WP3 meets WP2

eXtreme DataCloud

Data Management for extreme scale computing

eXtreme DataCloud All Hands Meeting, Hamburg 11 - 13 Sept. 2018

Doina Cristina Duma, Alessandro Costantini (INFN)
Pablo Orviz (IFCA/CSIC)
Rachid Lemrani (IN2P3)
Baptiste Greniter (EGI Foundation)

European Commission

eXtreme DataCloud is co-funded by the Horizon2020 Framework Program – Grant Agreement 777367

Goals and Objectives

- Goals
  - of the **project**:
    - *“releasing services able to provide many of ... lacking functionalities at the infrastructure level, with an approach general enough to fit the requirements of many user communities. The **final goal** is to significantly lower the access barriers to distributed computing, **release more usable, more reliable, functionality-rich and still scalable** data management services to **cope with the most demanding scientific use cases**”*
  - of the **session**:
    - prepare for first release - XDC-1

- Objectives of the session
  - present state-of-affairs
    - what is available from WP3
  - towards XDC-1
    - explain & understand **“How to … use JIRA for …”**

WP3 vs WP2

WP3 deployed a set of services aimed at assisting and supporting the activities carried out project partners

- ***Integration infrastructure***
  - resources to assess the software interoperability and quality assurance
  - software validation environment for the execution of the automated tests performed through the continuous deployment pipeline
- ***Pilot infrastructure***
  - stable production-like environment, where the user communities can deploy and test their use-cases under realistic conditions
  - enables the exploitation of the project services

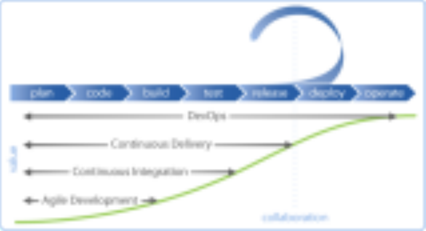
WP3 vs WP2

WP3 deployed a set of services aimed at assisting and supporting the activities carried out by project partners

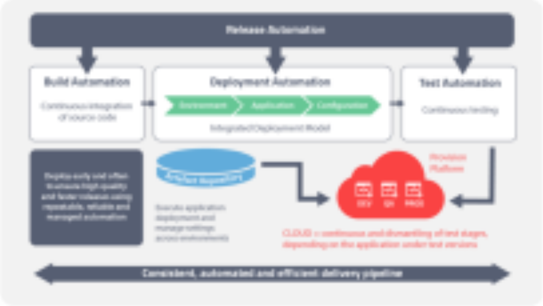
- Collaborative Tools and Repositories
  - a selection of open source tools have been put in place in order to cover
    - all the stages of the DevOps and Quality Assurance process
    - the communication among users, developers and boards
    - user requirements needs and their evolution
    - SW development tracking issues
  - Helpdesk service (GGUS)
    - To support users and developers asking for specific resources
    - To support resource providers contributing to the pilot testbed

WP3 - State-of-affairs

WP3



"WP3 will provide software lifecycle management services together with pilot e-Infrastructures"


Release Automation
Build Automation
Continuous integration of source code
Deployment Automation
Environment
Application
Configuration
Integrated Deployment Model
Test Automation
Continuous testing
Artifact repository
Execute application deployment and manage settings across environments
Provision Platform
DEV
QA
PROD
CLOUD = continuous and dismantling of test stages, depending on the application under test versions
Consistent, automated and efficient delivery pipeline

❖ ***T3.1 - Software Lifecycle Management***

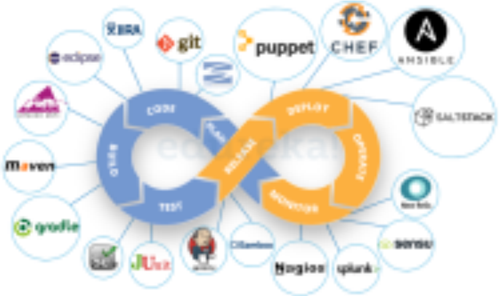
➢ Quality Assurance

➢ Software Lifecycle Management

❖ ***T3.2 - Pilot e-Infrastructures and support services***

➢ Integration & Testing (aka Preview pilot) infrastructures

➢ Setup and Maintenance of Tools and Repositories



❖ ***T3.3 - Exploitation***

➢ Staged-Rollout & SPB

➢ EOSC, WLCG, EGI, EUDAT, other EINFRA-12

WP3

Initial Plan - D3.1

eXtreme-DataCloud

SOFTWARE LIFECYCLE MANAGEMENT AND PILOT TESTBEDS - INITIAL PLAN

DELIVERABLE: D3.1

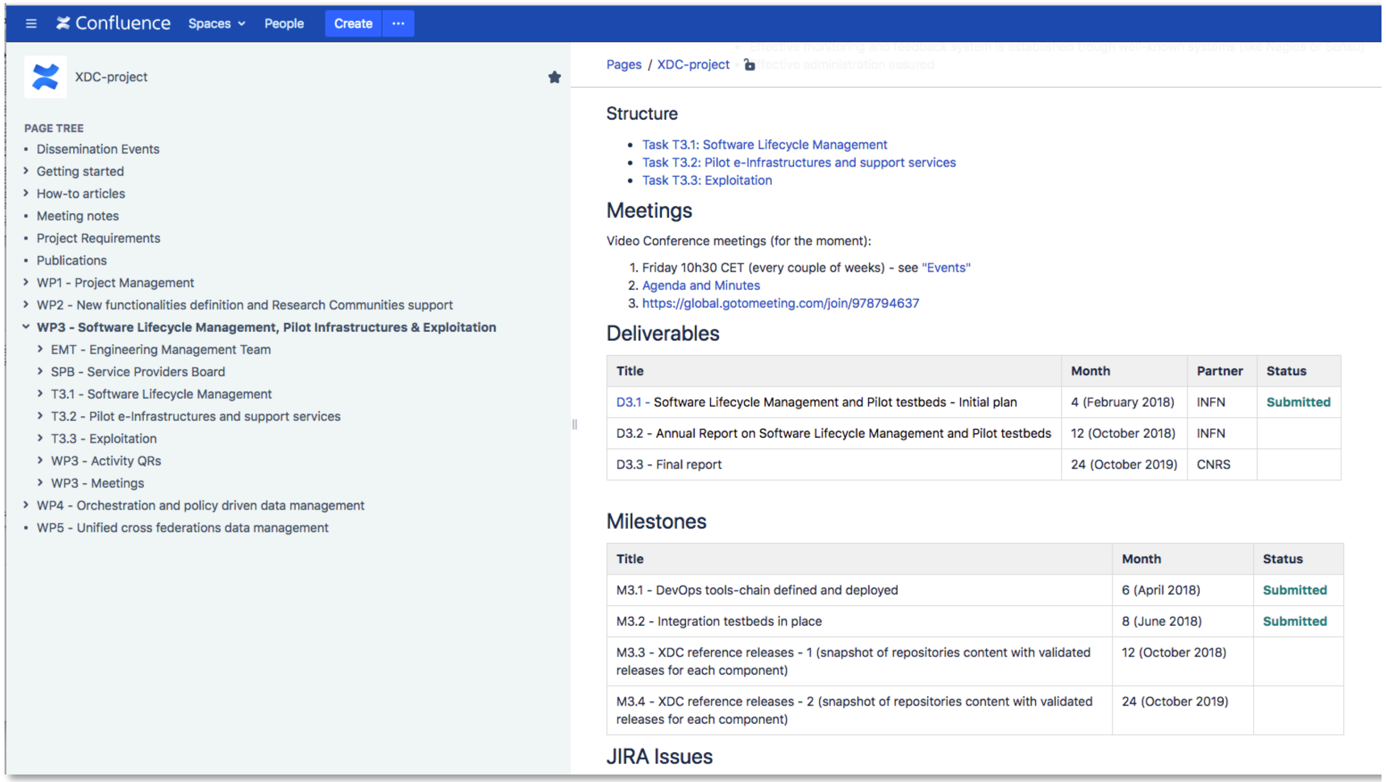
Document identifier:	XDC-D3.1
Date:	02/03/2018
Activity:	WP3-SA1
Lead partner:	INFN
Status:	Final
Dissemination level:	PUBLIC
Permalink:	

Abstract

This document describes the Software Lifecycle Management process - the Quality Assurance, Maintenance, Release and Support processes, the roles and responsibilities, and the definition of the architectures of the Pilot infrastructures according to use cases to be supported. It includes the selection of tools to support the Task 3.1 and the architecture of the Continuous Integration/Delivery infrastructure supporting the DevOps model. It describes the initial plan for liaison with external infrastructures and resource providers.

	Name	Partner/Activity	Date
From	D. C. Duma	INFN/WP3	02/03/2018
Reviewed by	Daniele Cesini, Giacinto Donvito	INFN/PC INFN/TC	19/06/2018
Approved by	PEB		21/06/2018

Confluence - WP3



XDC-project

PAGE TREE

- Dissemination Events
- Getting started
- How-to articles
- Meeting notes
- Project Requirements
- Publications
- WP1 - Project Management
- WP2 - New functionalities definition and Research Communities support
- **WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation**
  - EMT - Engineering Management Team
  - SPB - Service Providers Board
  - T3.1 - Software Lifecycle Management
  - T3.2 - Pilot e-Infrastructures and support services
  - T3.3 - Exploitation
  - WP3 - Activity QRs
  - WP3 - Meetings
- WP4 - Orchestration and policy driven data management
- WP5 - Unified cross federations data management

Pages / XDC-project

Structure

- Task T3.1: Software Lifecycle Management
- Task T3.2: Pilot e-Infrastructures and support services
- Task T3.3: Exploitation

Meetings

Video Conference meetings (for the moment):

1. Friday 10h30 CET (every couple of weeks) - see "Events"
2. Agenda and Minutes
3. https://global.gotomeeting.com/join/978794637

Deliverables

Title	Month	Partner	Status
D3.1 - Software Lifecycle Management and Pilot testbeds - Initial plan	4 (February 2018)	INFN	Submitted
D3.2 - Annual Report on Software Lifecycle Management and Pilot testbeds	12 (October 2018)	INFN	
D3.3 - Final report	24 (October 2019)	CNRS	

Milestones

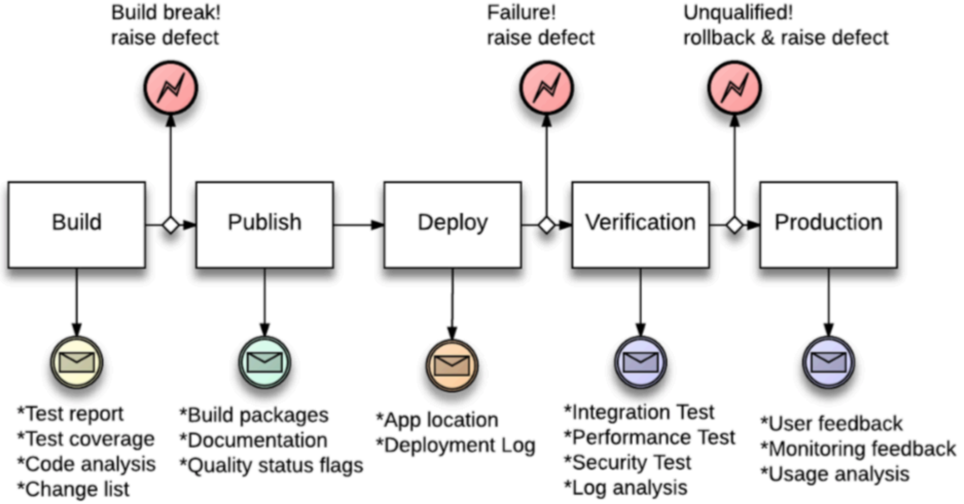
Title	Month	Status
M3.1 - DevOps tools-chain defined and deployed	6 (April 2018)	Submitted
M3.2 - Integration testbeds in place	8 (June 2018)	Submitted
M3.3 - XDC reference releases - 1 (snapshot of repositories content with validated releases for each component)	12 (October 2018)	
M3.4 - XDC reference releases - 2 (snapshot of repositories content with validated releases for each component)	24 (October 2019)	

JIRA Issues

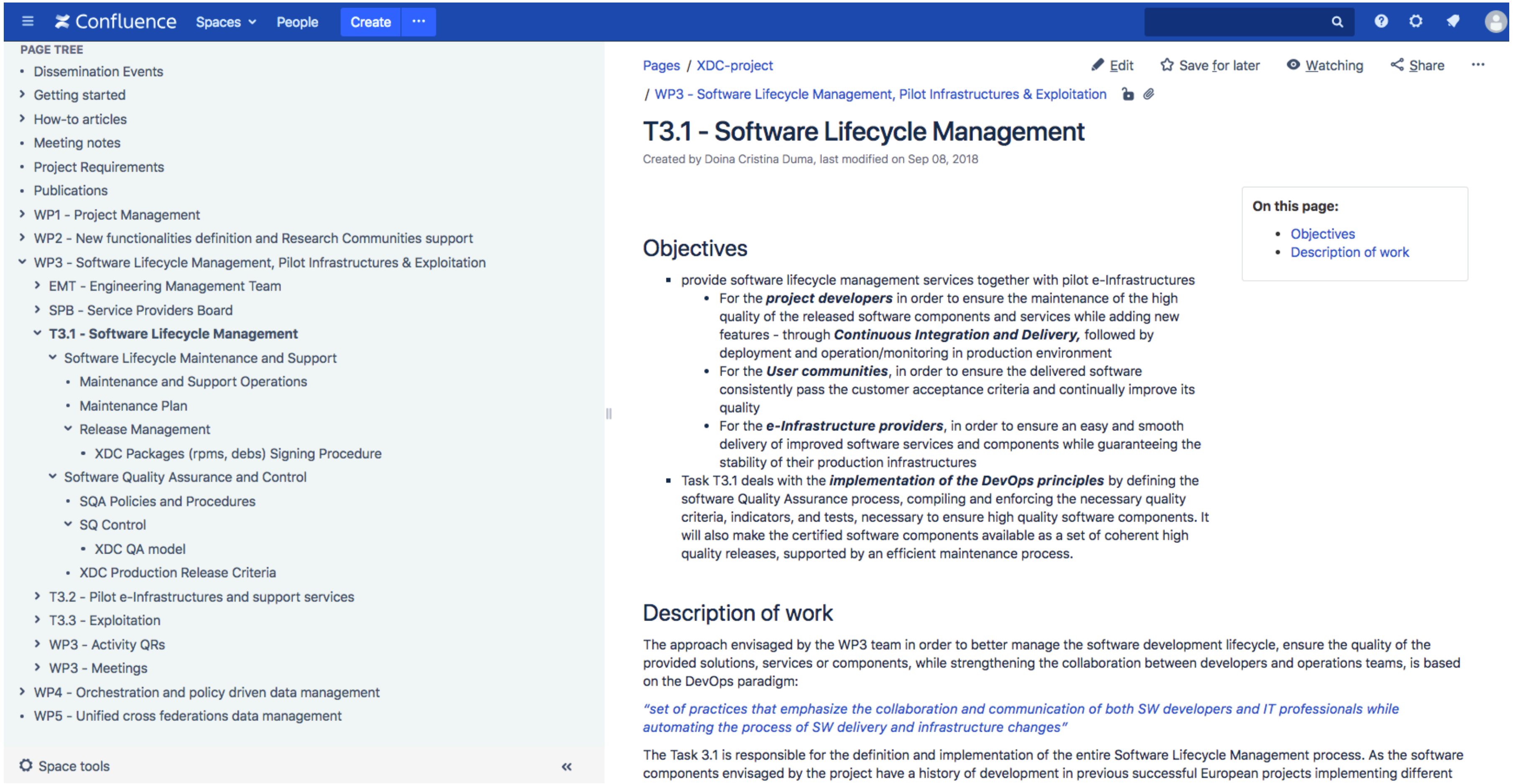
T3.1 – Software Lifecycle Management

T3.1

- *Definition and implementation of the entire Software Lifecycle Management process*
- ***Complement** the previous, individual, Continuous Development and Integration services with a **Continuous Testing, Deployment,** through **Configuration Management,** and **Monitoring***



T3.1



PAGE TREE

- Dissemination Events
- Getting started
- How-to articles
- Meeting notes
- Project Requirements
- Publications
- WP1 - Project Management
- WP2 - New functionalities definition and Research Communities support
- WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation
  - EMT - Engineering Management Team
  - SPB - Service Providers Board
  - **T3.1 - Software Lifecycle Management**
    - Software Lifecycle Maintenance and Support
      - Maintenance and Support Operations
      - Maintenance Plan
      - Release Management
        - XDC Packages (rpms, debs) Signing Procedure
    - Software Quality Assurance and Control
      - SQA Policies and Procedures
      - SQ Control
        - XDC QA model
      - XDC Production Release Criteria
  - T3.2 - Pilot e-Infrastructures and support services
  - T3.3 - Exploitation
  - WP3 - Activity QRs
  - WP3 - Meetings
- WP4 - Orchestration and policy driven data management
- WP5 - Unified cross federations data management

Space tools

Pages / XDC-project / WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation

T3.1 - Software Lifecycle Management

Created by Doina Cristina Duma, last modified on Sep 08, 2018

On this page:

- Objectives
- Description of work

Objectives

- provide software lifecycle management services together with pilot e-Infrastructures
  - For the ***project developers*** in order to ensure the maintenance of the high quality of the released software components and services while adding new features - through ***Continuous Integration and Delivery,*** followed by deployment and operation/monitoring in production environment
  - For the ***User communities***, in order to ensure the delivered software consistently pass the customer acceptance criteria and continually improve its quality
  - For the ***e-Infrastructure providers***, in order to ensure an easy and smooth delivery of improved software services and components while guaranteeing the stability of their production infrastructures
- Task T3.1 deals with the ***implementation of the DevOps principles*** by defining the software Quality Assurance process, compiling and enforcing the necessary quality criteria, indicators, and tests, necessary to ensure high quality software components. It will also make the certified software components available as a set of coherent high quality releases, supported by an efficient maintenance process.

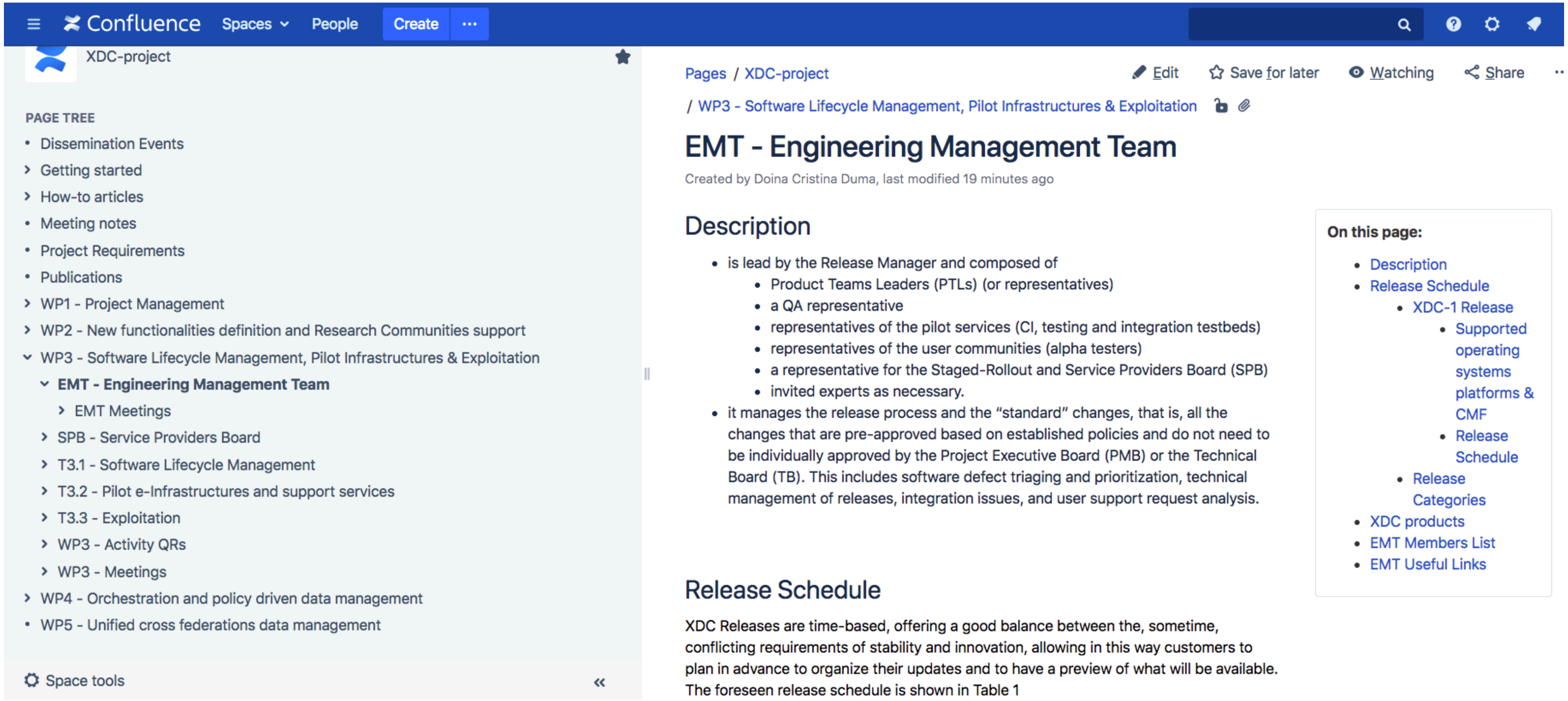
Description of work

The approach envisaged by the WP3 team in order to better manage the software development lifecycle, ensure the quality of the provided solutions, services or components, while strengthening the collaboration between developers and operations teams, is based on the DevOps paradigm:

"set of practices that emphasize the collaboration and communication of both SW developers and IT professionals while automating the process of SW delivery and infrastructure changes"

The Task 3.1 is responsible for the definition and implementation of the entire Software Lifecycle Management process. As the software components envisaged by the project have a history of development in previous successful European projects implementing different

T3.1 - EMT



XDC-project

PAGE TREE

- Dissemination Events
- Getting started
- How-to articles
- Meeting notes
- Project Requirements
- Publications
- WP1 - Project Management
- WP2 - New functionalities definition and Research Communities support
- WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation
  - EMT - Engineering Management Team
    - EMT Meetings
  - SPB - Service Providers Board
  - T3.1 - Software Lifecycle Management
  - T3.2 - Pilot e-Infrastructures and support services
  - T3.3 - Exploitation
  - WP3 - Activity QRs
  - WP3 - Meetings
- WP4 - Orchestration and policy driven data management
- WP5 - Unified cross federations data management

Pages / XDC-project / WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation

Edit Save for later Watching Share

EMT - Engineering Management Team

Created by Doina Cristina Duma, last modified 19 minutes ago

Description

- is lead by the Release Manager and composed of
  - Product Teams Leaders (PTLs) (or representatives)
  - a QA representative
  - representatives of the pilot services (CI, testing and integration testbeds)
  - representatives of the user communities (alpha testers)
  - a representative for the Staged-Rollout and Service Providers Board (SPB)
  - invited experts as necessary.
- it manages the release process and the “standard” changes, that is, all the changes that are pre-approved based on established policies and do not need to be individually approved by the Project Executive Board (PMB) or the Technical Board (TB). This includes software defect triaging and prioritization, technical management of releases, integration issues, and user support request analysis.

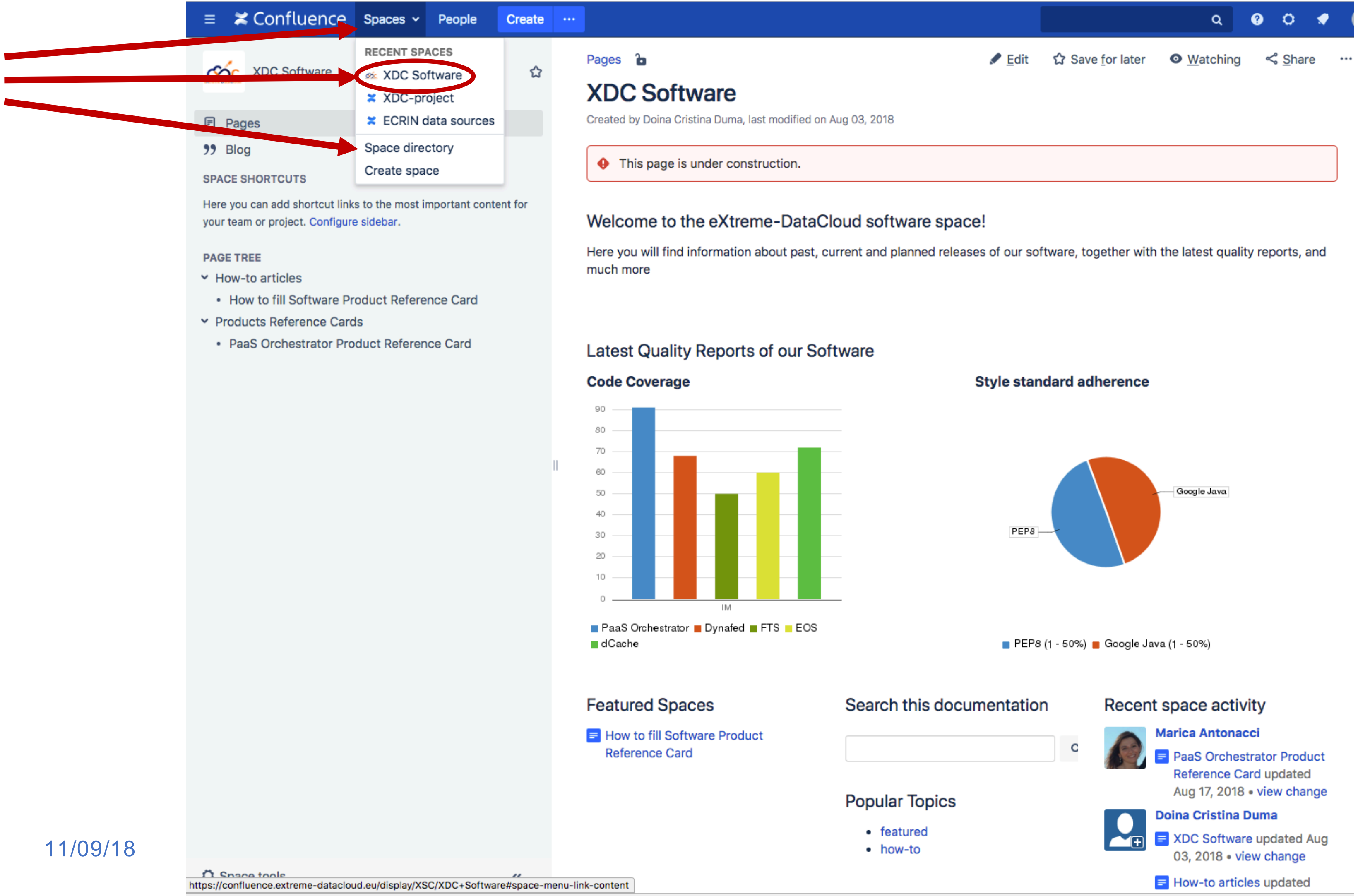
Release Schedule

XDC Releases are time-based, offering a good balance between the, sometime, conflicting requirements of stability and innovation, allowing in this way customers to plan in advance to organize their updates and to have a preview of what will be available. The foreseen release schedule is shown in Table 1

On this page:

- Description
- Release Schedule
  - XDC-1 Release
    - Supported operating systems platforms & CMF
    - Release Schedule
  - Release Categories
- XDC products
- EMT Members List
- EMT Useful Links

T3.1 – XDC Software Space



RECENT SPACES

- XDC Software
- XDC-project
- ECRIN data sources

Space directory

Create space

XDC Software

Pages

Blog

SPACE SHORTCUTS

Here you can add shortcut links to the most important content for your team or project. Configure sidebar.

PAGE TREE

- How-to articles
  - How to fill Software Product Reference Card
- Products Reference Cards
  - PaaS Orchestrator Product Reference Card

Space tools

Pages

XDC Software

Created by Doina Cristina Duma, last modified on Aug 03, 2018

This page is under construction.

Welcome to the eXtreme-DataCloud software space!

Here you will find information about past, current and planned releases of our software, together with the latest quality reports, and much more

Latest Quality Reports of our Software

Code Coverage

Style standard adherence

Featured Spaces

How to fill Software Product Reference Card

Search this documentation

Popular Topics

- featured
- how-to

Recent space activity

Marica Antonacci

PaaS Orchestrator Product Reference Card updated Aug 17, 2018 • view change

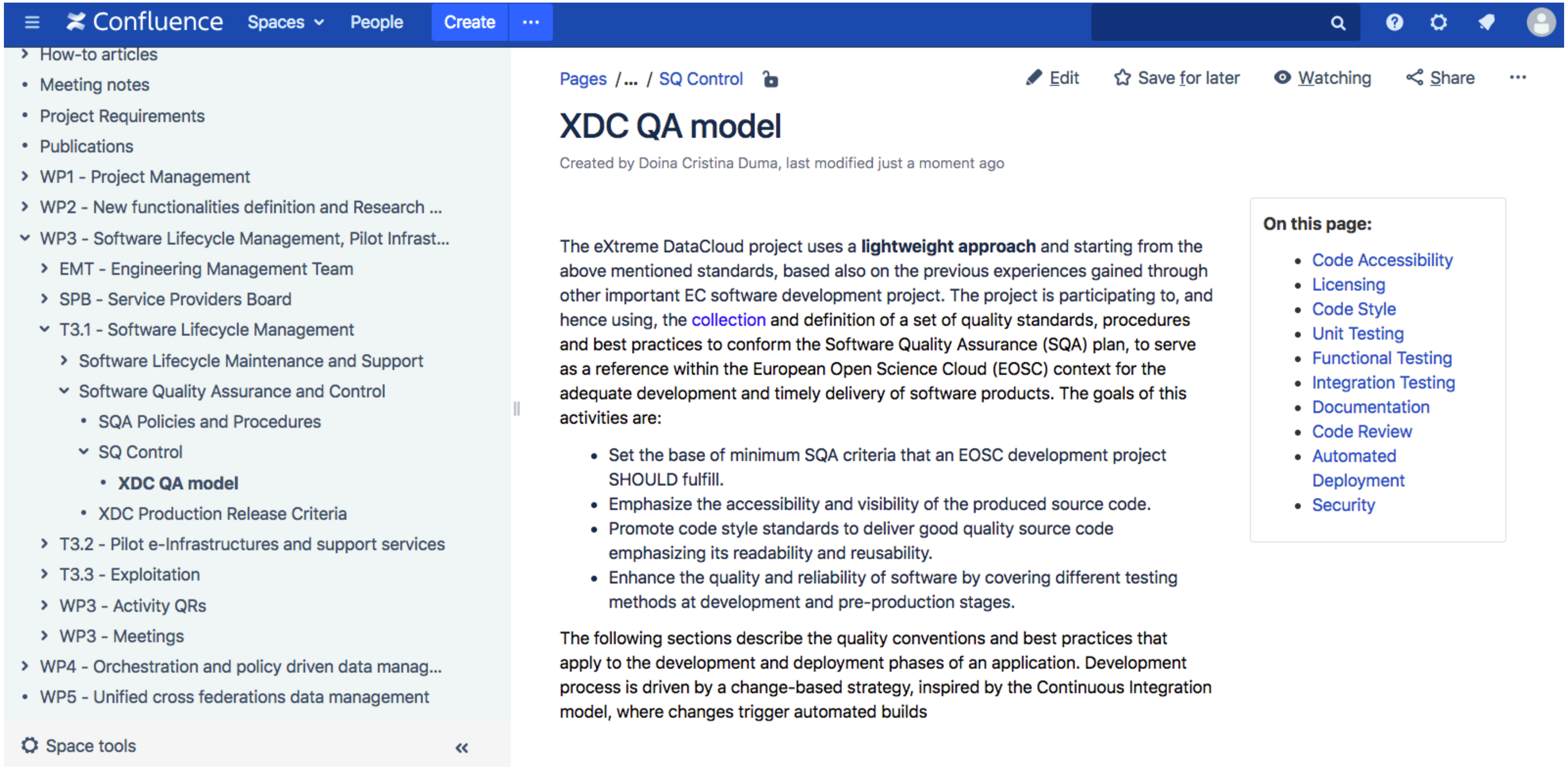
Doina Cristina Duma

XDC Software updated Aug 03, 2018 • view change

How-to articles updated

https://confluence.extreme-datacloud.eu/display/XSC/XDC+Software#space-menu-link-content

T3.1 – QA Model


≡ Confluence
Spaces
People
Create
···
How-to articles
Meeting notes
Project Requirements
Publications
WP1 - Project Management
WP2 - New functionalities definition and Research ...
WP3 - Software Lifecycle Management, Pilot Infrast...
EMT - Engineering Management Team
SPB - Service Providers Board
T3.1 - Software Lifecycle Management
Software Lifecycle Maintenance and Support
Software Quality Assurance and Control
SQA Policies and Procedures
SQ Control
XDC QA model
XDC Production Release Criteria
T3.2 - Pilot e-Infrastructures and support services
T3.3 - Exploitation
WP3 - Activity QRs
WP3 - Meetings
WP4 - Orchestration and policy driven data manag...
WP5 - Unified cross federations data management
Space tools
Pages / ... / SQ Control
Edit
Save for later
Watching
Share

XDC QA model

Created by Doina Cristina Duma, last modified just a moment ago

The eXtreme DataCloud project uses a **lightweight approach** and starting from the above mentioned standards, based also on the previous experiences gained through other important EC software development project. The project is participating to, and hence using, the collection and definition of a set of quality standards, procedures and best practices to conform the Software Quality Assurance (SQA) plan, to serve as a reference within the European Open Science Cloud (EOSC) context for the adequate development and timely delivery of software products. The goals of this activities are:

- Set the base of minimum SQA criteria that an EOSC development project SHOULD fulfill.
- Emphasize the accessibility and visibility of the produced source code.
- Promote code style standards to deliver good quality source code emphasizing its readability and reusability.
- Enhance the quality and reliability of software by covering different testing methods at development and pre-production stages.

The following sections describe the quality conventions and best practices that apply to the development and deployment phases of an application. Development process is driven by a change-based strategy, inspired by the Continuous Integration model, where changes trigger automated builds

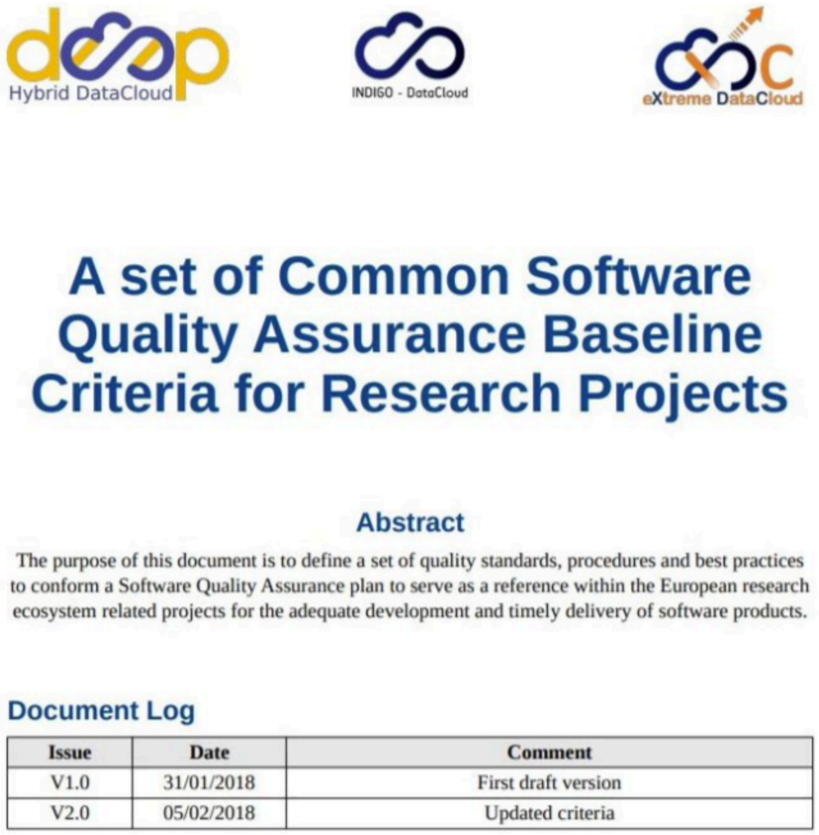
On this page:

- Code Accessibility
- Licensing
- Code Style
- Unit Testing
- Functional Testing
- Integration Testing
- Documentation
- Code Review
- Automated Deployment
- Security

T3.1 – SQA Baseline

http://digital.csic.es/handle/10261/160086

- What?
  - Set of conventions and recommendations for software development, aiming “to serve as a reference within the European research ecosystem related projects“
- Why?
  - Adequate development
  - Timely delivery
  - Reliable operation



A set of Common Software Quality Assurance Baseline Criteria for Research Projects

Abstract

The purpose of this document is to define a set of quality standards, procedures and best practices to conform a Software Quality Assurance plan to serve as a reference within the European research ecosystem related projects for the adequate development and timely delivery of software products.

Document Log

Issue	Date	Comment
V1.0	31/01/2018	First draft version
V2.0	05/02/2018	Updated criteria

T3.1 – Quality Criteria (1)

- Code accessibility
  - **open and publicly available**
- Licensing
  - Common open-source - Open Source Definition from the Open Source Initiative (OSI)
- Code-style
  - Comply with a de-facto **code style standard**
- Unit testing
  - **Automated**
  - Minimum acceptable code coverage **threshold**
- Functional testing
  - Check “full” set of functionalities
  - Automated, if possible
  - Regression testing

T3.1 – Quality Criteria (2)

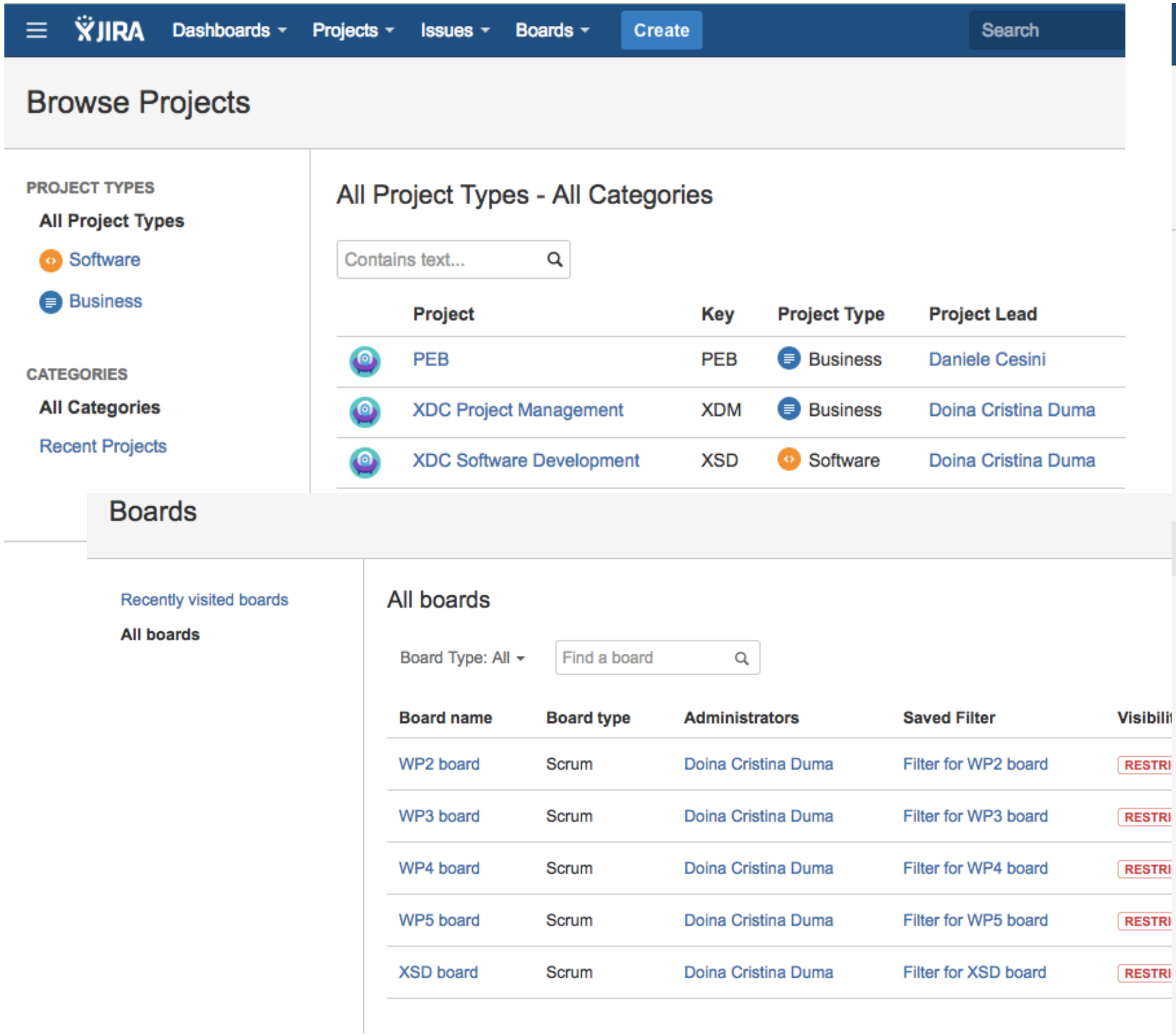
- Integration testing
  - On lack of automation, **pilot service infrastructures or local testbeds** MAY be used
- Documentation
  - Treated as **code**
  - **Target audience:** Developer, Deployment & Admin, User
- Code review
  - Agreed peer review
- Automated Deployment
  - leveraging software configuration management (SCM) tools
  - SCM treated as code
- Security
  - Compliance with Open Web Application Security Project (**OWASP**) secure coding guidelines
  - Automatic SAST, DAST
  - ESOC-Hub Information Security Management (T4.4)

T3.1 – Quality Criteria (2)

- Integration testing
  - On lack of automation, **pilot service infrastructures or local testbeds** MAY be used
- Documentation
  - Treated as **code**
  - **Target audience:** Developer, Deployment & Admin, User
- Code review
  - Agreed peer review
- Automated Deployment
  - leveraging software configuration management (SCM) tools
  - SCM treated as code
- Security
  - Compliance with Open Web Application Security Project (**OWASP**) secure coding guidelines
  - Automatic SAST, DAST
  - ESOC-Hub Information Security Management (T4.4)

Base for (XDC-1) Release Criteria

XDC - JIRA



Browse Projects

PROJECT TYPES

- **All Project Types**
- Software
- Business

CATEGORIES

- **All Categories**
- Recent Projects

All Project Types - All Categories

Contains text...

Project	Key	Project Type	Project Lead
PEB	PEB	Business	Daniele Cesini
XDC Project Management	XDM	Business	Doina Cristina Duma
XDC Software Development	XSD	Software	Doina Cristina Duma

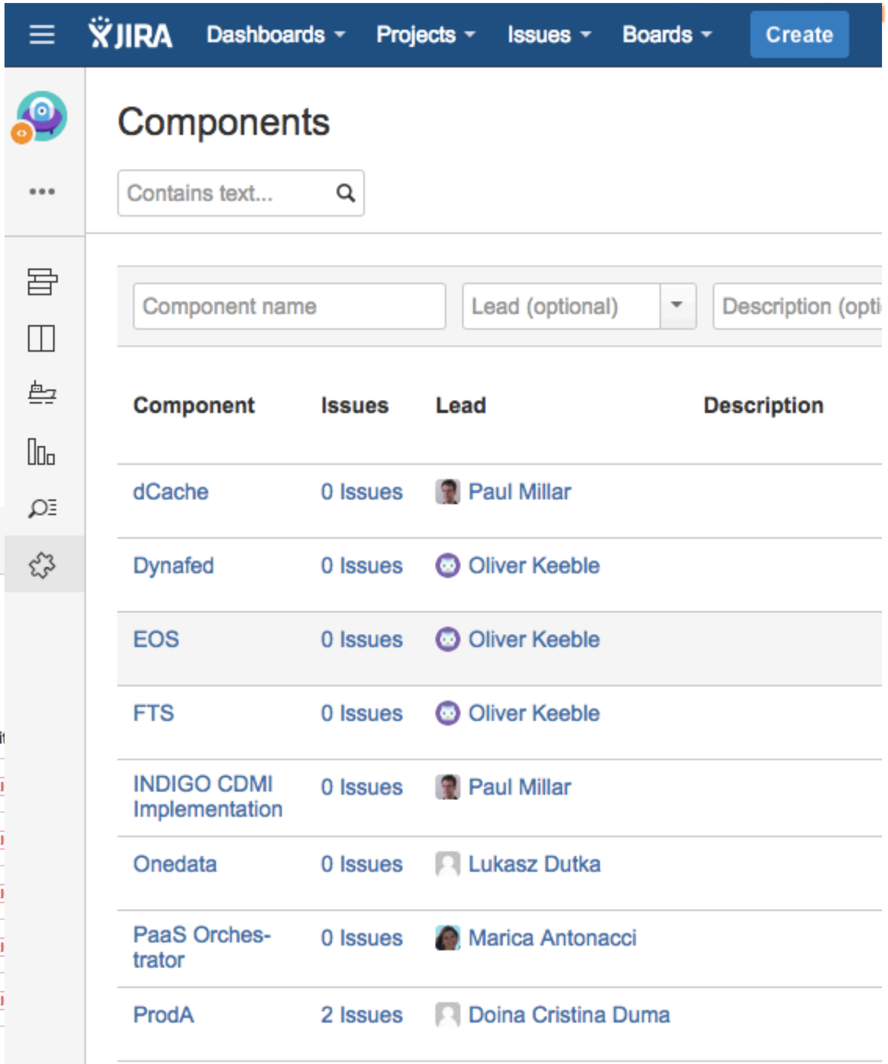
Boards

- Recently visited boards
- **All boards**

All boards

Board Type: All ▾ Find a board

Board name	Board type	Administrators	Saved Filter	Visibilit
WP2 board	Scrum	Doina Cristina Duma	Filter for WP2 board	RESTRI
WP3 board	Scrum	Doina Cristina Duma	Filter for WP3 board	RESTRI
WP4 board	Scrum	Doina Cristina Duma	Filter for WP4 board	RESTRI
WP5 board	Scrum	Doina Cristina Duma	Filter for WP5 board	RESTRI
XSD board	Scrum	Doina Cristina Duma	Filter for XSD board	RESTRI



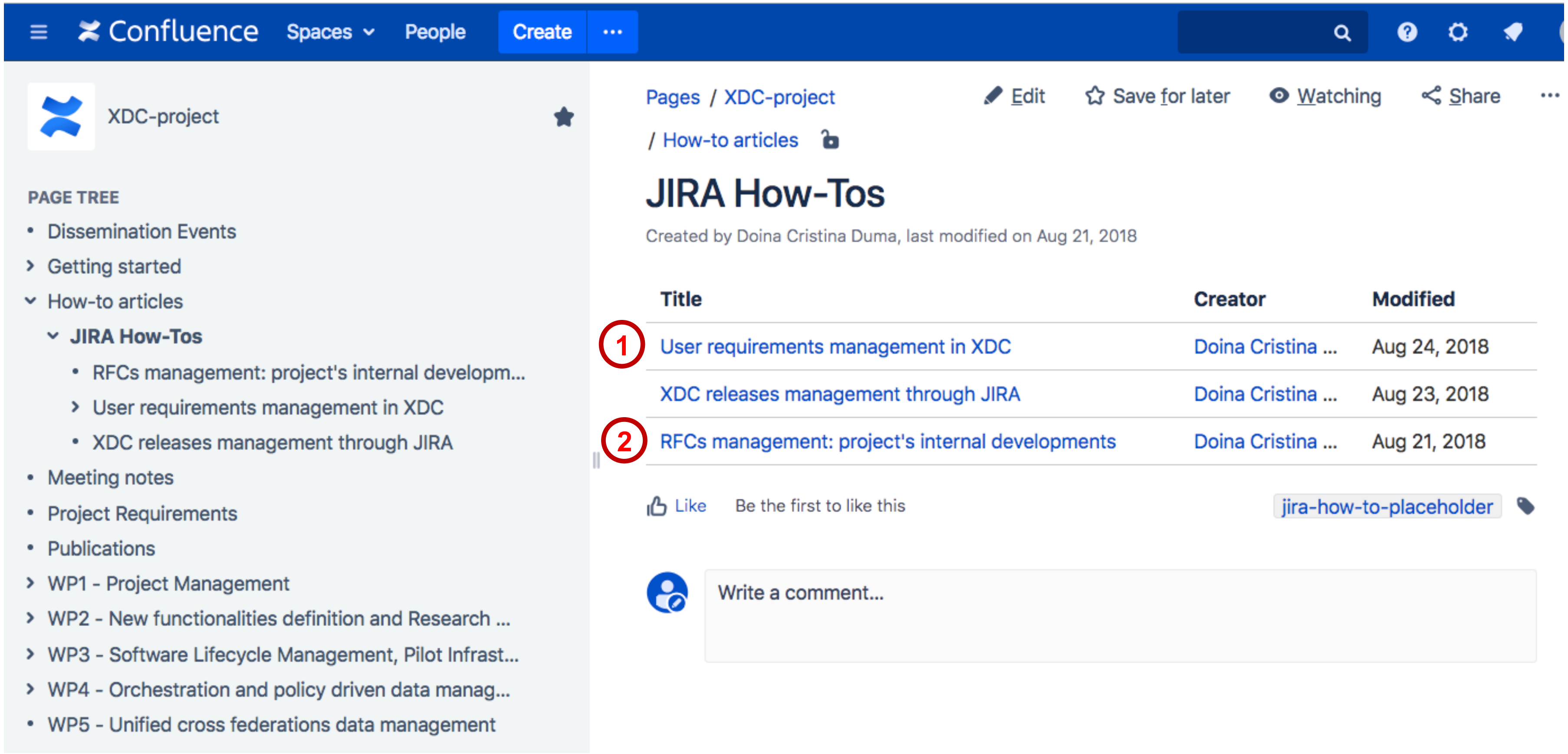
Components

Contains text...

Component name | Lead (optional) | Description (opti

Component	Issues	Lead	Description
dCache	0 Issues	Paul Millar	
Dynafed	0 Issues	Oliver Keeble	
EOS	0 Issues	Oliver Keeble	
FTS	0 Issues	Oliver Keeble	
INDIGO CDMI Implementation	0 Issues	Paul Millar	
Onedata	0 Issues	Lukasz Dutka	
PaaS Orchestrator	0 Issues	Marica Antonacci	
ProdA	2 Issues	Doina Cristina Duma	

How - Tos



- Dissemination Events
- Getting started
- How-to articles
  - JIRA How-Tos
    - RFCs management: project's internal developm...
    - User requirements management in XDC
    - XDC releases management through JIRA
- Meeting notes
- Project Requirements
- Publications
- WP1 - Project Management
- WP2 - New functionalities definition and Research ...
- WP3 - Software Lifecycle Management, Pilot Infrast...
- WP4 - Orchestration and policy driven data manag...
- WP5 - Unified cross federations data management

JIRA How-Tos

Created by Doina Cristina Duma, last modified on Aug 21, 2018

	Title	Creator	Modified
1	User requirements management in XDC	Doina Cristina ...	Aug 24, 2018
	XDC releases management through JIRA	Doina Cristina ...	Aug 23, 2018
2	RFCs management: project's internal developments	Doina Cristina ...	Aug 21, 2018

How-Tos ①

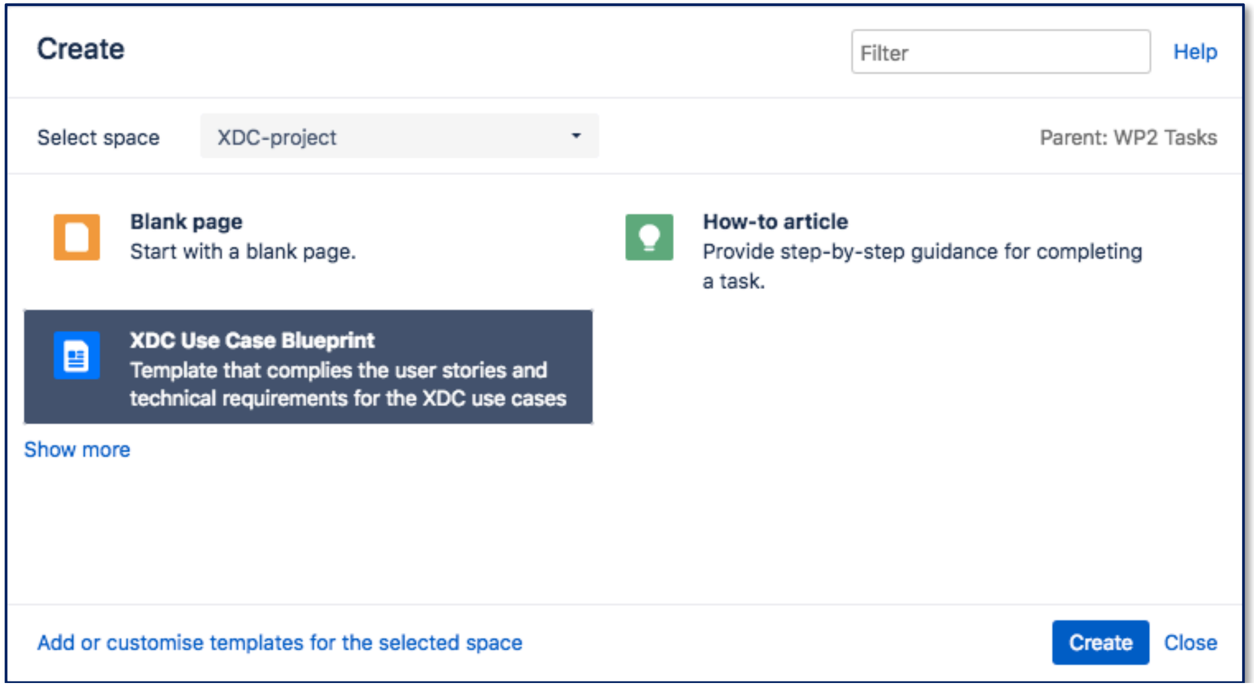
User requirements management

- The process describes the life cycle of the requirements coming from the use cases in WP2
  - Tools
    - **JIRA**: the issue tracking tool. This is where the main software development work takes place.
    - **Confluence**: the documentation repository. JIRA issues can be referenced within Confluence pages.
  - Terminology
    - **Epic** : represents the largest amount of work. An Epic groups multiple Tasks.
    - **Story** : software requirement expressed using non-technical language.
    - **Task** : unit of work that represents a technical activity.

How-Tos ①

User requirements management

1. WP2 representative creates a Use Case blueprint in Confluence
   - Different user stories and technical requirements are outlined



Create

Filter

Help

Select space: XDC-project

Parent: WP2 Tasks

Blank page
Start with a blank page.

How-to article
Provide step-by-step guidance for completing a task.

XDC Use Case Blueprint
Template that complies the user stories and technical requirements for the XDC use cases

Show more

Add or customise templates for the selected space

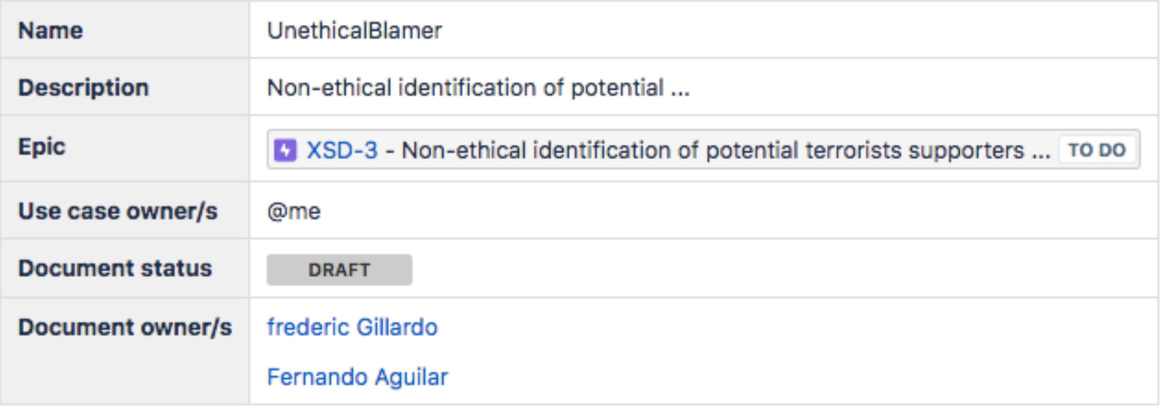
Create

Close

How-Tos ①

User requirements management

2. Each Use Case blueprint has to identify
   - The use case
   - The relevant actors
   - The JIRA Epic issue

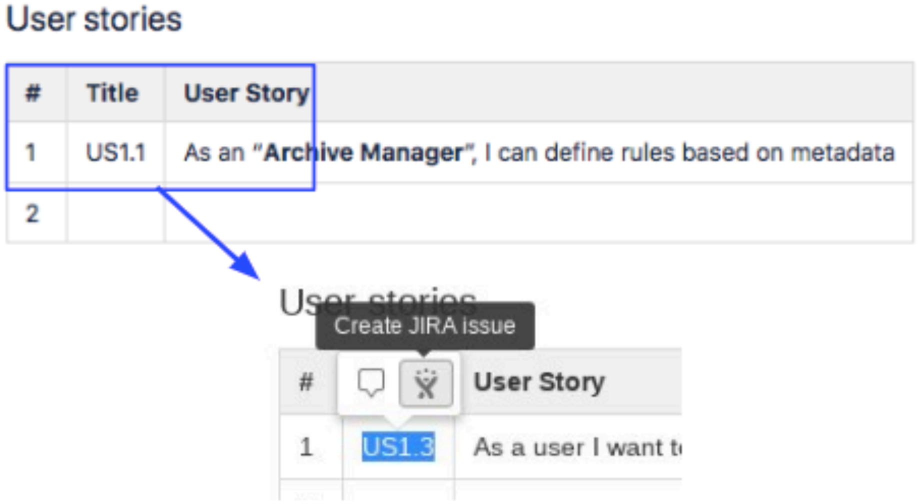


Name	UnethicalBlamer
Description	Non-ethical identification of potential ...
Epic	XSD-3 - Non-ethical identification of potential terrorists supporters ... TO DO
Use case owner/s	@me
Document status	DRAFT
Document owner/s	frederic Gillardo Fernando Aguilar

How-Tos ①

User requirements management

3. Create user stories and technical requirements by tracking the actual development progress in JIRA

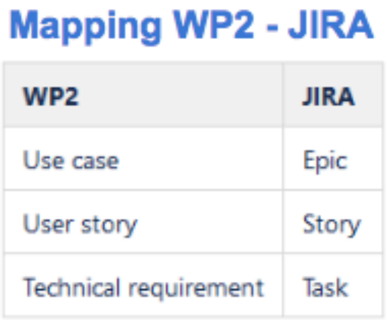


User stories

#	Title	User Story
1	US1.1	As an "**Archive Manager**", I can define rules based on metadata
2		

User stories

#		User Story
1	US1.3	As a user I want t



Mapping WP2 - JIRA

WP2	JIRA
Use case	Epic
User story	Story
Technical requirement	Task

The relationship between these 3 types of JIRA issues is one-to-many
(*1 Epic -> n Stories* and 1 *Story -> n Tasks*)

How-Tos ①

User requirements management

- Interaction among WP2, JRAs and WP3
  - **WP2** creates the required hierarchy of Epics/Stories/Tasks issues.
  - **WP4/5** take care of the development work to complete the user requirements
    - The workflow is the one described in the Software product requirement management in XDC.
  - **WP3** monitors the good progress of the requirement completion

How-Tos ②

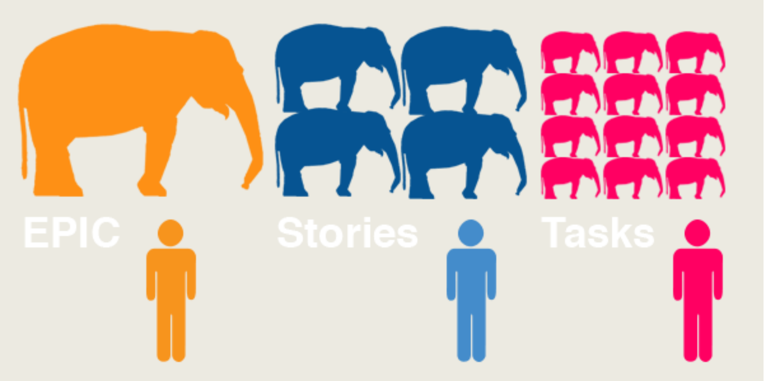
RFCs management: project's internal developments

- The process describes the lifecycle of a development requirement coming from the XDC development plans, based on XDC project proposal
- Tools
- Terminology
- XDC Software Development project
  - ***Important***:
    - All the software developments will be only tracked under the ***XDC software development*** JIRA project.
    - The **Label** field is mandatory when creating a new issue (=WP4/5)

How-Tos ②

RFCs management: project's internal developments

- The **minimum workflow required** must cover the usage of the following JIRA issues:
  - **Epics**, to identify the high-level requirements identified for each software product.
  - **Tasks/SubTaks**, to handle the units of work for each Epic.
  - **Bugs**, to track the detected defects found in the software.
- **Project reporting**
  - **JIRA issues will be used for project reporting**. Any improvement/feature/bugfix that is meant to be accounted **has to have a corresponding JIRA issue**
    - However**…**when using GitHub issues …release notes for the releases will contain only the JIRA issues.



How-Tos ②

XDC releases management through JIRA

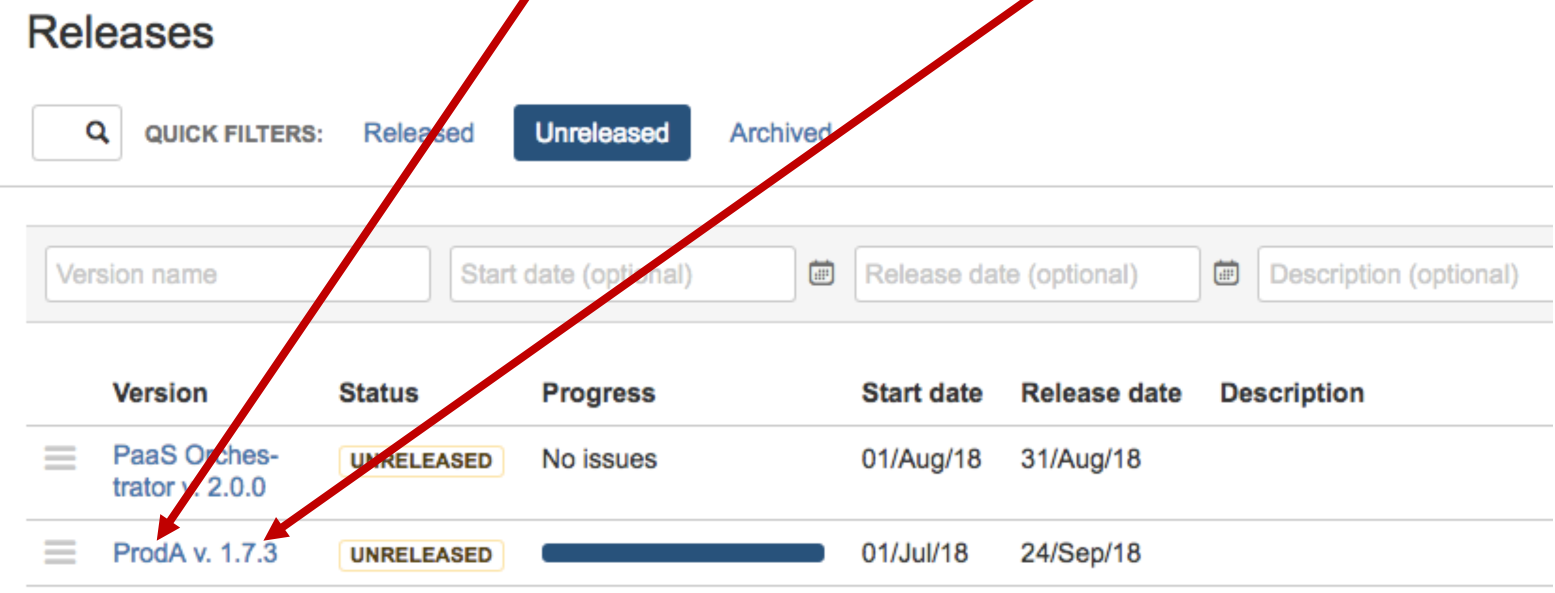
- The process describes how the PTs can create their own releases. The Release and Deployment Management (RDM) team will then utilize these versions to create the final releases
- Two levels
  - ***Product*** <- developers
    - Major, minor, revision
  - ***Project*** <- RDM
    - Major - Two major releases with fixed timelines
    - Updates – no fixed dates
- JIRA Version
  - set of ***features*** and ***fixes*** released together as a single update to your product

Workflow (short version)

- PT
  - Announces the release to RDM
- RDM
  - validates the release
  - generates a report
  - announce in XDC Software space (public)

How-Tos ②

- Workflow
  1. PT creates a new Version =“Name_of_product” + “”tag_version”
     - Tentative ***start and end dates***



Releases

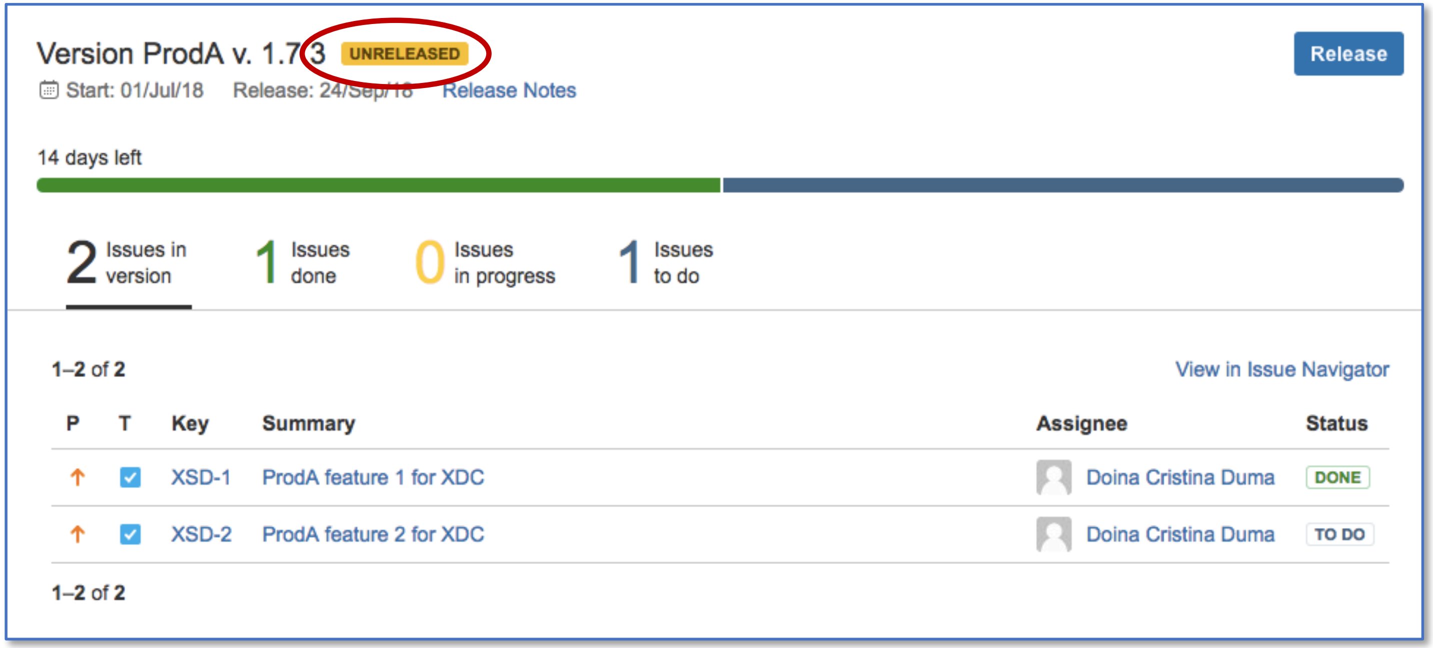
QUICK FILTERS: Released **Unreleased** Archived

Version name | Start date (optional) | Release date (optional) | Description (optional)

Version	Status	Progress	Start date	Release date	Description
PaaS Orches-trator v. 2.0.0	UNRELEASED	No issues	01/Aug/18	31/Aug/18	
ProdA v. 1.7.3	UNRELEASED		01/Jul/18	24/Sep/18	

How-Tos ②

- Workflow
  2. PT works on “version” – create issues, work on issues, add them to Version
  3. Notifies RDM



Version ProdA v. 1.73 UNRELEASED

Start: 01/Jul/18 Release: 24/Sep/18 Release Notes

14 days left

2 Issues in version | 1 Issues done | 0 Issues in progress | 1 Issues to do

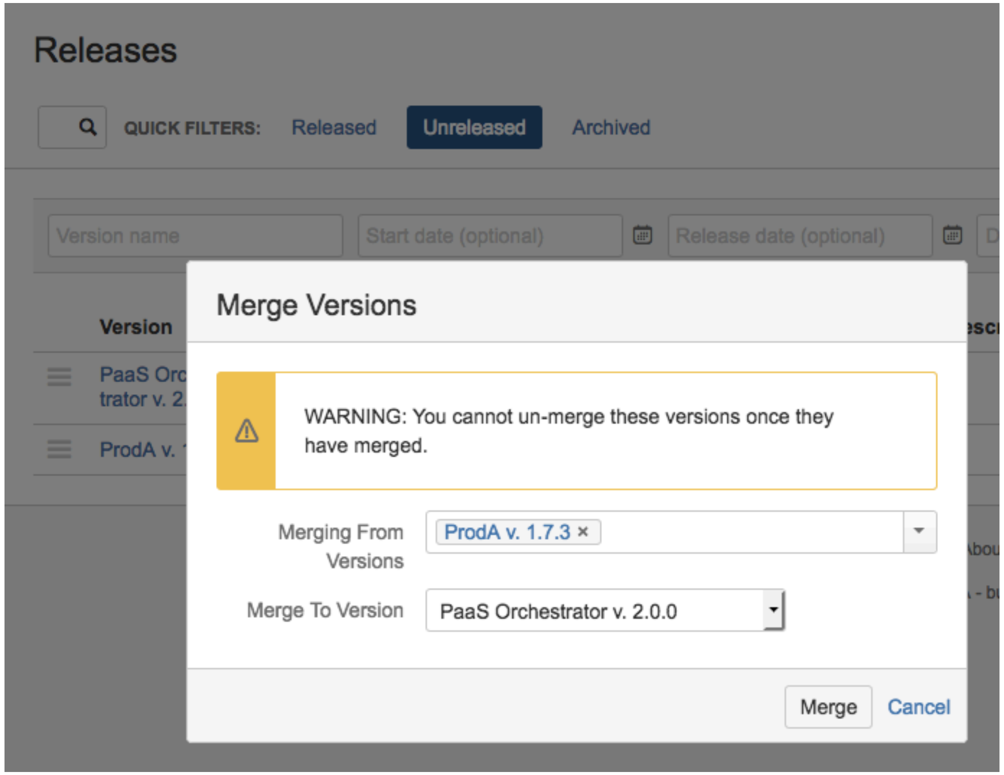
1–2 of 2

P	T	Key	Summary	Assignee	Status
↑	☑	XSD-1	ProdA feature 1 for XDC	Doina Cristina Duma	DONE
↑	☑	XSD-2	ProdA feature 2 for XDC	Doina Cristina Duma	TO DO

1–2 of 2

How-Tos ②

- Workflow – RDM
  1. ***Checks*** the status of the Issues involved in the Version/s.
     - RDM team might contact PTs to resolve any problem/doubt.
  2. ***Checks release criteria are met***
  3. ***Decides*** whether the given Version/s will be part of an update or major release.
     - In case of handling more than one Version, the *RDM team will **merge** them* to create a unique Version.
     - Release Notes from the different *Versions* are merged as well, listing all the *Issues* involved.
  4. Prepares announcement



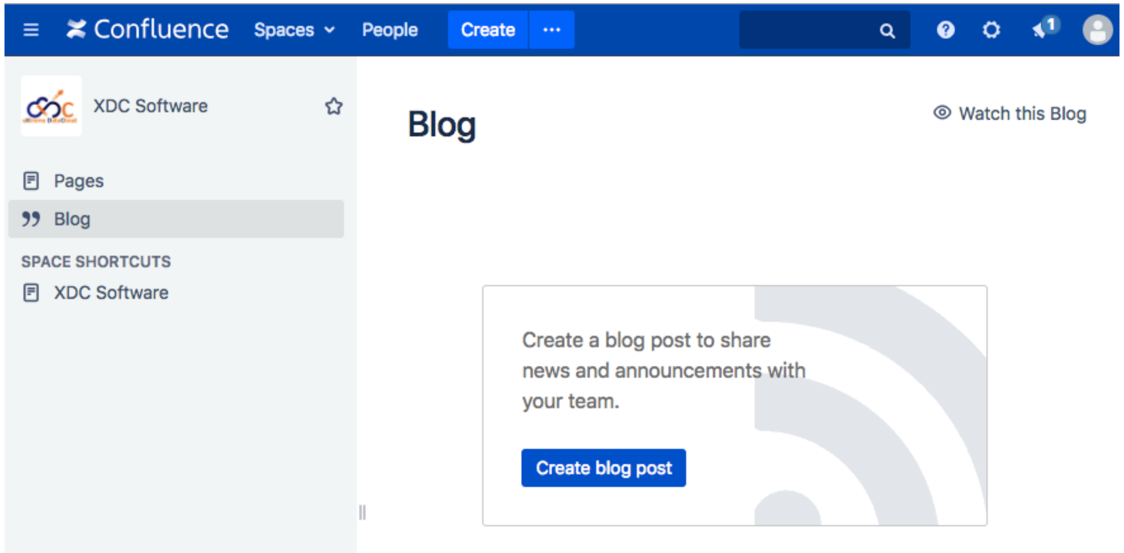
WARNING: You cannot un-merge these versions once they have merged.

Merging From Versions: ProdA v. 1.7.3 ×

Merge To Version: PaaS Orchestrator v. 2.0.0

Things to remember

- **JIRA issues are the primary source of information for tracking the work done on each software product**
- Versions status are set to “Released” by the RDM.



Create a blog post to share news and announcements with your team.

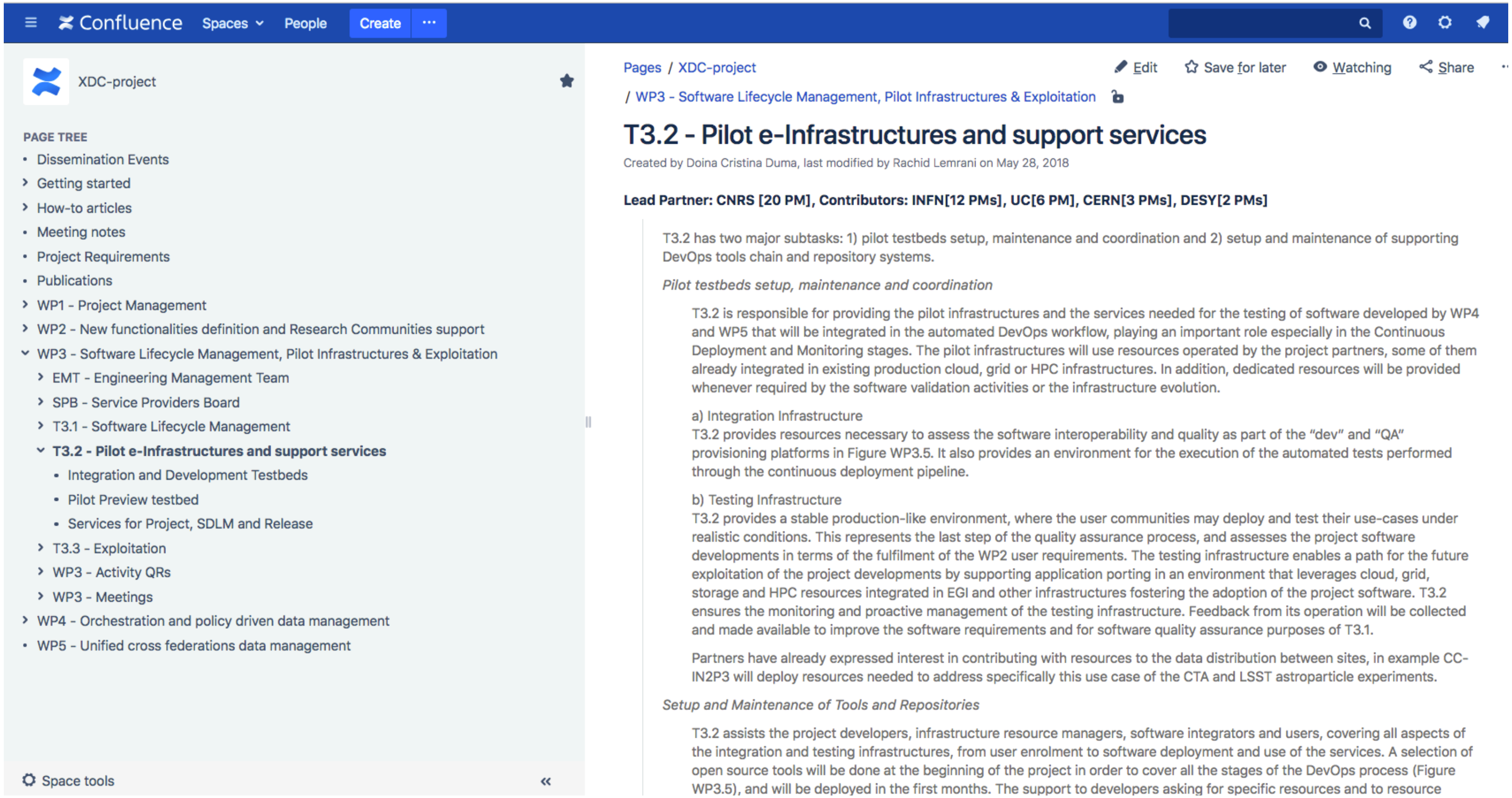
What about Support?

- For external users:
  - GGUS SU:
    - INDIGO DataCloud Catch-all
    - FTS Development
    - Dynafed Development
    - dCache Support
- For internal users:
  - XDM – WP3 Board
- For “all”:
  - support@extreme-datacloud.eu

- Request for resources both for use-cases and devels
- Request for services deployment, operations (upgrade, monitoring)

T3.2 - Pilot e-Infrastructures and support services

T3.2


Confluence
Spaces
People
Create
XDC-project
PAGE TREE
Dissemination Events
Getting started
How-to articles
Meeting notes
Project Requirements
Publications
WP1 - Project Management
WP2 - New functionalities definition and Research Communities support
WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation
EMT - Engineering Management Team
SPB - Service Providers Board
T3.1 - Software Lifecycle Management
T3.2 - Pilot e-Infrastructures and support services
Integration and Development Testbeds
Pilot Preview testbed
Services for Project, SDLM and Release
T3.3 - Exploitation
WP3 - Activity QRs
WP3 - Meetings
WP4 - Orchestration and policy driven data management
WP5 - Unified cross federations data management
Space tools
Pages / XDC-project
/ WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation
Edit
Save for later
Watching
Share

T3.2 - Pilot e-Infrastructures and support services

Created by Doina Cristina Duma, last modified by Rachid Lemrani on May 28, 2018

Lead Partner: CNRS [20 PM], Contributors: INFN[12 PMs], UC[6 PM], CERN[3 PMs], DESY[2 PMs]

T3.2 has two major subtasks: 1) pilot testbeds setup, maintenance and coordination and 2) setup and maintenance of supporting DevOps tools chain and repository systems.

Pilot testbeds setup, maintenance and coordination

T3.2 is responsible for providing the pilot infrastructures and the services needed for the testing of software developed by WP4 and WP5 that will be integrated in the automated DevOps workflow, playing an important role especially in the Continuous Deployment and Monitoring stages. The pilot infrastructures will use resources operated by the project partners, some of them already integrated in existing production cloud, grid or HPC infrastructures. In addition, dedicated resources will be provided whenever required by the software validation activities or the infrastructure evolution.

a) Integration Infrastructure
T3.2 provides resources necessary to assess the software interoperability and quality as part of the "dev" and "QA" provisioning platforms in Figure WP3.5. It also provides an environment for the execution of the automated tests performed through the continuous deployment pipeline.

b) Testing Infrastructure
T3.2 provides a stable production-like environment, where the user communities may deploy and test their use-cases under realistic conditions. This represents the last step of the quality assurance process, and assesses the project software developments in terms of the fulfilment of the WP2 user requirements. The testing infrastructure enables a path for the future exploitation of the project developments by supporting application porting in an environment that leverages cloud, grid, storage and HPC resources integrated in EGI and other infrastructures fostering the adoption of the project software. T3.2 ensures the monitoring and proactive management of the testing infrastructure. Feedback from its operation will be collected and made available to improve the software requirements and for software quality assurance purposes of T3.1.

Partners have already expressed interest in contributing with resources to the data distribution between sites, in example CC-IN2P3 will deploy resources needed to address specifically this use case of the CTA and LSST astroparticle experiments.

Setup and Maintenance of Tools and Repositories

T3.2 assists the project developers, infrastructure resource managers, software integrators and users, covering all aspects of the integration and testing infrastructures, from user enrolment to software deployment and use of the services. A selection of open source tools will be done at the beginning of the project in order to cover all the stages of the DevOps process (Figure WP3.5), and will be deployed in the first months. The support to developers asking for specific resources and to resource

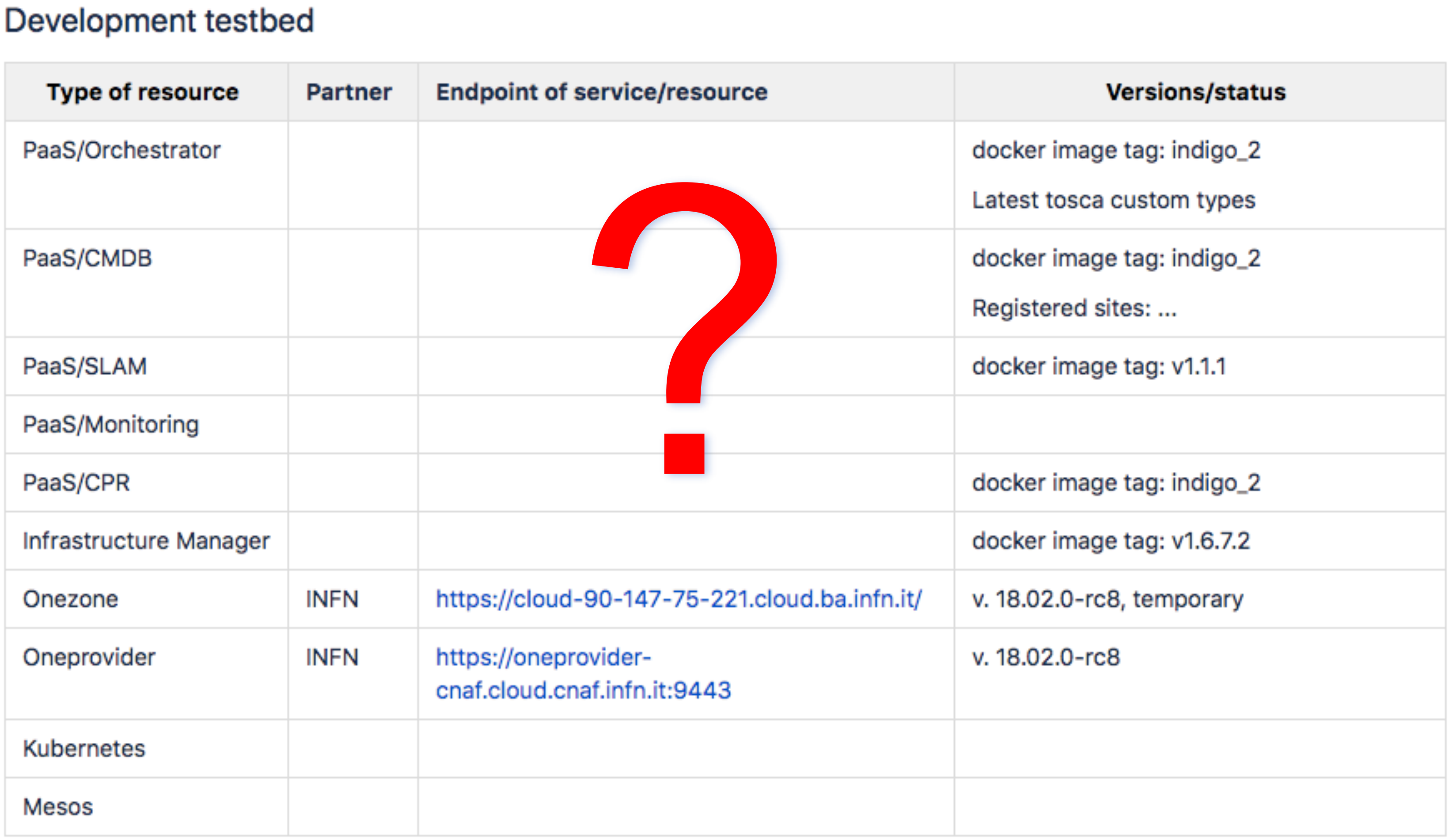
T3.2

- T3.2 has ***two major subtasks***:
  - Pilot testbeds setup, maintenance and coordination
  - Setup and maintenance of supporting DevOps tools chain and repository systems
- Activities are aimed at providing the following Services
  - ***Integration*** and ***Development*** Testbeds
  - Pilot ***Preview*** testbed
  - ***Services*** for Project, SDLM and Release

T3.2 - Integration and Development Testbeds

- An integrated platform where Services and Resources are made available for the projects activities
  - Developers and Champions can ***develop*** & ***test*** their workflows
  - provides an environment for the execution of the ***automated tests*** performed through the continuous deployment pipeline
- Most of the partners agreed on contributing to this platform
  - INFN, CERN, AGH, DESY, CNRS, UC
  - External services are also made available to the project: VOMS, WaTTs, others
- A complete list can be found in the XDC Confluence page:
  - https://confluence.extreme-datacloud.eu/display/XDCPROJ/Integration+and+Development+Testbeds

T3.2 - Integration and Development Testbeds

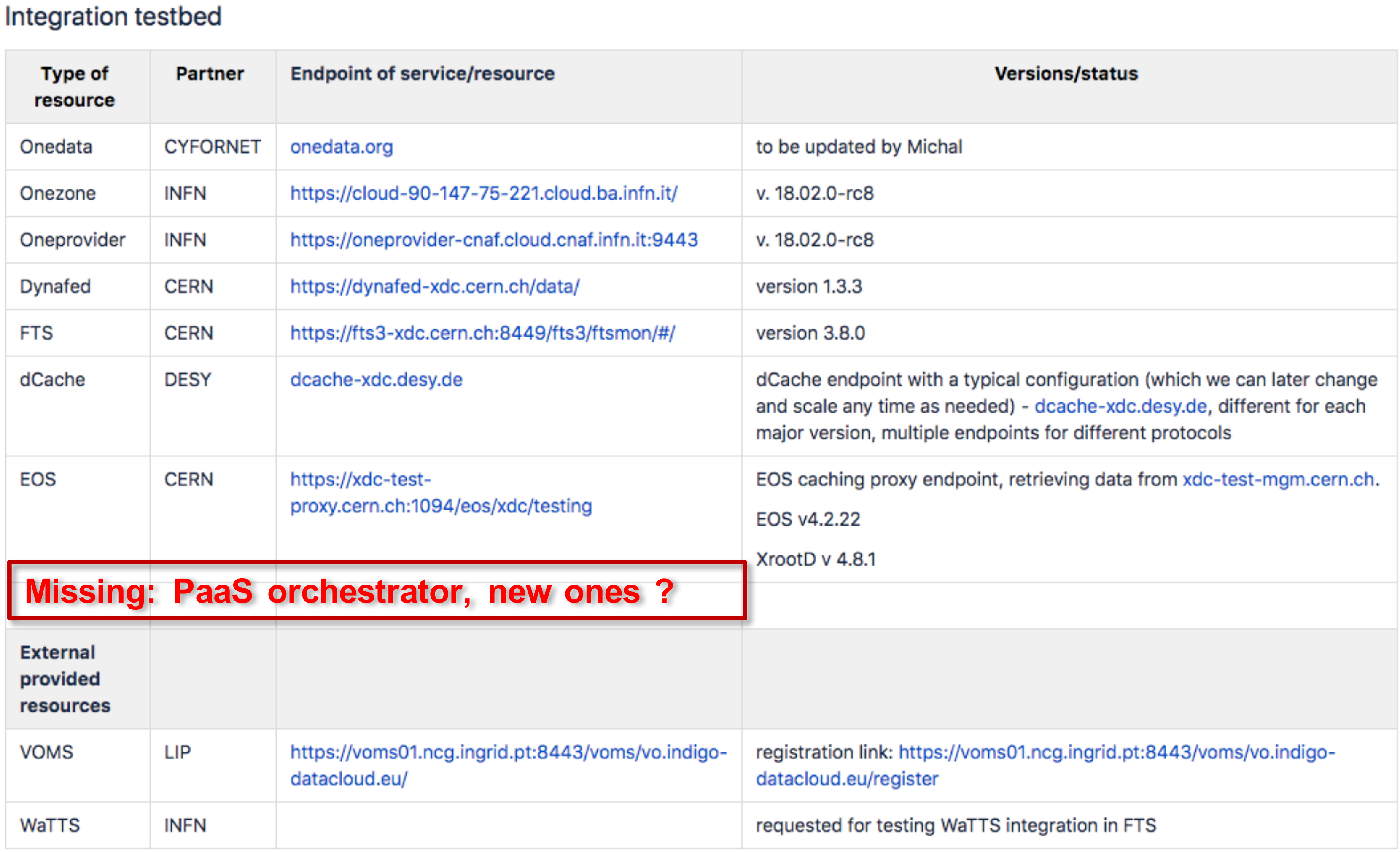


Development testbed

Type of resource	Partner	Endpoint of service/resource	Versions/status
PaaS/Orchestrator			docker image tag: indigo_2 Latest tosca custom types
PaaS/CMDB			docker image tag: indigo_2 Registered sites: ...
PaaS/SLAM			docker image tag: v1.1.1
PaaS/Monitoring			
PaaS/CPR			docker image tag: indigo_2
Infrastructure Manager			docker image tag: v1.6.7.2
Onezone	INFN	https://cloud-90-147-75-221.cloud.ba.infn.it/	v. 18.02.0-rc8, temporary
Oneprovider	INFN	https://oneprovider-cnaf.cloud.cnaf.infn.it:9443	v. 18.02.0-rc8
Kubernetes			
Mesos			

?

T3.2 - Integration and Development Testbeds

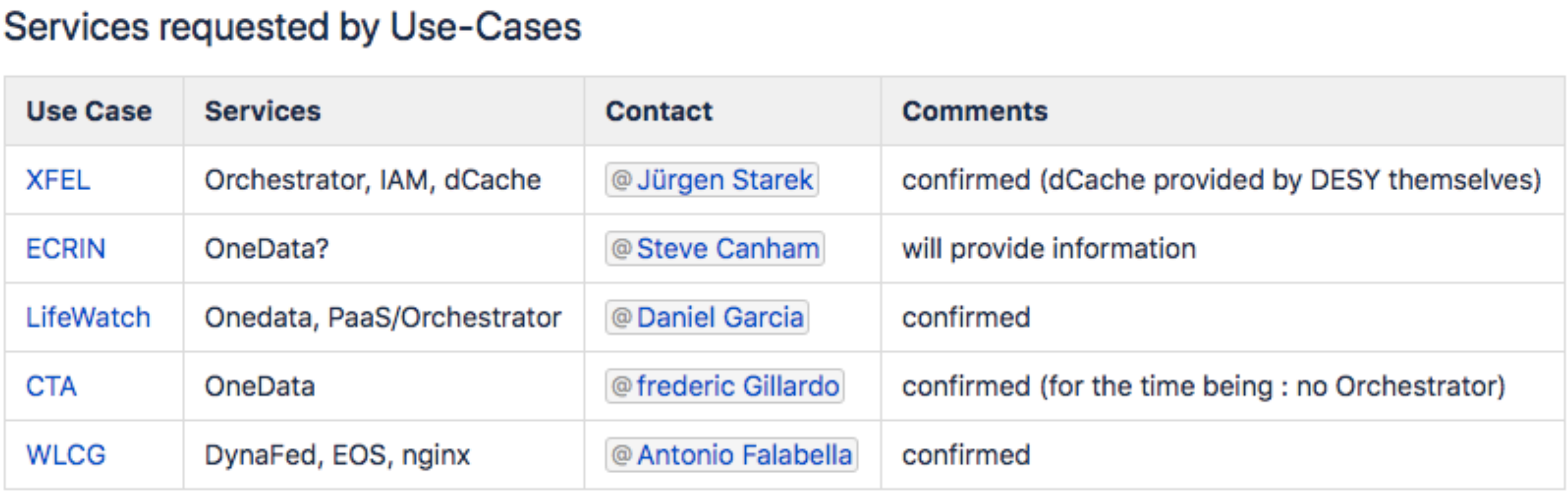


Integration testbed

Type of resource	Partner	Endpoint of service/resource	Versions/status
Onedata	CYFORNET	onedata.org	to be updated by Michal
Onezone	INFN	https://cloud-90-147-75-221.cloud.ba.infn.it/	v. 18.02.0-rc8
Oneprovider	INFN	https://oneprovider-cnaf.cloud.cnaf.infn.it:9443	v. 18.02.0-rc8
Dynafed	CERN	https://dynafed-xdc.cern.ch/data/	version 1.3.3
FTS	CERN	https://fts3-xdc.cern.ch:8449/fts3/ftsmon/#/	version 3.8.0
dCache	DESY	dcache-xdc.desy.de	dCache endpoint with a typical configuration (which we can later change and scale any time as needed) - dcache-xdc.desy.de, different for each major version, multiple endpoints for different protocols
EOS	CERN	https://xdc-test-proxy.cern.ch:1094/eos/xdc/testing	EOS caching proxy endpoint, retrieving data from xdc-test-mgm.cern.ch. EOS v4.2.22 XrootD v 4.8.1
External provided resources			
VOMS	LIP	https://voms01.ncg.ingrid.pt:8443/voms/vo.indigo-datacloud.eu/	registration link: https://voms01.ncg.ingrid.pt:8443/voms/vo.indigo-datacloud.eu/register
WaTTS	INFN		requested for testing WaTTS integration in FTS

Missing: PaaS orchestrator, new ones ?

T3.2 - Integration and Development Testbeds



Services requested by Use-Cases

Use Case	Services	Contact	Comments
XFEL	Orchestrator, IAM, dCache	@Jürgen Starek	confirmed (dCache provided by DESY themselves)
ECRIN	OneData?	@Steve Canham	will provide information
LifeWatch	Onedata, PaaS/Orchestrator	@Daniel Garcia	confirmed
CTA	OneData	@frederic Gillardo	confirmed (for the time being : no Orchestrator)
WLCG	DynaFed, EOS, nginx	@Antonio Falabella	confirmed

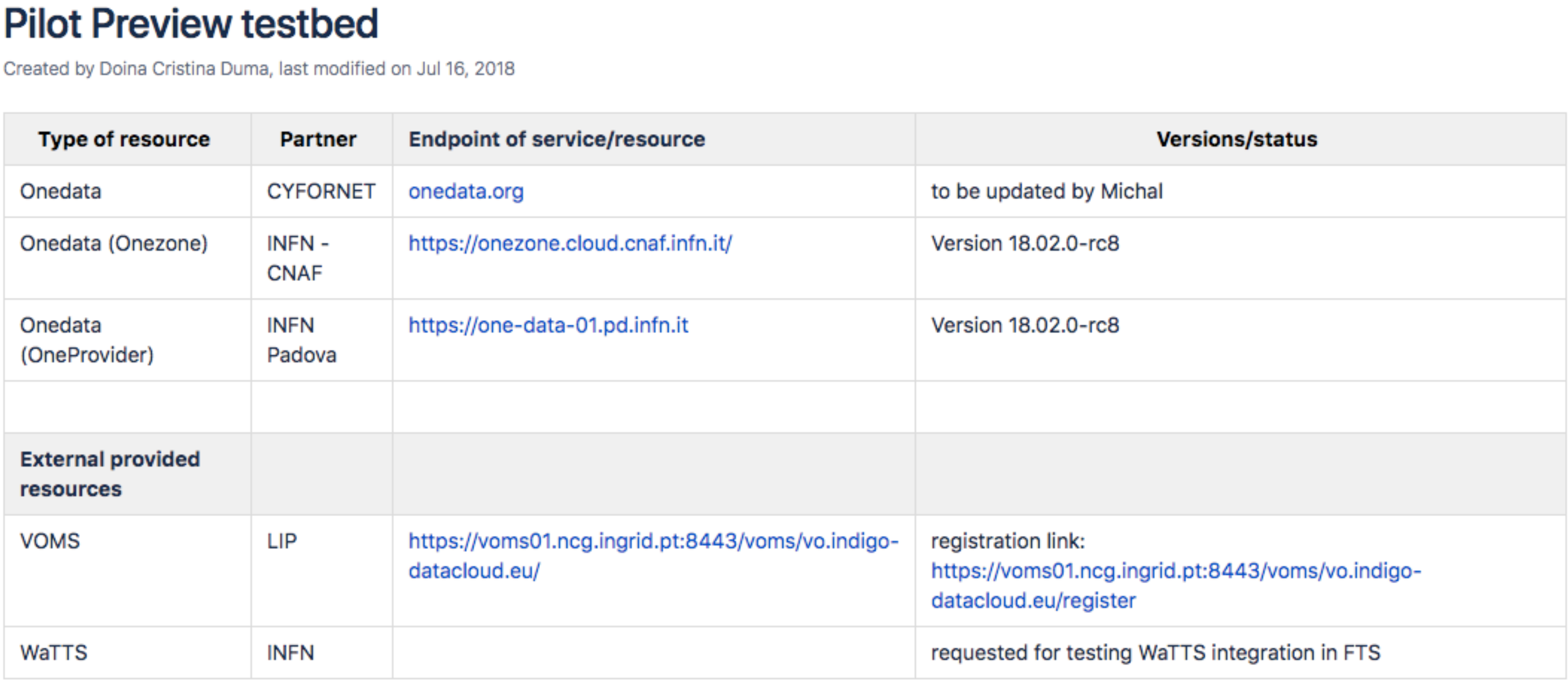
Issues:

- There are missing services
- How the services are accessed? IAM, VOMS?
- What deployment?
- What other resources needed to support the use-cases workflows?

T3.2 - Pilot Preview testbed

- It is the testbed providing the released services versions
- Aimed at supporting dissemination and exploitation
  - Transfer knowledge & results with the aim to enable others to use and take up results
  - Effectively use project results and made available for scientific and commercial purposes
- An updated list can be found in the XDC Confluence page
  - https://confluence.extreme-datacloud.eu/display/XDCPROJ/Pilot+Preview+testbed

T3.2 - Pilot Preview testbed

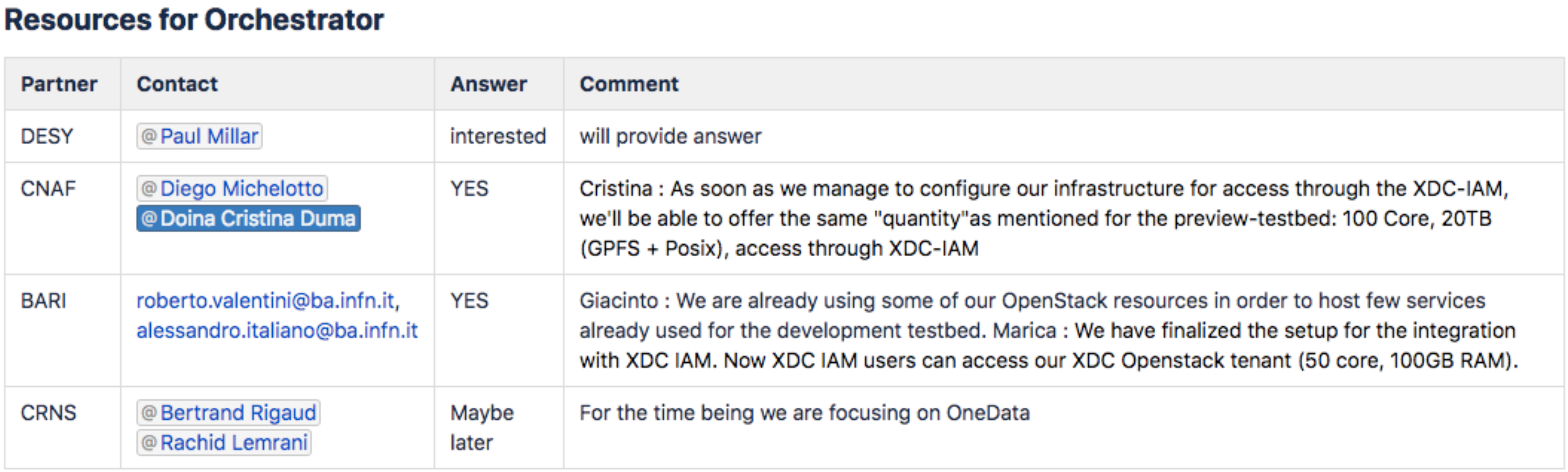


Pilot Preview testbed

Created by Doina Cristina Duma, last modified on Jul 16, 2018

Type of resource	Partner	Endpoint of service/resource	Versions/status
Onedata	CYFORNET	onedata.org	to be updated by Michal
Onedata (Onezone)	INFN - CNAF	https://onezone.cloud.cnaf.infn.it/	Version 18.02.0-rc8
Onedata (OneProvider)	INFN Padova	https://one-data-01.pd.infn.it	Version 18.02.0-rc8
External provided resources			
VOMS	LIP	https://voms01.ncg.ingrid.pt:8443/voms/vo.indigo-datacloud.eu/	registration link: https://voms01.ncg.ingrid.pt:8443/voms/vo.indigo-datacloud.eu/register
WaTTS	INFN		requested for testing WaTTS integration in FTS

T3.2 - Pilot Preview testbed

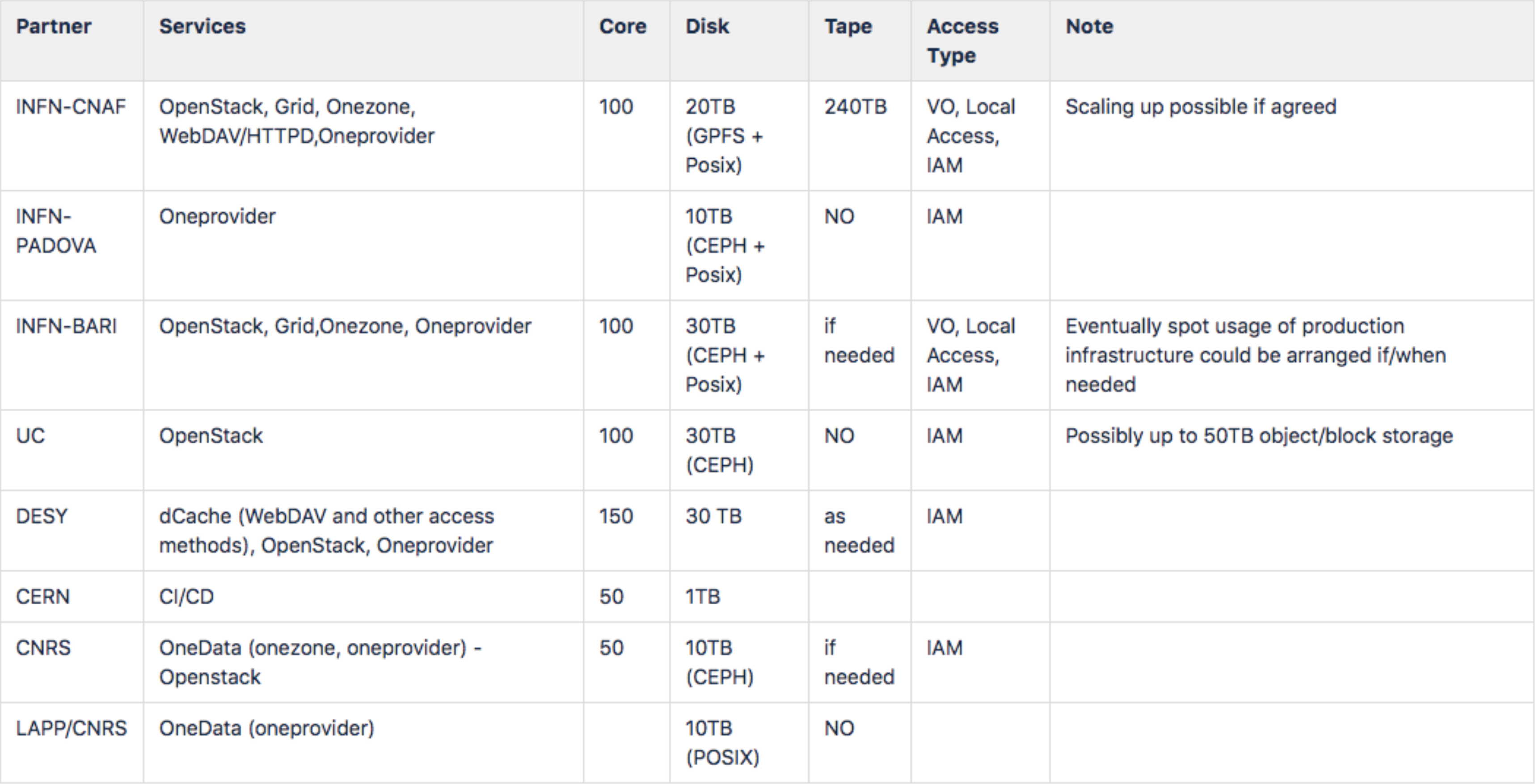


Resources for Orchestrator

Partner	Contact	Answer	Comment
DESY	@Paul Millar	interested	will provide answer
CNAF	@Diego Michelotto @Doina Cristina Duma	YES	Cristina : As soon as we manage to configure our infrastructure for access through the XDC-IAM, we'll be able to offer the same "quantity"as mentioned for the preview-testbed: 100 Core, 20TB (GPFS + Posix), access through XDC-IAM
BARI	roberto.valentini@ba.infn.it, alessandro.italiano@ba.infn.it	YES	Giacinto : We are already using some of our OpenStack resources in order to host few services already used for the development testbed. Marica : We have finalized the setup for the integration with XDC IAM. Now XDC IAM users can access our XDC Openstack tenant (50 core, 100GB RAM).
CRNS	@Bertrand Rigaud @Rachid Lemrani	Maybe later	For the time being we are focusing on OneData

T3.2

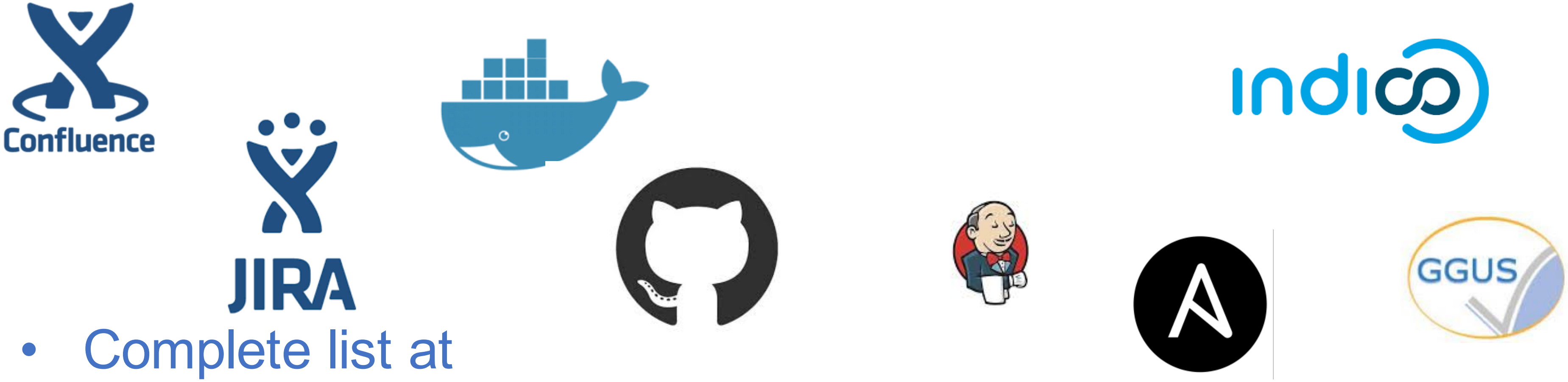
Services and Resources to be commited by partners



Partner	Services	Core	Disk	Tape	Access Type	Note
INFN-CNAF	OpenStack, Grid, Onezone, WebDAV/HTTPD,Oneprovider	100	20TB (GPFS + Posix)	240TB	VO, Local Access, IAM	Scaling up possible if agreed
INFN-PADOVA	Oneprovider		10TB (CEPH + Posix)	NO	IAM	
INFN-BARI	OpenStack, Grid,Onezone, Oneprovider	100	30TB (CEPH + Posix)	if needed	VO, Local Access, IAM	Eventually spot usage of production infrastructure could be arranged if/when needed
UC	OpenStack	100	30TB (CEPH)	NO	IAM	Possibly up to 50TB object/block storage
DESY	dCache (WebDAV and other access methods), OpenStack, Oneprovider	150	30 TB	as needed	IAM	
CERN	CI/CD	50	1TB			
CNRS	OneData (onezone, oneprovider) - Openstack	50	10TB (CEPH)	if needed	IAM	
LAPP/CNRS	OneData (oneprovider)		10TB (POSIX)	NO		

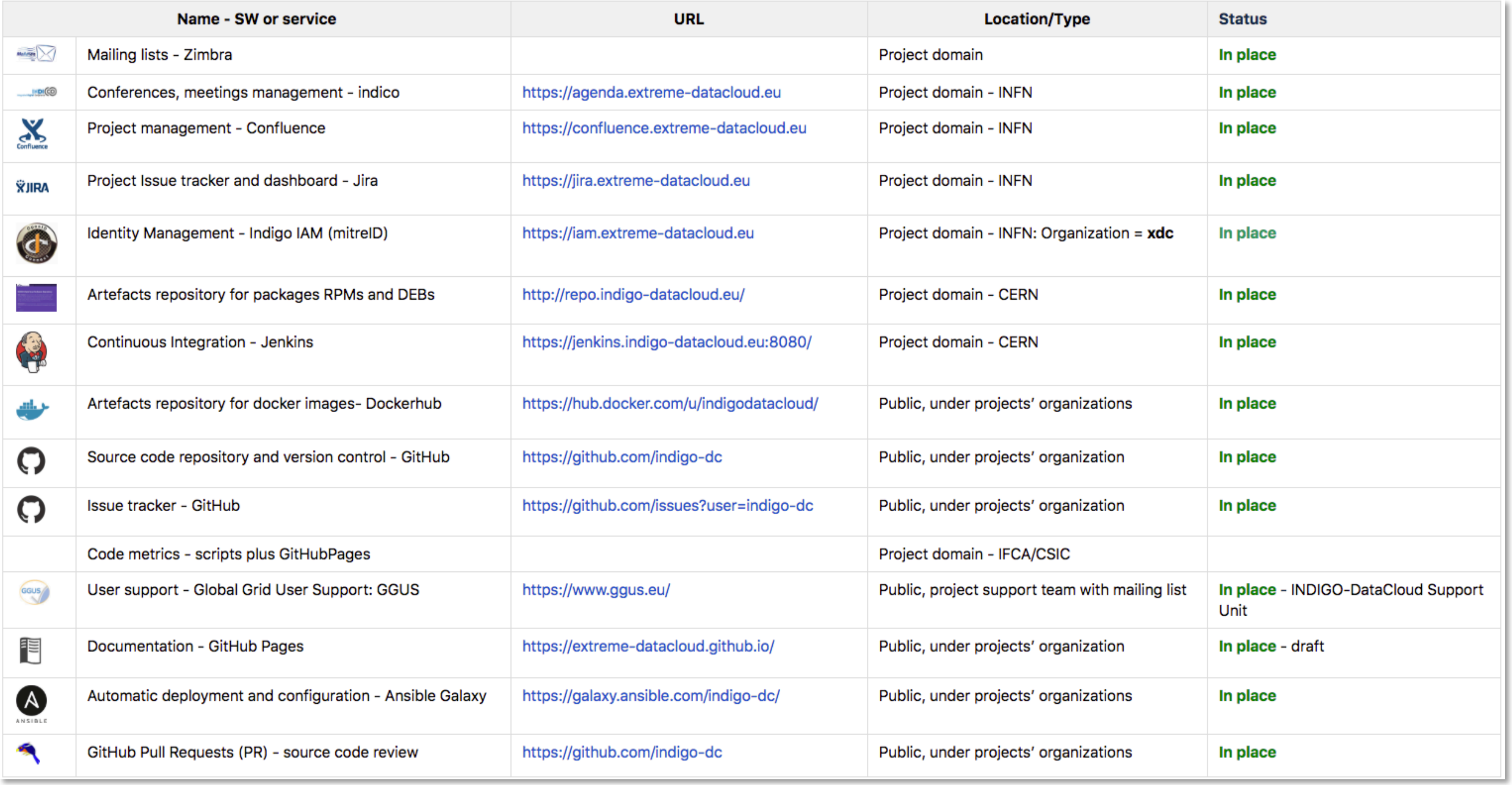
T3.2 - Services for Project, SDLM and Release

- A set of activities have been carried out to put in place a set services aimed at supporting and facilitate the project activities



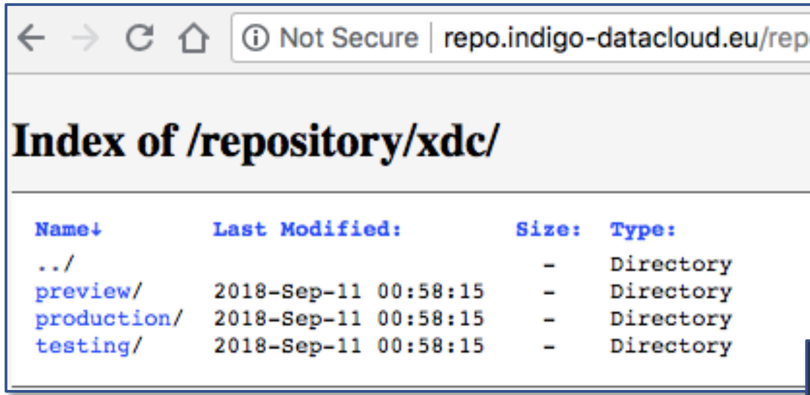
- Complete list at
  - https://confluence.extreme-datacloud.eu/display/XDCPROJ/Services+for+Project%2C+SDLM+and+Release

T3.2 - Services for Project, SDLM and Release



Name - SW or service		URL	Location/Type	Status
	Mailing lists - Zimbra		Project domain	In place
	Conferences, meetings management - indico	https://agenda.extreme-datacloud.eu	Project domain - INFN	In place
	Project management - Confluence	https://confluence.extreme-datacloud.eu	Project domain - INFN	In place
	Project Issue tracker and dashboard - Jira	https://jira.extreme-datacloud.eu	Project domain - INFN	In place
	Identity Management - Indigo IAM (mitreID)	https://iam.extreme-datacloud.eu	Project domain - INFN: Organization = **xdc**	In place
	Artefacts repository for packages RPMs and DEBs	http://repo.indigo-datacloud.eu/	Project domain - CERN	In place
	Continuous Integration - Jenkins	https://jenkins.indigo-datacloud.eu:8080/	Project domain - CERN	In place
	Artefacts repository for docker images- Dockerhub	https://hub.docker.com/u/indigodatacloud/	Public, under projects' organizations	In place
	Source code repository and version control - GitHub	https://github.com/indigo-dc	Public, under projects' organization	In place
	Issue tracker - GitHub	https://github.com/issues?user=indigo-dc	Public, under projects' organization	In place
	Code metrics - scripts plus GitHubPages		Project domain - IFCA/CSIC	
	User support - Global Grid User Support: GGUS	https://www.ggus.eu/	Public, project support team with mailing list	In place - INDIGO-DataCloud Support Unit
	Documentation - GitHub Pages	https://extreme-datacloud.github.io/	Public, under projects' organization	In place - draft
	Automatic deployment and configuration - Ansible Galaxy	https://galaxy.ansible.com/indigo-dc/	Public, under projects' organizations	In place
	GitHub Pull Requests (PR) - source code review	https://github.com/indigo-dc	Public, under projects' organizations	In place

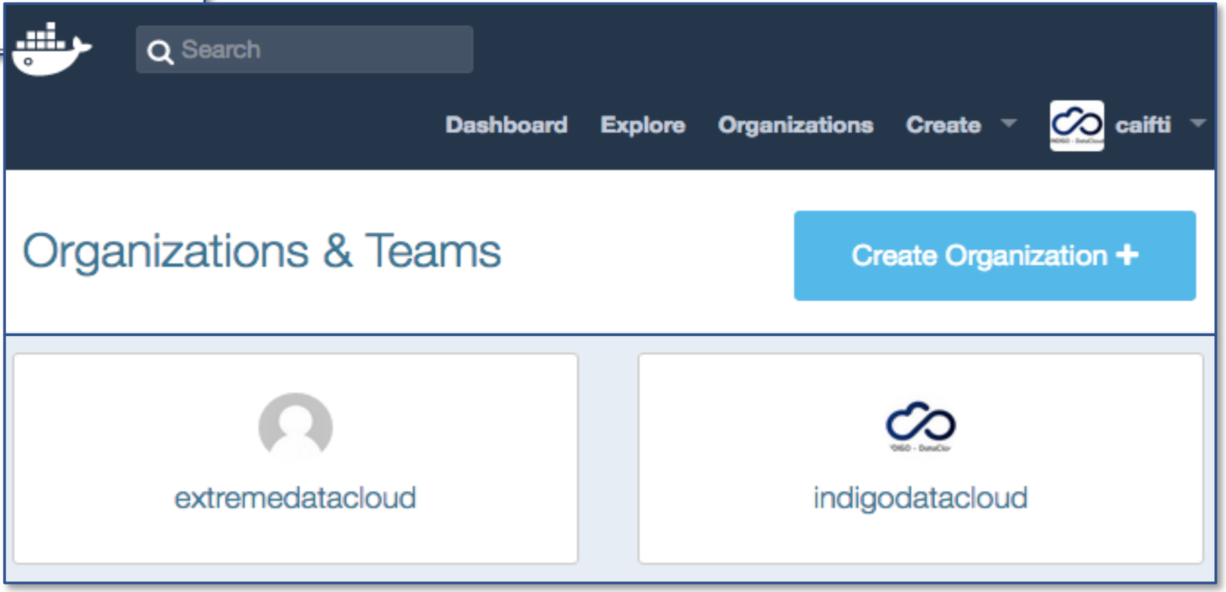
Repositories



Not Secure | repo.indigo-datacloud.eu/repository/xdc/

Index of /repository/xdc/

```
Name↓        Last Modified:           Size:  Type:
../                                     -    Directory
preview/     2018-Sep-11 00:58:15       -    Directory
production/  2018-Sep-11 00:58:15       -    Directory
testing/     2018-Sep-11 00:58:15       -    Directory
```

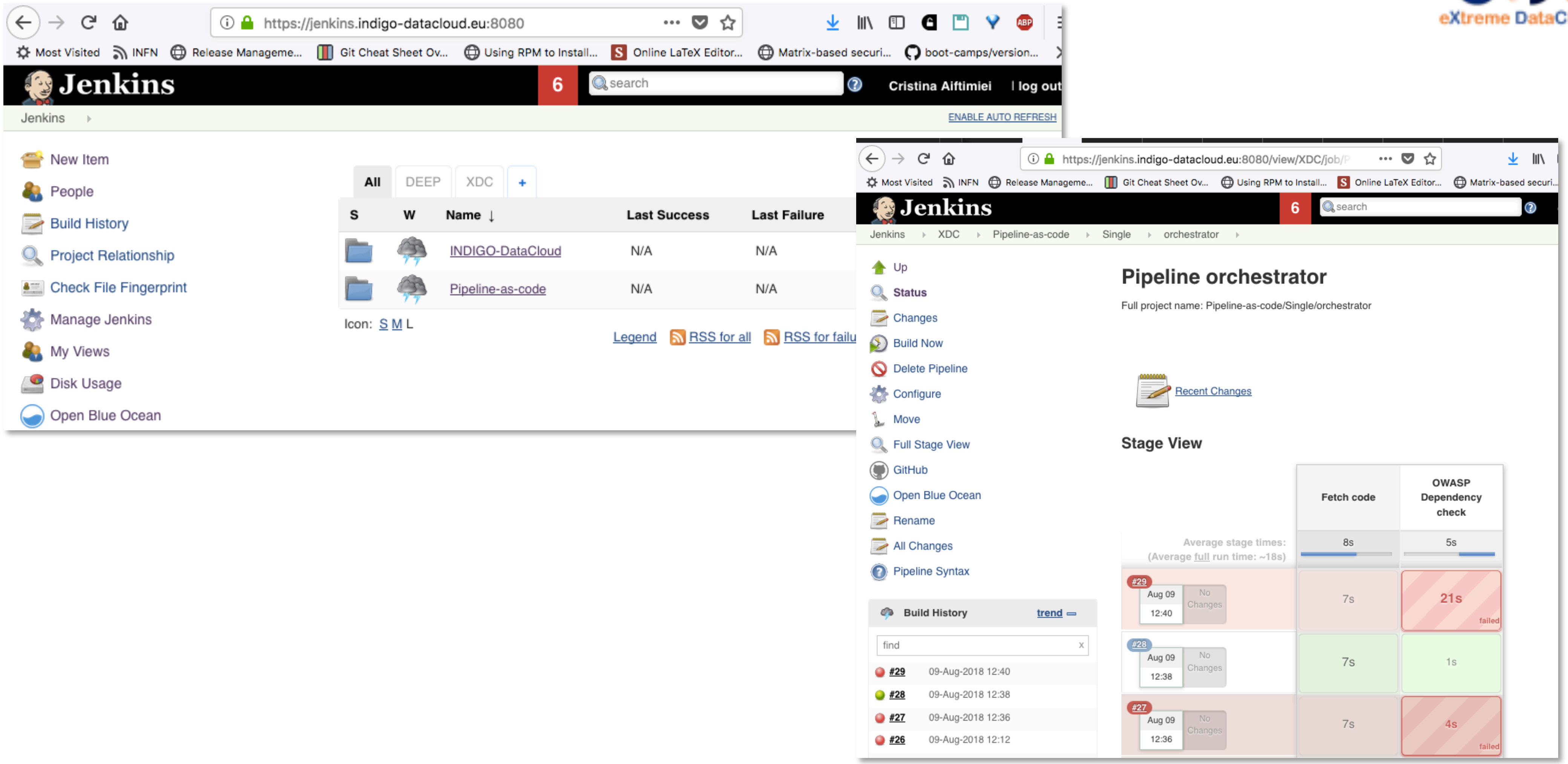


Organizations & Teams

extremedatacloud

indigodatacloud

CI/CD - Jenkins



https://jenkins.indigo-datacloud.eu:8080

Most Visited · INFN · Release Manageme... · Git Cheat Sheet Ov... · Using RPM to Install... · Online LaTeX Editor... · Matrix-based securi... · boot-camps/version...

Jenkins · 6 · search · Cristina Aiftimiei · log out

Jenkins › ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Disk Usage
- Open Blue Ocean

All · DEEP · XDC · +

S	W	Name ↓	Last Success	Last Failure
		INDIGO-DataCloud	N/A	N/A
		Pipeline-as-code	N/A	N/A

Icon: S M L

Legend · RSS for all · RSS for failu

https://jenkins.indigo-datacloud.eu:8080/view/XDC/job/P

Most Visited · INFN · Release Manageme... · Git Cheat Sheet Ov... · Using RPM to Install... · Online LaTeX Editor... · Matrix-based securi...

Jenkins · 6 · search

Jenkins › XDC › Pipeline-as-code › Single › orchestrator ›

- Up
- Status
- Changes
- Build Now
- Delete Pipeline
- Configure
- Move
- Full Stage View
- GitHub
- Open Blue Ocean
- Rename
- All Changes
- Pipeline Syntax

Pipeline orchestrator

Full project name: Pipeline-as-code/Single/orchestrator

Recent Changes

Stage View

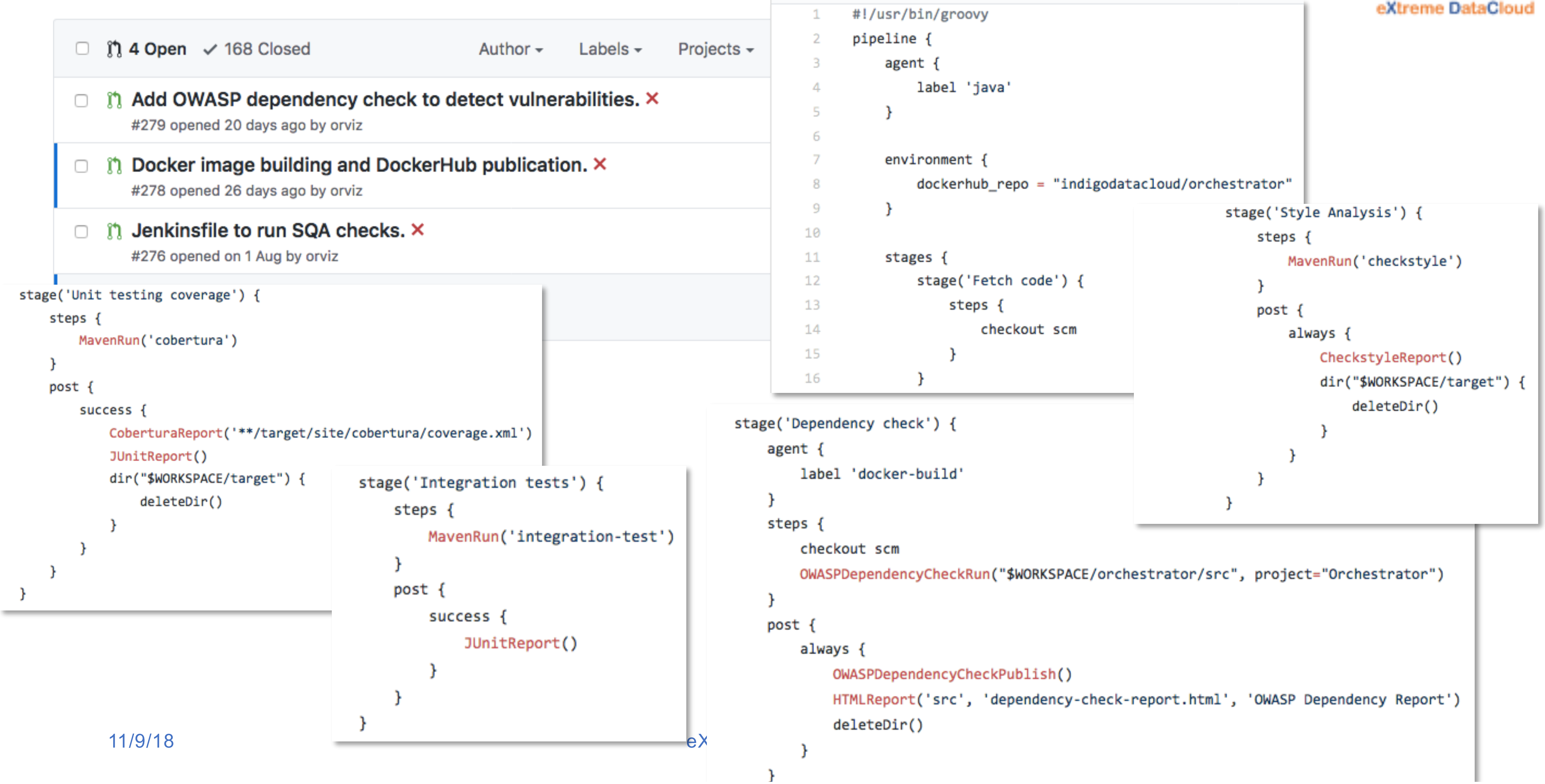
	Fetch code	OWASP Dependency check
Average stage times: (Average full run time: ~18s)	8s	5s
#29 Aug 09 12:40 No Changes	7s	21s failed
#28 Aug 09 12:38 No Changes	7s	1s
#27 Aug 09 12:36 No Changes	7s	4s failed

Build History · trend

find

●	#29	09-Aug-2018 12:40
●	#28	09-Aug-2018 12:38
●	#27	09-Aug-2018 12:36
●	#26	09-Aug-2018 12:12

Jenkinsfile – small example



□ 4 Open ✓ 168 Closed

□ **Add OWASP dependency check to detect vulnerabilities.** ✕
#279 opened 20 days ago by orviz

□ **Docker image building and DockerHub publication.** ✕
#278 opened 26 days ago by orviz

□ **Jenkinsfile to run SQA checks.** ✕
#276 opened on 1 Aug by orviz

```
#!/usr/bin/groovy
pipeline {
    agent {
        label 'java'
    }

    environment {
        dockerhub_repo = "indigodatacloud/orchestrator"
    }

    stages {
        stage('Fetch code') {
            steps {
                checkout scm
            }
        }
```

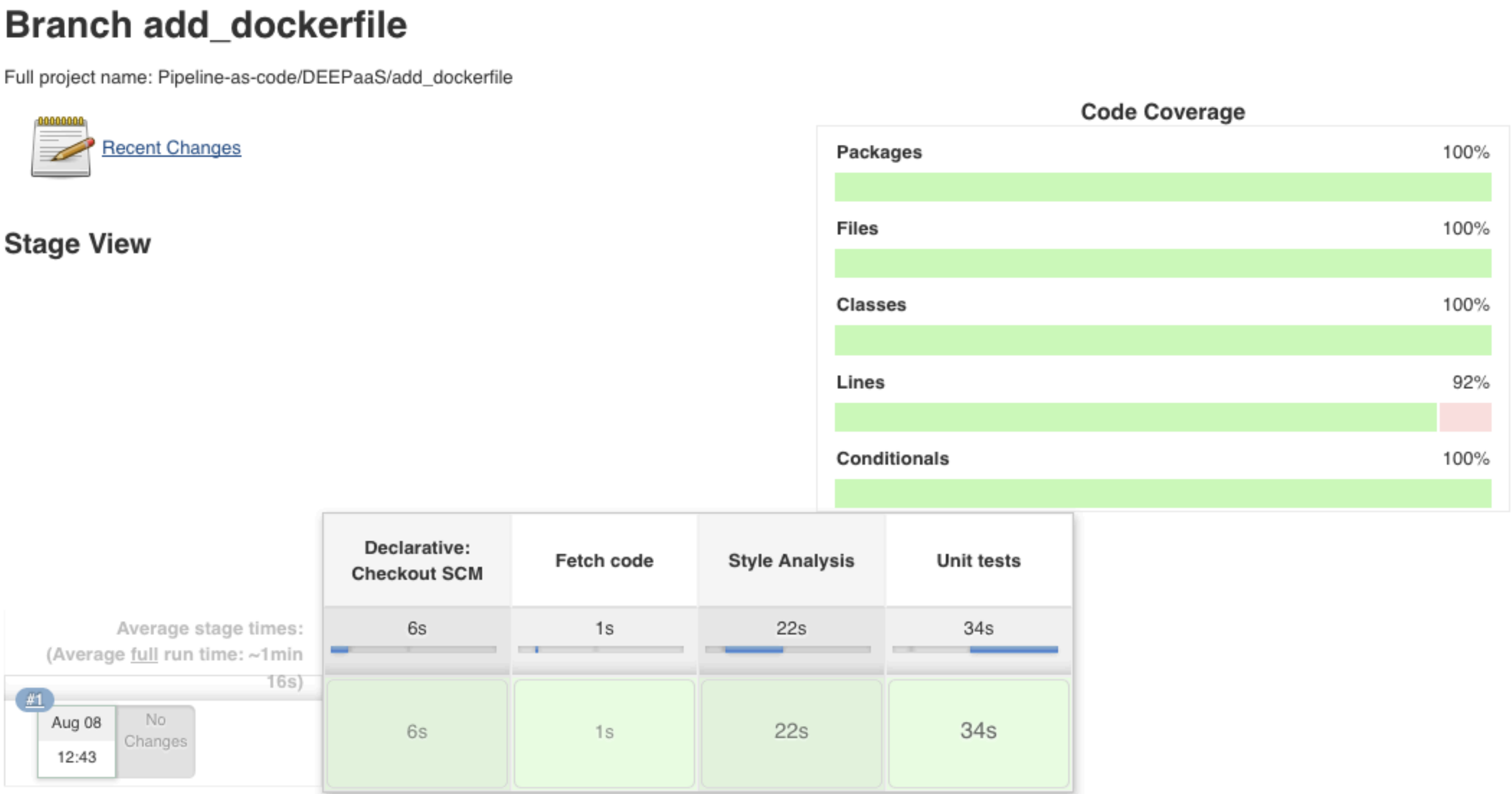
```
stage('Unit testing coverage') {
    steps {
        MavenRun('cobertura')
    }
    post {
        success {
            CoberturaReport('**/target/site/cobertura/coverage.xml')
            JUnitReport()
            dir("$WORKSPACE/target") {
                deleteDir()
            }
        }
    }
}
```

```
stage('Integration tests') {
    steps {
        MavenRun('integration-test')
    }
    post {
        success {
            JUnitReport()
        }
    }
}
```

```
stage('Style Analysis') {
    steps {
        MavenRun('checkstyle')
    }
    post {
        always {
            CheckstyleReport()
            dir("$WORKSPACE/target") {
                deleteDir()
            }
        }
    }
}
```

```
stage('Dependency check') {
    agent {
        label 'docker-build'
    }
    steps {
        checkout scm
        OWASPDependencyCheckRun("$WORKSPACE/orchestrator/src", project="Orchestrator")
    }
    post {
        always {
            OWASPDependencyCheckPublish()
            HTMLReport('src', 'dependency-check-report.html', 'OWASP Dependency Report')
            deleteDir()
        }
    }
}
```

Jenkinsfile – small example



Branch add_dockerfile

Full project name: Pipeline-as-code/DEEPaaS/add_dockerfile

Recent Changes

Code Coverage

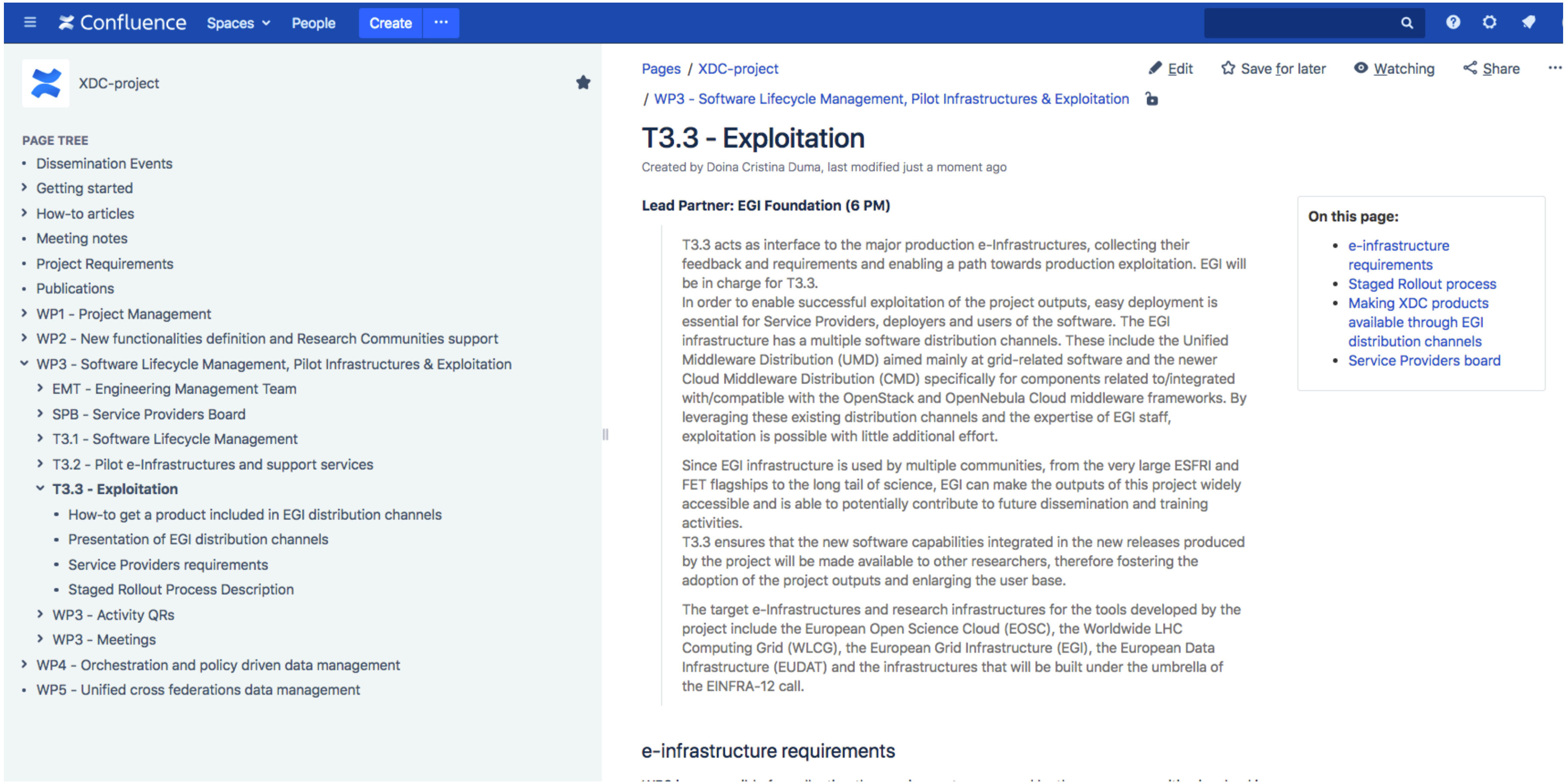
Packages	100%
Files	100%
Classes	100%
Lines	92%
Conditionals	100%

Stage View

	Declarative: Checkout SCM	Fetch code	Style Analysis	Unit tests
Average stage times: (Average full run time: ~1min 16s)	6s	1s	22s	34s
#1 Aug 08 12:43 No Changes	6s	1s	22s	34s

T3.3 - Exploitation

T3.3



XDC-project

PAGE TREE

- Dissemination Events
- Getting started
- How-to articles
- Meeting notes
- Project Requirements
- Publications
- WP1 - Project Management
- WP2 - New functionalities definition and Research Communities support
- WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation
  - EMT - Engineering Management Team
  - SPB - Service Providers Board
  - T3.1 - Software Lifecycle Management
  - T3.2 - Pilot e-Infrastructures and support services
  - **T3.3 - Exploitation**
    - How-to get a product included in EGI distribution channels
    - Presentation of EGI distribution channels
    - Service Providers requirements
    - Staged Rollout Process Description
  - WP3 - Activity QRs
  - WP3 - Meetings
- WP4 - Orchestration and policy driven data management
- WP5 - Unified cross federations data management

Pages / XDC-project / WP3 - Software Lifecycle Management, Pilot Infrastructures & Exploitation

T3.3 - Exploitation

Created by Doina Cristina Duma, last modified just a moment ago

Lead Partner: EGI Foundation (6 PM)

T3.3 acts as interface to the major production e-Infrastructures, collecting their feedback and requirements and enabling a path towards production exploitation. EGI will be in charge for T3.3.
In order to enable successful exploitation of the project outputs, easy deployment is essential for Service Providers, deployers and users of the software. The EGI infrastructure has a multiple software distribution channels. These include the Unified Middleware Distribution (UMD) aimed mainly at grid-related software and the newer Cloud Middleware Distribution (CMD) specifically for components related to/integrated with/compatible with the OpenStack and OpenNebula Cloud middleware frameworks. By leveraging these existing distribution channels and the expertise of EGI staff, exploitation is possible with little additional effort.

Since EGI infrastructure is used by multiple communities, from the very large ESFRI and FET flagships to the long tail of science, EGI can make the outputs of this project widely accessible and is able to potentially contribute to future dissemination and training activities.
T3.3 ensures that the new software capabilities integrated in the new releases produced by the project will be made available to other researchers, therefore fostering the adoption of the project outputs and enlarging the user base.

The target e-Infrastructures and research infrastructures for the tools developed by the project include the European Open Science Cloud (EOSC), the Worldwide LHC Computing Grid (WLCG), the European Grid Infrastructure (EGI), the European Data Infrastructure (EUDAT) and the infrastructures that will be built under the umbrella of the EINFRA-12 call.

On this page:

- e-infrastructure requirements
- Staged Rollout process
- Making XDC products available through EGI distribution channels
- Service Providers board

e-infrastructure requirements

T3.3 - Exploitation activities

- Getting requirements from Service Providers
  - Requirements mainly related to deployment and operations
    - Features requirements covered by WP2

- Presenting XDC products to Service Providers
- Collecting feedback from Service Providers

- Making XDC products available to EGI and e-infrastructures participants and users
  - Staged Rollout
  - Getting products pushed to distribution channels

T3.3 - Service Providers

- Service Provider ***Terms of Reference***
  - https://docs.google.com/document/d/1ExR9tKDI8whYVc7WKO7cDruSIDBoowiIGT8G6Dku2vs
- Service Providers ***members*** coming from
  - Project’s partners
  - INDIGO DataCloud Service Providers board
  - EGI participants
  - EOSC-hub partners
- ***Tools***
  - Mailing list: spb@extreme-datacloud.eu
  - Wiki: https://confluence.extreme-datacloud.eu/x/nQEB
  - Online meetings
- Initial ***requirements*** collection using an online form
  - https://goo.gl/forms/MteqIrRFO6ZHVWVA2

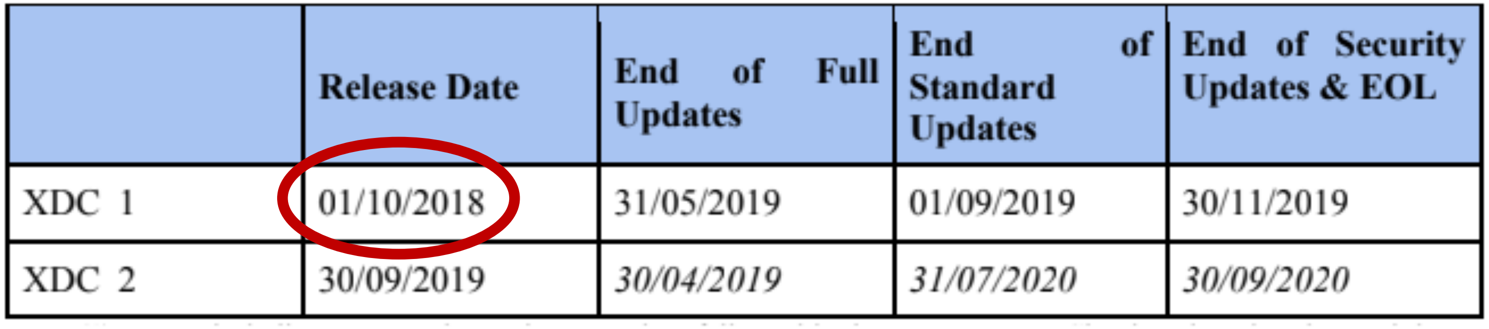
T3.3 - Exploitation status

- Done
  - SPB Terms of Reference validated by XDC PEB
  - Initial documentation and tools in place
  - Initial set of members registered
  - Requirements collection form v1 available for discussion #XDM-14
- Ongoing
  - Extending membership to external providers #XDM-11
  - Completing Wiki documentation #XDM-10
  - Jira/Confluence template for Staged Rollout reporting #XDM-9
  - Initial Requirements collection #XDM-12
  - Initial SPB meeting #XDM-13
  - Post-XDC Release meeting

Towards XDC-1

Schedule (1)

Initial Plan

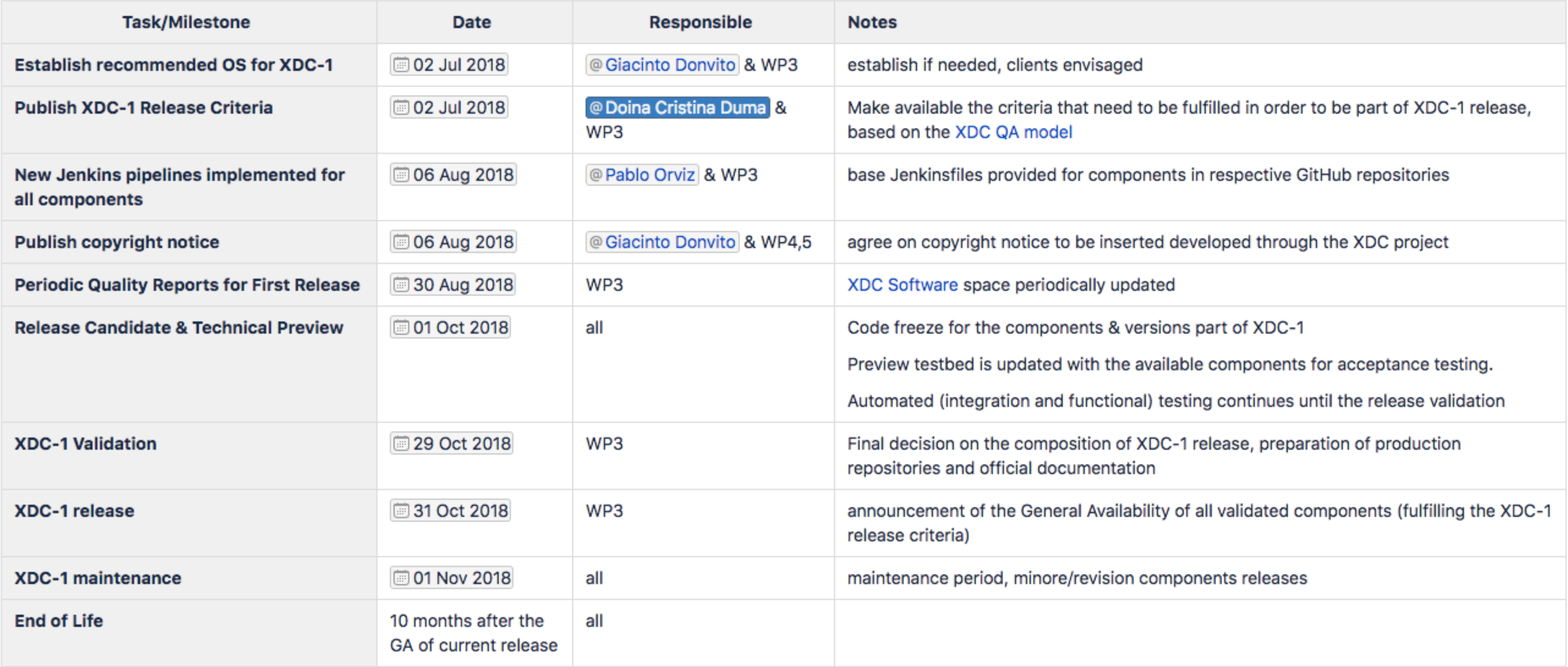


	Release Date	End of Full Updates	End of Standard Updates	End of Security Updates & EOL
XDC 1	01/10/2018	31/05/2019	01/09/2019	30/11/2019
XDC 2	30/09/2019	*30/04/2019*	*31/07/2020*	*30/09/2020*

Rationale:

- D3.2, Annual Report on Software Lifecycle Management and Pilot testbeds – **M12**
- M3.3, XDC reference releases - 1 (snapshot of repositories content with validated releases for each component) – **M12**

Schedule (2)



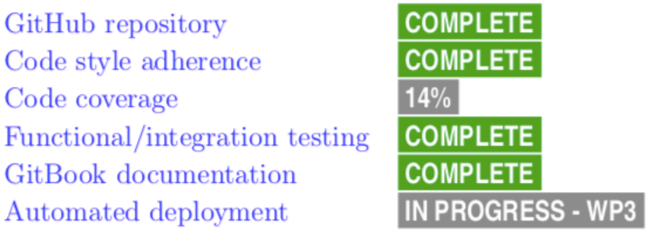
Task/Milestone	Date	Responsible	Notes
Establish recommended OS for XDC-1	02 Jul 2018	@Giacinto Donvito & WP3	establish if needed, clients envisaged
Publish XDC-1 Release Criteria	02 Jul 2018	@Doina Cristina Duma & WP3	Make available the criteria that need to be fulfilled in order to be part of XDC-1 release, based on the XDC QA model
New Jenkins pipelines implemented for all components	06 Aug 2018	@Pablo Orviz & WP3	base Jenkinsfiles provided for components in respective GitHub repositories
Publish copyright notice	06 Aug 2018	@Giacinto Donvito & WP4,5	agree on copyright notice to be inserted developed through the XDC project
Periodic Quality Reports for First Release	30 Aug 2018	WP3	XDC Software space periodically updated
Release Candidate & Technical Preview	01 Oct 2018	all	Code freeze for the components & versions part of XDC-1 Preview testbed is updated with the available components for acceptance testing. Automated (integration and functional) testing continues until the release validation
XDC-1 Validation	29 Oct 2018	WP3	Final decision on the composition of XDC-1 release, preparation of production repositories and official documentation
XDC-1 release	31 Oct 2018	WP3	announcement of the General Availability of all validated components (fulfilling the XDC-1 release criteria)
XDC-1 maintenance	01 Nov 2018	all	maintenance period, minore/revision components releases
End of Life	10 months after the GA of current release	all	

Before ….. Release Criteria

- Products
  - Do we know them all?
    - If not - Please use the JIRA XDM project and open tasks to WP3
  - Product Cards
    - see “How to fill….”
- Tracking
  - user requirements
  - Developments
  - *….releases*
- Agreement on
  - License
  - Copyright
  - OS Support
  - Packaging – rpms/debs or containers
- SQA reports definition

SQA Progress Status COMPLETE

85% done

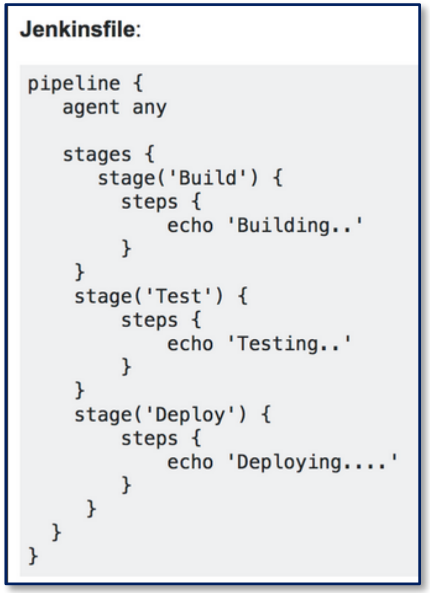


GitHub repository	COMPLETE
Code style adherence	COMPLETE
Code coverage	14%
Functional/integration testing	COMPLETE
GitBook documentation	COMPLETE
Automated deployment	IN PROGRESS - WP3

Release Criteria

1. Repository syncronisation
   - Do we miss someone? Yes!

- Code style specification – automated
  - Define threshold??
- Unit test coverage
  - Automated, threshold
- Functional & Integration testing
  - automated, what is missing
- Configuration Management
- Documentation
- Packaging – automated , even for dockers – deployment tests automated



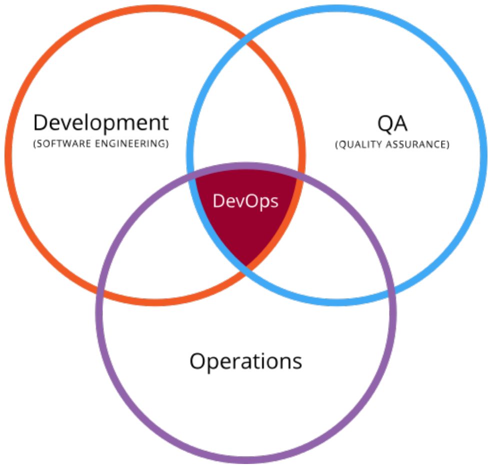
Jenkinsfile:

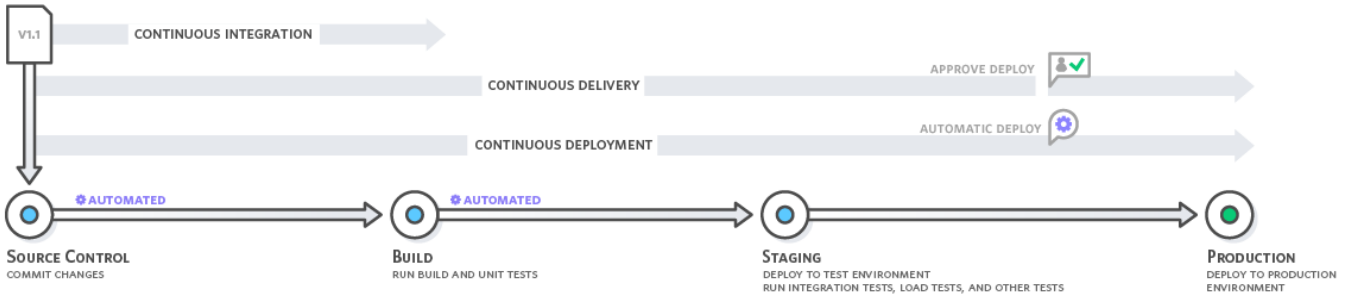
```
pipeline {
   agent any

   stages {
      stage('Build') {
        steps {
            echo 'Building..'
        }
    }
    stage('Test') {
        steps {
            echo 'Testing..'
        }
    }
    stage('Deploy') {
        steps {
            echo 'Deploying....'
        }
     }
  }
}
```

Developing and integrating an application: DevOps approach

DevOps culture - “set of practices that emphasize the collaboration and communication of both SW developers and IT professionals while automating the process of SW delivery and infrastructure changes” (source: wikipedia)

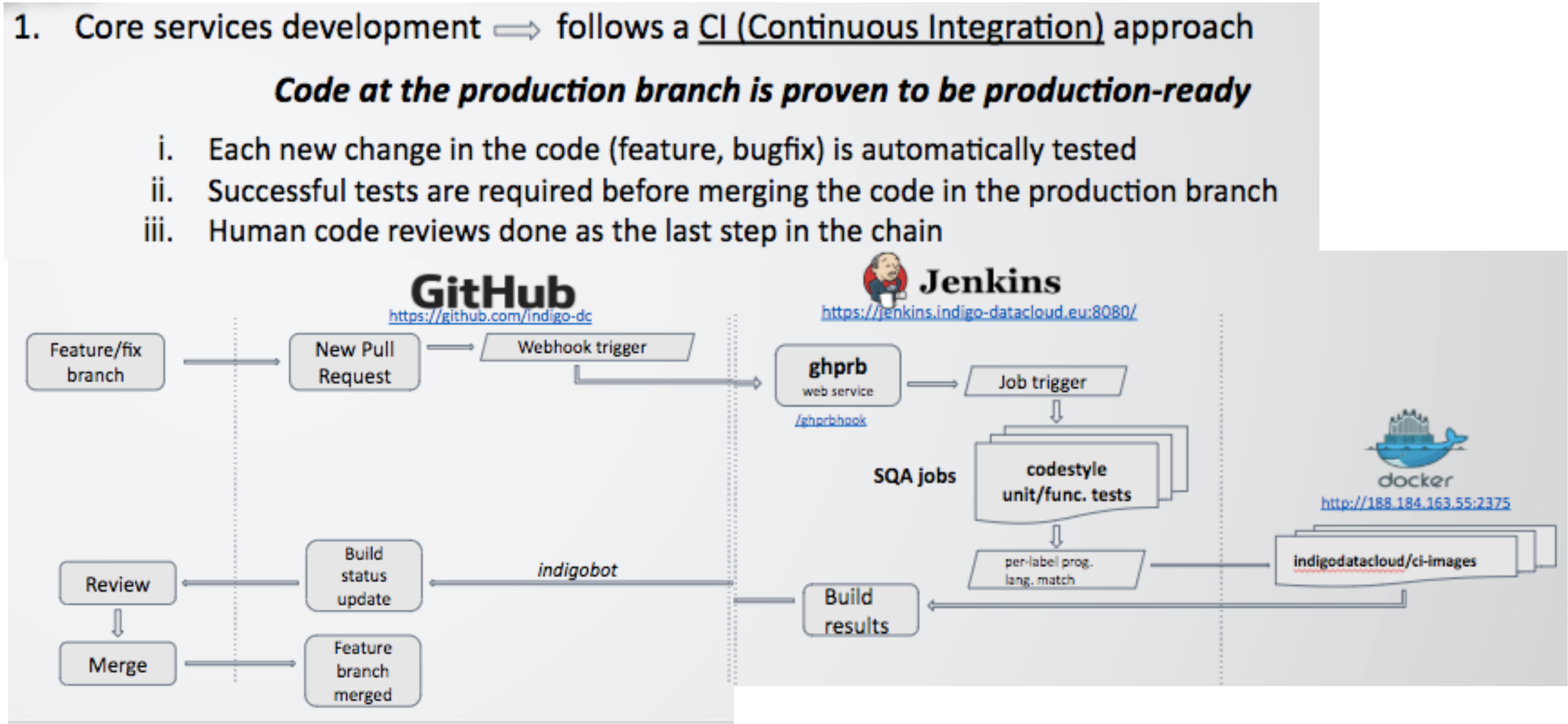
- **Dev**elopment and **Op**erations
  - traditionally distant → new features vs stability
  - QA procedures → maturity of SW trust/confidence
  - fast and frequent delivery → SW more reliable
- Approaches (automation)
  - Continuous Integration (CI)
  - **Continuous Delivery (CD)** *suitable for user community apps*
  - **Continuous Deployment (CDep)** *suitable for user community apps*





Courtesy Pablo Orviz

DevOps – Core services development



1. Core services development ⟹ follows a CI (Continuous Integration) approach

Code at the production branch is proven to be production-ready

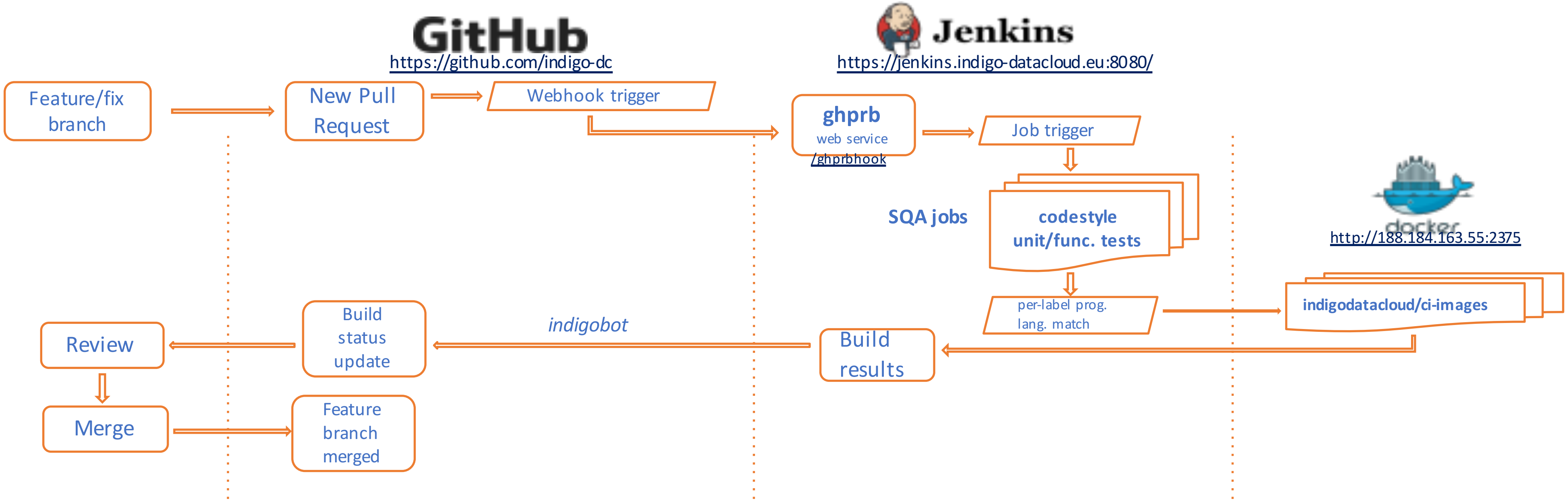
   i. Each new change in the code (feature, bugfix) is automatically tested
   ii. Successful tests are required before merging the code in the production branch
   iii. Human code reviews done as the last step in the chain

DevOps – Core services development

1. Core services development ⟶ follows a CI (Continuous Integration) approach

Code at the production branch is proven to be production-ready

i. Each new change in the code (feature, bugfix) is automatically tested
ii. Successful tests are required before merging the code in the production branch
iii. Human code reviews done as the last step in the chain



GitHub
https://github.com/indigo-dc

Jenkins
https://jenkins.indigo-datacloud.eu:8080/

Feature/fix branch

New Pull Request

Webhook trigger

ghprb
web service
/ghprbhook

Job trigger

SQA jobs

codestyle
unit/func. tests

docker
http://188.184.163.55:2375

per-label prog. lang. match

indigodatacloud/ci-images

Build results

indigobot

Build status update

Review

Merge

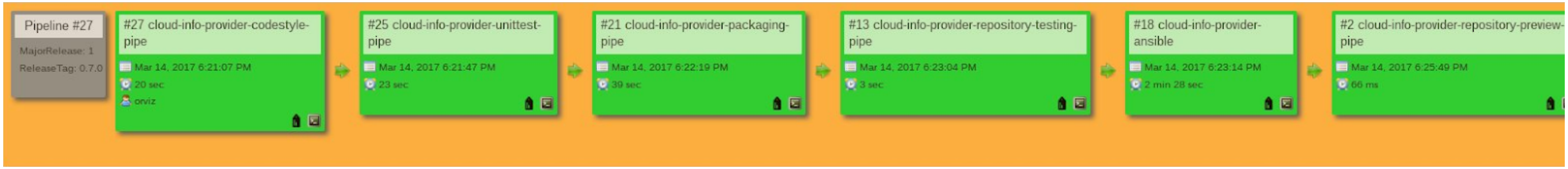
Feature branch merged

DevOps – Core services releases

2. Core services validation and release ⟶ conforms a CD (Continuous Delivery) scenario (pipeline)

    Software (packages, appliances) can be reliably released at any time

    i. Source code checks are re-ran (regression testing) and built
    ii. When releasing packages:
        1. RPMs/DEBs are created and uploaded to the *testing* repository
        2. Deployment is tested using available Ansible roles or Puppet modules
        3. RPMs/DEBs are moved to the *preview* repository - through a promotion/approval process -
    iii. When releasing containers (in progress)
        1. Docker image is built and uploaded to *indigodatacloud* DockerHub organization



Pipeline #27
MajorRelease: 1
ReleaseTag: 0.7.0

#27 cloud-info-provider-codestyle-pipe
Mar 14, 2017 6:21:07 PM
20 sec
orviz

#25 cloud-info-provider-unittest-pipe
Mar 14, 2017 6:21:47 PM
23 sec

#21 cloud-info-provider-packaging-pipe
Mar 14, 2017 6:22:19 PM
39 sec

#13 cloud-info-provider-repository-testing-pipe
Mar 14, 2017 6:23:04 PM
3 sec

#18 cloud-info-provider-ansible
Mar 14, 2017 6:23:14 PM
2 min 28 sec

#2 cloud-info-provider-repository-preview-pipe
Mar 14, 2017 6:25:49 PM
66 ms

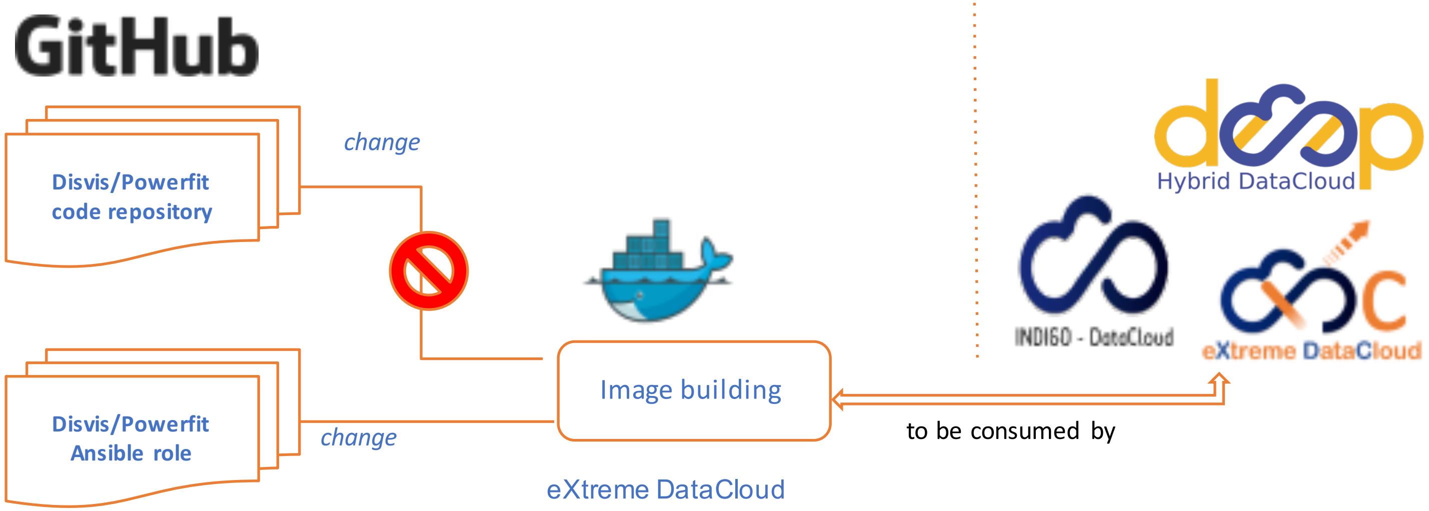
DevOps – user communities apps

3. Move a step forward and apply INDIGO insights on DevOps to the user community workflows
   - 2 approaches
     - Continuous Delivery (CD)
     - Continuous Deployment (CDep)
   - INDIGO prerequisites (INDIGO expert **= XDC Champion**)
     - A *TOSCA template* to describe the application composition
     - A Docker image to run the application, automatically deployed using *Ansible roles*
   - Actors involved
     - Source code repository (**GitHub *indigo-dc***): hosts application code
     - Continuous Integration Srv (**INDIGO Jenkins CI**): code testing and Docker image building
     - Container registry/catalogue (**DockerHub's *indigodatacloudapps***)
     - **INDIGO platform**: deploys and runs the application

Continuous Delivery (CD) on user community apps

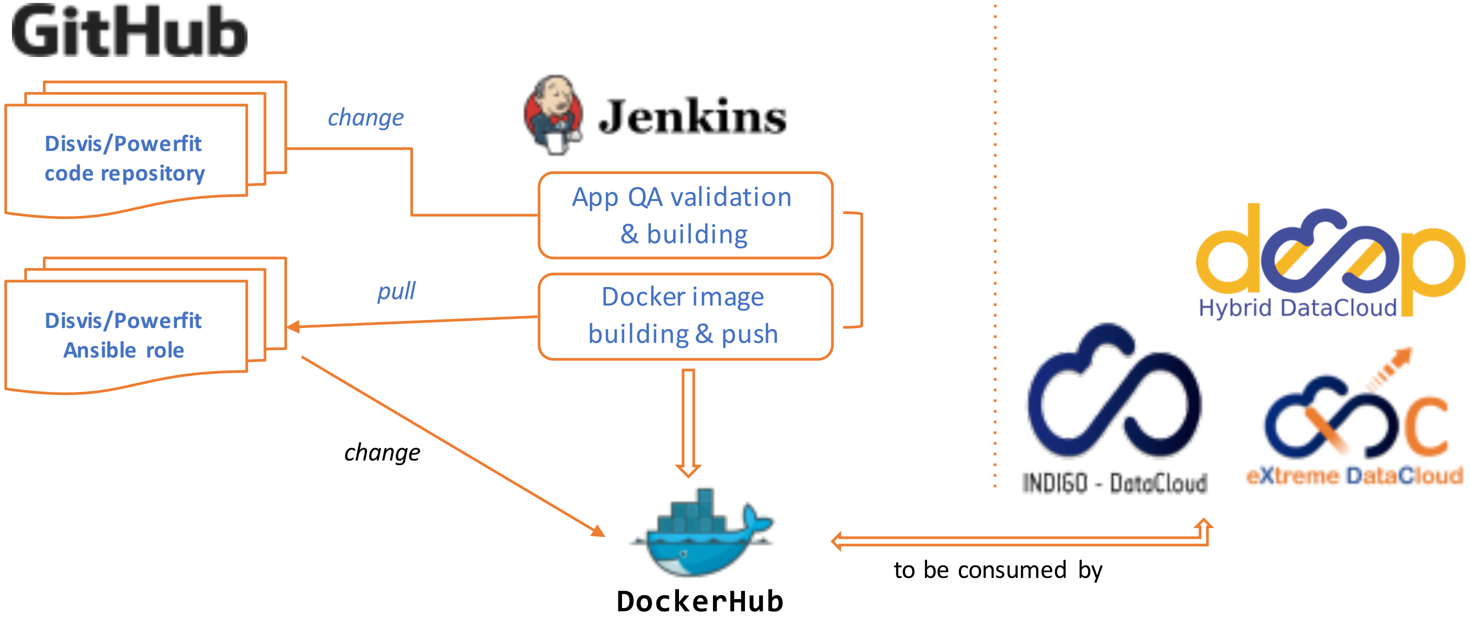
Target: ***Deliver fast, good-performing appliances***

- INDIGO-DataCloud used mainly Docker containers for user applications
  - Applications are deployed using **Ansible roles**, pushed to *indigodatacloudapps* DockerHub's organization
  - DockerHub <-> GitHub integration via *automated builds*
    - Images are automatically built when a change is done to the Dockerfile but not whenever a change is done in the application code repository
    - QA checks: the Ansible deployment itself



Towards a DevOps-like Continuous Delivery approach

Adding a CI service..


GitHub
Disvis/Powerfit code repository
change
Jenkins
App QA validation & building
Disvis/Powerfit Ansible role
pull
Docker image building & push
change
DockerHub
to be consumed by
deep Hybrid DataCloud
INDIGO - DataCloud
eXtreme DataCloud

- Docker image is built when a change has been made to the application code repository
- Application is QA-validated ⟶ DevOps compliant approach

Adopting DevOps on user apps: benefits & improvements

Benefits

- No user interaction
  - Application developer only needs to deal with its code
  - Application will be automatically deployed using INDIGO platform
- New features/bugfixes are available straightaway
- Application reliability enhanced
  - Applying QA procedures

Lines of improvement

- Jenkins plugin to interface directly with the PaaS orchestrator
  - Job definition would be simpler
- GitHub integrations: https://github.com/integrations
- DockerHub webhooks
  - React to automated builds of Docker images, triggering application submission to INDIGO platform, bypassing Jenkins (for apps not requiring testing)

XDC Releases Codenames

- sattelites, constellations, real codenames……

Thank

You !