

Eclipse oniro


Oniro - an open-source starter for
fast-paced IoT environments -
project retrospect

Speaker: Sebastian Serewa
Technical Product Manager

Guest: Ettore Chimenti
<http://www.seco.com>

Date: 2022-02-05



- 
- **A bit of history**
 - **How do we work?**
 - **Our product-driven approach**
 - **Systems, work packages, processes**
 - **Our lessons learned**
 - **What's next?**
 - **A partner's impressions**

▶ HISTORY – WHO WE ARE

- ~18-month young/old project
- Partner-driven organization
- Part of Eclipse Foundation since 2021
- Team members scattered across EU & India (core team ~20 members, community ~100 members)
- Multicultural environment: we live in ~16 different countries, talking natively different languages, yet, working full time on daily basis toward a common goal

▶ WHAT IS ONIRO?

- Oniro is an Eclipse Foundation's project focused on reducing fragmentation in the consumer and IoT industry, led by a working group of which Huawei (organization) is an active member.
- We are a FOSS project
- Oniro project's license: Apache-2.0

▶ HOW DID IT START?

- Huawei, as one of the funding members, made an initial investment into this project together with other members
- Other funding partners: Linaro, Noi, Seco, Synesthesia
- Each partner in this working group has the same & equal voice & rights
- From start we are focused on:
 - ✓ Licenses clearance / IP compliance
 - ✓ Transparency
 - ✓ Upstream-first principle

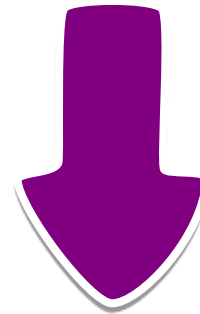
▶ WORK ORGANIZATION #1/3

We believe in Scrum and apply as much of it as it is possible.

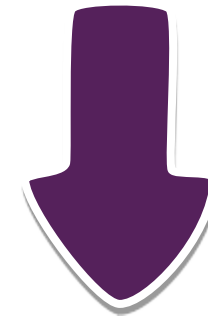
Empirical process control



Transparency



Inspection



Adaptation

▶ WORK ORGANIZATION #1/3

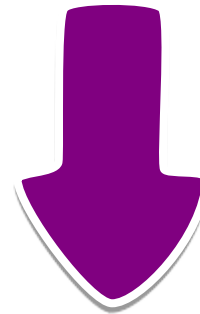
We believe in Scrum and apply as much of it as it is possible.

Empirical process control



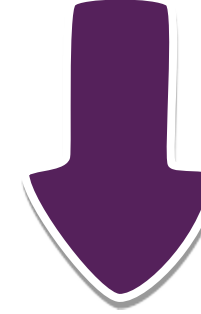
Transparency

- transparent information flow
- open work culture
- project vision, releases, DoDs



Inspection

- recurring backlog reviews
- peer reviews
- Demoing and prod. reviews



Adaptation

- retro meetings
- daily/weekly meetings
- monitoring

▶ WORK ORGANIZATION #2/3

What we took from Scrum?

- Events (review, planning, retro)
- Timeboxing
- “Weekly” instead of “Daily” (we are way too diverse!)
- Velocity measurements, though, still in hours rather than in points :/
- self-organizing, self-learning, self-adjustment
- Value-based prioritization
- ... iterative development

WORK ORGANIZATION #3/3

- Writing points (2 mins, try to not duplicate) - 4 votes (2 mins, 2 for Went Well, 2 for To Improve/Blockers)

🔍 Search Sort by order ▾ 🗃️ 📄 Add ▾ ➔ S

Went Well	To Improve	My Main Blockers	Action Items
<p>+ We have done 803h even though severely reduced workforce and hot holiday season. 🗳️ 5 🗳️ 0</p> <p>We're opening for risc-v architecture (e.g. toolchain support). 🗳️ 3 🗳️ 0</p> <p>First visible progress on blueprints 🗳️ 2 🗳️ 0</p> <p>Smooth loading / planning phase. 🗳️ 1 🗳️ 0</p> <p>Good progress with OTA and its architecture understanding 🗳️ 0 🗳️ 0</p> <p>We streamline with processes and activities around them making development more pleasant. 🗳️ 2 🗳️ 0</p> <p>Code review done over the whole sprint, no big batch at the end 🗳️ 1 🗳️ 0</p> <p>Started cross-domain work, eg. security review for OTA 🗳️ 1 🗳️ 0</p>	<p>+ Sometimes another unplanned "extra" tasks appear deviating deliverables from the current sprint scope. 🗳️ 7 🗳️ 0</p> <p>On features delivery and visibility for the closest sprints (e.g. what is going be delivered in 1 or 2 or 3 sprints). 🗳️ 3 🗳️ 0</p> <p>Tasks are too large (sometimes 40+ hours) making whole process imprecise and unpredictable (in terms of timeframe & features landed). 🗳️ 2 🗳️ 0</p>	<p>+ </p>	<p>✓ [monitoring] Zero-based budgeting, being more assertive on new/extra features taking, trust our judgements @team</p> <p>✓ [monitoring] 1. To make tasks/ decomposition as granular as possible, however, it's not always possible. 2. possibly to switch into points instead of hours</p> <p>To consider more pair-work, team up - so there is no single "point of failure" that may break any features delivery in the future. @team, to be addressed partially at Eclipse via projects structure (maintainers, developers etc)</p> <p>✓ Missing visibility of some team members availability (eg. L1 work) @Adam</p>

▶ OUR PRODUCT-DRIVEN APPROACH

Project mindset:

- Scope
- Time
- Budget

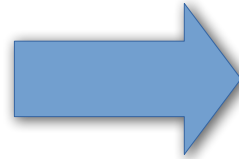
▶ OUR PRODUCT-DRIVEN APPROACH

Project mindset

- Score

- Time

Not applicable to us



Product mindset:

- Users adoption / retention
- Revenue
- Cost savings
- More creativity
- Number of designs

▶ DEVELOPMENT SYSTEMS

	Pros	Cons (*)
GitLab	<ul style="list-style-type: none">- de facto a standard for s/w development- a tool very close to development- REST API available	<ul style="list-style-type: none">- missing custom fields feature e.g. lists, extra value placeholders- missing custom, „enforced” workflows
OpenProject	<ul style="list-style-type: none">- easiness to implement new workflows- easy to manage users' rights- REST API available	<ul style="list-style-type: none">- another system to use along with GitLab- disjoint from our mainstream development

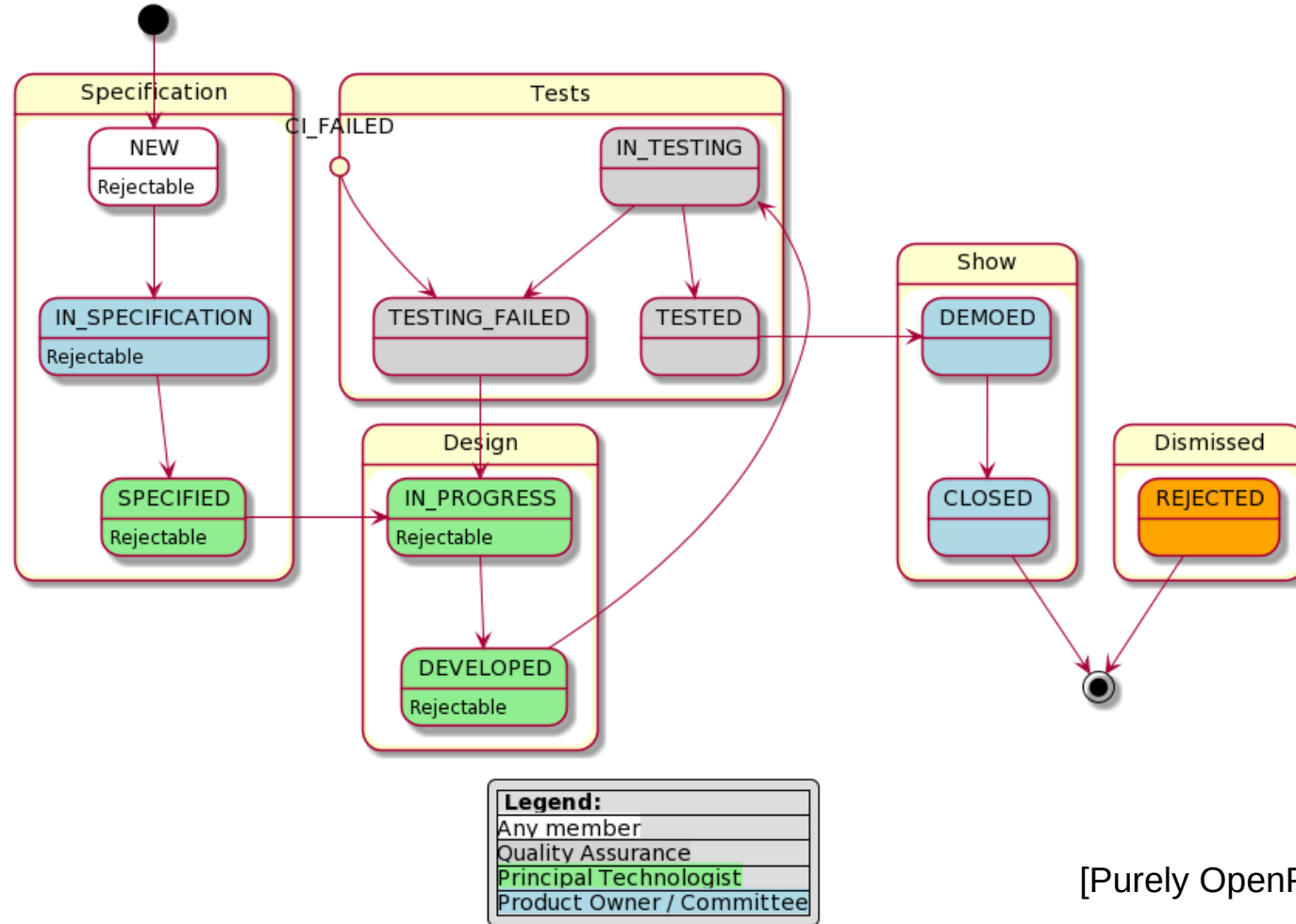
(*) - subjective, the most missing features for us

A few of Python modules used:

- <https://github.com/Flying-Free/pyopenproject> (OP API handling)
- <https://gitlab.com/python-gitlab/python-gitlab> (GitLab API handling)
- <https://matplotlib.org/> (data visualisations, graphs drawing)
- <https://numpy.org/> (computations)
- base: requests, regex, json etc

▶ WORK PACKAGES #1/3

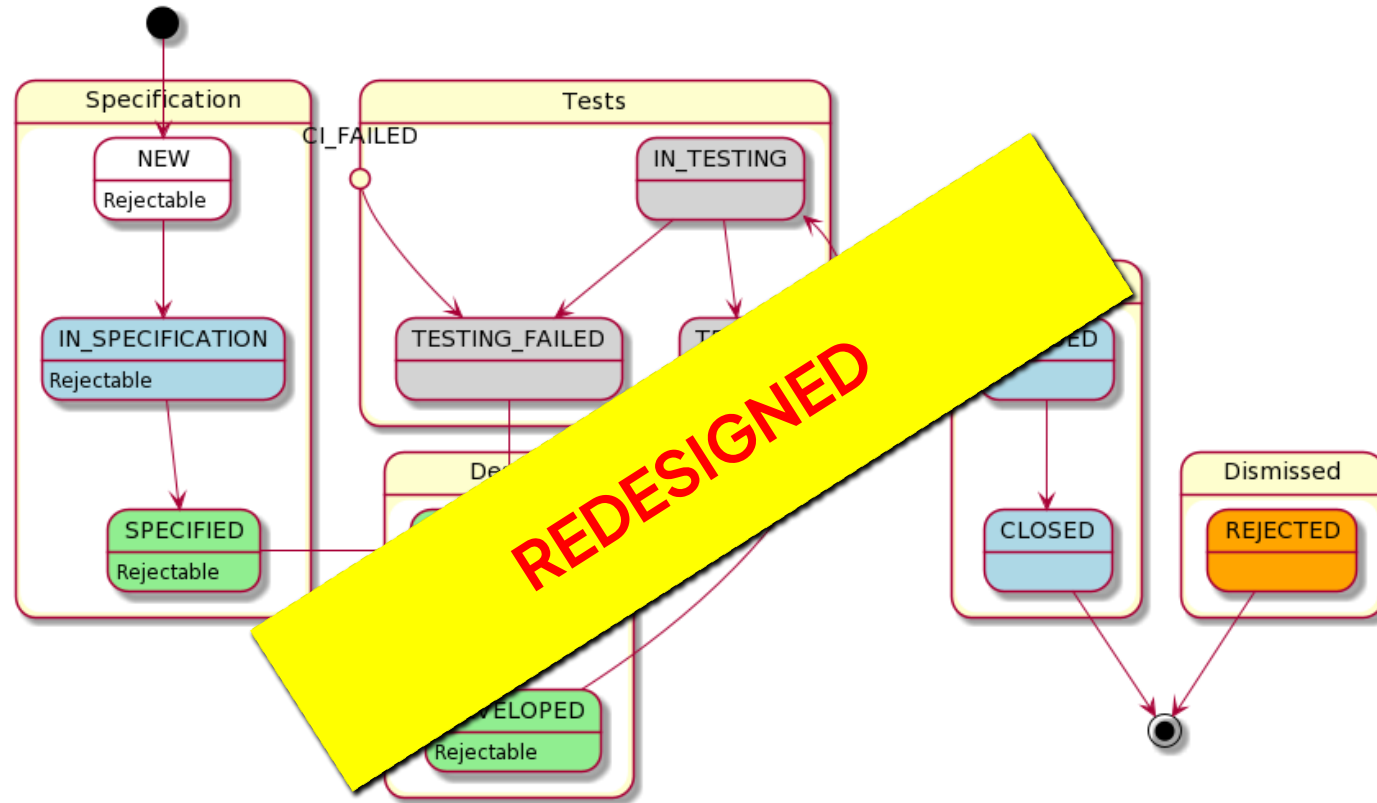
Lean requirements flow



[Purely OpenProject-based]

▶ WORK PACKAGES #1/3

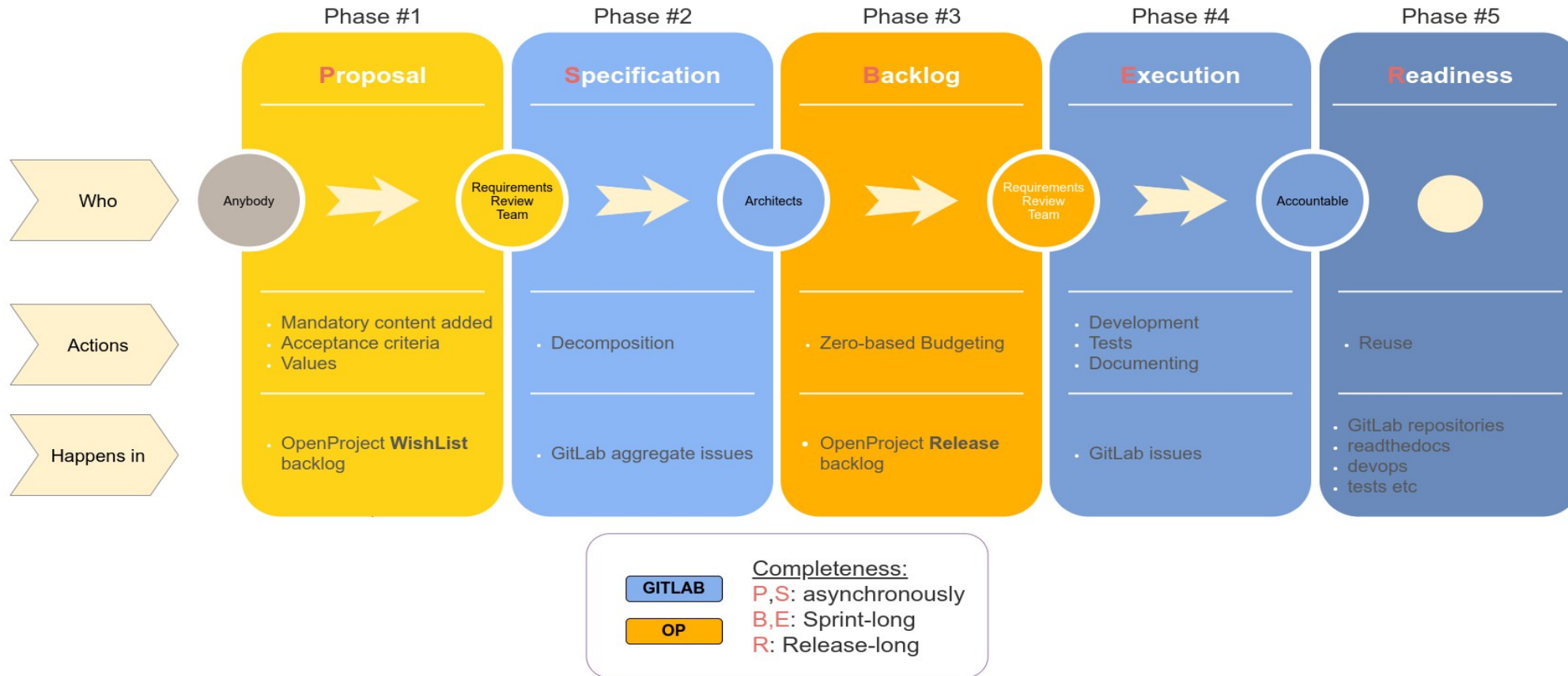
Lean requirements flow



Legend:
Any member
Quality Assurance
Principal Technologist
Product Owner / Committee

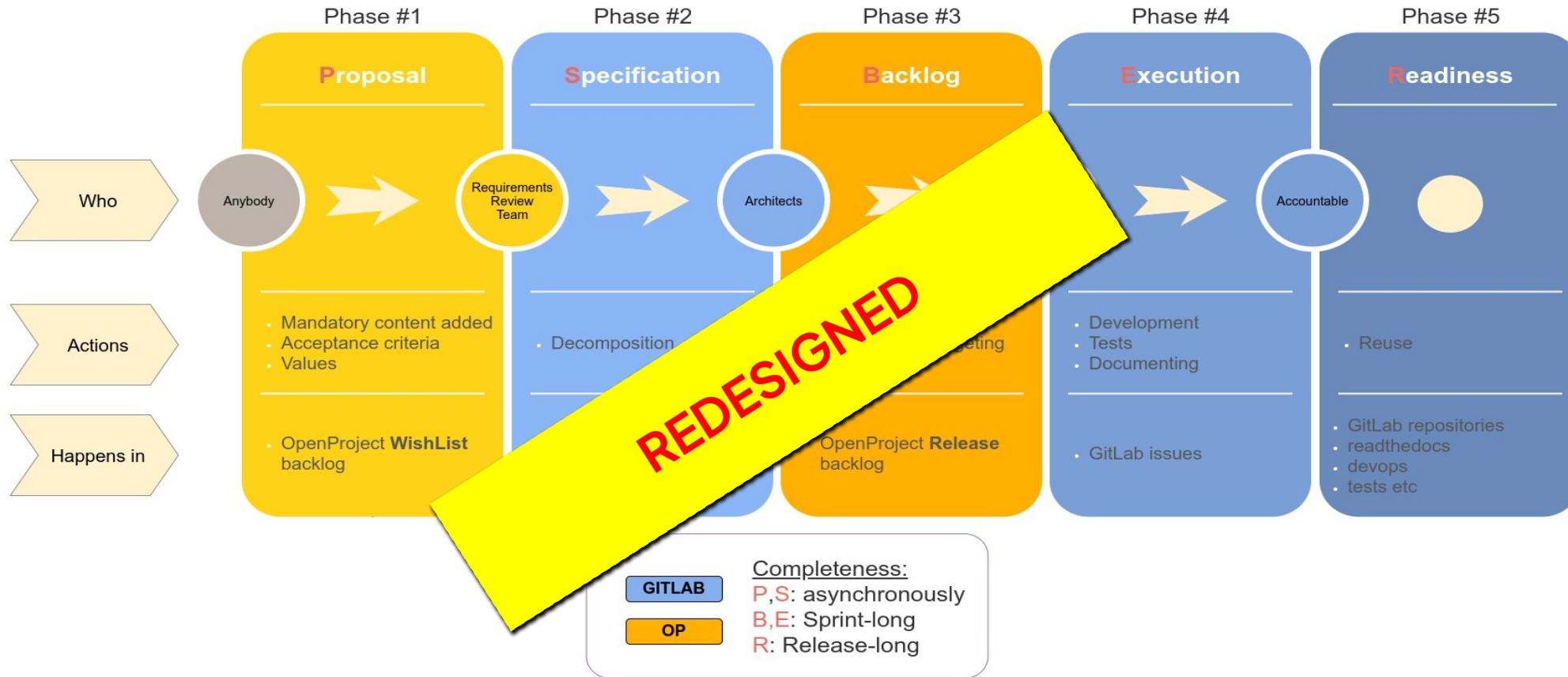
[Purely OpenProject-based]

▶ WORK PACKAGES #2/3



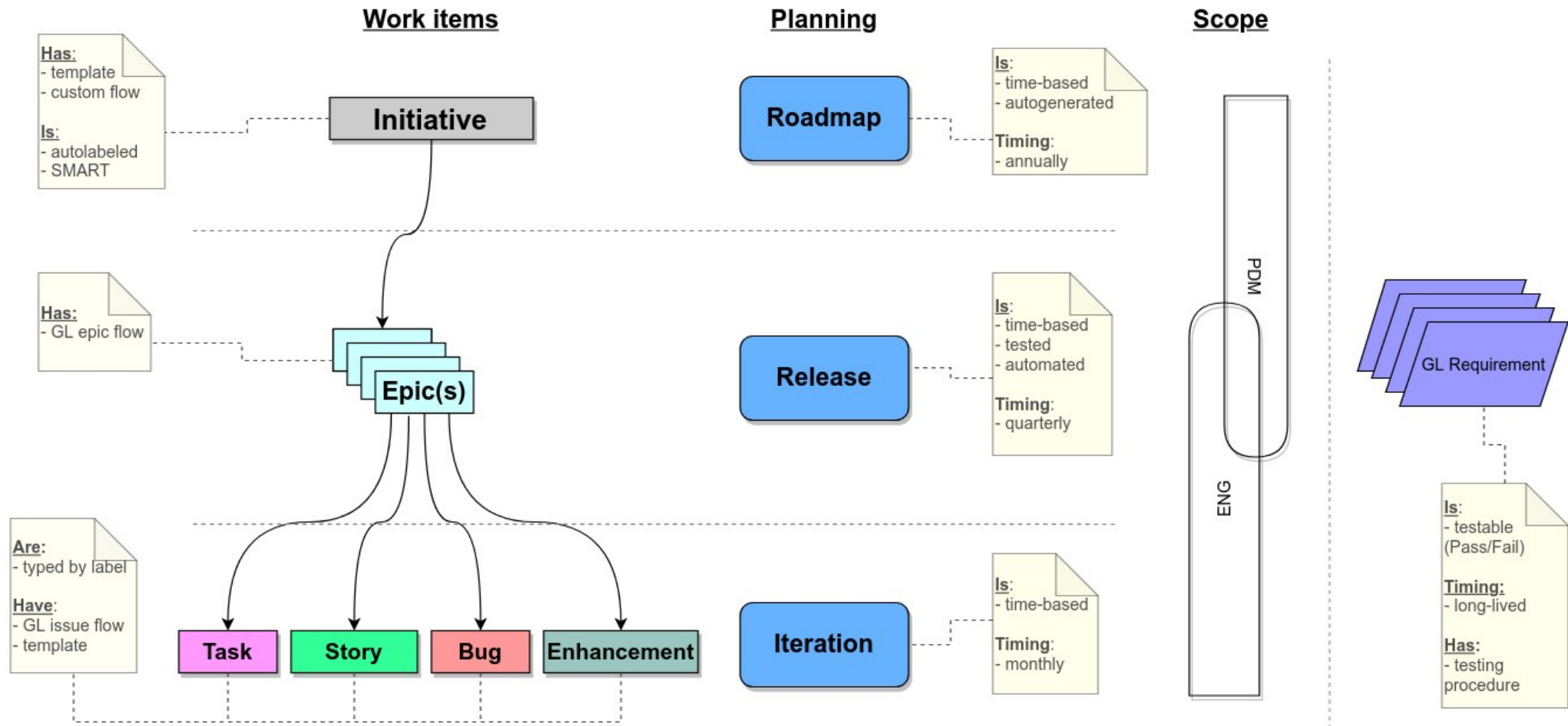
[Mix of OpenProject & GitLab]

▶ WORK PACKAGES #2/3



[Mix of OpenProject & GitLab]

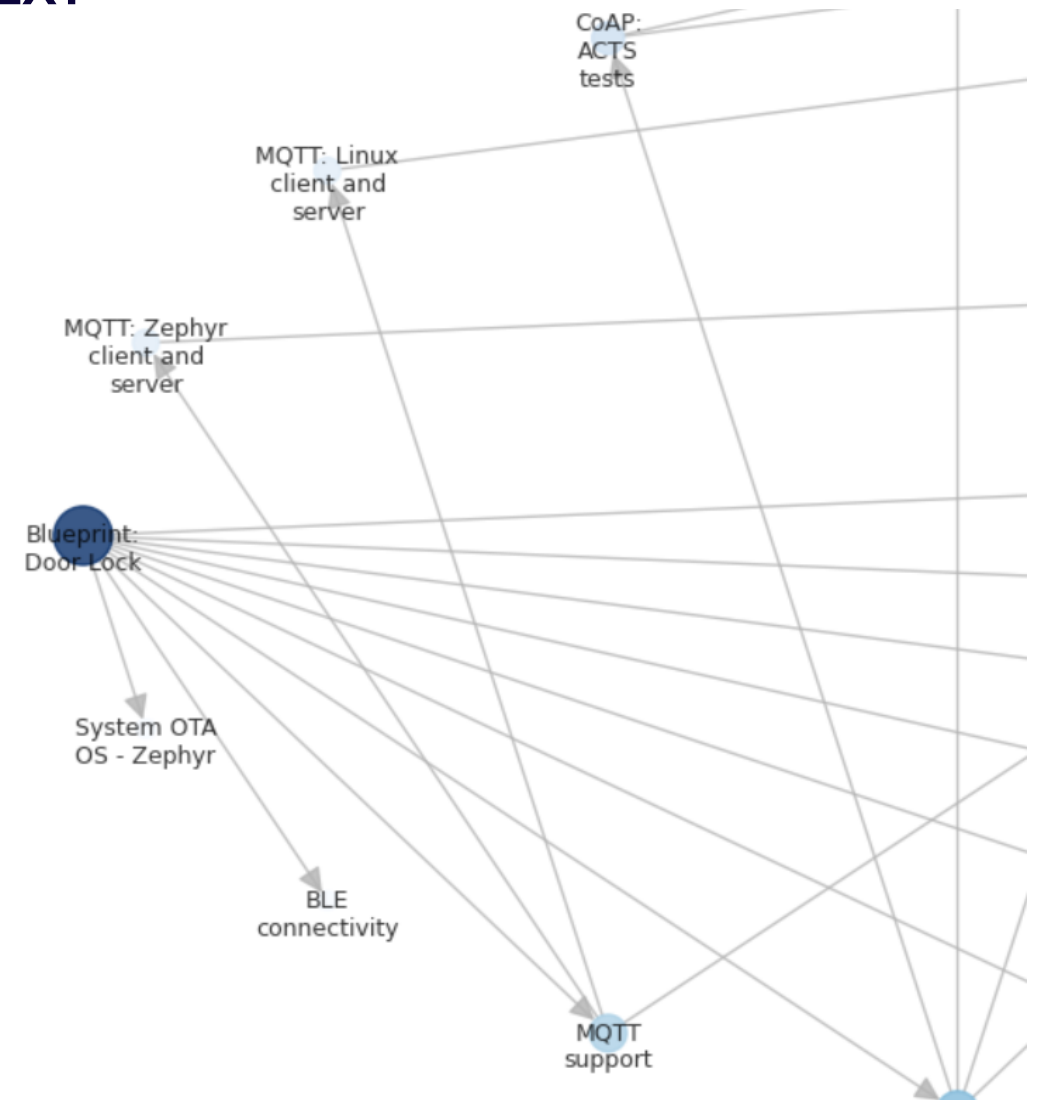
▶ WORK PACKAGES #3/3



[Based on: "Multi-team planning with GitLab Ultimate"]

▶ SOME OF IDEAS WE CAME UP WITH IN THIS CONTEXT (PROTOTYPES)

- „Labels hygiene” and the extension of the idea – e.g. tests, administrative work
- „Auto labels” assigned based on issue title & content
- Dependency graph drawing based on focused architectural part – here „Door lock” Blueprint.

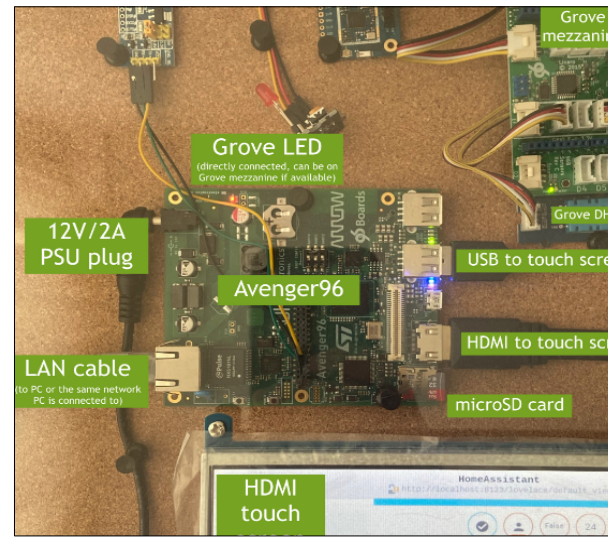


▶ BLUEPRINT as MVP

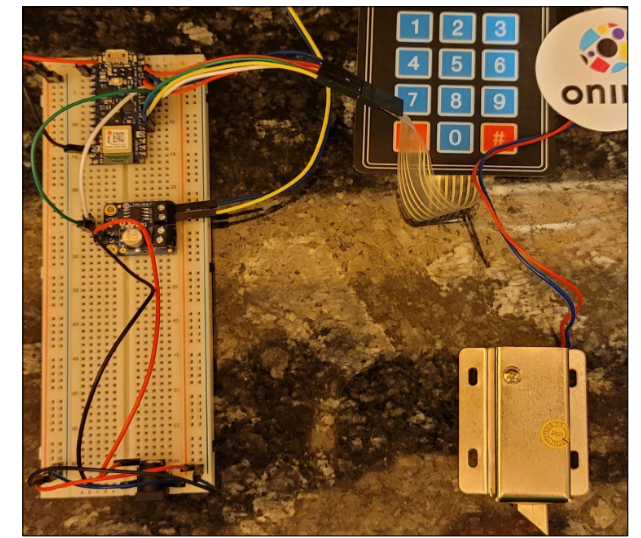
We have 4 blueprints:

- Smart Panel blueprint
- DoorLock blueprint
- Transparent Gateway blueprint
- Vending Machine blueprint

Anyone can join and extend with new blueprints. We see this as a way of the system adoption and growth.



Smart Panel blueprint



DoorLock blueprint



Transparent Gateway blueprint



Vending Machine blueprint

[Visit: <https://docs.oniroproject.org/projects/blueprints/en/latest/>]

▶ LESSONS LEARNED – TAKE AWAYS

- Not to be afraid of changes if they are business-justified
- Demo as early as possible
- Be explicit rather than implicit
- Listen to your „Customers” (both internal and external)
- Be brave and challenge yourself
- Love what you do and look positively into the future :)

▶ WHAT'S NEXT?

- **For the project:**
 - ✓ Setting and further merging within Eclipse Foundation
 - ✓ Become API-compatible with OpenHarmony
 - ✓ More partners on-boarding
 - ✓ Extend a list of Blueprints and Design Wins

▶ A PARTNER'S IMPRESSIONS

- Problems and how they were solved?
- What was the biggest challenge?
- What was new for you?
- What next?

▶ REFERENCES

The latest version of documentation:

- <https://docs.oniroproject.org/en/latest/>

Oniro's Developer mailing list:

- <https://accounts.eclipse.org/mailling-list/oniro-dev>

Oniro's WorkingGroup mailing list:

- <https://accounts.eclipse.org/mailling-list/oniro-wg>

Community's IRC:

- <https://web.libera.chat/#oniroproject>

How to get in touch with us (more options)

- <https://docs.oniroproject.org/en/latest/community-chat-platform.html>

Contact us directly (Oniro's staff):

- <https://oniroproject.org/join-us/>



Thank you!
(any questions?)