OCS

Contribution Guidelines

1 Introduction

This document describes the guidelines to contribute in the OCS open source project.

It is important to read and understand the agreements and strive to meet the standards expected.

Some procedures may appear a little complex, but they are there to protect you as well as the OCS project.

Before starting to code, the contributors must read the corresponding documentation.

1.1Disclaimer

Any information in here that might be perceived as legal information is informational only.

1.2 Licensing

The OCS project is open source and developed/distributed under the European Union Public Licence – EUPL (http://joinup.ec.europa.eu/software/page/eupl)

If you wish to contribute to OCS (which you're very welcome and encouraged to do so) then you must agree to release the rights of your source to us under this license.

2 Contributors

Anyone can be a contributor to the OCS project.
Being a contributor simply means that you take an interest in the project and want to develop new features.

3 How to contribute

3.1 Code conventions

Each OCS developer must follow code conventions.

The code conventions to be follow by the contributors can be found here:
http://www.oracle.com/technetwork/java/codeconvtoc-136057.html
(*Code Conventions for the Java Programming Language -revised April 20, 1999*)

3.2 Developer check list

When submitting a patch, each contributor should follow the developer check list.

1. A Jira ticket should exist for this feature.
   Jira is the task/bug tracking system use by OCS.
2. The Jira ticket should be assigned to the contributor.
3. The added code complies with coding standards.
4. The added code is always based on the most recent SVN version. It's the responsibility of the developer to keep his project up to date.
5. The added code compiles and runs on Java SE 1.6 (latest version) and Java EE 5
6. The new java files begin with a reference to the European Union Public Licence (EUPL) statement: http://joinup.ec.europa.eu/software/page/eupl.
7. The code includes the @author tag on any files you've altered or created.
8. The existing test cases succeed.
9. The new test cases written and succeed (unit, functional, performance).
10. A unit test line coverage report (with 80 % line coverage).
11. The diff files generated using SVN diff.
12. *The documentation page extended as appropriate.*

3.3 Contributors workflow

The OCS project don't allow commit on its source control system (Subversion). The Subversion commit are restricted to "core developers".

In order to commit his modification, a developer must send a patch to the OCS team.

This can be done by attaching the patch to an existing Jira issue (new feature).

The patch file is an archive (patch.tar.gz) containing at least an unified diff of files that have been modified and also include files that have been added.
The developer must review the file for completeness and correctness.

The validated (tested) and reviewed patch is committed inside subversion by a core developer.

During the testing phase, the tested functionalities are those who are described in the Jira ticket.

The new contribution is available for the entire community.

3.3.1 Contributors workflow steps

The contributors' workflow steps are:

1. The proposal for new feature/bug fix is accepted by the boards.
2. If a contributor promises he will implement this, he becomes the owner of this task.
3. The decision board + contributor + core developers decide on a due date.
4. A Jira ticket is created containing the needed information/requirements.
5. The contributor can start working on the ticket.
6. When the contributor finishes implementing, he completes the checklist and attaches the needed artefacts to the JIRA ticket (filled checklist and unified diff).
7. A core developer needs to review the work done by contributor.
8. The core developer marks the Jira ticket as "In Review".
9. The core developer integrates the patch into the local version.
10. The core developer checks the patch against the checklist document.
11. If the check fails:

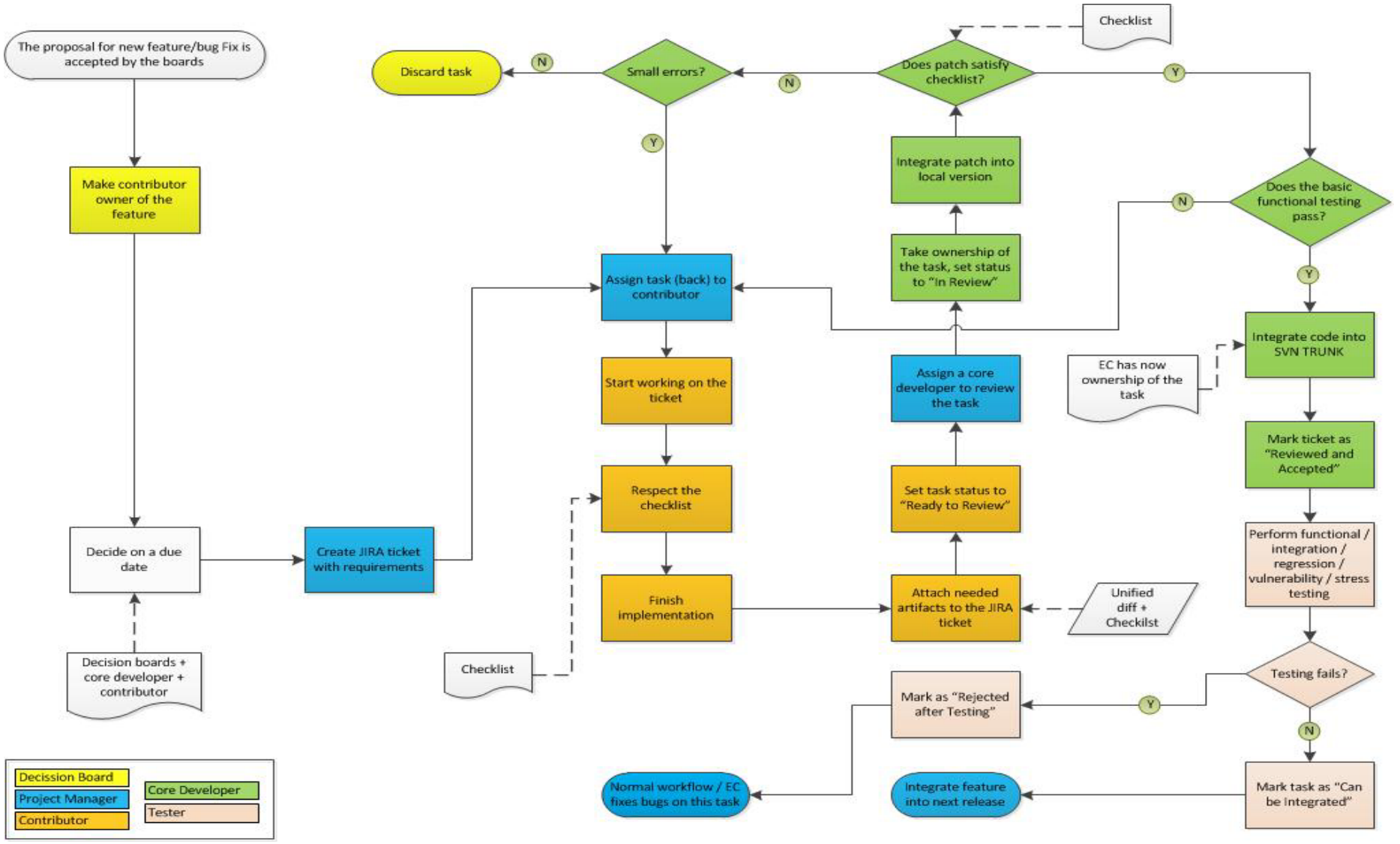
a. Because of small mistakes, the issue is marked as "Return to Assignee" and the workflow resumes from point 10.

b. Because of bigger mistakes (security issues, requirement violations, etc.), the issue will be discarded.

12. If the check passes, the core developer must do some basic functional testing.
13. If the functional testing fails, the task is returned to assignee and the process continues from 10.
14. If the functional testing passes, the core developer can commit the code to the SVN repository (TRUNK).
15. The ticket is marked as "Reviewed and Accepted".
16. A functional/regression/integration/vulnerability/stress testing must be performed by a tester.
17. If this testing fails, the issue is marked as "Rejected after testing" and EC takes the ownership of the task fixing any bugs which were discovered.
18. If functional/regression testing passes, the ticket marked as "Can be integrated" and closed.
19. This feature can now be integrated into the new version.

The diagram bellow explains how a contributor can submit his code to OCS.

3.3.2 Contributions workflow diagram



3.4 Class header template

Each new class must begin with the following header template:

```
/* ============================================================
* Licensed under the European Union Public Licence (EUPL v1.2)._
* [https://joinup.ec.europa.eu/community/eupl/topic/public-consultation-draft-eupl-v12|https://joinup.ec.europa.eu/community/eupl/topic/public-consultation-draft-eupl-v12]
* ============================================================
*
*
*<p>
* General description (documentation) of the class.
*</p>
*
* @author JDoe <a
href="[mailto:johndoe@emailaddress.com]"><johndoe@emailaddress.com></a>
* @created: 02/05/2013
*
*/
```

The author name could be replaced by the complete name.
The @created tag could be replaced by a @since tag.

4 OCS tools

OCS uses several tools to make the developer life better:

1. Subversion
2. Jira
3. Confluence

4.1 Subversion

Subversion is the source version control of OCS.

The OCS Subversion is only available in checkout mode.
You can reach it here: https://webgate.ec.europa.eu/CITnet/jira/browse/OCS

The up to date code will be always present in the trunk.

4.2 Jira

Jira is the bug/issue/task tracking system of OCS

It can be reached at the following address:
https://webgate.ec.europa.eu/CITnet/jira/browse/OCS

4.3 Confluence

Confluence is the wiki used by OCS for documentation purpose.

The OCS documentation space can be reached at the following address:
https://webgate.ec.europa.eu/CITnet/confluence/display/OCS

4.4 Access

The accesses of those tools are protected with a user and a password.
To obtain an access, you must send an email to DIGIT-ECI@ec.europa.eu containing yours:

- First name
- Last name
- Email Address

(OCS doesn't allow anonymous contributors)

You will receive a confirmation by email.