WFT
Workflow Tool - version 4

User Manual

Copyright © 2016 Nokia Networks. All rights reserved.

Table of contents

1 Introduction

1.1 General

The Workflow Tool (WFT) is a responsive web application which is used to support the management of Central Builds, SC builds and external deliveries for the Load & Build process.
It acts as an umbrella for SCM and I&V tasks like the complete Load Planning and Load Building process.

The Workflow Tool models the Load & Build process as a finite state machine, where only predefined state changes are possible. The tool converts the incoming planning information into tasks, deadlines and production triggers.

It assists creating Load Plans, generating Freezing Reminders and Release Notes (Output format: HTML and XML). Information has to be entered only once and is re-used to create needed output.
The WFT contains a list of features, change notes and faults which can be assigned to one or more builds.

Due to multisite development and distributed I&V across disparate geographical sites, complex product structures and the acute need for improved handover handling, the Workflow Tool was introduced.
Without the Workflow Tool all of the coordinating work has to be done via email and excel sheets. To avoid this error-prone manual procedure the Workflow Tool uses a predefined chain that unifies this process.

Additional the functionality of the complete WFT is also available via API which makes it possible to interact via CLI, e.g. doing a build handover via an XML Release Note.

The tool is independent from any product / project version.

The technical realization of the Workflow Tool is based on the web frameworks "Ruby on Rails" and "Bootstrap". For hosting the production environment of the Workflow Tool several Linux servers with Nginx and MySQL are used.

1.2 Prerequisites

To use the WFT the user has to own a Nokia account and must have access to the Nokia-intranet, at least via VPN.

1.3 What is "Ruby on Rails"

Ruby on Rails is a web framework enabling you to create web applications very quickly. Ruby itself is not a new programming language, it first appeared in 1995. It was quite unknown, until Rails was offered as an add-on to Ruby. From this moment "Ruby on Rails" became widely accepted and subject to a certain amount of hype.
Rails is what is known as "a gem" (or add-on) for Ruby and consists of several parts: ActionMailer, ActionPack, ActiveSupport, ActiveRessource, ActiveRecord and Rails 3. All these parts together comprise the Rails framework.

1.4 Supported browsers

WFT uses the framework Bootstrap for developing the page layout. Therefore it supports the latest versions of the following browsers and platforms: Chrome, Firefox, Internet Explorer (on Windows IE 8 - 11), Opera and Safari. More specific support information is provided here: http://getbootstrap.com/getting-started/#support-browsers. Though they are not officially supported WFT should look and behave well enough in Firefox for Linux, Internet Explorer 7 and Chromium and Chrome for Linux.

1.5 Used Technology

- Framework: Ruby on Rails (Finite State Machine Plug in)
  - Ruby version: 2.2.4
  - RoR (Ruby on Rails) version: 4.2.5.1
- Framework Bootstrap (see http://getbootstrap.com).
  - Bootstrap version: 4.0.0-alpha.2
- Database: MySQL
- Web server
  - Nginx to deliver static content
  - Nginx Module "Phusion Passenger" with "Ruby Enterprise Edition"

1.6 Access

To access the Workflow Tool a web browser has to be used.

The WFT is accessible via the following address: https:/wft.int.net.nokia.com. Collaborators have to use https://esworkflow.emea.nsn-net.net (this link can also be used by internal staff; e.g. in case of some server problems).

Users can login with their Nokia-Intra Account through WAM (Nokia Web Access Management) Login Page. It is possible to use "Single Sign on" (SSO).

If the user enters the first time the WFT or if SSO is not active he will be automatically be redirected to the WAM Login Page.

To enable "Silent SSO" one has to modify some settings described here:
https://intranet.nokia.com/global/MyServices/IT/ITforIT/WAM/Pages/WAMServiceFAQ.aspx#HowtoenableSilentSSOinFireFox
(Look for the caption "How to enable Silent SSO in Firefox?". On the same page one also gets more information about SSO and Silent SSO)

1.7 WFT Server Setup

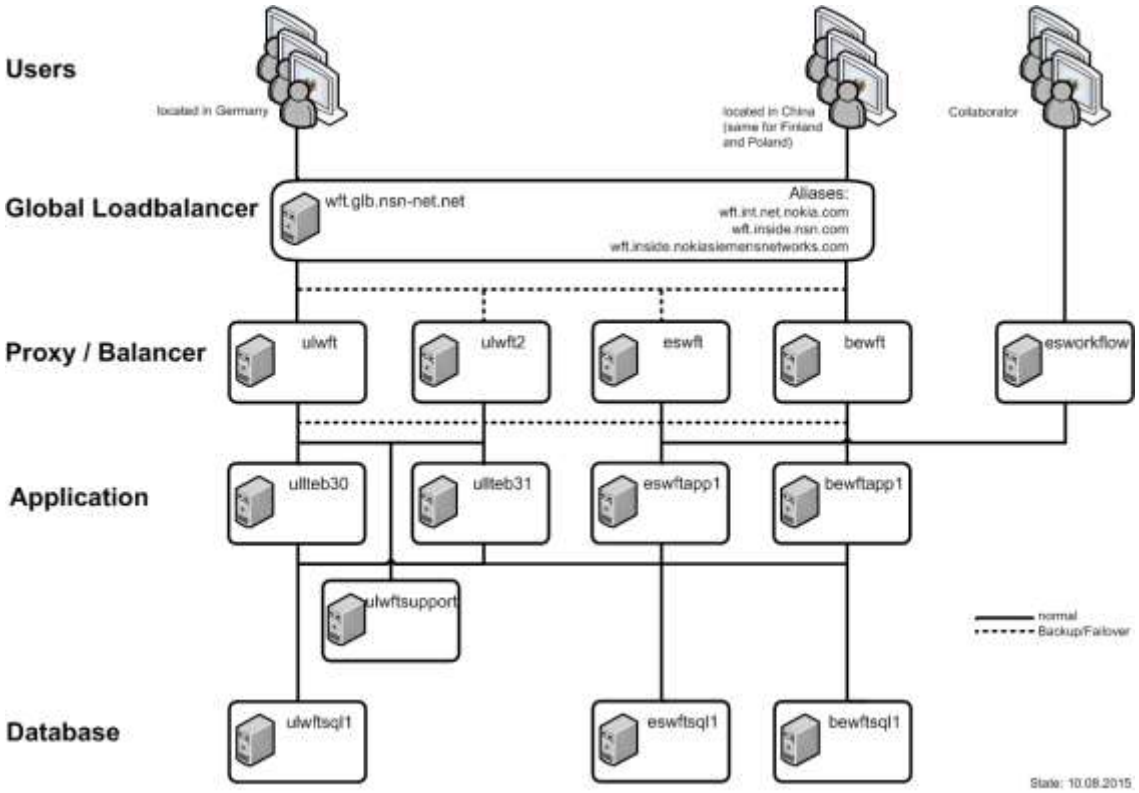


Figure 1: WFT Server Setup

1.8 Contacts

For questions regarding the Workflow Tool please contact:

WFT Support Mail: support.wft@nokia.com

Andreas Hutstein
Mobil: +49 160 97872153 Mail: andreas.hutstein@nokia.com

Dawid Goslawski
Mobil: +48 668182004 Mail: dawid.goslawski@nokia.com

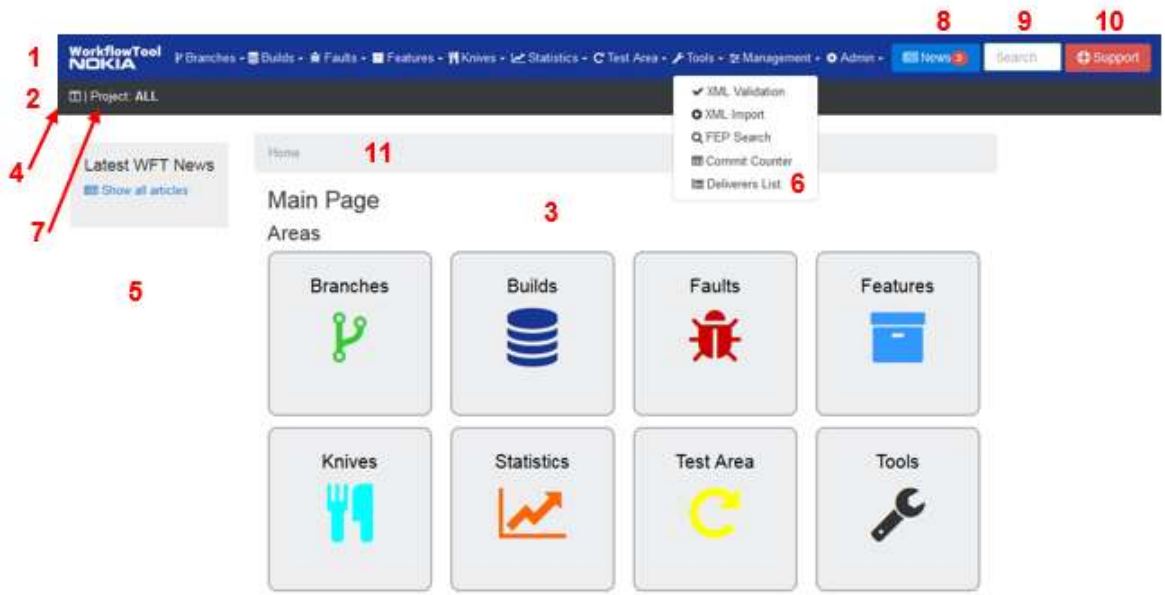
Mariusz Kowalski
Mobil: +48 784054316 Mail: mariusz.1.kowalski@nokia.com

Miao Yun Guo (Klarke)
Mobil: +86 14727136521 Mail: miaoyun-klarke.guo@nokia.com

Wolfgang Elster
Mobil: +49 170 2256731 Mail: wolfgang.elster@nokia.com

2 Overview

2.1 Main Page



The menu bar is divided into a main navigation (1) and a sub-navigation (2). The sub-navigation is used e.g. within the detailed build view (see chap. 3.2.2).

Hint: a standard user (no permission for Quicktest; see also chap. 2.2) does not see the navigation and the box for area "Test Area".
Also "Management" and "Admin" will be offered for users only with appropriate permissions.

The first sub-area of each area (for details see chap. 2.2) can be selected using the relevant box within the main area (3).
All other sub-areas have to be selected via the pulldown menus (6).

The sidebar (5) (see also chap. 2.6) can be switched on and off using the button (4) within the sub navigation.

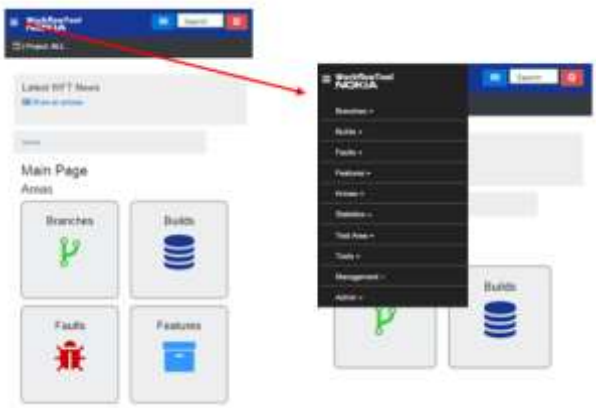
Any news about WFT can be found using the News button (8). A counter within that button indicates the user's unread news.

Details about the search functionality (9) (see chap. 2.7) as also the Support menu (10) (see chap. 2.3) are explained later on.

For navigation the bread crumbs (11) also can be used.

For project selection the link "Project" (7) has to be used (see also chap. 2.4).

On a mobile or in case the window is too small the menu selection looks like that:



2.2 Areas

The Workflow Tool (WFT) is in the first place a web application. The GUI is divided into several areas (see below) to fulfill the given tasks.
Additionally the complete functionality offered by GUI is also available via API which makes it possible to interact via CLI (e.g. via scripts).
Some of these areas are hidden for a standard user and need permission. These permissions have to be requested by means of "Permission Request" (see button "Support") and are manually granted by the WFT administration.

Area	**Sub-area**	**Function**
Branches		- overview of branches
Builds		- perform and release loads - download loads - load planning: view and schedule loads
Faults		- search for a specific pronto - pronto report - pronto diff
Features		- search for a specific feature
Knives		- viewing knives - requesting a knife
Statistics		- graphical view of several statistics
Test Area [1)]		- release to I&V and send release mail for - Quicktest Phase 1 and - Quicktest Phase 2
Tools		- XML validation / import - Deliverers list - FEP search
Management [1)]	Branches Branch Types Releases System Release Knife Settings Knife Servers Permissions RN-Templates	- user management - permission handling for Key-User

	Release Settings Build Configuration	- build conf. spec. configuration
Admin		for WFT Administration only

[1] permission needed

2.3 Support menu

Using the button "Support" the user
- has access to various documentations like this user manual, the functional specification and the WFT internal Wiki (user information, managed by WFT administration).
- is able to subscribe for RSS Feeds
- is able to write a new support request or watch existing support requests.
- can request permissions for permission groups.
- has the opportunity to edit his profile.
- gets information about the current WFT version

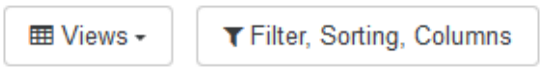
2.4 Project selection

Using the link "Project" (see (7) within chap. 2.1) the user selects the project he wants to work on.
The project selection works per browser tab, i.e. different browser tabs can be used for different projects.

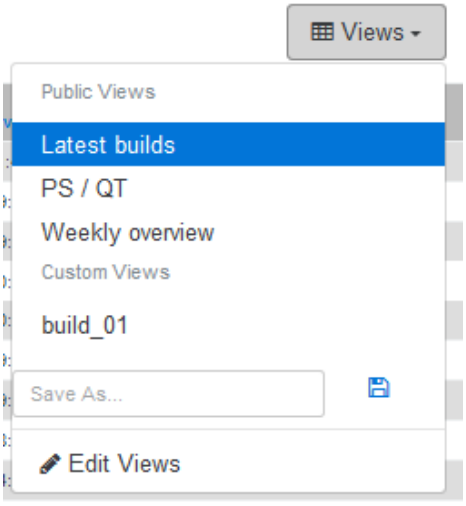
The selected project is also visible within the URL, which can be used to set needed bookmarks. E.g. (project WMP selected): https://wft.int.net.nokia.com:8081/WMP/builds/ chronological.

2.5 Customized views

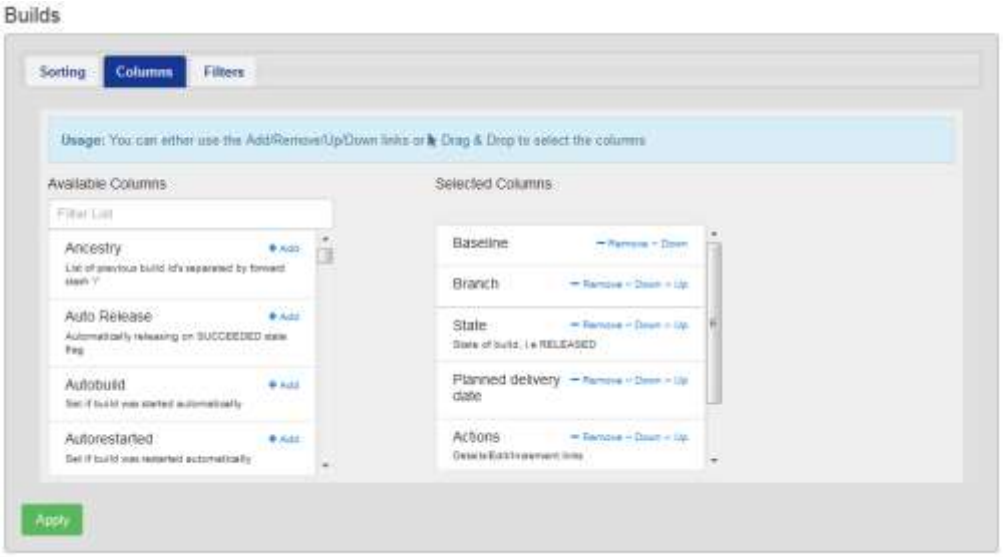
Where applicable the user is able to create customized views using the buttons "Views" and "Filter, Sorting, Columns".



Clicking on "Views" opens a menu showing:
- all available views of the used area (predefined views as also user defined ones).
- offering the possibility to save a currently defined view.
- possibility to edit available user defined views.

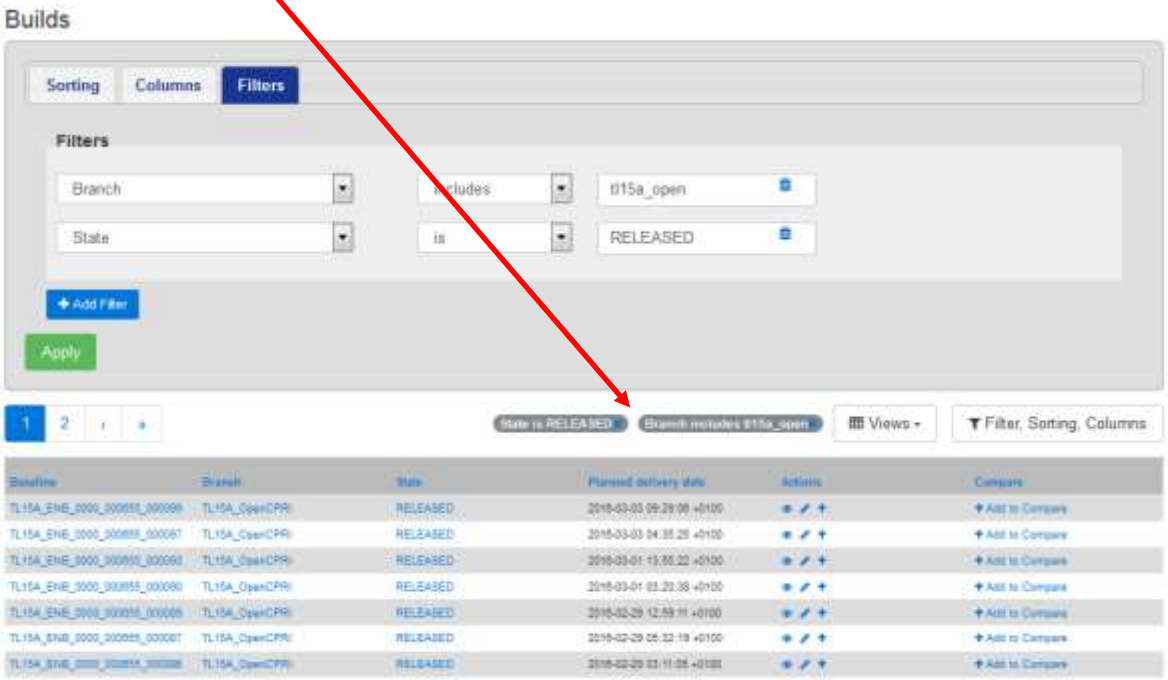


Selecting the button "Filter, Sorting, Columns" the user has various possibilities to set the current view.



- using the tab "Sorting" the user is able to set the sorting and count of the view
- within the tab "Columns" all available DB columns for the view can be selected and ordered (see figure above).
- Within tab "Filters" filters for also all available DB columns can be defined.

After applying the chosen setting the concerned view is shown. This view can be saved (as described above).
The chosen filtering is displayed on top of the page.



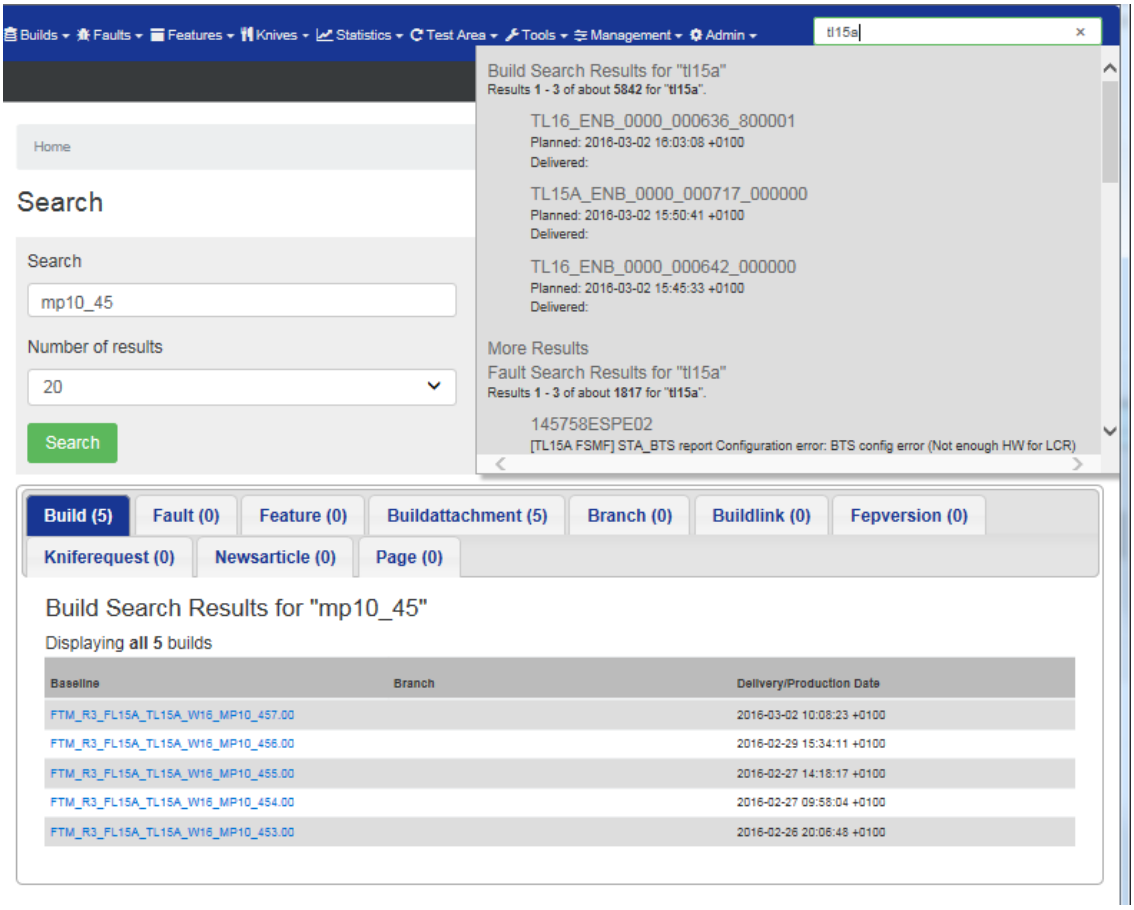
2.6 Sidebar

Several pages within WFT have on the left-hand side a sidebar which offers additional information and/or links to other pages.
For hide / unhide of this sidebar, e.g. because of space requirements (4) (see chap. 2.1) has to be used.

2.7 Search functionality

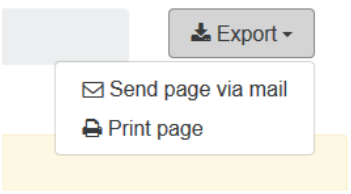
With the extensive search functionality (see (9) out of chap. 2.1) an user can find an expression he is looking for separated within several categories (e.g. Build, Fault, Feature, Knife Requests, Wiki, etc.)
Be aware, that the search algorithm is not case sensitive.

As soon as he has entered the expression a pre-selection is showing the latest 3 hits of each category.
Pressing "Enter" will show all hits and the user can walk through the displayed tabs to have a look to them.



2.8 Publishing a page

Every page of the WFT can be published by printing it or sending the content to the user. Where applicable an XML file can be generated.



3 Using the different Areas

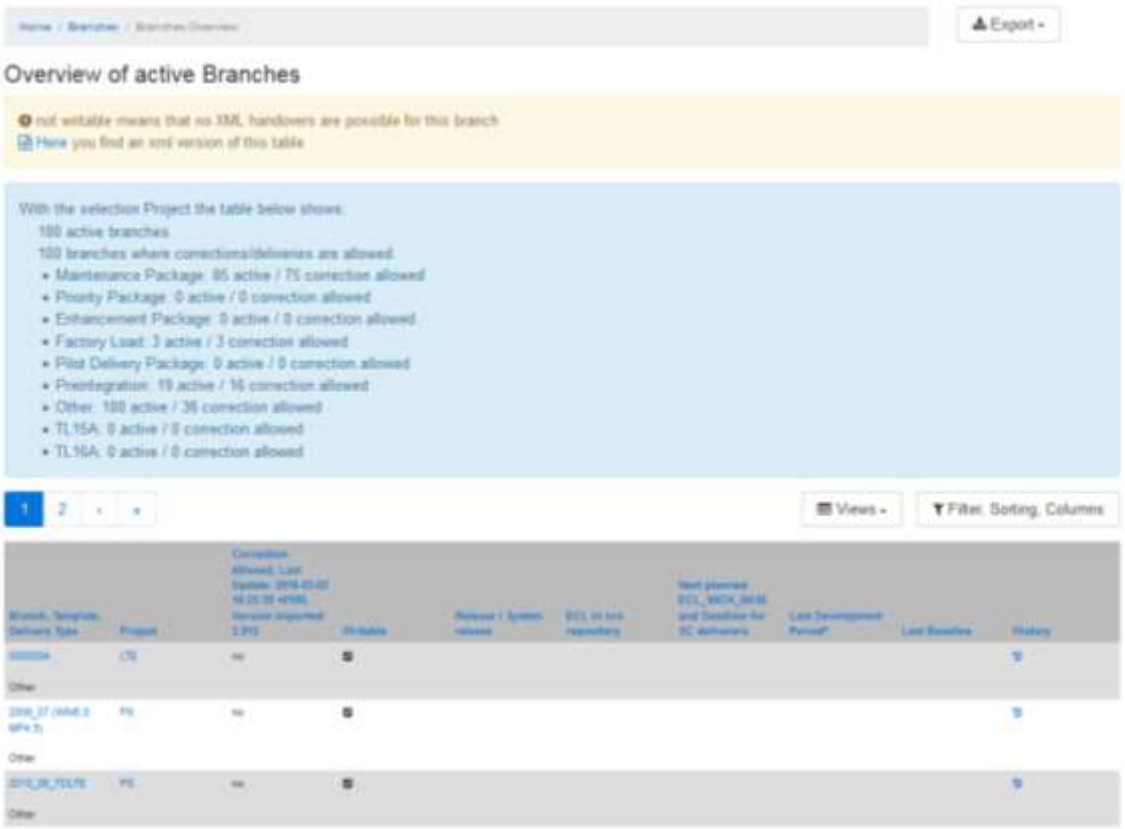
The WFT is structured into several areas, which sometimes are also divided into several sub-areas (see also chap. 2.2).

3.1 Branches

Within area Branches the use gets an overview of all builds using various views.

3.1.1 Overview of active branches

Within that view the use gets an overview about all active branches. The branches can be filtered and sorted.



3.1.2 All branches

Here the user gets an overview of all branches within the selected project.
Per branch the last 5 eNB builds are shown.

On request (see link "Show more") the user can select a list with all eNB builds of the required branch.

3.1.3 Trunk Baselines

This view shows the last 5 eNB builds of each Trunk branch.

On request (see link "Show more") the user can select a list with all eNB builds of the required Trunk branch.

3.2 Builds

This area is the central and most used area of the WFT. Via that page all builds managed by the WFT can be accessed.
Within the Load & Build process there exists three different categories of builds. Each build category has different requirements and characteristics which have been modeled (see chap. 5.6) within the Workflow Tool.

Central builds are the highest level of builds, e.g. LTE / eNB builds. They are compiled out of the SC builds and External builds. This load will be delivered to Quicktest (QT).

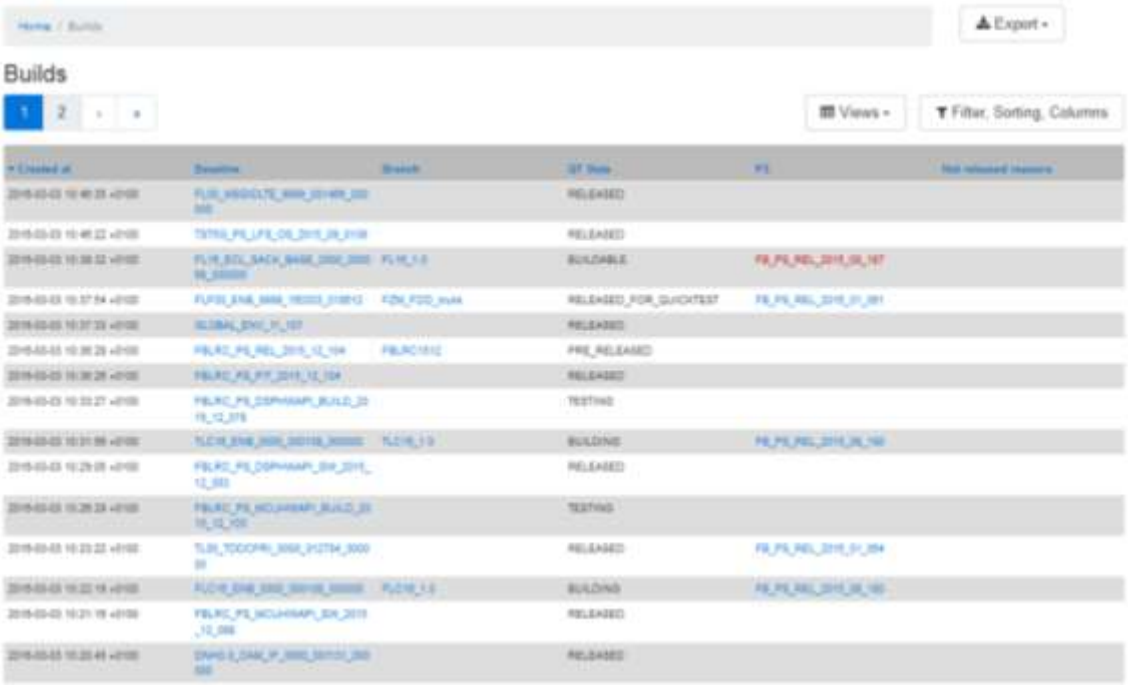
SC builds are those deliveries (components), which are delivered to a central build (as also to other SC builds, or External deliveries) and are planned and controlled via WFT.

External deliveries are software packages needed for builds which are not planned in detail and are not controlled via WFT. They are produced outside WFT and provide information about the build via API and XML Release Note to WFT. These are SCs like BTSOM, MAC, TUP, PHY_RX, Linux OS, parts of Platform, etc.
External deliveries deliver to Central Builds, to SC builds, or to other External deliveries.

General remark: the expressions "builds" and "loads" are used synonymously throughout this document.

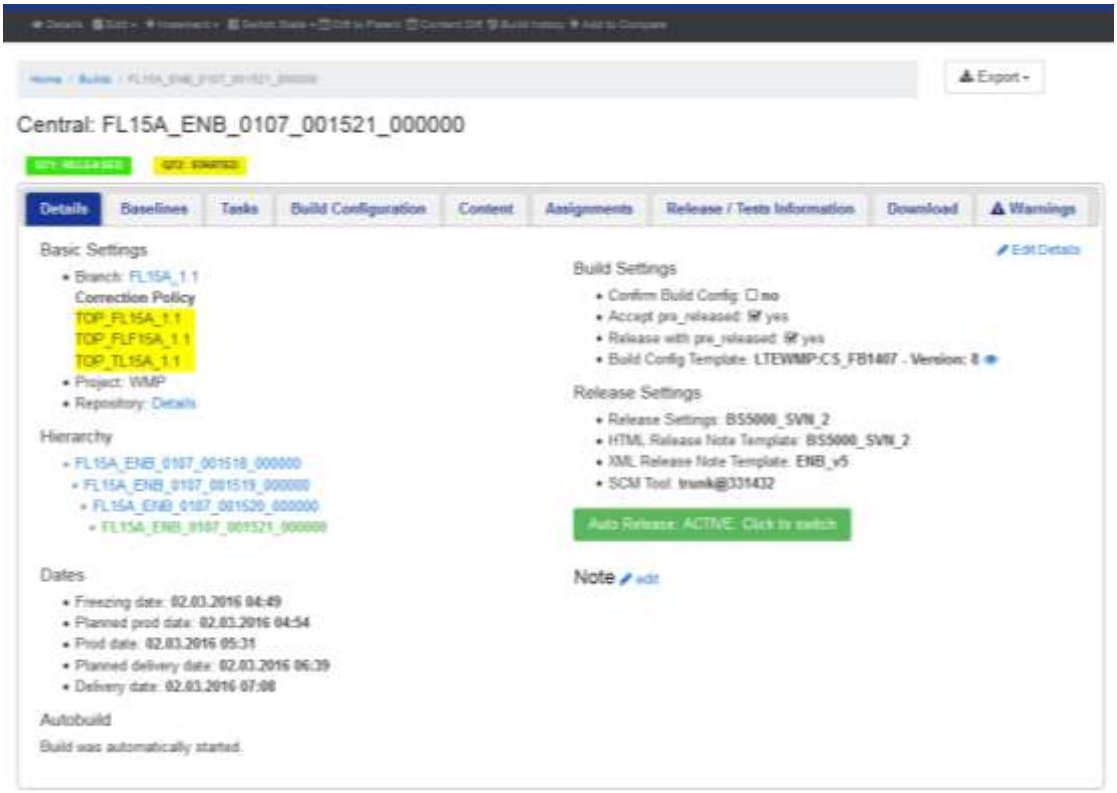
On the detailed view of a single build (see chap. 3.2.2) the user is able to manage, edit and display various information about this build.
The state of a build can be switched depending on the category and according the state machine (see chap. 5.6) using menu "Switch State".

3.2.1 Latest Builds



Using this view the user get an overview of all available builds depending on the selected view (for details see chap. 2.5).

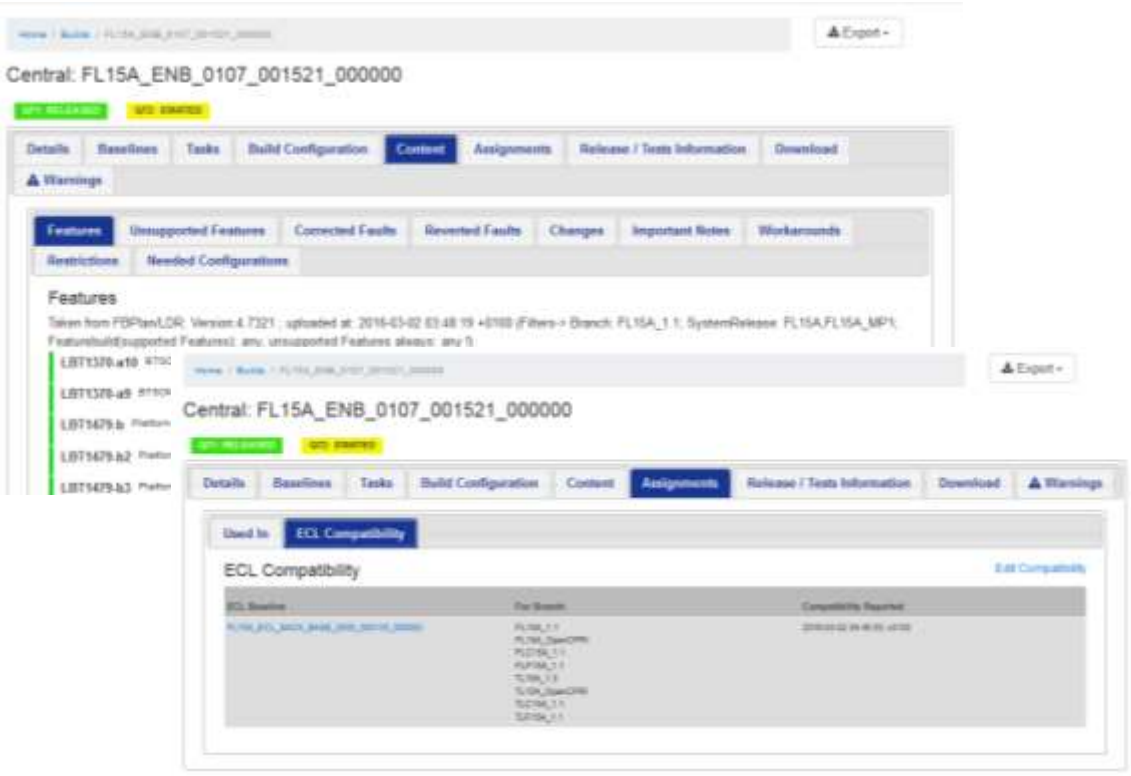
3.2.2 Detailed view of a build

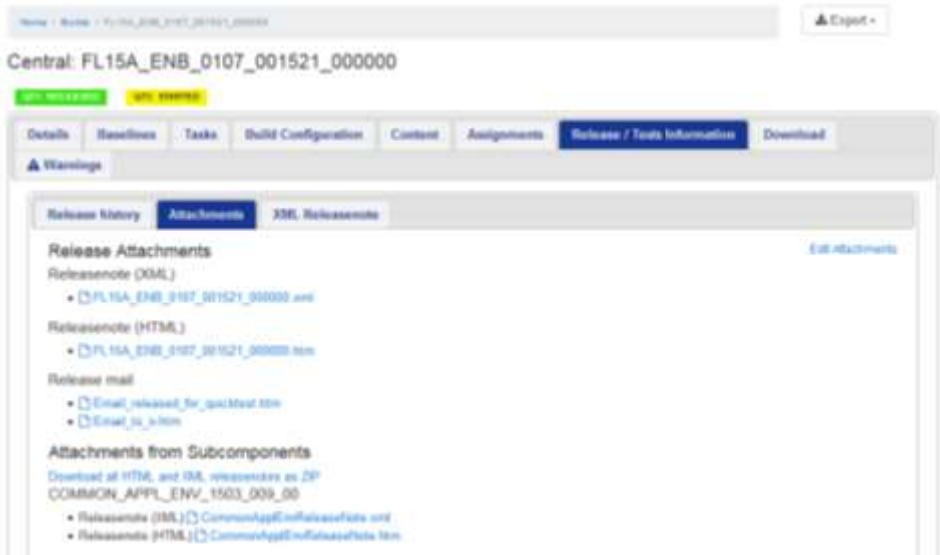


The ability to edit the content of a build depends on the owned permissions.
Also the visibility of the links ("Details", "Edit", "Increment", "Switch State", etc.) within the sub-navigation line is depending on the associated permissions.

For each build the following data can be downloaded:
- release notes of the contained components (tab "Release / Test Information ").
- the build results, e.g. the complete load, MAP & OUT files, etc. (tabs "Download" / "Build results") and
  the used tools (for testing and/or installation) for that build (tabs "Download" / "Tools")

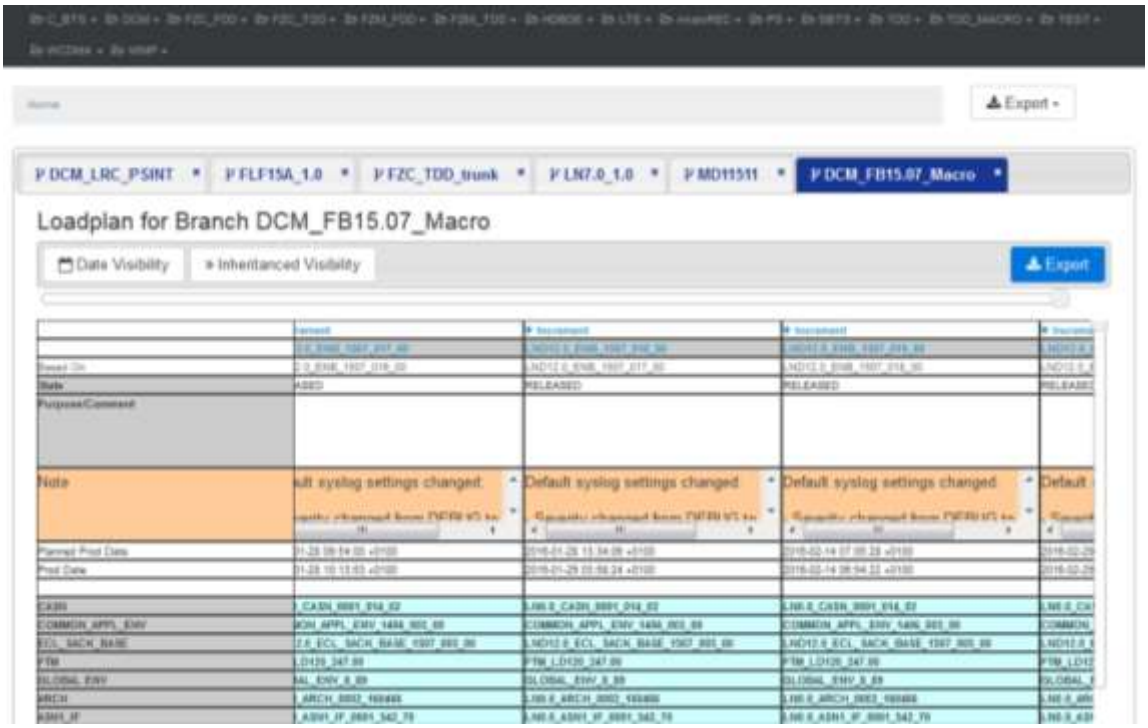
The tabs "Content" "Assignments", "Release / Test Information" and "Warnings" contain sub-tabs as some examples below are showing.





3.2.3 Load Planning

This area is restricted to the load coordination.
People of the load coordination are writing to the planning sheet and after fixing a new build configuration the build team gets informed by email to start the build. All other users may read the planning sheet.



3.3 Faults

Within that area the user:

- Is able to search for specific faults and gets detailed information about it.
- gets on a daily basis detailed information about the corrected faults within each build and about possible differences to the Pronto tool (link "Pronto Report").
- gets all those faults listed which have been corrected between two builds (link "Pronto Diff"). The faults corrected within the "from" build are excluded (precon-ditions for "Pronto Diff" are that the baselines are within the same branch and have an uninterrupted hierarchy chain; or they have a common predecessor).
- can view the list of the "Pronto Group Mapping".

3.4 Features

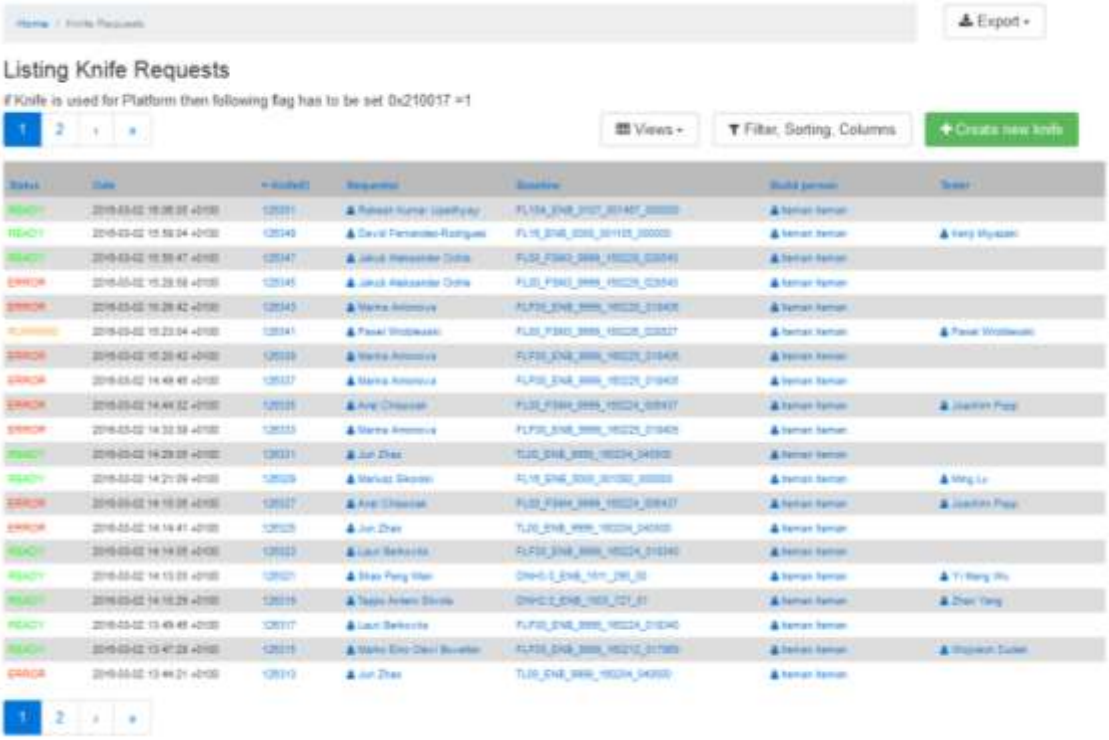
Within this area the user can search for features.

3.5 Knives

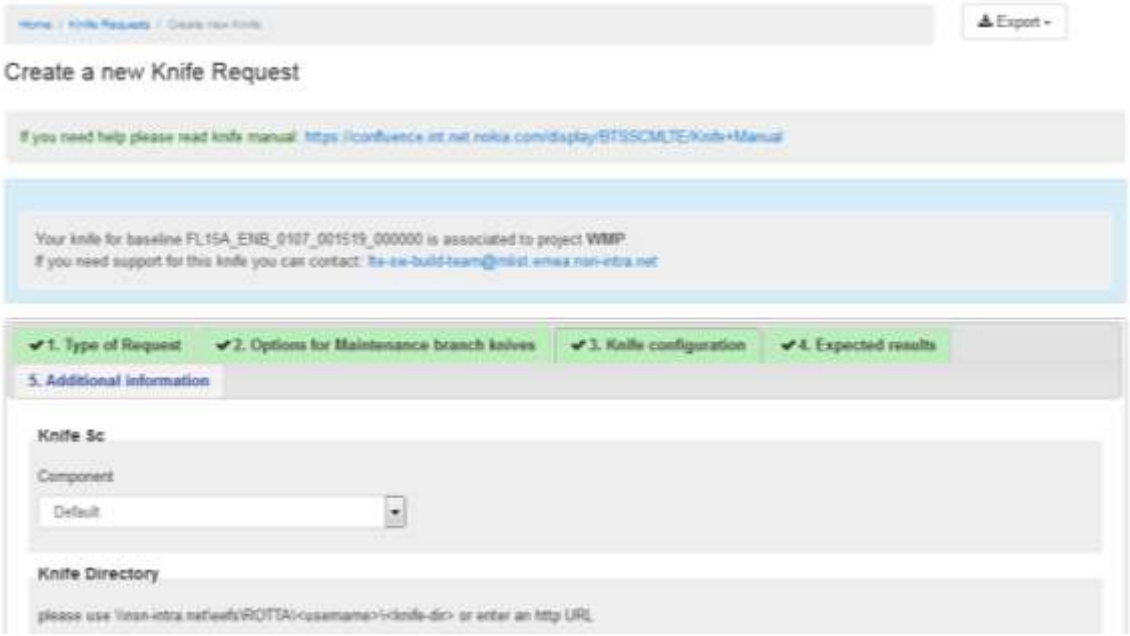
Within this area every user can request a knife (see chap. 3.5.2), finds a list of the past knife requests and can search for knife requests.
Additionally the user can select the knives done at a specific date, or to those done by himself and he can have a look to several statistics of the past knife requests.
The detailed proceeding of a knife request is explained within the WFT internal Wiki.

Hint: please be aware that WFT is the front end for requesting a knife. Knife building itself will be done by build-script of build teams.
In case you have any problems (e.g. about the status of the knife build) reply to the mail sent to you and the concerned SW Build Team will help.

3.5.1 Knife overview

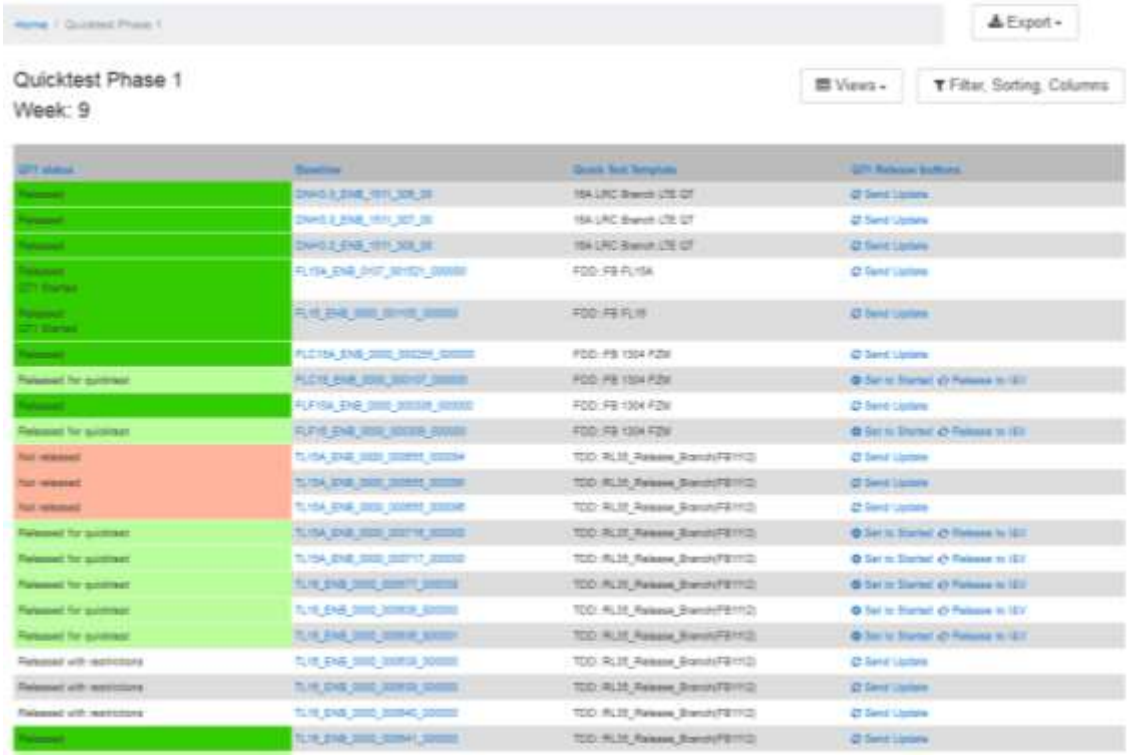


3.5.2 Knife request creation



3.6 Test Area

The "Test Area" contains the areas for Quicktest Phase 1 and Quicktest Phase 2.



For using this area the relevant permission has to be granted by the WFT administration.

3.6.1 Quicktest Phase 1

This page is used by the Quicktest Team to release a specific eNB build to I&V after testing it within "Phase 1". It shows all builds according the selected view.
The default view shows all relevant information necessary for the Quicktest Team.
The Quicktest Team can define release note templates based on their own needs.
Using the link "Manage Templates" the members of the Quicktest Team are able to create and/or to edit the quicktest release note templates.

A build can have one of the following states:

- Error: Build is in state error
- Succeeded: Build succeeded
- Released for quicktest Released for quicktest
- Released: Released to I&V
- Not Released: Not Released
- Released with restrictions: Released to I&V with Restrictions

To release a specific eNB load to I&V or to ask for an update of the release the user (Quicktest Team) can use the relevant link ("Release to I&V" or "Send Update") displayed within the column "Release".
This link opens a formula where Quicktest Team has to enter the test results and can manage the Quicktest release mail.

With the link "Manage Templates" the member of the quick-test team is able to edit the template of the Quicktest release note and manage the addresses for the emails to be sent.

3.6.2 Quicktest Phase 2

This page is also used by the Quicktest Team and shows all loads performed for Quicktest Phase 2 according the selected view. The default view shows all relevant information necessary for the Quicktest Team.

The functionality is similar to that of the area "Quicktest Phase 1" (see chap. 3.6.1). Additionally the Phase 2 Result of the concerned builds is displayed.

3.7 Statistics

Within that area one find several statistic views concerning the eNB Central Build results. One can select the production type, the increment / feature build and the time frame.
The user can choose between a total view of all productions (pie), a view on a weekly basis, he can get information about the Central Build times, can have an overview about the "days count since last working build" and the faults grouped by "group in charge".

3.8 Tools

Within that area the user is able to:
- use the "XML Validation Check", which validates a selected XML file.
- do an XML Import.
- find the eNB baseline version string belonging to the entered FEP-no using the (DCM specific) "FEP Search".

3.9 Management

Nominated Key-Users have access to that area and depending on their permission they are able to:
- create/edit branches.
- manage the permission group they are responsible for (e.g. they can grant permissions, e.g. adding or removing of users from that group and they can create an access-key to be used for the whole group)
- create and change build config. templates
- create and change HTML release note templates
- manage various knife settings / knife servers settings.

4 Functionality

4.1 Build Checks

The building of a load is supervised by several checks.
For details please refer to chap. 3.4.12 of /1/.

4.2 Tasks / Task Types

Tasks are script calls or other executions which are performed in the scope of a build. Using a form these tasks can be created, can be edited and can be deleted from every build. Details like actions or the result of the added task can be displayed. Tasks are based on defined task types using different options.

For details please refer to chap. 3.4.6.3 of /1/.

5 Interfaces

5.1 Overview

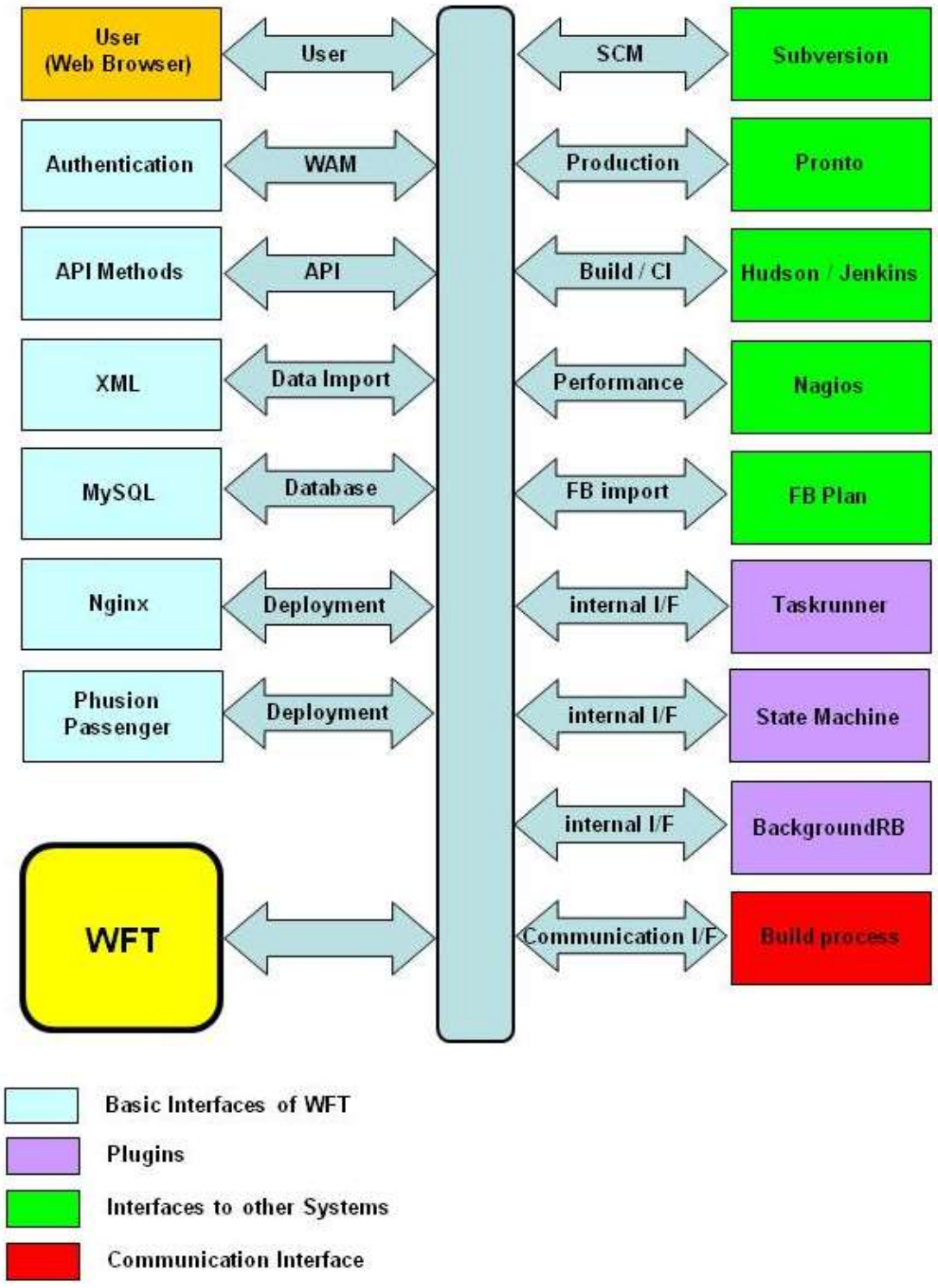


Figure 2: Interfaces of the Workflow Tool

5.2 Basic Interfaces

5.2.1 User (Web browser)

The user has to use a web browser to interact with the Workflow Tool. For details about supported browsers please see chap. 1.4.

5.2.2 Authentication

Users can login with their Nokia-Intra Account through WAM (Nokia Web Access Management) Login Page, see also chap. 1.6.

5.2.3 Available API methods

Needs to be reworked.

The complete functionality offered by GUI is also accessible via API which makes it possible to interact with WFT also via CLI, resp. via scripts.
A description of the possible API methods (CLI commands) is offered on the applicable WFT pages to the users via a link (button) "API DESCRIPTION".

Up to now not for all possible pages such an "API DESCRIPTION" is existing, but will be added step by step by WFT administration.
In the following an overview of the currently existing implementation is described.

Method	**Description**	**Para-meters**	**Arguments**
xml	Creates a baseline from uploaded XML file in predefined format	none (Baseline is detec-ted from XML)	File sent with post request as variable file
xml_ validate	Validates the uploaded XML file against xsd schema	none	File sent with post request as variable file
upload	Uploads an attachment to a baseline	Baseline	File sent with post request as variable file. Optional you can provide type of uploaded file with parameter "type=", where type is one of rn, rn_xml, needed_config or other.
link	Adds a link to a baseline	Baseline	name=IDA2 link=http://...
repository	Changes repository URL for this build	Baseline	repository=
build_ content	Shows baseline infor-mation in XML format	Baseline	

For additional information refer to internal WFT Wiki.

5.2.4 MySQL

MySQL is a relational database management system. The MySQL database contains all data (production as also process) from the Workflow Tool.

MySQL was chosen because of its propagation for web applications and the good accessibility to the Nginx web server.

5.2.5 Nginx

Nginx is the frontend web server and passes all dynamic requests through to the passenger module.

5.2.6 Phusion Passenger

Phusion Passenger is a module (plug-in) for the Nginx web server for running of Rails web applications. It is available as a Gem package and allows easy deployment of Ruby on Rails applications. There is no Ruby (on Rails)-specific server configuration required.
Using the Phusion Passenger the Nginx should never crash, even in case of crashing Rails applications.
The Phusion Passenger distributes the different requests of the users to the various processes of the WFT application.

5.3 Plug-ins

5.3.1 Task Runner

The Task Runner is a ruby script which is responsible for executing all tasks in the context of a specific UNIX user. The ruby script communicates with the Workflow Tool across the database.

The status of a task is responsible for it's execution.

NEW	Every task starts with this state. The Task Runner searches for tasks in state NEW and executes depending on the task type some kind of code/script. As soon as the execution succeeds the task will switch to PREPARED, otherwise to BLOCKED.
PREPARING	This state shows that the prepare step for that task is being currently executed.
PREPARED	Tasks in state PREPARED can be started by the user or automatically as element of a chain.
STARTED	Tasks which were started are in state STARTED. This is the indicator for the Task Runner to execute the script for this task.
RUNNING	In case of the script is executed the Task Runner switches the task to state RUNNING.
SUCCESS or ERROR	Depending on the result of the execution the task is switched to SUCCESS or ERROR.
BLOCKED	If there are problems with the task preparation it will switch to state BLOCKED.
TAINTED	If the task configuration has to be rebuilt this is done by switching to state TAINTED. The Task Runner will switch the task to state NEW afterwards the cleanup has finished.

For more details refer to /1/.

5.3.2 State Machine

Needs to be reworked.

During a build's lifecycle the software build moves through several predefined states. There are predefined state changes that will be triggered by a user, a cron-job or an external tool via a curl call. To track a builds state history, each state change is stored in the database.

For more detail about the states / state changes of the build process refer to chap. 5.6.

5.3.3 Background Worker

The Background Worker is a self written ruby script and performs all actions which need to be done periodically. It is responsible for the following tasks:

- State Switch PLANNED->ANNOUNCED
- State Switch ANNOUNCED->BUILDABLE
- State Switch BUILDABLE->FROZEN
- Task State Calls
- Autorelease

For more details refer to /1/.

5.4 Interfaces to external SW products

5.4.1 Subversion

The data of the SCs as also the build results are stored within the VCS Subversion. Fetching this data out of Subversion or importing data into it is supported by WFT (see chap. 3.2).

5.4.2 Pronto

The WFT polls the error tracking tool Pronto with a cron-job to fetch and store all contained faults. The faults can be assigned to one or more builds.

5.4.3 Nagios

Nagios is an open source software application for network monitoring. It watches services and alerts users when things go wrong and again when they get better. Nagios monitors the system performance and load of the Workflow Tool.

5.4.4 Jenkins / Hudson

Jenkins / Hudson is an extendable, web based system for CI (Continuous Integration). It supports different build tools and different VCSs.
The WFT starts all Central Builds via Hudson whereas Jenkins is used within the development.

5.5 Communication Interfaces to Build Process

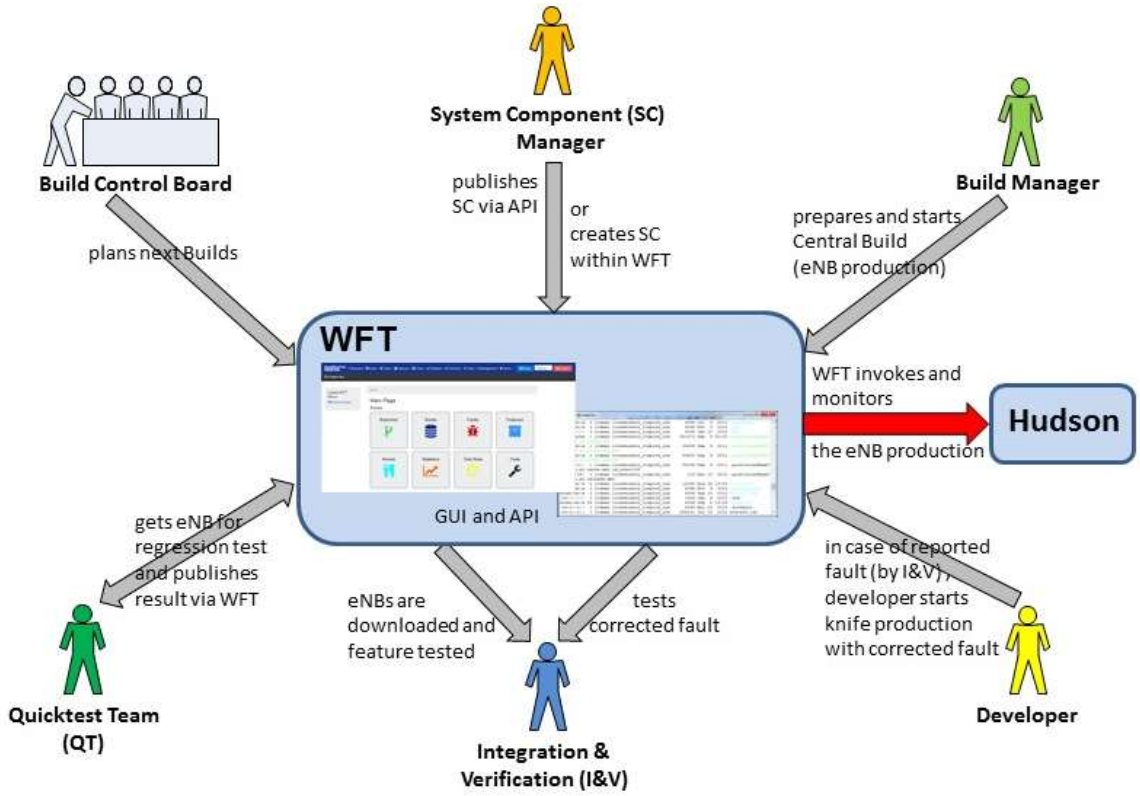


Figure 3: Communication Interfaces of the Workflow Tool to Build Process

5.6 Build Process states

Needs to be reworked.

During a build's lifecycle the software build moves through several predefined states. The state changes will be triggered by a user, a cron-job or an external tool via a curl call (see also chap. 5.3.2).

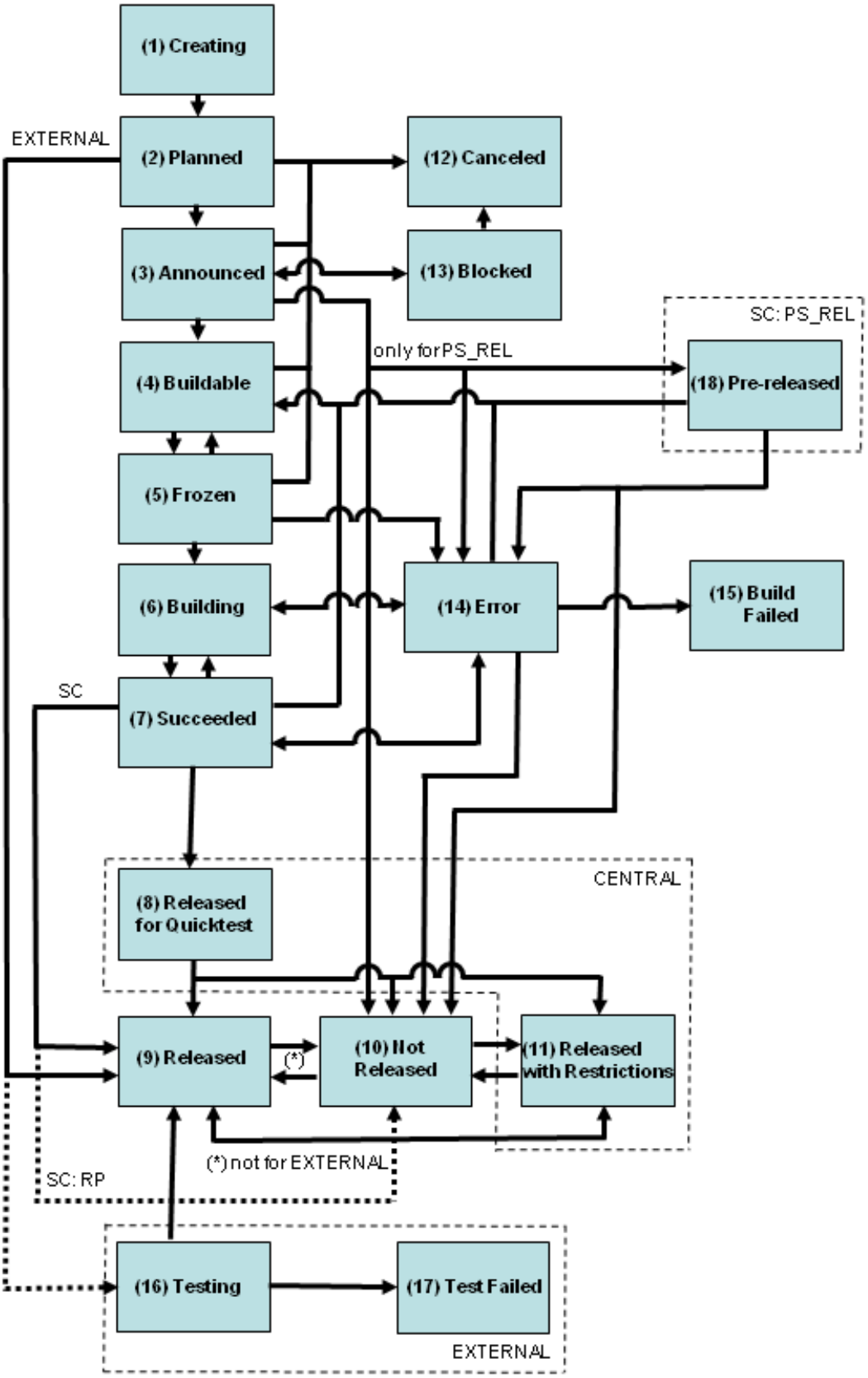


Figure 4: Build Process states

Following build states are implemented:

(1) Creating	This build state is used for planning new builds, that shouldn't be visible to other users (not used at the moment).
(2) Planned	Builds that are officially planned and visible to other users. A build in this state can't no more be changed.
(3) Announced	As soon as the build has entered this state, the freezing reminder could be sent and handovers are possible. (the possibility to send a freezing reminder is up to now not used)
(4) Buildable	When all handovers are performed the build will switch to this state and the build responsible will be informed. (No notification will be sent.)
(5) Frozen	In case of the flag "Confirm Build Config" is not set the build will automatically switch from buildable to frozen once the freezing date is reached. Otherwise this has to be done by the build responsible. As soon as the build is frozen the configuration files for the build configuration are provided and the build configuration is invoked. In case of the build responsible notifies that something was forgotten it is possible to unfreeze the build and to return to state "Buildable".
(6) Building	The transition into that state can be started automatically or manually and will be set by some task which is configured to switch the state of the build.
(7) Succeeded	Changing to this state can only be carried out by some task which is configured to switch the build to this state. The build responsible gets notified about this event. In case of the build responsible notifies that something was forgotten it is possible to return to state "Buildable". It is also possible to restart the build (switch to state "Building").
(8) Released for Quicktest	Quicktest Team gets informed that the CENTRAL build is released for Quicktest Phase 1. A release note will be sent to Quicktest Team.
(9) Released	An EXTERNAL build will be automatically set to Released after entering into WFT. An SC build reaches this state after it has been successfully finished. A release note will be sent. A CENTRAL build has successfully been tested and released by Quicktest Team. A release note will be sent.
(10) Not Released	The CENTRAL build failed during Quicktest Phase 1 or Phase 2. A release note will be sent. An SC build can be switched to that state.
(11) Released with Restrictions	The CENTRAL build is partly successful tested and is released by Quicktest Team. A release note will be sent.
(12) Canceled	Builds can only be deleted if they are in the state "Creating", later they can only be canceled. A build which has not yet reached the state "Building" (all builds coming from states (2), (3), (4), (5) and (13)) can be canceled.

	A notification will be sent.
(13) Blocked	A build gets automatically into this state if the freezing date is reached and handovers are missing. The build responsible is notified and he can decide to postpone or to re-announce this build.
(14) Error	If the build configuration or Hudson reports an error, the build will change to this state and the build responsible is notified. The build responsible can a) restart the build (go to state "Building"; e.g. the VCS was not reachable) b) change the configuration and restart the build (go to state "Buildable"). c) set the build to "Succeeded" (in case of e.g. a build script has stopped).
(15) Build Failed	If a build ran on an error, the build responsible can mark this build as failed. A notification will be sent.
(16) Testing	Testing of a system component sub-component. Depending on the test result state can be changed to "Released" or "Test Failed".
(17) Test Failed	If testing of a system component sub-component failed this component can be marked as failed.
(18) Pre-released	Interim state for handling the "Fast-Track" mechanism of PS_REL.

6 Terms and Abbreviations

6.1 Abbreviations

API	Application Programming Interface
BTS	Base Transceiver Station
DCM	DoCoMo
ECL	Environment Control List
eNB	enhanced Node B
FB	Feature Build
FDD	Frequency Division Duplex
I&V	Integration and Verification
LinSEE	Linux-SEE (SW Engineering Environment)
LTE	Long Term Evolution
SC	System Component
SCM	SW Configuration Management
SSO	Single Sign on
SVN	Subversion
tbd	to be defined
TDD	Time Division Duplex
WAM	Nokia Web Access Management
WFT	Work Flow Tool
WMP	World Market Product

6.2 Definitions

branch	A branch describes the various development lines of a SW, e.g. for different products (e.g. WMP or DCM) and/or for different versions, i.e. feature builds (FB).
component	See deliverer.
deliverer	Set of baselines, sometimes also called "component".
fallback	A minor SW version of a sub-baseline has to be used within the super ordinate build (due to various reasons). In case of a fallback this has to be regarded for e.g. no longer corrected faults, etc.
knife	A knife is a kind of a patch, it is an eNB load containing the corrected SW parts. It is built on base of an existing eNB load including the corrections, which have to be tested.
pronto	Synonym for a fault stored within the Pronto Tool.

6.3 List of Figures

7 References

/1/ *WFT Functional Specification, see IMS,*
https://sharenet-ims.int.net.nokia.com/Open/D431870360

8 Revision history

Version	Date	Modified by	Main changes
1.0	19.09.12	W. Elster	first issue
2.0	18.10.13	W. Elster	update (e.g. description for checks and task types added; new area "management")
2.01	12.11.13	W. Elster	update (some minor additions)
2.02	30.01.14	W. Elster	update
3.0	30.06.14	W. Elster	update for WFT 3.0
3.01	17.07.14	W. Elster	update (some minor additions / adaptations)
3.02	02.09.14	W. Elster	update (new contact and due to #11793 and #11847)
4.0	07.03.16	W. Elster	update according WFT 4.0; contacts updated

End of Document