Eclipse Scout 4 Decision Makers
Scout UGM 2016, Ludwigsburg

@ZimMatthias, Matthias Zimmermann, BSI Business Systems Integration AG
Agenda

➔ What is Scout
➔ Benefits of Scout
➔ Framework Features
➔ Two Lessons from the Past
➔ Take Home Message
What is Scout?
What is Scout?

Business Application Framework

→ Based on Java and HTML5
→ Multi Device support

Framework Goals

→ Long Term Strategy (enterprise apps live > 10 years)
→ Boosts Productivity (producing software in Switzerland ...)
→ Easy to learn (new team members productive in 1-2 weeks)
Eclipse Scout
Scout Benefits
Scout Benefits

- Efficient Development
- Sustainable Investment
- Professional Support
- Open Source
Efficient Development

Scout boosts Productivity

➤ Covers Communication, Security, Data Binding, Rendering
➤ Concentration on Business Functionality
➤ Concentration on User friendly UI
Sustainable Investment

Business Code keeps its Value

➔ Business Code written against Application Model
➔ Application Model is technology neutral
➔ The Framework keeps up with Technologies
Professional Support

Do more in Less Time

➔ Bug Fixes and Custom Builds on short Notice
➔ Guaranteed Response Times
➔ Training and Consulting
Open Source

Benefits of an Eclipse Project

- Free for commercial and personal Use
- World-class IP Management
- Yearly Releases on Time
Scout Features
Framework Features

» Multi Device Support
» Modular Applications
» Java Application Model
» HTML5 Rendering
Multi Device Support

Single Code Base

➔ Desktop
➔ Tablet
➔ Mobile
Modular Applications

Clean Modularization

→ Layers - Technical Modules
→ Slices - Business Modules
Scout Application Model

Java Application Model

- Clean Business Code
- Long Term Maintainability
HTML5 Rendering

Current Standards

➔ HTML5, CSS3, JavaScript
➔ Styling and Theming
2 Lessons for the Future
Two Lessons for the Future

1. UI Technologies are «short-lived»
2. Scout Code is maintainable
UI Technologies are «short-lived»

Scout UI Technologies
➔ 2002 Swing
➔ 2006 SWT
➔ 2011 RAP
➔ 2015 HTML5

Substantial Framework Efforts
New UI Technology every 4 years

Minor Project Efforts
Usually < 3% of Project Codebase affected
«Today’s Trend is tomorrow’s Legacy»
UI Dos and Don'ts

Don’ts

→ Don’t mix UI Technology and business code
→ Don’t «marry» specific UI Technologies

Dos

→ Think long Term
→ Abstract UI Technologies
→ Use a Framework like Scout that enforces this Abstraction
Scout Code is Maintainable

Example: BSI CRM
- Based on Scout since 1999
- Scout defines an Application Model
- CRM is based on this Model
- CRM UI over the years ...
... 1999
... 2002

<xml />
... 2005
... 2007
Eclipse Scout 2011
Eclipse Scout 2012
Eclipse Scout 2015/16
Scout Code is Maintainable

Example: BSI CRM

- Based on Scout since 1999
- Scout defines an Application Model
- BSI CRM is based on this Model

Observations

- CRM Code was never thrown away
- Migrations from Tech-Stack to Tech-Stack
- Scout had a big Part in this
Take Home Message
Scout is good for Business Applications

- Scout moves with current Technologies
- Robust, proven in Practice for many Years
- Business Code keeps its Value
  - Written against Scout model
  - Remains technology neutral
Thanks
@EclipseScout