Table of Contents

Change Management Policy

Latest approved version of this policy
28.03.2011: EMI_SA2_Change_v_2_0.pdf

Introduction

This document describes the EMI policy to be followed to manage software changes in EMI software components.

There is a project deliverable describing the Software Maintenance and Support Process. This policy is kept synchronised with this deliverable. For more information on the deliverable check DSA11.

Change Management Process

EMI Releases

The EMI distribution will be organized in periodic major releases, tentatively delivered once a year, providing a good balance between the conflicting requirements of stability and innovation.

An EMI major release is characterized by well-defined interfaces, behavior and dependencies for all included components, available on a predefined set of platforms. What is included in a new EMI major release is defined by the PTB and the implementation of the plan is coordinated by JRA1.

Backward-incompatible changes to the interface or to the behavior of a component that is part of the EMI distribution can be introduced only in a new EMI major release. Changes to interfaces that are visible outside the node where the component runs (e.g. a WSDL) need to be preserved even across major releases, according to end-of-life policies to be defined on a case-by-case basis.

The availability of a new major release of EMI does not automatically obsolete the previous ones and multiple major releases may be supported at the same time according to their negotiated end-of-life policies.

Component Releases (CR)

A component release is a new version of an EMI software component. The list of EMI software components can be found in Section 4 `EMI Components and Product Teams`, page 11 of DNA1.3.1 - Technical Development Plan.

An EMI distribution includes all the components that are developed within the project and that have reached production quality. Within an EMI major release, only one major version of a given component is maintained. Four types of releases have been identified for a given component:

- **Major Release**: A major release for a component is characterized by a well-defined interface and behavior, potentially incompatible with the interface or behavior of a previous release. New major releases of a component can be introduced only in a new major release of EMI. The contents of a new major release are endorsed by the PTB and included in the project technical plan. The implementation is coordinated by JRA1.
- **Minor Release**: A minor release of a component includes significant interface or behavior changes that are backwards-compatible with those of the corresponding major release. New minor releases of a component can be introduced in an existing major release of EMI. The contents of a new minor release are endorsed by the PTB and included in the project technical plan. The implementation is coordinated by JRA1. If the release is going to be introduced in an existing major release of EMI, the

implementation is also supervised by SA1 in order to guarantee that the production quality and the backwards-compatibility are preserved.
- **Revision Releases**: A revision release of a component includes changes fixing specific defects found in production and represents the typical kind of release of a component during the lifetime of an EMI major release.
- **Emergency Releases**: An emergency release of a component includes changes fixing only Immediate-priority defects found in production, typically security-related.

CR Tracking

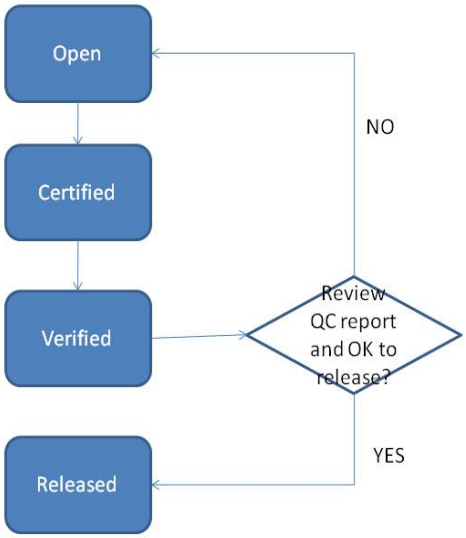
Component Releases are tracked in the EMI Release Tracker in Savannah and they should contain the following information:

- `Unique identifier`: This is automatically created by Savannah when a new task is generated.
- `Should Start On`: Ignore this field within the EMI context. It's a field that always appears in a Savannah task template. PTs can use for it internal purposes if they want to.
- `Should be Finished On`: The date by which the CR should be released in Production.
- `Category`: Type of tracker entry. Values can be `Component` or `Release`. `Component` refers to tasks tracking new component versions. `Release` refers to a set of new component versions.
- `Technical Area`: Lists one or more of the four EMI technical areas that are relevant to the component/release, that is: Compute Area, Data Area, Infrastructure Area and Security Area.
- `UMD Capability`: It specifies one or more of the UMD capabilities provided by the component/release. *More details to be confirmed by PEB and EGI.*
- `Priority`: `Normal` unless it's an emergency release. In that case it should be set to `High`.
- `Status`: see next section.
- `Assigned to`: savannah user name of the PT responsible person for the component. If it's a release, it's assigned to he release manager.
- `Open/Closed`: field that tracks whether the task is open and closed. It's automatically managed by Savannah.
- `Discussion Lock`: ignore this field. It's a field that is always set to `unlocked` and it's defined by the tracker manager.
- `Release`: EMI major release.
- `Name`: Name of the component or release.
- `Component Version`: version of the component/release.
- `List of elements`: A component/release can include several `elements` that are listed in the EMI Technical development plan. Official component table in Section 4, page 11.
- `List of RfCs`: links to the corresponding items in the different tracking systems describing which bugs/feature requests are fixed in this release.
- `Package list`: list of packages affected by the change and the URL from where they can be downloaded.
- `Documentation`: links to the relevant documentation. It should at least contain:
    - **General Information URL**
    - **User guide URL**
    - **System Administrator guide URL**
- `Component Release Notes`: non-technical text written in good english giving an overview of the change introduced by the packages. The text should be prepared by the PT responsible for the component. It should contain the following structure:
    - **What's new**: brief description of the main changes, both new features and bug fixes.
    - **Installation and configuration**: more details on installation and configuration stating very clearly if the service must be reconfigured and/or restarted.
    - **Know issues**: known issues present in the release, possibly with a workaround.
- `License`: link to the file describing the license under which the component is released.
- `Extended Release Notes`: link to the web page, if any, where an extended and more detailed version of the release notes can be found.

- `Test Plan Link`: link to the test plan used to test the CR.

Component releases are created in Savannah by the release manager although PTs will have to fill in the requested information as described in the next section.

CR state transition diagram



- **Open**: the Release manager is responsible for creating CR items in Savannah to track the different scheduled releases. This should be done with the assistance of the JRA1 activity leader making sure the different development plans are fully covered in the release schedule. Check the Release Management Policy for more details on this. All CRs should be either planned according to the different development plans, or in case they are needed because an unplanned change needs to be introduced, new CR items will be created in Savannah at any time by the Release manager. At this moment, the Release manager together with the PTs, defines a `Due Date` and some other mandatory fields.
- **Open to Certified**: the PT moves the CR to `Certified` once the development, testing and the certification of the CR has finished. Before this transition happens, the PT has to make sure all the fields in the CR described above are properly filled in.
- **Certified**: the work of the PT has finished at this stage. This stage triggers the SA1 QC team to verify that the CR complies with the EMI Production Release criteria.
- **Certified to Verified**: The SA1 QC team moves the CR to `Verified` after doing the evaluation of the information contained in the CR. QC will check whether the CR succeds to meet the EMI Production Release criteria. The SA1 QC team should include the result of its evaluation as an attachment to the CR.
- **Verified**: the work of the SA1 QC team has finished at this stage. This stage triggers the release manager.
- **Verified** to **Released**: If the release manager is happy with the QC report, the new packages are installed in the testbed for inter-component testing. If inter-component testing is successsful, the packages are copied in the EMI production repository.
- **Released**: the component release is available in the EMI production repository. This state automatically closes the savannah ticket.

Request for Change (RfC)

A Request for Change (RfC) is a formal request to change one or more software components. A change in the software may be motivated by:

- GGUS tickets where users report incidents or make requests.
- High level user requirements coming from DCIs.
- Technical objectives defined in the Technical Area work plans.
- PTs fixing defects or introducing unplanned improvements.

RfC Tracking

Changes are handled via a tracking tool. A change should be evaluated as soon as possible by the PTs to accept it or reject it. A period of two weeks has been defined to carry oit this evaluation.

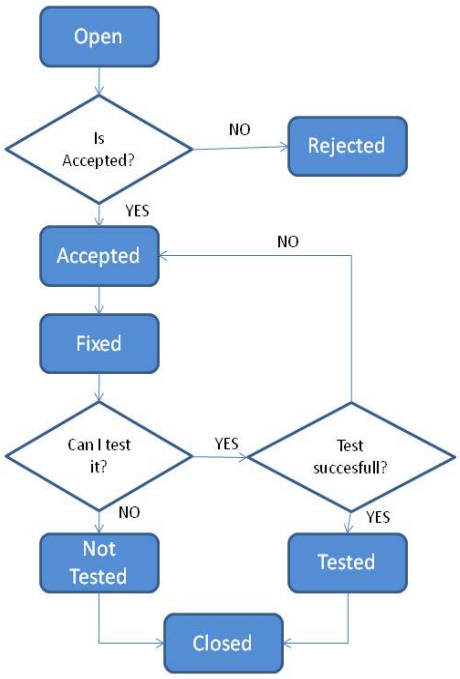
An RfC should be created per each EMI major releases. If within the same major release, only one of the supported platforms requires a change, then a RfC per platform should be created as well.

The tracking tool should contain the following information for an RfC:

- **Unique identifier**.
- **Title**.
- **Longer description**.
- **Software component** affected by the change (Category).
- **Priority** of the change. The priority of the change is based on severity, impact, urgency and cost.
    - **Inmediate**: The RfC needs to be addressed as soon as possible, in all affected EMI major releases. A release containing immediate- priority changes can contain only immediate-priority changes. Multiple immediate-priority changes can be included in the same release, provided that any change does not delay the release significantly.
    - **High**: The RfC will be addressed in a next release of the affected component, in all affected EMI major releases.
    - **Medium**: The RfC will be addressed in the release of the affected component that will be shipped with the next EMI major release.
    - **Low**: There is no target date for addressing the RfC.
- **Defect vs Feature**. To differenciate between software bugs and new features.
- **Status** of the change. It shows in which stage of the Change Management Process the RfC is. See the Change Status section below.
- The **area** where the change was orignally raised. The RfC may be motivated by a problem detected in the production infrastructure or somewhere else like development or certification. If the RfC comes from a GGUS ticket, the RfC should contain a link to the corresponding GGUS ticket.
- Associated **EMI major release**.

RfC state transition diagram

The following diagram represents the minimum set of states that should be present in any of the tracking systems:



- **Open**: The RfC is opened after all the necessary clarifications have been made. This may include discussions in a GGUS ticket to understand if there's an actual problem, internal discussions within the PT after a defect has been found or after an improvement has been proposed; etc.
- **Open to Accepted**: The clarification is complete and the RfC is accepted to be implemented.
- **Open to Rejected**: The clarification is complete and it's been decided in the end not to implement the RfC.
- **Accepted**: The RfC is ready to be implemented by the PT. This state triggers the implementation of the RfC within the PT.
- **Rejected**: The RfC won't be implemented and the PT should explain why it has been rejected.
- **Accepted to Fixed**: The implementation of the RfC has finished, that is, the code has been committed.
- **Fixed**: The code is committed. Once the packages are ready within the PT, testing of the new release can start, including the testing of all RfC within the release.
- **Fixed to Not Tested**: The PT is not able to test the RfC.
- **Not Tested**: The RfC can't be tested and the PT should explain why.
- **Fixed to Tested**: The PT tests the RfC by running the relevant tests. If the tests are succesful, the RfC can be moved to `Tested`.
- **Tested**: The RfC has been succesfully tested.
- **Fixed to Accepted**: The PT tests the RfC but the tests fail. This means the new feature/bug hasn't been properly implemented.
- **Tested to Closed** and **Not Tested to Closed**: Once the corresponding task where the RfC is included is released to production, the PT closes the RfC.
- **Closed**: The RfC is now available in the EMI production repository.

Roles

- Change Advisory Board: Although the priority of an RfC can be suggested by the responsible PT, oficially it is the PTB with the support of the SA1 activity leader who should. assess the RfCs and determine their priority and their association with the EMI major release(s) where they will be implemented. The PTB can also delegate the decisions concerning corrective and adaptive maintenance to the EMT.
- Change Manager: i.e. following the process of controlling the lifecycle of approved changes, is taken either by the SA1 Maintenance task leader or by the JRA1 leader, depending on whether that RfC is

going to be applied to a component release to be delivered within an existing EMI major release or within the next one.

Contact

- maria.alandes.pradillo@cernSPAMNOT.ch

References

No references.

Logbook

v1.0 (approved on 13.12.2010)

- Version 1.0 ready on 17.12.2010. To be announced on EMT 20.12.2010.

v2.0 (approved on 28.03.2011)

- *18th Feb 2011*: Changes requested by Technical Director:
  - Create new fields in CR tracker: `Test Plan Link`, `License`, `Extended Release Notes`.
- *23rd Feb 2011*: Changes requested by Technical Director:
  - Create new fields in the CR tracker: `UMD capability`, `Technical Area`, `List of elements`.
  - Rename the following fields: `Component version` to `Version` and `Component name` to `Name`.
  - Change the meaning of `Category` to only `release` and `component`.
- *1st March 2011*: Added changes provided by Technical director (definitions). Some clarifications still pending. Mail sent to Technical Director.
- *3rd March 2011*: All definitions are now clear.
- *10th March 2011*: After discussing with QC, some improvements are done in the RfC state and transition definitions. Use `tested` instead of `certified` to be aligned with the meaning of this terms within the project.
- *16th March 2011*: Added more specific information on the Documentation field of the CR task.

This topic: EMI > EmiSa2ChangePolicy
Topic revision: r15 - 28-Mar-2011 - 18:00:35 - MariaALANDESPRADILLO