Your first Application based on Eclipse - FAQ

Jonas Helming - EclipseSource

jhelming@eclipsesource.com
What are OSGi, bundles and features?

Java Application

```
main(String[] args) {
    ...
}
```

OSGi Application

- Bundle 1
- Bundle 2
- Feature
- Bundle 3
- Bundle 4
RCP Application vs. Tool vs. IDE Plugin?

- Java Development Tools (JDT) → Tool Platform
- Rich Client Platform → OSGi
- OSGi → JVM
- Custom IDE Plugin (e.g. Checkstyle) → Custom Tool (e.g. CDT)
- RCP Application → OSGi Application
- Java Application
What is a target platform?

IDE

Workspace

Bundle 1

Target Platform

Eclipse Installation

Existing bundle

Update Site

Existing bundle

Directory

Existing bundle

Running Application

Bundle 1

Existing bundle
Eclipse 3.x vs. Eclipse 4.x vs. e4?

3.x Application

- Eclipse runtime
- Workbench (org.eclipse.ui)

JFace

OSGi

SWT
Eclipse 3.x vs. Eclipse 4 vs. e4?

“pure” Eclipse 4 Application

Eclipse runtime

JFace

OSGi

SWT

e4: Application Model, Services, Dependency Injection
Eclipse 3.x vs. Eclipse 4 vs. e4?

- Eclipse 4.x Application (Compatibility Layer)
  - Workbench Compatibility Layer (org.eclipse.ui.e4)
  - e4: Application Model, Services, Dependency Injection
- Eclipse runtime
- OSGi
- JFace
- SWT
Eclipse 3.x vs. Eclipse 4 vs. e4?

- Don’t go for Eclipse 3.x anymore, choose between:
  - “pure” Eclipse 4
  - Compatibility layer (3.x API based on Eclipse 4.x)
- 3.x API and “pure” Eclipse 4 generally do not mix well
- Existing IDE related frameworks usually use the 3.x API
- Look at the frameworks you are going to use:
  - Many 3.x based => Go for the compatibility layer
  - A few 3.x based => evaluate their adaptation to Eclipse 4
  - No 3.x based => go for “pure” Eclipse 4
- If you develop for the 3.x API, do it in an Eclipse 4 friendly way (for more details see hand-out)
How to define APIs?

Bundle 1

MyCallingClass

Bundle 2

MyClass1

MyClass2

MyClass3

MyClass4
How to define APIs?

- MyCallingClass
  - Bundle 1

API Package
- MyInterface
  - Bundle 2

- MyClass2
- MyClass3
- MyClass4
  - Internal Package
How to set up a build?

- Maven
- Tycho

- pom.xml
- pom.xml
- pom.xml

- Product
  - FeatureA
    - Bundle 1
    - Bundle 2
  - FeatureB
    - Bundle 3
    - Bundle 4
How to select additional technologies

- Take time for your technology decisions
- Do not reinvent the wheel
- Look at articles, blogs, websites
- Visit events, such as EclipseCon or Democamps
- Have a detailed look at a project:
  - Committer activity and diversity
  - Newsgroup activity
- Understand the stakeholders of a project
- Get in contact with the people behind a project
- Do not expect anything to happen without your influence
- Be ready to invest in support, training, sponsored development and maintenance
Even more Questions?

- How to manage dependencies?
- How to deal with version ins OSGi?
- How to split bundles
- (...)

- Get a textual version of this talk: http://goo.gl/ATJeUt

- Contact me: jhelming@eclipsesource.com
Evaluate the Sessions

Sign in and vote at eclipsecon.org
How to split bundles?

Guidelines for designing a bundle structure:

- Feature-based
- Split UI and core
- Keep entities separate
- Use Cases for exchanging/extending things?
- Use Cases for reusing things?
- Optional features?
- Dependency-based