



LFN Developer & Testing Forum

Tungsten Fabric Release Process

Marek Chwal, Szymon V. Gołębiewski



Agenda

- Structure of TF Community
- Project structure
- Release process principles
- How to become a TF member



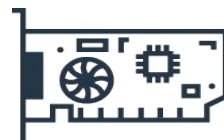
The Tungsten Fabric Project is an open source project to build all the necessary components for network virtualization and network security



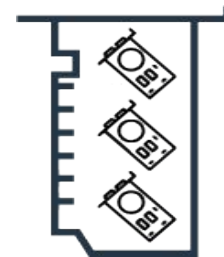
orchestrators or virtualization platforms



smartNIC devices



SR-IOV clusters



Cloud Platforms



3rd party solutions



Structure of TF Community

Technical Steering Committee

- elected from active community members
- consist of 10 seats
- fostering cross-project collaboration
- defining technical best practices and community norms
- responsible for setting high level architecture goals
- coordinating technical community engagement

Release Manager

- responsible for setting and monitoring release process
- coordinating overall transparency of performed work



Structure of TF Community

Project Technical Leaders

- provide direction for sub-project
- spokesperson for the project
- responsible for transparency of the project work

Feature Leads

- lead the work within the epic
- responsible for the transparency of epic artifacts



Structure of TF Community

Committers

- good knowledge about module
- ability to modify (“commit”) source code, documentation or other artifacts
decision maker on all matters for a project (design, code, patches)
- rights per repository/module

Contributors

- contributes to a project
- work with a project’s committers
- contributions could take the form of code, code reviews, or other artifacts



Project structure

Tungsten Fabric sub-projects¹:

- **TF Core** - umbrella project, covers almost all modules²
- **CI/CD** - continuous integration/delivery tasks
- **Documentation & Training** - materials, release notes, etc
- **TF Operator** - well defined separated scope of the product

1. <https://wiki.tungsten.io/x/6AELAQ>
2. <https://wiki.tungsten.io/x/0oVdAQ>



Release process - why & what for?

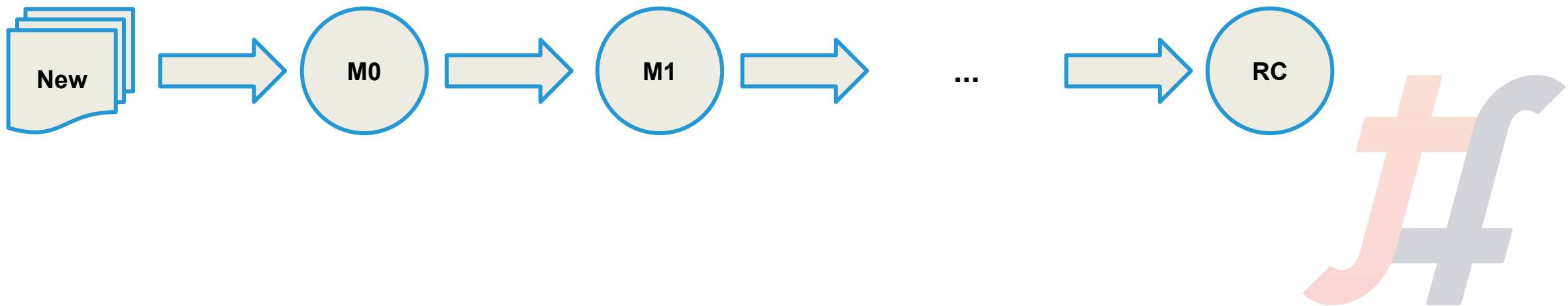
- Plan and schedule
- Structured and standardized way of presenting partial results and final output
- Checklist of expectations
- Transparency

Release process defines the Community!



Release milestones

- Acceptance criteria for a well-defined piece of work
- Checklist for each stage
- Clear expectation of partial result on a particular stage



Release Process - M0

- Declaration of participation in the simultaneous release
- Blueprint proposal represented as a Jira EPIC ticket
- Feature lead assigned to Jira Epic ticket
- The initial version of Increment Blueprint stored on github release folder
- The blueprint presented, discussed and approved by related committers
- Release Scope Overview updated accordingly
- Dependencies between projects/components/areas identified

Declaration of participation in the release, initial scope of release



Release Process - M1

- All dependencies between existing functionality and requested change/feature discussed and agreed by PTL and related approvers
- Jira Epic ticket broken down into stories in related Jira project (backlog, plan) - presented and discussed on TWS meeting
- The final version of Blueprints, Release Scope and Release Plans for participating projects discussed and approved by approvers

Dependencies between modules, new functionalities, release plan



Release Process - M2

- Technical design fully provided, agreed (documentation on the Confluence/github/tf-spec available)
- Jira Epic ticket updated with link to technical design
- Documentation started
- Feature tests started
- Feature/Functionality Freeze - Technical Design approved by relevant approvers (Modules/Committers)

Feature/Functionality technical design freeze



Release Process - M3

- External API available for beta-tests
- Jira Epic ticket updated with information about API availability (documentation link - tf repo)
- Documentation in progress
- Tests in progress
- API Freeze confirmed by Feature Lead (comment on Jira Epic ticket with repo/commit ID link)

API freeze (beta available)



Release Process - M4

- All functionality and APIs available for testing
- Jira Epic ticket updated with information about final API documentation, executed UI tests documentation
- Documentation provided - confirmed by Documentation project PTL
- Code freeze confirmed by Feature Lead (comment on Jira Epic ticket) - only bug fixing allowed

Code freeze - only bug fixing allowed



Release Process - RC

- Release candidates agreed, approved, tested
- System tests conducted, quality confirmed by CI/CD PTL
- Final documentation provided, reviewed, approved by Documentation PTL
- Marketing information provided to Marketing Advisory Council
- Release branch cut off - branch stabilization from now on
- Release Candidates freeze, confirmed by PTLs and TSC

Release candidates freeze - branch stabilization



Release Process - Deployment

- Tested Release branch
- Quality confirmed
- Known errors list prepared
- Release notes prepared
- Date of official publish agreed and approved by PTLs and TSC

Official Release Issued



Important links

- [Wiki of Tungsten Fabric release process](#)
- [Tungsten Fabric jira](#)
- [General writing style guidelines](#)
- [Cheat sheet for reStructuredText](#)



TF Release process

Release Process Proposals

Utworzone przez Casey Cain, ostatnia modyfikacja przez Marek Chwal przed momentem

- [The purpose of the release process](#)
- [Prerequisites](#)
- [Release Process](#)
 - [Milestones goal and acceptance criteria](#)
- [Release Cadence, Life Time & Support](#)

The purpose of the release process

The Release Process is dedicated to creating a plan and then following and monitoring that plan to deliver an increment to the product. It is extremely important, especially in a community where there are lots of developers from different companies, countries, to have a common understanding of what and how we would like to achieve. The release plan will help us to coordinate our efforts to achieve agreed goals and to deliver proper, coded, tested, and working as an expected increment of product. To achieve that, we need to act in accordance with agreed rules, time schedule, and milestones.

Prerequisites

All who would like to attend community effort and contribution must follow bellow prerequisites:

- [Read the Getting Started as a Contributor page](#) (including Registration and User Creation, environment installation)
- [Check/assign to one of the existing projects](#) or create a [proposal for a new project](#)

Release Process

The unit piece of functionality/work which can be included in the release is represented by the Epic ticket in Jira. The release process consists of several Milestones in the pipeline of epics with clearly defined acceptance criteria. One can treat these acceptance criteria as a checklist that has to be followed to say that particular epic achieved this Milestone. Overall flow is outlined as follows:

1. [Invitation of proposals for features, enhancement requests](#) - approved till M0
2. [Blueprints/Design/Architectural approval cut-off date](#) (4 months before release date)
3. [Code delivery cut-off date](#) (2 months before release date) - till M4
4. [Branch cut-off date](#) (1 month before release date)






Release process in Jira

Release TFF board

Kanban board

QUICK FILTERS: [Only My Issues](#) [Recently Updated](#)

SELECT FOR TSC DISCUS... 2 M0 ACHIEVED 16 M1 ACHIEVED 0 M2 ACHIEVED 0 M3 ACHIEVED 0

SELECT FOR TSC DISCUS... 2	M0 ACHIEVED 16	M1 ACHIEVED 0	M2 ACHIEVED 0	M3 ACHIEVED 0
<p>TFF-15 TF Operator Framework</p> 	<p>TFF-1 Provision for a knob (per service chain) to prevent reset of AS_PATH and maintain it</p> 			
<p>TFF-17 New website for TF</p> 	<p>TFF-3 Ubuntu 20.04 Support</p> 			
	<p>TFF-4 Charmed OpenStack Ussuri Support</p> 			

Source: <https://jira.tungsten.io/secure/RapidBoard.jspa?rapidView=5&projectKey=TFF>



Release process in Jira

TF Features / TFF-2

Collapsed Spine L2/3 fabric

Edit Comment Assign More Backlog Withdrawn Workflow

Details

Type:	Epic	Status:	RELEASE CANDIDA... (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	R2011
Labels:	None		
Epic Name:	Collapsed Spine L2/3 fabric		

Description

Collapsed Spine L2/3 fabric

ReleaseNotes: https://www.juniper.net/documentation/en_US/contrail20/information-products/topic-collections/release-notes/topic-150638.html#jd0e54

Attachments

Drop files to attach, or browse.

Source: <https://jira.tungsten.io/projects/TFF/issues/>



General writing style

Writing style

This section describes general writing style guidelines that documentation contributors should follow to ensure consistency throughout all technical publications.

- **General writing guidelines**
 - Use standard English
 - Write in active voice
 - Use the present simple tense
 - Write in second person
 - Use appropriate mood
 - Keep sentences short
 - Avoid ambiguous titles
 - Be clear and concise
 - Write objectively
 - Describe the most common use case first
 - Do not humanize inanimate objects
 - Write positively
 - Avoid prepositions at the end of sentences
 - Do not overuse this, that, these, and it
 - Do not split infinitives
 - Avoid personification
 - Do not use contractions
 - Eliminate needless politeness
 - Use consistent terminology
 - Use spelling and grammar checking tools
- **Word choice**
- **Punctuation**
- **Headings**
 - **General guidelines**

Source: [General writing style guidelines](#)



reStructured Text

RST conventions

Tungsten Fabric documentation uses reStructuredText (RST) markup syntax with Sphinx extensions.

RST is a powerful and straightforward markup language that, in combination with Sphinx, provides a wide range of facilities for intelligent and appealing documentation creation. It uses simple and implicit syntax to introduce a variety of content elements such as titles, code blocks, vertical lists, and many others. All the source content formatted using RST is stored in files with the `.rst` extension.

To keep the documentation format consistent, follow the guidelines included in this chapter for all the RST source content. When in doubt, use simpler formatting.

Note

All the examples provided in this chapter are for illustration purposes only and cannot be regarded as pieces of true technical information.

- General guidelines
 - Lines length
 - Space and tab characters
 - Indentation
- File naming and structure
 - File naming conventions
 - Directory structure
- Titles
- Inline elements
 - Application
 - Code
 - Command
 - File name and path
 - GUI element



How to join TF Community

1. **Create an account** at Linux Foundation
myprofile.linuxfoundation.org
2. **Login** to jira.tungsten.io and wiki.tungsten.io so your **account will be populated** to those systems
3. **Go to** tungsten.io/community/ to see community **meetings calendar** and learn about **different ways** you can connect with the community
4. **Join** community meetings, slack channels and mailing lists
5. **Start contributing**



Join us!



Marek Chwal
Release Manager
Tungsten Fabric, **codilime**
CREATING VALUE

marek.chwal@codilime.com



Szymon V. Gołębiewski
Documentation & Training PTL
Tungsten Fabric, **codilime**
CREATING VALUE

szymon.golebiewski@codilime.com