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
SCOPE

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

**Empirical process control**

- Transparency
- Inspection
- Adaptation
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 reviewes
- peer reviews
- Demoing and prod. reviewes

**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

### EasyRetro

**Jasmine Sprint 5 (14.07 - 03.08)**

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

<table>
<thead>
<tr>
<th>Went Well</th>
<th>To Improve</th>
<th>My Main Blockers</th>
<th>Action Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>We have done 80% even though severely reduced workforce and hot holiday season.</td>
<td>Sometimes another unplanned “extra” tasks appear deviating deliveries from the current sprint scope.</td>
<td></td>
<td>⚫️ monitoring: Zero-based budgeting, being more observant new extra features taking, trust our judgements @team</td>
</tr>
<tr>
<td>We’re opening for rise-v architecture (e.g. toolchain support).</td>
<td>On features delivery and visibility for the closest sprints (e.g. what is going be delivered in 1 or 2 or 3 sprints).</td>
<td>⚫️ monitoring: To make tasks/decomposition as granular as possible; however, it’s not always possible. If possible to switch into points instead of hours.</td>
<td></td>
</tr>
<tr>
<td>First visible progress on blueprints</td>
<td>Tasks are too large (sometimes 40+ hours) making whole process unrealistic and unpredictable (in terms of timeframe &amp; features landed).</td>
<td>⚫️ To consider more pair work, team up - so there is no single “big failure” that may break any features delivery in the future. @team, to be addressed partially at Eclipse via projects streams (mainstream, developers etc.)</td>
<td></td>
</tr>
<tr>
<td>Smooth loading / planning phase</td>
<td></td>
<td>⚫️ Missing visibility of same-team members’ availability (e.g. in work)</td>
<td></td>
</tr>
<tr>
<td>Good progress with GTA and its architecture understanding</td>
<td></td>
<td>@Adam</td>
<td></td>
</tr>
<tr>
<td>We streamline with processes and activities around them making development more pleasant.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code review done over the whole sprint, no big batch at the end</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Started cross domain work, eg. security review for GTA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*Building together a fully-connected all-scenario intelligent ecosystem*
Project mindset:
- Scope
- Time
- Budget
Project mindset:
- Scope
- Time
- Budget

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

Not applicable to us
## DEVELOPMENT SYSTEMS

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons (*)</th>
</tr>
</thead>
</table>
| **GitLab** | - de facto a standard for s/w development  
- a tool very close to development  
- REST API available | - missing custom fields feature e.g. lists, extra value placeholders  
- missing custom, „enforced” workflows |
| **OpenProject** | - easiness to implement new workflows  
- easy to manage users’ rights  
- REST API available | - another system to use along with GitLab  
- disjoint from our mainstream development |

A few of Python modules used:
- [https://github.com/Flying-Free/pyopenproject](https://github.com/Flying-Free/pyopenproject) (OP API handling)  
- [https://gitlab.com/python-gitlab/python-gitlab](https://gitlab.com/python-gitlab/python-gitlab) (GitLab API handling)  
- [https://matplotlib.org/](https://matplotlib.org/) (data visualisations, graphs drawing)  
- [https://numpy.org/](https://numpy.org/) (computations)  
- base: requests, regex, json etc

(*) - subjective, the most missing features for us
WORK PACKAGES #1/3

Lean requirements flow

Specification
NEW
Rejectable
IN_SPECIFICATION
Rejectable
SPECIFIED
Rejectable
DEVELOPED
Rejectable
Tests
IN_TESTING
TESTING_FAILED
TESTED
Show
DEMOED
CLOSED
Dismissed
REJECTED

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

[Purely OpenProject-based]
Building together a fully-connected all-scenario intelligent ecosystem
WORK PACKAGES #2/3

Phase #1
- Proposal
  - Requirements
    - Mandatory content added
    - Acceptance criteria
    - Values
  - OpenProject WishList backlog

Phase #2
- Specification
  - Decomposition

Phase #3
- Backlog
  - Zero-based Budgeting
  - OpenProject Release backlog

Phase #4
- Execution
  - Development
  - Tests
  - Documenting

Phase #5
- Readiness
  - Reuse
  - GitLab repositories
  - READMEs
  - devops
  - tests etc

[Mix of OpenProject & GitLab]
WORK PACKAGES #2/3

[Image of a diagram showing phases 1 to 5 with actions and stakeholders.

- Phase 1: Proposal
  - Who: Anybody
  - Requirements: Review, frame
  - Actions: Mandatory content added, Acceptance criteria, Values, OpenProject WishList backlog
  - Happens in: Work Packages #2/3

- Phase 2: Specification
  - Who: Architects
  - Actions: Decomposition, Testing
  - Happens in: Work Packages #2/3

- Phase 3: Backlog
  - Who: OpenProject Release backlog
  - Actions: Development, Tests, Documenting
  - Happens in: Work Packages #2/3

- Phase 4: Execution
  - Who: Accountable
  - Actions: GitLab issues
  - Happens in: Work Packages #2/3

- Phase 5: Readiness
  - Who:
  - Actions: Reuse, GitLab repositories, readthedocs, devops, tests etc
  - Happens in: Work Packages #2/3

[Mix of OpenProject & GitLab]
Building together a fully-connected all-scenario intelligent ecosystem
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.

[ 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/mailing-list/oniro-dev

Oniro’s WorkingGroup mailing list:
• https://accounts.eclipse.org/mailing-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?)