Agenda Testing and Building Eclipse RCP Applications

- **Eclipse Plug-in and application design**
  - Importance of automatic builds
  - Automatic build validation via tests
  - Eclipse plug-in and application design for tests

- **Introduction Maven / Tycho**
  - Brief history
  - Overview, benefits, examples
  - Brief introduction to Maven
  - Relationship with other build technologies

- **Using Maven**
  - Introduction to Maven
  - Using Maven for Java programs
  - Maven dependency management
  - Using Maven repositories

- **Building an Eclipse plug-in**
  - Introduction of pom.xml
  - Review of the Tycho configurations
  - Running our first build on the command line
  - Dealing with dependencies - target platform & repositories

- **Building multiple plugins**
  - Aggregation & parent pom
  - Build order / reactor

- **Building features and p2 repositories**
  - The role of features
  - Creation of feature & pom

vogella GmbH, Haindaalwisch 17a, 22395 Hamburg, T +49 40 78 80 4360, F +49 40 52476322, Email info@vogella.com
HypoVereinsbank Hamburg, IBAN: DE82200300000015620690 , BIC/Swift-Code: HYVEDEMM300
Steuernummer: 50/766/01072; USt-IdentNr.: DE-284122352, Amtsgericht Hamburg: HRB 127058, Sitz der Gesellschaft: Hamburg
Geschäftsführer: Lars Vogel, Jennifer Nerlich de Vogel

- **Building p2 repositories**
  - The role of p2 repositories
  - Categorizing content in a repository
  - Good practices to manage p2 repositories

- **Building products**
  - What are RCP app / products
  - Building a product archive
  - Building a product repository
  - Multi-platform builds

- **Unit Testing with JUnit**
  - JUnit 4.x
  - Using JUnit test runner
  - Using JUnit Rules and parameterized tests

- **Developing unit tests for Eclipse components**
  - Plug-in tests
  - Overview of the Eclipse test API
  - Developing unit tests without Eclipse dependencies
  - Developing plug-in test tests

- **Using Mockito for replacing object dependencies**
  - Using Mockito to create test fixtures
  - Mocking method call and parameter access
  - Mocking method calss and parameter access

- **Using advanced Assert frameworks**
  - Using and extending Hamcrest Matcher
  - Using and extending AssertJ Matcher
  - Developing custom matchers and actions with Hamcrest and AssertJ

- **User interface testing within one application**
  - Using SWTBot
  - Using RCP Testing Tool (RCPTT)

- **Running tests**

- Create pom for plug-in test project
- Running JUnit tests
- Overview of SWTBot
- Dealing with the details (start level / additional dependencies)

- **Using non-OSGi libraries**
  - Why using non-OSGi libraries
  - Strategies for dealing with non-OSGi libraries

- **Continuous Integration**
  - Integration into Jenkins / Hudson
  - Jenkins / Hudson Quality plug-ins (sonar...)
  - Git Commit Hook

- **Updates with Eclipse p2**
  - Overview: the p2 provisioning platform
  - Creating p2 update sides
  - Using the p2 API
  - Updating products and features
  - Automatic product updates during startup