1st OpenCms Developer Sprint
September 9 – 10, 2008

Alexander Kandzior,
CEO Alkacon Software GmbH
2 Day Schedule

• Tuesday, September 9, 2008
  - 9:30h Start / Discussion of Topics
  - 13:00h Lunch Break
  - 17:00h End
  - 19:00h (Optional) Evening Activity

• Wednesday, September 10, 2008
  - 9:00h Start
  - 13:00h Lunch Break
  - 17:00h End
  - Coffee Breaks: 11:00h, 16:00h
How do we go ahead?

- Each participant picks a topic of interest
  - If 2 people want to work on the same topic, then we do Extreme Programming for this
- The topic should be good for the 2 days and probably beyond
  - We certainly hope to increase the level of outside contributions to the OpenCms core
- If we can not finish a topic within these 2 days, a “proof of concept” or “First Milestone” should be reached
- Have fun working together
Recording the Sprint proceedings

- We have set-up an OpenCms Sprint Server to record the proceedings of the sprint
  - This will later get published online at opencms.org
- We hope to have one or more pages describing the findings for a topic at the end
  - Proceedings could also contain code fragments or other material created during the Sprint
- OpenCms Sprint Server runs OpenCms with template², each team will have its own folder
Suggestions for Sprint Topics

- Creating navigations with XML Content
- Using jQuery for the OpenCms Explorer
- Integrating JavaScript with OpenCms
- OpenCms with Google Chrome / Gears
- Integrating Workflow in OpenCms
- OpenCms database layer with JPA
- Integrating OpenCms with the Maven Plugin
- Updating OpenCms 3rd party libraries
- Using foreign key constraints in the OpenCms DB schema
In this sprint: The existing API could be enhanced to support both approaches to provide maximum backward compatibility

- OpenCms “CmsJspNavElement” instances could be created from default XML content in folders, if unavailable fallback to current properties

Advantages: Change in Navigation does not require publishing of the content file, “Direct Edit” could be used to edit the navigation
Topic: Using jQuery for the Explorer

• OpenCms Explorer is using “Ajax like” technologies
• Modern Ajax APIs like jQuery are now quite stable and offer a range of potential new features that could be leveraged
• Using jQuery for the OpenCms Explorer could improve usability and user experience
  – Required changes to OpenCms core could be rather small
  – Data for explorer is already handled similar to JSON, maybe directly use JSON for the data transfer

• In this sprint: Create a POC for a jQuery based OpenCms Workplace explorer
OpenCms 8 should include a scripting language interface to make it easier for non-Java developers to create standard templates and other basic functionality.

JavaScript seems a natural choice for the scripting language (?)

- Script integration with the JS Rhino interpreter is already available in the Java JRE since version 6.

*In this sprint*: POC for integration of JavaScript enabled templates with access to simple OpenCms functions.
OpenCms has so far not been much tested with Google Chrome
  - General functionality seems to work
  - External components (FCKeditor, Editarea) don’t

It would be interesting to investigate what can be done with Google Gears

In this sprint: Test OpenCms on Chrome, check new versions of external components, evaluate possible use cases for Gears
What is the general approach
- Alkacon has already done some initial work on the subject, it would be great if we could build on this basic infrastructure

In this sprint: Define a minimal interface / API for the OpenCms core
- Apparently, a “Workflow engine” could be somehow plugged into OpenCms
- Adding Workflow will require some changes to the OpenCms Workplace GUI, e.g. in the Explorer view we should see a “status info”
- General Triggers / Communication from the Workflow engine with the Core need API
JPA (Java Persistence API) is a JSR standard that aims to replace Hibernate, JDO and other “non standard” solutions.

OpenCms 8 should offer a JPA DB driver, either as replacement or alternative to the individual DB drivers.
- We would like to drop individual drivers where performance of JPA is close enough.

Georgi Naplatanov has already developed full JPA support for OpenCms.

In this sprint: Testing and potentially enhancing his work.
A Maven plugin has already been developed for OpenCms. To my knowledge, it is so far not compatible with the standard OpenCms core but requires patches to work.

In this sprint: We should find a way to make the plugin work on a future release of the OpenCms core by changing / adding the required APIs in either the Maven plugin or the core.
OpenCms uses a number of external libraries
Not all libraries are of the current version
In rare cases we use a non-standard patched version
  - Example: commons-email.jar has been patched to remove dependencies to other .jars we don’t want to ship

In this sprint: Try to replace patched libraries with newer standard versions, update external libraries in general
OpenCms does so far not make of “foreign key” constraints in the OpenCms DB schema
Adding foreign key constraints could improve DB consistency
Alkacon has tried this some time ago but found that this will also require changes to the OpenCms core
Moreover, the OpenCms DB Schema needs to be adjusted for the properties
In this sprint: Try to add foreign key constraints to the DB, find out where changes to the core are required and try to fix these issues if possible
Evening activity (optional)

- You are invited for a Dinner at the “Osman³⁰” Restaurant
- Start at 19:00h
- [http://www.osman-cologne.de](http://www.osman-cologne.de) for more info
Thank you very much for your attention

Alexander Kandzior
Alkacon Software

http://www.alkacon.com
http://www.opencms.org