#Sperasoft_Talks:

JIRA Development

What is JIRA?
What can we do with it?

Agenda

- Intro: what is JIRA?
- Custom system over JIRA in comparison
- JIRA plugin development
- Plugin module types
- Plugin module description
- Interesting problems:
  - Configuration migration
  - Building data model scheme in JIRA

SPERASOFT

Intro

- What is JIRA?
  - bugtracker
- What does JIRA have?
  - issues
  - workflows
  - user management
  - security
  - great configuration & customization ability

Intro (cont’d)

- What does custom System have?
  - entities
  - workflows
  - user management
  - security
  - reports
  - other systems integration

Custom system over JIRA

- Habitual interface
- Simple Atlassian products integration
  - other JIRA
  - Confluence
- Flexible configuration (issue types, custom fields, workflows, events & mail)
- Powerful plugin system
- Community with developers (sometimes)

SPERASOFT

Custom system over JIRA (cont’d)

- Proprietary source code
- Issues linking scheme
- No consistency (custom fields implementation)
- Velocity templates hell

Plugin Development

- Atlassian Plugin SDK (with Maven2 & Tomcat7)
- Plugin - *.JAR, contains of modules
- FastDev saves your time
- atlassian-plugin.xml – core plugin configuration file
- Lots of plugin module types

Plugin Module Types

- Web item, Web section
- Web panel
- REST
- Servlet
- Workflow customization (post functions, conditions, validators)
- Custom field type
- Component
- Many others…

Plugin Module Description

- Unique key
- Name, description
- Implementation class name
- Template files
- Parameters

Plugin Module Description

```
<web-panel key="issue.budgetcost.items" location="atl.jira.view.issue.left.context" weight="140">
    <label key="issue.budgetcost.items.name"/>
    <resource name="view" type="velocity" location="/templates/issue/budgetCostItems.vm"/>
    <context-provider class="com.sperasoft.erp.provider.BudgetCostItemsContextProvider"/>
    <condition class="com.sperasoft.erp.webcondition.BudgetCostIssueTypeCondition"/>
</web-panel>
```

```
<rest key="rest" path="/erp" version="1.0">
    <description>ERP REST service</description>
</rest>
```

```
<workflow-function key="postfunction.createremoteissue" name="Create remote issue"
                   class="com.sperasoft.erp.postfunction.CreateRemoteIssueFunctionFactory">
    <description>Creates an issue in remote JIRA for current issue</description>
    <function-class>com.sperasoft.erp.postfunction.CreateRemoteIssueFunction</function-class>
    <orderable>true</orderable>
    <unique>false</unique>
    <deletable>true</deletable>
    <resource type="velocity" name="view" location="/templates/workflow/postfunction/cri-view.vm"/>
    <resource type="velocity" name="input-parameters" location="/templates/workflow/postfunction/cri-edit.vm"/>
    <resource type="velocity" name="edit-parameters" location="/templates/workflow/postfunction/cri-edit.vm"/>
</workflow-function>
```

SPERASOFT

Follow us on Twitter
@Sperasoft

Visit our site:
sperasoft.com

SPERASOFT

Interesting Problems

- Configuration migration
- Data Model. One-to-many, many-to-many schemes for issues

SPERASOFT

Configuraion Migration

- Projects
- Issue types & schemes
- Statuses
- Resolutions
- Workflows & schemes
- Screens
- Screen configurations & schemes
- Custom fields
- Custom field configurations & schemes
- Project roles
- Security schemes
- Events

SPERASOFT

Configuraion Migration

- Goals:
  - migrate configuration Dev -> Test -> Sandbox -> Production
  - apply configuration to a clean JIRA instance
- Complexities:
  - replace all IDs with names
  - JIRA configuration API hell

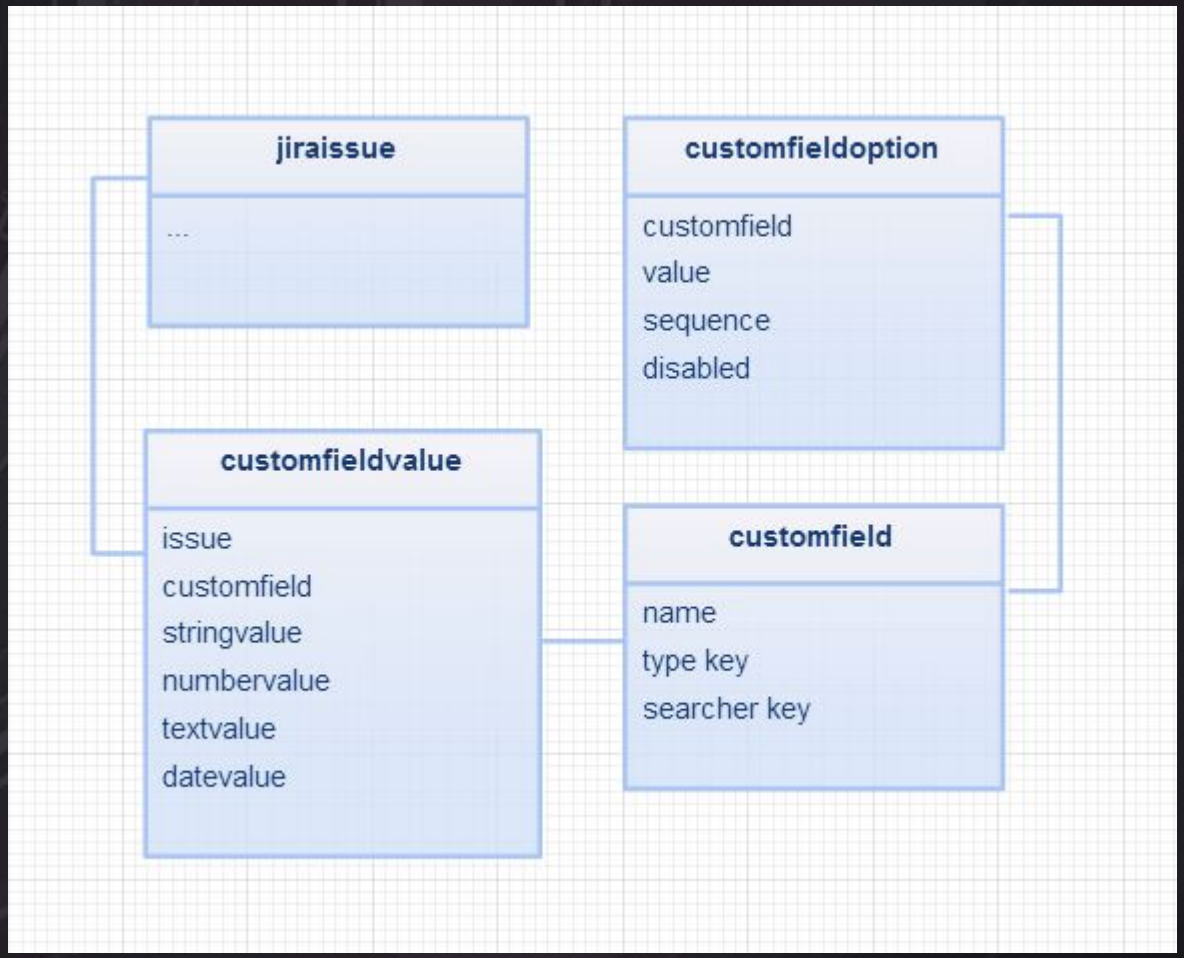
Configuration Migration (cont'd)

- Schema:
  - configuration model with JAXB annotations to produce XML
  - converters for configuration entities
  - replacers for IDs and names
  - reader and Writer that iterate configuration entities and read or write them
- Result:
  - 1 MB xml file

Data Model

- Database tables:
  - customfield
  - customfieldoption
  - customfieldvalue
- Value types storage:
  - Limited text
  - Unlimited text (clob)
  - Numeric
  - Date

Data Model (cont'd)



Data Model (cont’d)

- Solution:
  - Field: using AbstractSingleFieldType class with SortableCustomField interface
  - store issue’s ID
  - many-to-one – custom field with one value
  - many-to-many – custom field with multiple values
  - Searcher: using AbstractInitializationCustomFieldSearcher class and lots of utility classes providing search info.

Docs

- https://developer.atlassian.com
- https://answers.atlassian.com
- http://j-tricks.com
- JIRA Development Cookbook

SPERASOFT

Thanks!

Have a question?
Like this deck?

Just follow us on twitter
@Sperasoft

SPERASOFT