OSGi in the TK - The BundleMaker

Nils Hartmann, Techniker Krankenkasse
Gerd Wütherich, Freelancer
Initial Situation
The Project TKeasy

- Main application at Techniker Krankenkasse (TK)
  - One of Germany’s largest health insurance funds

- 3-Tier architecture
- Started in 1998
- More than 100 applications
- 65,000 Java classes
- 6500 packages
- 5 mio lines of code
- 100 active developers
Dependency problems
Why Modularity?

» Need to modularize the application
  » Current focus: the module layer
  » Easier maintainance
  » Reuse of components in other contexts

Why OSGi?

» „The“ module system for Java
» Open infrastructure: Extender Pattern, Hooks
» Future: introduce service layer
The Migration Path
The Development Cycle

~ 100 Developer

TKeasy V. 5.18

~ 100 Developer

TKeasy V. 5.19
Continuous Migration I

~ 100 Developer

TKeasy V. 5.18

~ 100 Developer

TKeasy V. 5.19
Continuous Migration II

~ 100 Developer

TKeasy V. 5.18

~ 100 Developer

TKeasy V. 5.19

~ 2-3 Developer

~ 2-3 Developer

~ 2-3 Developer

~ 2-3 Developer
Continuous Migration III

~ 100 Developer

TKeasy V. 5.18

Automated with BundleMaker

~ 100 Developer

TKeasy V. 5.19

Automated with BundleMaker
Continuous Migration IV
The „Final Transformation“
The BundleMaker
BundleMaker „Open Source Edition“

» Alpha Version

» http://www.bundlemaker.org

» Source code available
  » http://github.com/wuetherich/bundlemaker
  » EPL

» Goal: Open platform for software analysis applications
BundleMaker Concepts

1. Parse
   - Classes (Folder, JAR)
   - Source-Files (Folder, ZIP)
   - Bundles (Folder, JAR)
   - Eclipse-Projects (JDT, PDE)
   - Maven-Projects

2. Analyse

3. Transform

4. Export
   - Bundles
   - PDE-Plug-in-Projects
   - JDT/BND-Projects
   - Maven-OSGi-Projects
BundleMaker - Analyze
BundleMaker – Transform
BundleMaker - Export
Thank you very much!