topics for a variety of experience levels."
—Carol Rusch, Systems Analyst, Associated Bank

Technical
Classes

approach will be presented through real life examples by applying it to a number of commonly used applications. Dependency analysis will be used to extract the architecture for applications such as Ant, JUnit, jEdit and the Eclipse platform. We will look at real examples of software development spanning several years to see how architecture evolves, how it often begins to erode and how regular testing can prevent this erosion.

109 Effectively Training Your Offshore Test Team *By Michael Hackett*

Working with offshore teams is a fact of life now for domestic test leads and managers, but many are still struggling to make their global test team work effectively. Training your offshore test team is critical to the success of your projects. If done right, training can help minimize your stress and late-night phone calls and ensure that you are getting the right information from the offshore team to make sure your testing effort is successful. Training of the offshore team needs to focus on a broad range of topics and must be specifically designed to the unique needs of that team.

In this class, we will discuss the key elements of successful offshore testing, including training in the areas of process, product/domain knowledge and testing techniques, and how training can be used as a retention tool for offshore staff. The class will include several real world examples based on the speaker's experiences working with teams in the most common offshoring locations.

Wednesday, Nov. 8
10:30 am – 12:00 pm

NEW ### 201 Prevent Showstopper Overruns With Risk-Based Proactive Testing
By Robin Goldsmith

Project budget and schedule overruns frequently are caused by late, unplanned significant redesign and rework to fix showstopper errors. Traditional, reactive testing misses too many showstoppers or catches them too late to fix easily. In contrast, Proactive Testing's powerful risk analysis techniques identify many of the up to 75 percent of showstoppers that are ordinarily overlooked. Moreover, Proactive Testing can drive development to build systems in a truly agile different way that avoids much of the rework that showstoppers traditionally would have necessitated. The class will cover risk-based testing fundamentals and the limitations of traditional reactive testing approaches. You will learn how to continually refocus on testing higher-level risks more frequently and earlier in the testing cycle, as well as methods for identifying ordinarily overlooked showstoppers and reducing overruns.

202 Just-in-Time Testing Techniques and Tactics, Part 2 *By Robert Sabourin*

Please see description under Class 102.

203 How to Turn Your Testing Team Into a High-Performance Organization *By Michael Hackett*

All development managers, test managers and their organizations are looking for ways to improve quality. Quality improvement can come in many forms: reducing risks by delivering higher and predictable quality products; optimizing time-to-market; increasing productivity; and building a more manageable organization. Some managers look for quality improvement by attempting to implement a more standard or formal process. This sounds good, but where is the road map for how to get there? This class will help! You'll learn how to evaluate your test process and strategy, create a culture for change, implement change and use effective methods for measuring improvement.

204 Lessons Learned in Test Automation, Part 2 *By Elfriede Dustin*

We continue to explore automated testing lessons learned from actual experiences and from feedback based on real projects to help you to avoid some typical false starts and roadblocks when you implement test automation efforts. In Part 2 of this class, attendees will learn when automated testing doesn't speed up the testing effort. Attendees will also learn how to create mini development life cycles, how to maintain automated unit and system tests, how to implement smoke tests, and the pitfalls of using automated performance testing tools.

205 Database Security: How Vulnerable Is Your Data? *By Mary Sweeney*

There are many levels of software security. But how secure is the most important component of your application: your database? Quality control organizations must step up to the challenge of ensuring data security with appropriate tests that focus on this vital area. In this class, you will learn what the test team needs to know about protecting the server, the database connections, controlling access to database tables and restricting access to the database server itself. If data is in jeopardy, the entire system is at risk. We will also discuss the basics of security testing to ensure protection for the critical database component.

NEW ### 206 Foundations of GUI Test Automation Using C#, Part 2 *By BJ Rollison*

C# is a powerful programming language quickly becoming commonly used to develop test automation. But to effectively test Windows (Win32) applications written in C/C++, we must also learn to use common Win32 Application Programming Interfaces (APIs).

Part 2 of this class discusses how to use process invocation services to use Windows APIs to perform common testing tasks illustrated with well-commented code examples, and developed with free tools available on the Internet. In this session, attendees will learn how to:
• Import Win32 API library functions and marshal data types
• Get and set AUT focus

Register at www.stpcon.com

- Manipulate controls
- Manipulate menus
- Create custom methods to perform repetitive tasks
- Create a reusable test library

207 Analyze the Return on Your Testing Investment *By Rex Black*

Testing is not just a good idea—it's a good investment. This class demonstrates the value of solid testing through quantifiable returns on the investment. Through effective developer testing, skilled development managers deliver solid, quantifiable benefits in four ways:

- Find bugs that get fixed—or even prevent them
- Find bugs that don't get fixed but are known
- Run tests that mitigate (potentially expensive) risks
- Guide the project with timely, accurate, credible information

Attendees will learn about all of those benefits—and how to measure them. A hands-on exercise teaches you how to estimate the return you could—or currently do—achieve on your test investment.

208 Using Scrum to Manage the Testing Effort
By Robert Galen

Many testing efforts succumb to management and project pressures and become chaotic in terms of their focus and work quality. It's simply the nature of the endgame phase of software development projects, where anything goes in pushing for the delivery of a product, and it's usually quality that goes first. Beyond the product quality impacts, the team usually suffers, too, with low morale and little empowerment.

Scrum is one of the agile methodologies, and it focuses on project management in agile and iterative development efforts. It can be successfully applied to testing efforts to renew their focus and drastically improve overall results. In this presentation, we will explore the Scrum methodology and learn to apply it practically to your testing cycles. In this class, attendees will learn how the Scrum methodology applies to the testing effort.

NEW 209 Quality Throughout the Software Life Cycle *By Jeff Feldstein*

Software quality is everybody's job. Quality cannot be tested into the product; it must be emphasized, monitored and measured from the beginning of the project. Each team involved in the project, including product marketing managers, program managers, development engineering, documentation and test engineering, plays a key role in assuring software quality. A carefully planned application development life cycle is a key requirement to successful delivery of on-time quality software.

The application development life cycle consists of four broad phases: requirements, development, test and post-customer ship. Each phase has important activities that directly affect the quality of the delivered software. This presentation will explore each phase in detail from a software quality perspective. It will describe activities that need to happen at each step and the role of the test or software quality engineer, and will enumerate many common mistakes made. In addition you will learn how to catch bugs earlier in the life cycle when they are cheaper to fix.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

Wednesday, Nov. 8
1:15 pm – 2:30 pm

NEW 301 Hacking 101: Donning the Black Hat to Best Protect Applications From Today's Hacking Threats *By Tom Stracener*

Applications have become fertile ground for attackers to uncover seemingly innocuous features and utilities in today's complex systems and gain unauthorized access. Hackers seek out weaknesses in the many modules and components of complex systems, looking for hidden fields, embedded passwords, exposed parameters to manipulate and ways to tinker with input strings and steal data. At a time when security is highly coveted, understanding the enemy's mindset is more crucial than ever. Therefore, one of the best defense mechanisms against hackers is to understand how they think—and recognize your network's Achilles' heels before a hacker exploits them.

In this class, you'll learn the thinking, strategies and methodologies commonly used by hackers and how to effectively implement a sound defensive plan that will help mitigate multiple attacks. Top Web application security flaws, including invalidated input, broken authentication and management, buffer overflows, injection flaws, insecure storage, denial of service and insecure configuration management, will be addressed so as to arm you with the right knowledge to protect your company's infrastructure.

302 Five Core Metrics to Guide Your Software Endgames *By Robert Galen*

By its very nature, the endgame of software projects is a hostile environment. Typical dynamics include tremendous release pressure, continuous bug and requirement discovery, exhausted development teams, frenzied project managers and long hours. Testing teams are usually in the thick of this battle and accustomed to these dynamics. However, project managers may not be proactive enough in working with their testing teams to understand the change and repair workflows within their projects. Yes, we work hard at managing bug reports, but we can do so much more to influence and focus a project's direction. In this presentation, attendees will learn how project managers can focus the entire team on a few key performance metrics to improve the overall endgame experience and increase the probability of delivering on time. And yes, to also survive yet another endgame.

"Best concentration of performance testing presentations/professionals I've seen."
—Nathan White, Manager, Testing Services, AG Edwards

Technical
Classes

303 Overcoming Requirements-Based Testing's Hidden Pitfalls *By Robin Goldsmith*

Testing based on requirements is a fundamental method that is relied on extensively. However, its thoroughness frequently can be compromised by traps that testers are not aware of. In this interactive presentation, you'll learn key sources of requirements-based testing oversights, including distinguishing business requirements from system requirements; assessing the extent to which the requirements are complete; the premise of one test per requirement; the appropriate level of test case detail; and developers' inclusion of requirements-based unit tests. The class will also focus on the strengths and often unrecognized weaknesses of requirements-based tests; the importance of testing based on business as well as system requirements and determining how many tests a requirement needs.

NEW 304 Managing Acceptance Testing Cycles More Efficiently *By Yuri Makedonov*

Once in a while a supposedly "almost completed" project requires one more cycle of acceptance testing and then just one more, then another and so on. Surprised management looks on in disbelief as the project spirals out of control down into a bottomless pit of "acceptance testing cycles." Yet, often management does not have a 100 percent clear and correct understanding of what is happening and why it's happening. As a result, their actions might not necessarily be very effective in controlling the situation. In this presentation, you will learn the different reasons for why these testing cycles can happen and specific techniques for getting a project like this out of a tailspin.

305 How to Optimize Your Web Testing Strategy *By Hung Q. Nguyen*

One of the key strategic challenges of Web testing is the dominance of change. Another key challenge is interdependence. Web applications are fundamentally dependent on cooperating tools and processes. Many of the processes, tools and standards in use by groups that do Web testing were originally developed with simpler and less dynamic situations in mind. Used by skilled and thoughtful people, in the context of a clear strategy, these processes and tools can add value. But if we allow them to drive our testing practices, they can easily do more harm than good. In this talk, you will learn how to analyze and optimize your Web testing strategy by selecting the right types of tests, how to execute them at the right time with a balanced number of cycles, and how to drive changes to improve your team's testing throughput.

NEW 306 Model-Based Testing for Java and Web-Based GUI Applications *By Jeff Feldstein*

Classic test automation simply repeats the same tests (with optionally varying data) until it stops failing or the application ships. The problem with this approach is that customers rarely flow through the application in the same sequence as the automation, and thus they are likely to find bugs that the automation missed. Model-based testing is a form of automated testing that brings random and flexible behavior to your automated test cases.

Model-based testing can be used for many types of software or application testing. This class will teach how to implement model-based testing, specifically as applied to Java and Web applications. Part of the course includes a demonstration of model-based testing; you will be able to download the XDE Tester source code used in the demonstration. Although the example application tested by this source code is fairly simple, it contains all of the data structures, concepts and program flow for implementing a large-scale, industrial strength, model-based test system.

NEW 307 Taking AIM—Using Visual Models for Test Case Design *By Robert Sabourin*

Designing test cases is a basic fundamental skill all testers master over time. This workshop teaches a fun graphical technique to help design powerful test cases and choose test data that will surface important bugs fast. These skills can be used in exploratory, agile or engineered contexts—anytime you need to design a test.

Mindmaps are powerful graphical tools used to help visualize complex paths and relationships between concepts. The workshop shows how Mindmaps can be used to visualize test designs and help understand variables being tested, alone and in complex combinations with other variables and conditions. The AIM (Application Input Memory) heuristic is taught through a series of interactive exercises. Real recent project examples are used to demonstrate these techniques. We will look at using some widely available free open-source tools to help implement great test cases and to help focus our testing on what matters and quickly hone in on critical bugs! If you are new to testing, these techniques will remove some of the mystery of good test case design. If you are a veteran tester, these techniques will sharpen your skills and give you some new test design approaches.

NEW 308 Elements of Software Design for Unit Testing *By Thierry Ciot*

Too often, software projects are built around classes and components that incorporate a tangled web of dependencies that can hinder—or even prevent—unit testing. What steps can you take to keep this headache out of your project? This course, for beginner to intermediate programmers/testers and managers, will introduce you to the design guidelines that facilitate unit testing. By building applications around classes and components with clearly defined dependencies, code will be more unit-testable by design. Following these guidelines can also greatly reduce the number of stubs or mock objects needed to perform unit testing. The guidelines presented in this course are based on solutions that have been used in actual projects, and each guideline is illustrated with code examples

Register at www.stpcon.com

"I learned things I didn't know existed! I met people from all ranges of QA, all of whom were brimming with information they were willing to share."

—Rene Howard, Quality Assurance Analyst, IA Systems

that emphasize practical solutions. The class will end with a review of a complete real-world example.

NEW

309 Models for Security Testing in the Software Development Life Cycle
By Ryan Berg

How should security testing be implemented during software development to ensure a more secure product? There is general agreement in the industry that improving software security is a valuable endeavor, but implementing programs that generate positive, measurable results has eluded most companies. Questions arise about the lack of security expertise among development teams and lack of development expertise among security teams, and there is a misconception that the addition of security reviews will ultimately extend development schedules. At the same time, centralized decisions must be made to define security policies, determine what constitutes a vulnerability, and prioritize remediation efforts according to available resources. Organizations need a concrete model for security evaluation and a comprehensive task list detailing the roles and responsibilities for each group involved.

The class will include practical models that give testing responsibility to developers, QA staff or security teams, explaining the specific requirements for each approach as well as expected outcomes.

Special Track!

Secure Your Software!

Securing the network is fine, but it's not enough! These classes will help you test your software for security.

205	Database Security: How Vulnerable Is Your Data? – **Sweeney**
301	Hacking 101: Donning the Black Hat to Best Protect Applications From Today's Hacking Threats – **Stracener**
309	Models for Security Testing in the Software Development Life Cycle – **Berg**
401	The Secure Software Development Life Cycle – **Dustin**
606	Exploiting Web Application Code: The Methodologies and Automation of SQL Injection – **Fisher**
701	The Five Most Dangerous Application Security Vulnerabilities—And How to Test for Them – **Basirico**

Wednesday, Nov. 8
3:00 pm – 4:15 pm

NEW

401 The Secure Software Development Life Cycle *By Elfriede Dustin*

According to Gartner and Symantec, most business security vulnerabilities are now at the application layer. Attackers are focusing their efforts on regional targets, desktops and Web applications that potentially give attackers access to personal, financial or confidential information. Consequently, companies responsible for developing software

Software Test & Performance CONFERENCE

Register at www.stpcon.com

must build security into their products as they are being developed.

This class focuses on application security throughout the software development life cycle. Attendees will learn secure coding guidelines that will help prevent defects from getting into code and that they must adhere to during the development process. They will learn about the Secure Software Development Life Cycle (SSDL) and its relationship to system development starting with the guidelines for security implementations.

The class covers the security program review and assessment activities that need to be conducted throughout the testing life cycle and the secure deployment considerations that have to be implemented. It also addresses the metrics and final review and assessment activities that need to be conducted to allow for adequate and informed decision making.

402 Accelerate Testing Cycles With Collaborative Performance Testing *By Rick Cavallaro*

Testing and tuning the performance of enterprise Web applications is a complex task, undertaken by a team of individuals that may include performance engineers, QA testers, architects, developers, database administrators and related project team members. Fostering communication among these individuals can be challenging and can often lead to testing delays. The process is especially difficult when testers and developers are distributed around the building, around the country or even around the globe.

This session will provide a new methodology for collaborative load testing—an antidote to the iterative, multiweek process based on e-mail and conference calls that most organizations are forced to use today. You will learn:

- The drawbacks of traditional approaches to performance testing
- How to incorporate a team-based methodology for performance testing
- A new solution for collaborative load testing in a Web-based environment
- How outsourcing can impact QA efforts, and what you can do to mitigate that impact

NEW

403 Deciding What Not to Test *By Robert Sabourin*

Software project schedules are always tight. There is not enough time to complete planned testing. Do not stop just because the clock ran out. This presentation explores some practical and systematic approaches to organizing and triaging testing ideas. Testing ideas are influenced by risk and importance to your business. Information is coming at you from all angles—how can it be used to prioritize testing and focus on the test with the most value? Triage of testing ideas, assessing credibility and impact estimation can be used to help decide what to do when the going gets tough! Decide what not to test *on purpose*—not just because the clock ran out!

404 Putting the User Back in User Acceptance Testing *By Robin Goldsmith*

User acceptance testing (UAT) is often a source of consternation. Even though the process takes up considerable user time, too many defects continue to slip through, and users increasingly beg off from participating with claims that they don't have the time. Both effects may be symptoms of professional testers' mistaken conventional wisdom about the nature and structure of UAT. In this eye-opening presentation, you'll learn ways to gain user confidence, competence and cooperation. Plus, you'll learn how to create user-driven UAT that increases user testing competence and confidence.

405 Getting a Handle on Risk: Risk-Based Testing Strategies *By Clyneice Chaney*

With the rapid pace of application development, testing has become

a challenging proposition. Trying to meet tight deadlines and deliver products that meet customer requirements is the greatest challenge testers face today. This presentation discusses a risk assessment tool that is used to assess risks associated with product testing. The assessment tool provides an alternative to "guesses" about what should be tested and helps test managers determine where they should concentrate their efforts.

The proposed risk strategy for testing moves us from the informal approach experienced testers often use to a more formal and systematic way of assessing risk that allows you to base your test strategy on the assessment as well as address the quality concerns of the stakeholder.

NEW 406 Java EE Performance Tuning Methodology: Wait-Based Tuning
By Steven Haines

Java EE performance experts know that the key to success is to focus application tuning effort where it's needed most—at the wait-points, where the delays happen. But what are the wait-points, how can you find them in your application, and what can you do about them? That's what you'll learn in this intermediate-level class.

Wait-points can encompass Web and business tier thread pools, external dependency connection pools, persistence caches, object pools and even garbage collection. We'll show you how to disassemble your application call stack, identify wait-points and tune from the inside-out to optimize throughput and suspend requests where they are best suited to wait. You'll leave this class knowing how to focus your application tuning efforts to immediately improve the performance of your Java EE systems.

NEW 407 Agile Test Development
By Hans Buwalda

Agile methods have become standard in the software development world. The emphasis on short, iterative cycles, constant feedback and a team-based approach to quality has proven effective for delivering software on time and on budget. The same approach can be applied to developing your tests and test automation, even if your development project is using a traditional "waterfall" life cycle. Good test design, especially good automated test design, requires constant feedback from project stakeholders outside the QA team, including the development team, management and customers. The tests should go through several "iterations" of review before being put into "production" against the system under test.

This class will discuss an agile approach to building tests and test automation, so that the QA team can ensure that the system is tested early and often, thereby taking testing off the critical path to releasing the product. This class will present a methodology and case study to illustrate how agile test development can be implemented in real world projects.

NEW 408 Techniques for Testing Packaged Application Performance *By Michel Feaster*

Enterprises are investing more and more in ERP/CRP applications like SAP and Oracle. While these are more critical to the business than ever, new paradigms such as SOA, agile development and offshoring are creating an added layer of complexity for QA organizations. This presentation will drill down on performance testing techniques, including workload modeling, endurance and stress testing, diagnosis and problem isolation and automated script creation. In addition, the speaker will address how to create an optimal team structure and enhanced reporting and communications techniques.

Thursday, Nov. 9 8:30 am – 10:00 am

501 Strategies and Tactics for Global Test Automation, Part 1 *By Hung Q. Nguyen*

We automate software testing to gain speed. We organize our distributed teams globally to maximize round-the-clock coverage and cost efficiency. Both solutions fulfill legitimate objectives. However, implementing them successfully while keeping the risks contained with a high degree of certainty proves to be an enormous challenge. In this class, through a series of technical and management case studies and real life examples, you will learn about seven steps that will deliver return on investment through a global test automation program. Attendees will learn how to: assess testing strategy and needs; minimize the costs and risks of global resources; select the right test automation technology for the job; align testing with business processes and development practices; and measure, analyze and optimize for continuing improvement.

NEW 502 Creating and Leading the High-Performance Test Organization, Part 1
By Robert Galen

Issues such as fewer people, less time, constantly changing technologies and increasing business expectations are clearly the norms for what software teams must face today. Nowhere is this more evident than within testing teams, since the pressure increases as we move down through the life cycle. This pressure poses a tremendous leadership challenge for testing team managers, group leaders or anyone chartered with directing testing. However, this challenge creates the opportunity for effective test leaders to differentiate themselves and their teams as they meet and exceed organizational expectations.

This two-part class focuses on acquiring the fundamental skills to *become* that outstanding test leader. It will explore such issues as how to: build, motivate and lead great testing teams; create impact driven communications on testing state; properly plan and execute your team's evolution, growth and ability to meet project challenges; handle the toughest "people" challenges facing good managers; and be agile and adaptable—learning to change with the organizational landscape.

503 Pinpointing and Exploiting Specific Performance Bottlenecks *By Scott Barber*

One part of the system is always slowest—the bottleneck. Until you remedy that bottleneck, no other tuning will improve performance along that usage path, but before you can tune it, you must first conclusively identify it. Once the bottleneck has been identified, the resolution can be reached more quickly if you modify your existing tests to eliminate distraction from ancillary issues. Pinpointing the bottleneck precisely is an art all its own.

Designed for technical performance testers and developers/architects, this class will show how the performance testing team and the development team can work collaboratively to analyze results and identify bottlenecks by tier, component and object. Then you'll learn how to design tests to exploit those bottlenecks for tuning purposes with examples using IBM Rational and free tools.

Register at www.stpcon.com

504 Performance Tuning ASP.NET 2.0 Applications, Part 1 *By Thomas O'Mara*

This class will provide an intuitive understanding of how to set up a solid testing infrastructure, will help you gain an in-depth understanding of critical .NET and ASP.NET components, and will show you how to monitor the operating system and the ASP.NET application in real time. You will also get a good overview of statistical measurements and their meanings as applied to the data.

The first part of this class will take an in-depth look at the .NET Framework 2.0 that includes the Common Language Runtime, Windows 2003 Server with Internet Information Server 6.0 and ASP.NET 2.0 architecture as it relates to performance tuning.

Part 2 of this class will offer an in-depth real time look at the critical performance counters on Windows 2003 Server, IIS 6.0 and ASP.NET 2.0 as they provide feedback on the health of the application running under load.

505 Identify and Mitigate Risks Through Testing, Part 1 *By Rex Black*

Over the past 10 years, professionals working in software and system development have learned how to apply the powerful techniques of risk analysis and risk management to their projects. In this class, you will learn:

- How to apply risk analysis techniques ranging from informal discussions to ISO 9126 to failure mode and effect analysis
- How risk prioritization can tell system development professionals where to focus development and test resources
- How the project team can improve the accuracy of the risk analysis—and thus the effectiveness and efficiency of testing—throughout the system development life cycle

Part 1 of this class will discuss the various techniques and illustrate them through real case studies. Part 2 includes a hands-on exercise to prepare you to apply these powerful techniques to your next project.

506 Rapid Business-Driven Testing *By Clyneice Chaney*

Structured testing is a vital part of any development project. The problem is that almost no one is given the time and resources to properly execute a thorough test process. In an ideal world, rapid testing would not be necessary, but with most development projects there are schedule crunches and times when a quick assessment of the product quality is necessary.

Rapid testing is a way to scale thorough testing methods to fit arbitrarily compressed schedules. "Rapid" doesn't mean "not thorough," but it does mean as thorough as is reasonable given constraints on time. In this class, you will learn how to use new rapid business-driven testing techniques, methods and templates that will increase product quality in rapid development projects.

507 Unit Testing for Agile Development, Part 1 *By Robert Sabourin*

With the increasing popularity of agile development methods, the role of testing is starting earlier in the software development cycle. Testers and developers are challenged to develop software at lightning speed, often using new and untested technologies. The class will show you how development and testing teams can work together to promote and implement improved unit testing. Attendees will learn how to save your company money by finding and fixing bugs long before system testing even starts. Get the

ammunition you need to convince management of the economic and business benefits of comprehensive unit testing.

This two-part class addresses unit testing issues within the context of different development life-cycle models, especially new agile approaches, and demonstrates the tools and techniques needed to organize for and implement unit testing. The class is taught in workshop style and includes many hands-on group and team exercises, examples and unit testing tool demonstrations.

Due to the interactive nature of these workshops, class size is limited to 30 people.

Special Track!

Black *is Back!*

Rex Black was one of our most popular speakers at the first STPCon, and he's back. Sign up for one or more of these classes, and you won't be disappointed!

T-1 Assessing Your Test Team Effectiveness, Efficiency and More (Full-Day Tutorial)
105 Code Coverage Metrics and How to Use Them
207 Analyze the Return on Your Testing Investment
505 Identify and Mitigate Risks Through Testing, Part 1
605 Identify and Mitigate Risks Through Testing, Part 2

508 Using Code Metrics for Targeted Code Refactoring *By Andrew Glover*

Oftentimes, candidate code for refactoring is based on subjective determinations. The proper uses of code metrics, such as cyclomatic complexity, fan-in, fan-out and depth of inheritance, can also facilitate the discovery of candidate code that is in need of refactoring.

For example, cyclomatic complexity is adept at spotting methods containing a high degree of conditional logic, which, consequently, can be replaced with polymorphism as elaborated by Martin Fowler. Additionally, excessively deep hierarchy trees create problematic testing targets, which can be broken out into separate objects with Fowler's Replace Inheritance with Delegation and Collapse Hierarchy patterns. Fan-in and fan-out are quite effective at pinpointing brittle code, which can be refactored into a more stable state with numerous patterns, including Extract Hierarchy and Extract Class.

You'll leave this class with an understanding of seven industry-standard code metrics; moreover, you will have the ability to utilize these metrics to spot "complex" code and will have a grab bag of techniques with which to improve the code.

509 Recruiting, Hiring, Motivating and Retaining Top Testing Talent *By Jeff Feldstein*

The expectations today are for increasingly high-quality software, requiring more sophisticated automation in testing. Test and QA teams must work more closely with development to ensure that this sophisticated automation is possible. This has led to software engineers applying creativity, talent and expertise to not just application development, but testing as well. This transition from manual to scripting to highly engineered test automation changes the way we recruit, hire, motivate and retain great test engineering talent.

The speaker uses examples of how his team at Cisco changed the way it tests over the past six years. In this class, he'll review eight reasons why test is a better place for software developers than software development, and he'll show how and when to express

Software Test & Performance CONFERENCE

Register at www.stpcon.com

"This conference helps testers and developers as well as managers and leaders. There is enough variety and content for everybody."

—Michael Farrugia, Software Engineer, Air Malta

these points to hire, motivate and retain top talent. You'll see how to inspire greater innovation and creativity in your testing processes and how to manage and inspire test and development teams that are spread across different locations. You'll also learn the place of manual testing in the new environment.

Thursday, Nov. 9
10:30 am – 12:00 pm

601 Strategies and Tactics for Global Test Automation, Part 2 By Hung Q. Nguyen

Please see description under Class 501.

NEW 602 Creating and Leading the High-Performance Test Organization, Part 2
By Robert Galen

Please see description under Class 502.

NEW 603 SOA Performance Testing Challenges
By Scott Barber

Officially, SOA stands for service-oriented architecture, though Martin Fowler quips that maybe service-oriented ambiguity would be more apropos due to the diversity of technical methods being used to implement the SOA concept. The great thing about this ambiguity is that while the developers, architects, vendors and standards groups struggle to narrow the range of technologies, we testers have a chance to get ahead in our preparations for testing these applications. This presentation will introduce the core concepts of SOA, discuss the challenges these concepts present to performance testing and finally map out a performance testing strategy that allows us to use SOA as a springboard to move the state of performance testing significantly forward in your organization and the software industry as a whole.

604 Performance Tuning ASP.NET 2.0 Applications, Part 2 By Thomas O'Mara

Please see description under Class 504.

605 Identify and Mitigate Risks Through Testing, Part 2 By Rex Black

Please see description under Class 505.

606 Exploiting Web Application Code: The Methodologies and Automation of SQL Injection By Matthew Fisher

SQL injection is a technique for exploiting Web applications that use client-supplied data in SQL queries without stripping potentially harmful characters first. Despite being remarkably simple to protect against, there are an astonishing number of production systems connected to the Internet that are vulnerable to this type of attack, due to the simple fact of improper input validation. Developers and quality assurance professionals who design, build and test business-enabling applications generally lack the security knowledge necessary to avoid creating common defects that are so easily exploited by hackers.

In this class, you'll learn about the techniques that can be used to take advantage of a Web application that is vulnerable to SQL injection. The session addresses proper mechanisms that should be put in place to protect against SQL injection, as well as overall improper-input validation issues.

607 Unit Testing for Agile Development, Part 2
By Robert Sabourin

Please see description under Class 507.

608 Verifying Software Robustness
By Ross Collard

Do you like breaking things? If so, this session's for you!

It's not enough to design systems for dependability; we have to verify that reliability as well. Software is robust if it can tolerate such problems as unanticipated events, invalid inputs, corrupted internally stored data, improper uses by system operators, unavailable databases, stress overloads and so on. Systems that include both hardware and software are robust if they can tolerate physical problems such as equipment damage, loss of power and software crashes. Since these problems can and do occur in live operation, this session examines how to evaluate a system's robustness within the relative sanctity of the test lab.

609 Metrics: How to Track Things That Matter
By Clyneice Chaney

Metrics programs have often been a dirty word, misused and poorly implemented. This class discusses ways to provide metrics that really matter to organizations and provide visibility into their or their customers' organizations. The class will begin with discussions about why metrics programs fail and will move on to discuss keys to successful metrics programs, developing quality metrics that matter and ways to implement and maintain these metrics over time.

Thursday, Nov. 9
2:00 pm – 3:15 pm

NEW 701 The Five Most Dangerous Application Security Vulnerabilities— and How to Test for Them By Joe Basirico

The most difficult problems of IT security are found at the application layer. Exploitability of applications due to poor design has reached epidemic levels. Perimeter/network defenses are not enough to protect organizations from attacks and most software teams possess neither the tools nor the expertise to properly secure their applications. This class highlights the top five security vulnerabilities that face testers today. You will learn practical how-to tips for testing your applications with a security mindset to attack their applications before a hacker does.

NEW 702 S-Curves and the Zero Bug Bounce: Plotting Your Way to More Effective Test Management By Shaun Bradshaw

The use of objective test metrics is an important step toward improving the ability to effectively manage any test effort. Two significant test metrics concepts, the S-Curve and Zero Bug Bounce, allow test leads and test managers to easily track the progress of a test effort, improve the ability to communicate test results and test needs to the project team, and make better decisions regarding when an application is ready to be delivered.

You will learn:
- How to establish an S-Curve: What an S-Curve is, why it is important in testing, how to develop a theoretical S-Curve, what metrics can be tracked using the S-Curve and how to track them.
- How to create a Zero Bug Bounce: What the Zero Bug Bounce (ZBB) is, why it is important in tracking defects, and how to generate a ZBB.
- Interpreting the graphs: Finally, we discuss a method for examining and interpreting the graphs created by these

Register at www.stpcon.com

13

metrics to make improvements to the current test efforts as well as future development efforts.

NEW 703 Testing Java Programs—Memory Management Issues By Averil Meehan

It is a myth that Java's garbage collection has solved memory management problems. These can still cause space leaks severe enough to crash a program or for a thread of execution to stop. Unfortunately, often such errors occur only when a program runs for a long time, so they may not show up until the testing stage of development. Such errors are extremely difficult to debug as the error can appear in one section of code yet be caused by another. The nature of memory errors means that the problem can manifest itself in different ways and at different times in code execution, something that also makes detection and solution problematic. In this class, we will discuss what memory problems can occur and consider different approaches to detecting the code that caused them.

NEW 704 Defining Test Data and Data-Centric Application Testing By Chris Hetzler

As applications grow larger and more complex and as automated testing of these applications is increasingly adopted, the data that is used during the execution of the automated tests needs to be clearly defined and identified early in the development life cycle. Currently, the Software Engineering Body of Knowledge does not contain a definition for "test data," nor do any of the top books on the subject of software testing. This presentation will propose a definition for the term "test data" and outline what testers can do to ensure that teams are considering it in their planning phases and provide useful ideas on how to isolate what those data needs might be. The presentation will use examples of software development projects that have defined their test data early and those that have not to provide you with context and examples.

705 Coding Standards and Unit Testing—Why Bother? By Sergei Sokolov

Many developers think that the industry "best practices" of coding standards and unit testing are a waste of time: They require additional effort, but they don't seem to make your life any easier, or your code any better. This is not surprising.

This class explains how developers can apply coding standards and unit testing to improve their code and prevent the number of problems they need to identify, diagnose and fix over the course of the project. The first half teaches you how to apply coding standards to prevent errors related to code functionality, security and performance. The second part focuses on how you can extend traditional unit testing to expose reliability problems that could lead to instability, unexpected results or even crashes or security vulnerabilities. Discussion will include how these test cases can be leveraged to build a projectwide automated regression system that runs in the background each night and immediately alerts the team when code modifications or additions break previously verified functionality.

NEW 706 The 5% Challenges of Test Automation By Hans Buwalda

Some of the most common problems with software test automation projects are that too much effort is spent on developing test scripts where the percentage of tests that is actually automated is only a meager 20% to 30% because too much time is spent on script maintenance. These issues cause numerous pains, including skyrocketing costs, a lack of management visibility and questionable quality of tests.

These universal problems in test automation have led this speaker to introduce the 5% challenges of test automation: No more than 5% of tests should be executed manually, and no more than 5% of the test effort should be spent creating automation. The speaker will present the keys to meeting the 5% challenges and illustrate them through a case study of a successful project that maximized test automation while minimizing the time spent automating tests.

707 Performance Testing for Managers By Scott Barber

Performance testing as an activity is widely misunderstood, particularly by managers and others not directly involved in doing it. This presentation details the most critical things for managers to know about the performance testing process and ways to improve it. Learning, understanding and applying these nuggets of knowledge to your current or future performance testing projects will dramatically increase your team's chances of success. In this class you will learn how to work with experienced performance testers to get the results you need, why performance testing should begin well before the application is fully functional and how to do it, and ways to better integrate performance-testing personnel into the development team effort. In addition, you will learn how to recognize the difference between "delivery" and "done" as they relate to performance testing and how to assess and balance the risks inherent in each. Also covered is how to create and maintain a program that will ensure not only that your performance testers have the tools they need, but that they will know how to use them and when to put them away.

NEW 708 What Hollywood Can Teach You About Software Testing By Robert Sabourin

Powerful lessons can be learned from some of the great epic movies of our day: "Star Wars" and bug triage, "Indiana Jones" and requirements, "Monty Python" and configuration management, "Jurassic Park" and unit testing, "The Usual Suspects" and teamwork, and "Star Trek" and SLAs.

There are important metaphors within these movie stories that you can apply to real test management problems. Robert Sabourin's entertaining talk ties practical real-world experiences to lessons learned from the movies, offers tips to manage your team and provides his unique insight into how to get things done.

You will learn:
- How to identify your testing project genre—epic saga, action thriller, mystery or comedy
- How to storyboard a testing project
- How to draw lessons from an unlikely source

Thursday, Nov. 9
3:45 pm – 5:00 pm

NEW 801 Effective Metrics for Managing a Test Effort By Shaun Bradshaw

When managing a test effort, test leads and test managers sometimes find it difficult to empirically convey the impacts of scope changes, delays and defects to upper management. This class introduces a set of well-defined test metrics related to tracking and managing a testing effort. It demonstrates how to consistently apply these metrics to software projects and how to improve communication of your findings to the rest of the organization.

The following concepts will be taught as a part of this class:
- Test Metrics Philosophy – A four-step strategy for creating, tracking and interpreting test metrics
- Base Test Metrics – Fourteen simple metrics easily tracked

by test analysts that can be used to derive more sophisticated test management information

- Management Test Metrics – Ten metrics and two graphs that make managing and communicating the status of a test effort easier for test managers and test leads
- Interpreting Test Metrics – A method for examining and interpreting the data to make improvements to the current test effort as well as future development efforts

NEW 802 Software Testing a Service-Oriented Architecture By Laura Rose

The presence of service-oriented architectures (SOA) has grown significantly in the past few years, and test professionals must stay abreast of the latest technologies in order to continue providing significant value-add to the software development process.

This class describes briefly what SOA is, methodologies that can be employed when testing in an SOA environment, common and potential pitfalls that can be avoided, a comparison and contrast of testing in an SOA environment versus a "traditional" test environment and a look ahead at future opportunities.

NEW 803 Real-World Performance Testing Lab for (Almost) Free By Aaron Flint

This course will discuss how to set up and maintain an effective performance test lab without spending a great deal of money on specialized performance testing hardware or software. Topics that will be covered include: what is needed to set up an effective performance lab; the hardware needed to get the maximum performance from the lab; and the software that must be installed and run in the lab, including tuning options and software configurations. We will also discuss how to manage and run performance tests and generate results. Finally, we will talk about maintenance issues such as how to preserve the environment.

804 Testing Java Applications Using the Eclipse Test and Performance Tools Platform (TPTP) By Paul Slauenwhite

The Eclipse Test and Performance Tools Platform (TPTP) provides a flexible and extensible framework for creating and managing tests, deployments, datapools, execution histories and reports with extensions for performance, JUnit, GUI and manual testing of Java applications. In this class, you will learn how to use the performance, JUnit, GUI and manual test tooling in the TPTP for testing Java applications. Intended for developers and testers who want to test their Java applications, the class begins by providing an overview of the motivation, history and architecture for the TPTP project. Then the TPTP test framework is explained with a focus on extending the framework for vendor-specific purposes. Finally, the performance, JUnit, GUI and manual test tooling in the TPTP project is demonstrated to illustrate the life cycle of an application's test assets. Attendees are provided with the sample configuration and code used in the technical class.

NEW 805 Building a Bridge Between Functional Test Automation and Performance Testing By Peter Sody

In a lot of software development organizations there is a clear distinction between functional test automation and performance testing. This often leads to different teams testing either the functionality or performance of the same software systems. This class highlights the commonalities between functional test automation and performance testing and shows how these two areas can benefit from each other.

This class gives practical examples on how to benefit from a collaboration, covering aspects of the whole test cycle. Applied in the right way, this can go far beyond reusing common artifacts. With the right approach the teams can complement their skill sets and significantly increase the efficiency and coverage of their testing. You will learn how

to determine the common areas that can boost the testing performance for both sides and how to recognize the limits of such efforts.

NEW 806 Best Practices for Managing Distributed Testing Teams By Dean Stevens

Software testing projects are now often completed using a distributed model to leverage offshore labs to lower costs or decrease project delivery time. But without building certain cultural, methodological and technological foundations into your organization's core values, it can be difficult or impossible to effectively manage projects being completed by dispersed virtual teams.

Managing a distributed testing project requires a highly disciplined approach. Developing an effective framework for the execution of distributed projects goes hand in hand with requiring consistently documented processes and standards ("best practices"). However, by architecting a work model to accommodate a distributed model, you ensure a rigorous approach that will assist the successful execution of projects according to a well-defined set of best practices, regardless of the execution model employed.

Once methodological and technical frameworks are in place, the bulk of the remaining challenges are related to "soft issues," including cultural incompatibility, leadership problems, trust issues and negative competitiveness. These are actually the major obstacles to successful completion of distributed projects. However, there are concrete ways to alleviate these problems, including redefining your corporate culture, improving project planning and adjusting project staffing and technical infrastructure. In this session, attendees will learn methodologies and best practices designed to build "teamness" between distributed groups and set up both the physical and project structure so your teams can succeed.

NEW 807 Diagnosing and Resolving J2EE Application Issues Before Deployment By Ferhan Kilical and Stanley Au Yeung

J2EE offers many advantages to developers but introduces many challenges to the development, performance diagnosis, tuning, deployment and management of applications in the enterprise network. In this class, attendees will learn how to proactively detect problems and isolate them at different layers and tiers. They will acquire the tools to be able to pinpoint root causes to specific application components.

The class begins with the end-user business process, then drills down to application layers, communication between different layers, transaction trend-request data, method calls and class data. During the class, the presenters will share some of the hands-on samples that they mastered during their own J2EE tuning efforts.

NEW 808 GUI Test Automation for Eclipse RCP Applications By Dan Rubel and Phil Quitslund

The Eclipse IDE and Eclipse RCP applications are taking the Java world by storm, and along with that wave is the need for automated user interface testing. This class introduces open-source tools and methodologies such as Abbot for SWT and Eclipse TPTP GUI Recorder for automated testing of Eclipse RCP applications. Attendees will be taken step-by-step through the process of creating and running GUI tests using these tools, and learn the strengths and weaknesses of each approach. This intermediate-level course assumes you have knowledge of Eclipse technology, building user interfaces and testing user interfaces.

Register at www.stpcon.com

Scott Barber is software test manager at AuthenTec and a member of the technical advisory board for Stanley Reid Consulting. Previously, he was a consultant who focused on teaching and performing practical performance testing and analysis. His project-level experience was evenly split between testing and analyzing performance for complex systems and mentoring organizations in the development of customized corporate methodologies based on his performance testing approach.

Mr. Barber has a master's degree in IT from American Intercontinental University. He writes Peak Performance, the performance testing column in Software Test & Performance magazine, and he also speaks at many technical conferences.

Joe Basirico has spent most of his educational and professional career studying security and developing tools that assist in the discovery of security vulnerabilities and general application problems. His primary responsibility at Security Innovation is to deliver the company's Security Training Curriculum to software teams in need of application security expertise. Mr. Basirico has trained developers and testers from numerous world-class organizations, such as Microsoft, HP, EMC, Symantec and ING. He holds a B.S in computer science from Montana State University.

Ryan Berg is a co-founder and lead security architect of Ounce Labs, an innovator of software vulnerability risk management solutions, based in Waltham, Mass. Prior to Ounce Labs, Mr. Ryan co-founded Qiave Technologies, a pioneer in kernel-level security, which was sold to WatchGuard Technologies in 2000. He also served as a senior software engineer at GTE Internetworking, leading the architecture and implementation of new managed firewall services. Mr. Berg holds patents, and has patents pending, in multilanguage security assessment, intermediary security assessment language, communication protocols and security management systems.

A 20-year-plus software and systems engineering veteran, **Rex Black** is president and principal consultant of RBCS, which offers training, assessment, consulting, staff augmentation, insourcing, offsite and offshore outsourcing, test automation and quality assurance serv-

ices. Mr. Black has published several books, including "Managing the Testing Process" and "Critical Testing Processes." He has also written more than 20 articles, presented hundreds of papers, workshops and seminars, and given more than a dozen keynote speeches at conferences and events around the world. Mr. Black is the president of both the International Software Testing Qualifications Board and the American Software Testing Qualifications Board.

Shaun Bradshaw serves as director of quality solutions for Questcon Technologies. He is responsible for managing Questcon's team of senior practice managers in the areas of quality solutions development and service delivery, and also works with clients to improve their quality assurance and software test processes.

Hans Buwalda leads LogiGear's action-based testing (ABT) research and development, and oversees the practice of ABT methodology. Mr. Buwalda is an internationally recognized expert specializing in action-based test automation, test development and testing technology management. He's also a speaker at international conferences, delivering tutorials and workshops, as well as presenting testing concepts such as ABT, the three Holy Grails of test development, soap-opera testing and testing in the cold. Recently, Mr. Buwalda co-authored "Integrated Test Design and Automation." He holds an M.S. in computer science from Free University, Amsterdam.

In his five years at Empirix, **Rick Cavallaro,** senior applications engineer, has worked with hundreds of companies helping to ensure the performance of their most critical Web applications. A 10-year veteran of the software industry, he specializes in testing and application development. Prior to joining Empirix, Mr. Cavallaro served in engineering roles at Aviv, Workstation Solutions and Revelation Software. He holds a BSEE degree from the University of Massachusetts, Lowell.

Clyneice Chaney, quality manager at Project Performance, brings more than 16 years of testing, quality assurance and process improvement experience. She is an American Society for Quality Certified Quality Manager and a Quality Assurance Institute Certified Quality Analyst. Focusing on process improvement and procedure development in the software testing and quality assurance areas, Ms. Chaney has successfully led process improvement, methodology development and re-engineering projects for organizations wishing to improve their software development, testing processes and tools implementation.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

Thierry Ciot is a senior software engineer with more than 15 years of experience in software design and implementation. His experience spans many operating systems, programming languages (ADA, C, C++, Java, .NET) and a multitude of application domains, from mail and messaging to debugging tools. Mr. Ciot currently works at Compuware as a technical lead on DevPartner Studio. He is also the lead developer for System Compare, a tool that enables users to quickly find why an application works on one system but not on another.

Ross Collard is president of Collard & Co., a New York consulting firm. While he specializes in software testing and quality assurance, his consulting assignments have included strategic planning on the use of information technology for competitive advantage, the facilitation of quality improvement teams, management of large software development projects and the development of software engineering practices.

Mr. Collard has made keynote presentations at major software conferences, published articles and conducted seminars on information technology topics for businesses, governments and universities, including George Washington University, Harvard, New York University and U.C. Berkeley. He holds a B.E. in electrical engineering from the University of Auckland, New Zealand, an M.S. in computer science from the California Institute of Technology and an M.B.A. from Stanford.

Elfriede Dustin is an SQA manager at Symantec, author of the book "Effective Software Testing" and lead author of "Automated Software Testing" and "Quality Web Systems." She is currently writing the "Security Testing Handbook," along with two security experts, to be published by Symantec Press (spring 2006). She has also authored various white papers on the topic of software testing and is a frequent speaker at various software testing conferences.

Ms. Dustin has a B.S. in computer science and more than 15 years of IT experience in various positions, such as QA director for BNA Software and assistant director for integration test and deployment at CSC on the IRS modernization effort.

Michel Feaster is the director of product management for Mercury Interactive. She has seven years of experience as a systems engineer, and has worked closely with hundreds of Mercury Interactive software developers and QA manager customers helping them to streamline QA operations. She has spent a total of 12 years in the enterprise software industry. Ms. Feaster's in-depth technical knowledge and engaging style have made her a sought-after speaker at numerous industry events.

Jeff Feldstein is currently a manager of software development at Cisco Systems Inc. During his 24-year career, he has been a software developer, tester, development manager and computer consultant; for the past five years, he has been involved with software testing or has managed a team of developers who write software test tools. His specialties have included internetworking, real-time embedded systems, communications systems, hardware diagnostics and firmware, databases and test technologies. Mr. Feldstein has been one of the highest-rated speakers at previous Software Test & Performance conferences.

Matthew Fisher is a senior security engineer for SPI Dynamics and has been specializing in Web application security assessments for many years. A native Washingtonian, he has performed countless assessments of Web applications within the DoD and the federal government, as well as some of the largest commercial institutions around the globe, and is registered as a subject matter expert to the Defense Information Services Agency. Prior to joining SPI Dynamics, Mr. Fisher worked at Computer Sciences and Digex, where he acted as lead technical adviser on large-scale enterprise Web applications for Fortune 500 companies. He is a contributing author to the book "Google Hacking for Penetration Testers" and is currently working on his own book, titled "Web Application Security: A Guide for Developers and Penetration Testers." In addition, Mr. Fisher leads the Washington, D.C., OWASP chapter.

Aaron Flint has more than 10 years of increasingly senior quality assurance experience, primarily focused on testing enterprise-level server software, and responsible for functional testing, performance testing and building teams for quality assurance. For the past three years, Mr. Flint has been the QA manager at Layer7 Technologies. Prior to this, he worked at GTE Enterprise Solutions as QA team lead for multiple-listing-service real estate system software, and InfoWave Software as QA team lead and manager for enterprise wireless connectivity software.

Bob Galen is a principal consultant of RGalen Consulting Group LLC, and has nearly 25 years of experience working in a wide variety of domains at companies including Bayer, Bell & Howell Mail Processing, EMC, Lucent, Unisys and Thomson.

Mr. Galen regularly speaks at international conferences and professional groups on software development, project management, software testing and

Register at www.stpcon.com

team leadership. He is a certified Scrum Master and a member of the Agile Alliance. Mr. Galen is author of "Software Endgames: Eliminating Defects, Controlling Change, and the Countdown to On-Time Delivery."

Andrew Glover is president of Stelligent, where his primary responsibilities include the strategic development of Stelligent's products and services. Mr. Glover's career includes founding Vanward Technologies and leadership in software development for IBM, Philips Electronics and Procter & Gamble. He is a graduate of George Mason University in Fairfax, Va., and is a frequent speaker at industry events. Mr. Glover is a co-author of "Java Testing Patterns" (Wiley, 2004).

Robin F. Goldsmith has been president of the Go Pro Management consultancy since 1982. He works directly with and trains professionals in business engineering, requirements analysis, software acquisition, project management, quality assurance and testing. Previously he was a developer, a systems programmer/DBA/QA and a project leader with the City of Cleveland, leading financial institutions and a "Big 4" consulting firm. Author of numerous articles and the recent book "Discovering REAL Business Requirements for Software Project Success," Mr. Goldsmith was formerly international vice president of the Association for Systems Management and executive editor of the Journal of Systems Management. Mr. Goldsmith has an A.B. from Kenyon College, an M.S. from Pennsylvania State and a J.D. from Boston University.

Michael Hackett is a founding partner of LogiGear and is responsible for the direction and development of the company's training program. He has in-depth experience in software engineering and the testing of applications developed for deployment across multiple platforms. Mr. Hackett writes and teaches a software testing curriculum for LogiGear University, and for the U.C. Berkeley Extension. He is also co-author of "Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems," Second Ed., and holds a B.S. in engineering from Carnegie Mellon University.

Steven Haines started Quest Software's Java EE Performance Tuning professional services organization in 2002. Mr. Haines is the author of both "Java 2 Primer Plus" and "Java 2 From Scratch" and shares author credits on "Java Web Services Unleashed." He is also the author of two newly released books, "InformIT Java Reference Guide" and "Pro Java EE Performance Management." Mr. Haines has taught Java at the University of California in Irvine and at Learning Tree University. As the Java host on InformIT.com, he publishes a weekly column on everything from Java Web technologies to design patterns and performance tuning.

Linda Hayes is the CTO of Worksoft, and also the founder of three software companies, including AutoTester, which delivered the first PC-based test automation tool. Ms. Hayes holds degrees in accounting, tax and law, is an award-winning author on software quality, and has been a frequent industry speaker at numerous industry conferences and shows. She has been named one of Fortune Magazine's "People to Watch" and one of the "Top 40 Under 40" by Dallas Business Journal. She has been a columnist and contributor to StickyMinds and Better Software magazines, as well as a columnist for Computerworld and Datamation, author of the "Automated Testing Handbook" and co-editor of "Dare to Be Excellent" with Alka Jarvis on best practices in the software industry.

Chris Hetzler is employed in the Fargo, N.D., Microsoft Business Solutions development office as a software development engineer in test. On his latest project, Microsoft Dynamics GP Web Services, he was the lead technical tester and was responsible for the development of the product's test strategy. Mr. Hetzler has a B.S. in computer science and an M.S. in software engineering from North Dakota State University. He has won numerous public speaking awards throughout his academic career.

Ferhan Kilical is an experienced IT professional with more than 20 years of software development and engineering experience. She has successfully managed tuning efforts of many Web-based systems in government and nonprofit institutions. Early in her career, Ms. Kilical successfully managed high-profile projects for the Department of Defense. She has taught computer science courses in the Washington, D.C., metro area universities. Recently she started a Washington, D.C.-based consulting firm specializing in performance, load and stress testing, performance monitoring, application tuning and optimization.

Timothy Korson has had more than two decades of substantial experience working on a large variety of systems developed using modern software engineering techniques. This experience includes distributed, real-time, embedded systems as well as business information systems in an n-tier, client/server environment. Dr. Korson's typical involvement on a project is as a senior management consultant with additional technical responsibilities to ensure high-quality, robust test and quality assurance processes and practices. Dr. Korson has authored numerous articles, and co-authored a book on Object Technology Centers.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

Yury Makedonov was trained as a researcher and worked in an R&D organization dealing with composite materials. He has a Ph.D. degree in physics and mathematics. He is now using his skills and knowledge to improve software quality.

Dr. Makedonov has 10 years of testing experience. Currently, he is working as a QA manager, a test automation manager and a senior consultant for the Centre of Testing and Quality at CGI Group, a leading Canadian provider of end-to-end information technology and business process services.

Averil Meehan has a Ph.D. in the area of memory management of object-oriented programming languages. Currently she lectures at Letterkenny Institute of Technology, a third level college in Ireland with students at degree and masters levels. Dr. Meehan has presented conference papers and written articles for publications internationally, including "Performance Implications of Java Memory Management," "Computer

Programming in a Blended Learning Environment," "Java Garbage Collection—A Generic Solution" and "The Semantics of Garbage Collection Rules, A Denotational Approach."

Hung Q. Nguyen is CEO and founder of LogiGear, a software quality engineering firm offering training, testing services and test automation products. He is author and co-author of several software testing books, including "Testing Applications on the Web," Second Ed., and "Testing Computer Software," Second Ed.

Mr. Nguyen writes and teaches a software testing curriculum for LogiGear University, as well as for U.C. Berkeley and the U.C. Santa Cruz Extension. He holds a B.S. in quality assurance from Cogswell Polytechnical College, is a graduate of a Stanford Graduate School of Business Executive Program, and is a certified quality engineer.

Thomas O'Mara has more than 25 years of experience with PC-based computing, ranging from Fiber Optic Gyroscope data acquisition using the stack-based FORTH language to Web-based applications utilizing the .NET Framework and ASP.NET. In between, there were C, Visual Basic and various database and middleware initiatives. Mr. O'Mara has been working with and writing articles about .NET technology since early 2001. He has considerable direct per-

formance-tuning experience on a Web-based ASP.NET banking software application for large credit unions.

Phil Quitslund, window tester architect at Instantiations, is an expert in GUI testing and automation. He has been active in the Eclipse research community since 2002 and has devel-

oped numerous tools for supporting advanced programming language features and extensions such as aspect-oriented programming. Mr. Quitslund is also an expert in programming language design, static analysis and code generation.

BJ Rollison is a technical trainer in the Engineering Excellence Group at Microsoft, where he designs and develops an intensive, hands-on technical training curriculum for new and experienced test engineers. He started his professional career in the industry working on developing custom solutions for hospitals and local government agencies in Japan. In 1994 he joined the Windows 95 project at Microsoft, focusing on the internationalization of the Windows operating system.

In 1996, Mr. Rollison became a test manager in the Internet Client and Consumer Division, responsible for several client products and a Web server. He moved to Microsoft's Internal Technical Training group in 1999 as the director of test training, responsible for planning and organizing training for more than 6,000 test engineers. He also teaches software testing courses at the University of Washington and sits on the advisory boards for software testing certificate programs at the University of Washington and Lake Washington Technical College.

Laura Rose is a quality assurance expert and the product manager responsible for automated performance test tools at IBM Rational. In addition to leading projects in both software programming and testing environments, she has 13 years of programming experience and 10 in test management.

Ms. Rose has been a member of the American Society for Quality, the Triangle Quality Council and the Triangle Information Systems Quality Association.

She is published and has presented at various test and quality conferences, including IBM Test Symposium West, Practical Software Quality Conference (East and West), American Quality Society Conference, Better Software Conference & Expo and STARWest.

Dan Rubel, chief technology officer for Instantiations, has been designing object-oriented software since 1987. He is an expert in the design and implementation of commercial software products like CodePro Studio, jFactor and VA Assist, and frameworks used throughout the company and beyond, including jKit/GO and jKit/Grid. At Objectshare Systems, Mr. Rubel was one of the lead designers of WindowBuilder Pro and other products.

Robert Sabourin, P.Eng., has more than 20 years of management experience leading teams of software development professionals to consistently deliver projects on time, on quality and on budget.

Mr. Sabourin is an adjunct professor of software engineering at McGill University who often speaks at conferences around the world on software engineering, SQA, testing and management issues.

Register at www.stpcon.com

Neeraj Sangal is president of Lattix, a company specializing in software architecture management solutions and services. He has analyzed many large proprietary and open-source systems. Prior to Lattix, Mr. Sangal was president of Tendril Software, which pioneered model-driven EJB development and synchronized UML models for the Java programming language. Tendril was acquired by BEA/WebGain. Prior to Tendril, he managed a distributed development organization at Hewlett-Packard.

Paul Slauenwhite is a software developer on the IBM Autonomic Computing (AC) Tools and Technologies project and a committer to the Eclipse Test and Performance Tools Platform (TPTP) open-source project. After receiving a B.Sc. in computer science from Dalhousie University, Mr. Slauenwhite joined IBM in 2000 and worked on the WebSphere Object Level Trace (OLT) project. In 2001, he joined the IBM WebSphere Studio Team and developed logging and tracing technologies. He is currently working on an M.Math in software engineering at the University of Waterloo.

Over the past 15 years, **Peter Sody** has been a testing contractor, software developer, test engineer and development manager for several companies and organizations in Germany and the United States, most recently for Eclipsys and Vertex. Being an advocate of test-driven development, he is now using his skills for developing architectures and frameworks for both functional test automation and performance testing. Mr. Sody has authored publications on various aspects of software development, including presentations at conferences such as IEEE RE and Net.ObjectDays. He holds an M.S. in computer science from Kaiserslautern University, Germany.

Sergei Sokolov is the product manager of C/C++ solutions at Parasoft, where he has been a senior member of the Professional Services team for two years. He has more than 10 years of experience in developing large software systems in C and C++ in the Electronic Design Automation (EDA) industry. Mr. Sokolov has published a number of technical and research papers, and presented at technical conferences on the subjects of application security and best development practices. He has two master's degrees, and has been awarded a patent for a new technology in the EDA field.

Now an independent software consultant, **Alfred Sorkowitz** was a computer scientist with the Department of the Navy, responsible for developing real-time, software-intensive embedded systems. Prior to joining the Department of the Navy, he was director of the Standards and Quality Control Staff, U. S. Department of Housing and Urban Development. Mr. Sorkowitz has published papers and has presented seminars on software metrics, SQA and testing at conferences sponsored by the IEEE Computer Society, ACM and the British Computer Society.

Dean Stevens has been involved with developing and delivering world-class hardware, software and service products for more than 20 years. Prior to joining Symbio, Mr. Stevens founded and served as the CEO of China TechSource—an outsourcing broker for Chinese services firms. In addition, he has operated a consulting firm that worked with corporations to resolve executive management and execution issues. He has demonstrated success managing global remote development, multicompany projects and distributed virtual organizations. Mr. Stevens began his career writing FORTRAN code for a CDC mainframe. He is a graduate of the University of Idaho.

Tom Stracener was one of the founding members of nCircle Network Security. While at nCircle he served as the head of vulnerability research from 1999 to 2001, developing one of the industry's first quantitative vulnerability scoring systems, and co-inventing several patented technologies. Mr. Stracener is an experienced security consultant, penetration tester and vulnerability researcher. One of his patents, "Interoperability of vulnerability and intrusion detection systems," was granted by the USPTO in October 2005. He is the senior security analyst for Cenzic's CIA Labs and the architect of Cenzic's Application Penetration Testing Methodology.

Mary Sweeney has been developing, using and testing relational database systems for more than 20 years for such companies as Boeing and Software Test Labs. She's the author of "Visual Basic for Testers" (Apress, 2001) and "A Tester's Guide to .NET Programming" (Apress, 2006). Ms. Sweeney is a college professor with a degree in mathematics and computer science from Seattle University. She is an MCP in SQL Server, and is on the board of IIST (International Institute of Software Test).

Stanley Au Yeung has years of IT experience in database applications, performance testing and quality assurance. After getting his M.S. degree from George Washington University in Information Technology, he was heavily involved with research and application development projects in the university. Then he joined a nonprofit organization where he mastered skills and knowledge in quality assurance and performance testing. He was involved with various high-profile Web systems tuning and optimization projects. He is currently working on a performance optimization project for a mission-critical Navy application as a senior consultant.

Software Test & Performance CONFERENCE

Register at www.stpcon.com

Hotel

The Software Test & Performance Conference has secured a special rate of $159/single per night with the Hyatt Regency Cambridge. To make your reservations, use the link on the show's Web site, www.stpcon.com. You can also call the hotel directly. Be sure to identify yourself as being with the Software Test & Performance Conference to receive the group rate.

YOU MUST MAKE YOUR RESERVATIONS BY OCTOBER 16 TO SECURE THIS RATE.

Directions

From Logan International Airport:
Follow signs to Boston via Sumner Tunnel. After you come out of the tunnel take exit 26A—(Storrow Drive/Back Bay/Cambridge). Keep left, go onto Storrow Drive. Go 3/4 mile and take exit on left-hand side (Government Center/Kendall Square). At the top of the ramp take a right over the Longfellow Bridge; at the end of the bridge turn right onto Memorial Drive West (Rt. 3 North). Stay on Memorial Drive approximately 1 1/2 miles; at the third traffic light turn right. The entrance to the Hyatt will be on your left.

Hyatt Regency Cambridge
575 Memorial Drive
Cambridge, MA 02139-4896
Tel: +1-617-492-1234
Fax: +1-617-491-6906

Transportation

From the Boston Airport:
- Taxi – Approximately $35
- Subway – $1.25 (T stops), 1 1/2 miles from hotel (taxi available)
  Above Ground: B.U. Central, Green Line (T stops)
  Underground: Harvard Square, Kendall Square Red Line

Amtrak:
- Back Bay Station is located 3 miles from hotel, approximately 10-15 minutes away. Amtrak also stops at South Station in the Financial District, which is approximately 5 miles, 20-30 minutes, from hotel. Taxi is available to and from the hotel. From Back Bay Station, cost is $7-$10. From South Station, cost is $12-$15.

Downtown Cambridge:
- Hotel offers complimentary scheduled shuttle service to locations in Cambridge. The shuttle service will take guests anywhere from Harvard Square to the Cambridgeside Galleria Mall area and all points between.
- For guests who would like to get into the city of Boston, the shuttle will stop at Kendall Square T Stop (Red Line) or the Boston University T Stop (Green Line). For more information or to sign up for the shuttle, contact the Guest Services Desk at in-house extension 51 or call +1-617-492-1234.

General Shuttle Information:
Traffic conditions may change pick-up times, forcing the shuttle off schedule, especially during rush hour and inclement weather. The Hyatt shuttle is for registered guests only. If requested, please show your room key to the driver as identification. T Stop schedules are available at the Guest Services Desk. For the safety of our passengers, luggage is not allowed on the shuttle. Shuttle schedule is subject to change without notice.

MBTA: (Subway) 1 1/2 miles from hotel.

Parking:
Self-parking is $28 per night with in/out privileges (clearance 6' 8").
Valet parking is $30 per night; RV and van parking is available at your own risk in the employee parking lot, based on availability. See the doorman or front desk upon arrival for details.

Register at www.stpcon.com

MONDAY November 6	TUESDAY November 7	WEDNESDAY November 8		
REGISTRATION OPEN 4:00 pm - 7:00 pm	REGISTRATION OPEN 7:30 am - 7:00 pm	REGISTRATION OPEN 7:30 am - 7:00 pm		
		INDUSTRY KEYNOTE 4:30 pm - 4:45 pm		KEYNOTE: REX BLACK
		EXHIBITS OPEN 2:30 pm - 7:00 pm		ATTENDEE RECEPTION
	TUTORIALS	**TECHNICAL CLASSES**		
	9:00 am - 5:00 pm	8:30 am - 10:00 am	10:30 am - 12:00 pm	1:15 pm - 2:30 pm
	T-1 Assessing Your Test Team Effectiveness, Efficiency and More - Black	**101** Seven Low-Overhead Software Process Improvement Methods - Goldsmith	**201** Prevent Showstopper Overruns With Risk-Based Proactive Testing - Goldsmith	**301** Hacking 101: Donning the Black Hat to Best Protect Applications From Today's Hacking Threats - Stracener
	T-2 Testing Techniques: Theory and Application - Rollison	**102** Just-in-Time Testing Techniques and Tactics, Part 1 - Sabourin	**202** Just-in-Time Testing Techniques and Tactics, Part 2 - Sabourin	**302** Five Core Metrics to Guide Your Software Endgames - Galen
	T-3 Testing in Highly Iterative, Quasi-Agile Projects—Practical Strategies for Mixed Culture Projects - Korson	**103** Automated Database Testing: Testing and Using Stored Procedures - Sweeney	**203** How to Turn Your Testing Team Into a High-Performance Organization - Hackett	**303** Overcoming Requirements-Based Testing's Hidden Pitfalls - Goldsmith
	T-4 Twenty-One Ways to Spot—And Fix—Requirements Errors Early - Goldsmith	**104** Lessons Learned in Test Automation, Part 1 - Dustin	**204** Lessons Learned in Test Automation, Part 2 - Dustin	**304** Managing Acceptance Testing Cycles More Efficiently - Makedonov
	T-5 Foundations of GUI Test Automation - Makedonov	**105** Code Coverage Metrics and How to Use Them - Black	**205** Database Security: How Vulnerable Is Your Data? - Sweeney	**305** How to Optimize Your Web Testing Strategy - Nguyen
	T-6 Software Endgames: How to Finish What You've Started - Galen	**106** Foundations of GUI Test Automation Using C#, Part 1 - Rollison	**206** Foundations of GUI Test Automation Using C#, Part 2 - Rollison	**306** Model-Based Testing for Java and Web-based GUI Applications - Feldstein
	T-7 Using Metrics to Improve Software Testing - Sorkowitz	**107** Designing for Testability - Feldstein	**207** Analyze the Return on Your Testing Investment - Black	**307** Taking AIM—Using Visual Models for Test Case Design - Sabourin
	T-8 Creating Agility and Effectiveness in Software Testing - Hayes	**108** Testing the Software Architecture - Sangal	**208** Using Scrum to Manage the Testing Effort - Galen	**308** Elements of Software Design for Unit Testing - Ciot
	BIRDS OF A FEATHER SESSION 8:00 pm - 10:00 pm	**109** Effectively Training Your Offshore Test Team - Hackett	**209** Quality Throughout the Software Life Cycle - Feldstein	**309** Models for Security Testing in the Software Development Life Cycle - Berg
		BIRDS OF A FEATHER SESSION 8:30 pm - 10:30 pm		

Software Test & Performance CONFERENCE

Register at www.stpcon.com

THURSDAY
November 9

REGISTRATION OPEN 7:30 am - 4:00 pm

| 4:45 pm - 5:30 pm |
| 5:30 pm - 7:00 pm |

EXHIBITS OPEN 12:00 pm - 4:00 pm

TECHNICAL CLASSES

3:00 pm - 4:15 pm		8:30 am - 10:00 am		10:30 am - 12:00 pm		2:00 pm - 3:15 pm		3:45 pm - 5:00 pm	
401	The Secure Software Development Life Cycle - Dustin	501	Strategies and Tactics for Global Test Automation, Part 1 - Nguyen	601	Strategies and Tactics for Global Test Automation, Part 2 - Nguyen	701	The Five Most Dangerous Application Security Vulnerabilities—and How to Test for Them - Basirico	801	Effective Metrics for Managing a Test Effort - Bradshaw
402	Accelerate Testing Cycles With Collaborative Performance Testing - Cavallaro	502	Creating and Leading the High-Performance Test Organization, Part 1 - Galen	602	Creating and Leading the High-Performance Test Organization, Part 2 - Galen	702	S-Curves and the Zero Bug Bounce: Plotting Your Way to More Effective Test Management - Bradshaw	802	Software Testing a Service-Oriented Architecture - Rose
403	Deciding What Not to Test - Sabourin	503	Pinpointing and Exploiting Specific Performance Bottlenecks - Barber	603	SOA Performance Testing Challenges - Barber	703	Testing Java Programs—Memory Management Issues - Meehan	803	Real-World Performance Testing Lab for (Almost) Free - Flint
404	Putting the User Back in User Acceptance Testing - Goldsmith	504	Performance Tuning ASP.NET 2.0 Applications, Part 1 - O'Mara	604	Performance Tuning ASP.NET 2.0 Applications, Part 2 - O'Mara	704	Defining Test Data and Data-Centric Application Testing - Hetzler	804	Testing Java Applications Using the Eclipse Test and Performance Tools Platform (TPTP) - Slauenwhite
405	Getting a Handle on Risk: Risk-Based Testing Strategies - Chaney	505	Identify and Mitigate Risks Through Testing, Part 1 - Black	605	Identify and Mitigate Risks Through Testing, Part 2 - Black	705	Coding Standards and Unit Testing—Why Bother? - Sokolov	805	Building a Bridge Between Functional Test Automation and Performance Testing - Sody
406	Java EE Performance Tuning Methodology: Wait-Based Tuning - Haines	506	Rapid Business-Driven Testing - Chaney	606	Exploiting Web Application Code: The Methodologies and Automation of SQL Injection - Fisher	706	The 5% Challenges of Test Automation - Buwalda	806	Best Practices for Managing Distributed Testing Teams - Stevens
407	Agile Test Development - Buwalda	507	Unit Testing for Agile Development, Part 1 - Sabourin	607	Unit Testing for Agile Development, Part 2 - Sabourin	707	Performance Testing for Managers - Barber	807	Diagnosing and Resolving J2EE Application Issues Before Deployment - Kilical, Au Yeung
408	Techniques for Testing Packaged Application Performance - Feaster	508	Using Code Metrics for Targeted Code Refactoring - Glover	608	Verifying Software Robustness - Collard	708	What Hollywood Can Teach You About Software Testing - Sabourin	808	GUI Test Automation for Eclipse RCP Applications - Rubel, Quitslund
		509	Recruiting, Hiring, Motivating and Retaining Top Testing Talent - Feldstein	609	Metrics: How to Track Things That Matter - Chaney				

Pricing and Registration

	eXtreme Early Bird	Super Early Bird	Early Bird	Full Price
Register By	By July 28	By Sept. 15	By Oct. 20	After Oct. 20
Full Event Passport November 7-9	$1,095 **Best Value**	$1,195	$1,365	$1,565
Two-Day Technical Conference Only November 8-9	$935	$1,035	$1,195	$1,345
Tutorials Only November 7	$695	$795	$845	$945
Exhibits Only November 8-9	FREE	FREE	FREE	$50

All prices are in US dollars

How to Register

REGISTER ONLINE. Register online at www.stpcon.com and use one of the following payment methods:

CREDIT CARD. You can use the secure online form to pay via credit card and get immediate confirmation of your classes. MasterCard, Visa and American Express are accepted cards. You'll receive a REGISTRATION RECORD and RECEIPT. Please print out these pages and bring them with you to the conference. Present them at the Registration Desk to pick up your badge and any course materials.

CHECK. Fill out the online registration form. Print out the REGISTRATION RECORD and RECEIPT and mail to BZ Media LLC, 7 High Street, Suite 407, Huntington, NY 11743. Be sure to include your payment. Online registrations that are mailed without payment will not be confirmed until payment is received.

P.O. If you register using a P.O., you will be invoiced immediately for the registration amount. Payment must be received before your registration can be confirmed.

GROUP DISCOUNTS. Registering four or more attendees from your company? You can receive a $100 discount off the Full Event Passport on each registration. Enter the word GROUP when asked for a code on our online registration form.

ALUMNI DISCOUNT. Have you attended the Software Test & Performance Conference in previous years? Welcome back! Enter the code ALUMNI to receive $100 off the Full Event Passport registration price.

REFUND POLICY. You can receive a full refund, less a $50 registration fee, for cancellations made by October 10, 2006. Cancellations after this date are non-refundable. Send your cancellation in writing to registration@bzmedia.com.

Paid Conference/Tutorial Registration Includes:
- Admission to tutorials and/or technical classes. Please make your class selections when registering.
- Admission to keynotes
- Admission to exhibits
- Conference materials
- Attendee reception
- Continental breakfast, coffee breaks

Exhibits-Only Registration Includes:
- Admission to keynotes
- Admission to exhibits
- Attendee reception

Register Online Today at www.stpcon.com!

Registration Questions
Contact Donna Esposito at +1-415-785-3419 or desposito@bzmedia.com.

Software Test & Performance CONFERENCE

BZ Media LLC
7 High Street
Suite 407
Huntington, NY 11743