

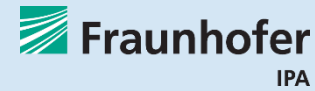
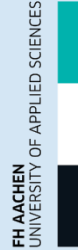
ROSIN SW QA MEASURES

ROS Industrial Workshop

Stuttgart, May 29

Jonathan Hechtbauer

ROSIN
Consortium



IT UNIVERSITY OF COPENHAGEN



rosin-project.eu



TOC

■ Status report on QA measures

- Influences and constraints of QA Practices
- Community working group, discussion forum and quality hub

■ Tools developed for improving code quality

- ROS node / package generator
- Rosinstall Time Machine
- Tools outlook: History-based Linting and Fuzz Testing

Qualitative Research in the ROS Community

- In-depth **interviews** with 10 community members (ROS/ROS-I)
- **Virtual ethnography** (wikis, forums, answers, etc.)
- **Community Events** (ROSCon, 2 ROS-I meetings, Danish ROS meetup)
- The analysis phase is **completed**.
- We are in the **implementation phase**.

ROSIN Consortium. *Quality Assurance Process and Community Management in ROS*. (116pages)

Alami. Dittrich. Wąsowski. *Influencers of Quality Assurance in an Open Source Community*. Submitted to CHASE'18



Influences and constraints of QA Practices in ROS

What do we see?

- Participation Motives (ideology, enjoyment)
- Priorities of the community (new features)
- Meritocratic culture (reputation through technical expertise)
- Lacks of sustainability strategy
- Complexity (interdisciplinary)
- Adaptability (informalism, ease of use)



What does it mean for a company developing systems on top of ROS?

- Difficult to **mandate QA tasks** (tests, code review) which is seen as “boring”
- QA work must be made **enjoyable** via secondary incentives (**recognition**, gamifications)
- Sustainability of ROS (packages) requires your willingness to **invest into maintenance**
- Importance to engage with the **community**

Community engagement: ROS QA Working Group

- 23 members
- From 7 countries
- From 3 continents, under European leadership
- From 20 organizations: 16 companies, 2 universities, 2 research institutions
- 5 meetings have taken place
- Discussion of possible QA initiatives in the community processes
- Ranking of initiatives

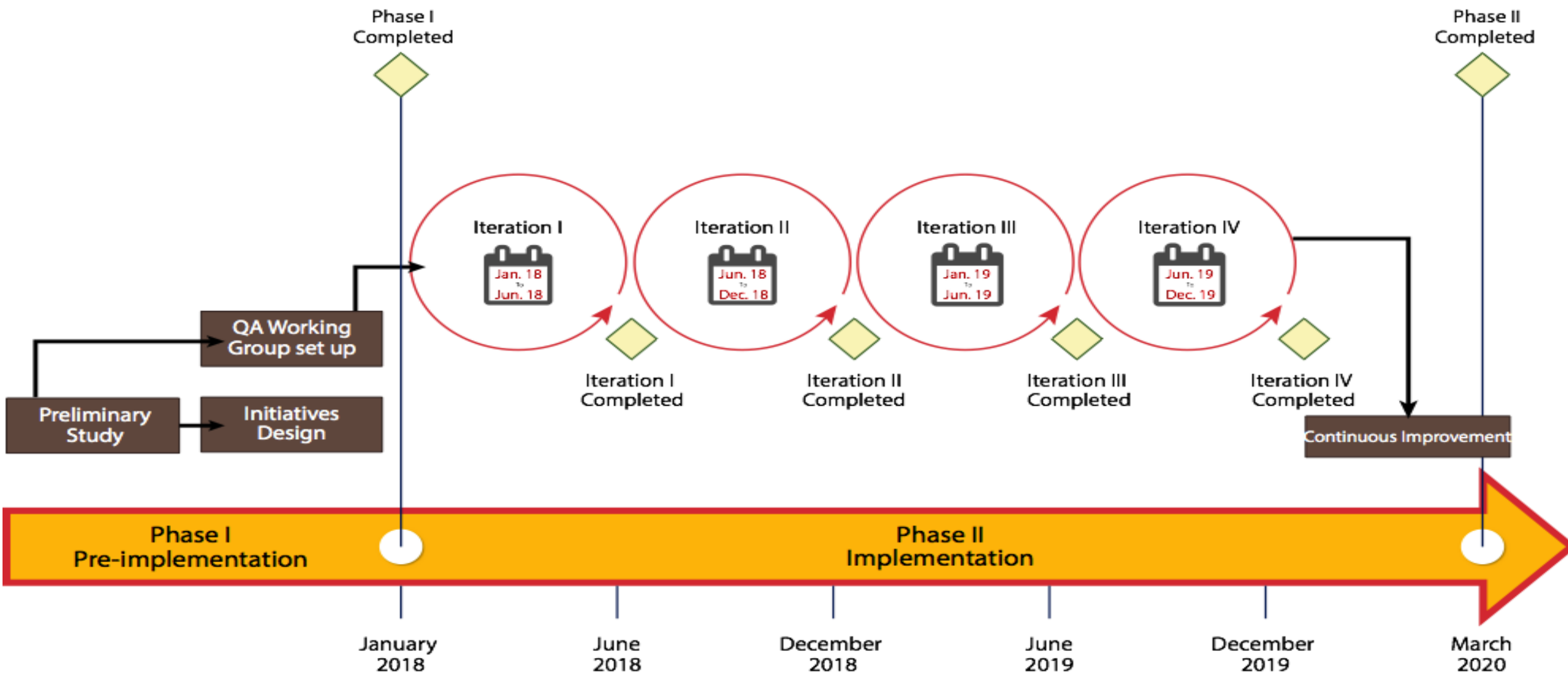


ROSIN Initiatives ranking

- What are the community priorities?
- Packaged into 4 iterations

Iterations	ROSIN Initiatives	No. Of vote points
Iteration 1	Make ROS packages quality visible.	42
	Appoint ownership	32
	Energize the code review process	32
	Implement a code scanning method and tool	31
	Maintenance issues	31
Iteration 2	Energize Continuous Integration	31
	Quality Hub website	30
	Formalize the code ownership process	29
	Onboarding process for core and non-core community members	29
Iteration 3	Model-in-the-Loop testing	28
	Implement a continuous improvement process	26
	Automated unit test generation	26
	Quality Discourse	23
Iteration 4	QA promotion events	19
	Model Driven Development	18
	#ROSQA	15

ROSIN Initiatives Implementation Timeline














Community engagement - Quality Discourse

- New category dedicated to quality
- Channel discussion
- Extension of the Quality Working Group

ROS Discourse

Quality Assurance ▾ Latest New Unread Top

+ New Topic

Topic	Users	Replies	Views	Activity
What quality metrics do we need to make packages quality visible?		2	60	16h
Input validation as a metric for quality		11	230	3d
Design By Contract		45	1.6k	3d
ROS Quality Assurance Working Group meeting minutes - Feb. 2018 Meeting		0	56	5d
Chaos testing tool for Docker		3	92	9d
Maintaining and releasing orphaned packages		0	92	10d
Quality guide for ROS2	 Latest Quality Assurance topics - Discourse.ros.org	2	82	10d
Documentation for maintainers		10	194	12d
Better Tagging of Issues to increase community involvement		0	60	20d
ROS Quality Assurance Working Group meeting minutes Kick Off Meeting - 10/01/2018		10	291	Jan 17
★ About the Quality Assurance category		0	77	Jan 4

Community engagement - Quality Hub

- Focal point to foster quality
- Tutorials for quality oriented practices
- We need your help to create solid resources

Quality

ROS Software Quality Hub

The [Quality Hub](#) is an online community hub that will provide a focal point and resources to foster quality related activity in ROS. Please help to grow it in the interest of the entire community.

This page is a central entry point for all quality assurance material for ROS. This includes tutorials and patterns on producing and contributing high quality code, as well as on testing, and other quality oriented practices. The particular focus are industrial users of ROS, but everybody is welcomed to benefit and to contribute.

Please create new quality oriented materials under this tree in the wiki.

Contents

1. [ROS Software Quality Hub](#)
 1. [Tutorials and articles](#)
 2. [Quality category on ROS Discourse](#)
 3. [Who is behind ROS Quality Hub? \(About Us\)](#)

1. Tutorials and articles

See [Quality/Tutorials](#) for tutorials & articles on quality assurance with ROS and for ROS.

2. Quality category on ROS Discourse

There is a dedicated software quality category on ROS Discourse: [discourse.ros.org/Quality](#).

Go there to discuss issues of quality policies in the ROS projects, and in ROS-based applications, including proposals of content for this webpage, offers to help, etc.

3. Who is behind ROS Quality Hub? (About Us)

So who is behind this page? The answer is simple: **It's you!** We need your help to create a solid resource on building high quality industrial level systems with ROS. We are a group of self-appointed editors who are generously funded by the European Union H2020 project [ROSIN](#).

We are trying to spin this effort with a good speed and trajectory, but it will die, if you don't join us. Talk to us on the discourse category for Quality (see above), or just grab an article devoted to Quality that needs improvement and work on it! You can take the article from anywhere in the wiki, and link it from here, as a sign that it has been edited. If you think an article on a topic is missing, create a new one, or ask others to do so in a discourse discussion.

Wiki

[Distributions](#)
[ROS/Installation](#)
[ROS/Tutorials](#)
[RecentChanges](#)
[Quality/Tutorials](#)
[Quality](#)

Page


[Edit \(Text\)](#)
[Edit \(GUI\)](#)
[Comments](#)
[Info](#)
[Subscribe](#)
[Add Link](#)
[Attachments](#)

More Actions:


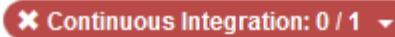
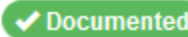
User

[Adam Alami](#)
[Settings](#)
[Logout](#)

First success – Improved quality badge

- Before: 
 - Green badge if at least one test exists
- Available in *ROS melodic*:
 - Build-statistic generated in buildfarm
 - Color of badge rewards successful tests
 - Displays number of tests & build history
 - Wiki user can directly access test results

Package Summary

Common code


- Maintained

Build history (last 2 of 2 builds):

#2	22-May-2018 18:30	0 / 1
----	-------------------	-------

ROS.org



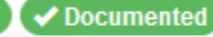
actionlib

indigo kinetic lunar melodic Show EOL distros: 

[Documentation Status](#)

[ros_base: actionlib](#) | [bond_core](#) | [dynamic_reconfigure](#) | [nodelet_core](#)

Package Summary

The actionlib package provides a framework for interfacing tasks. Example tasks include: [bond_core](#) for target location, [dynamic_reconfigure](#) for the handle of

Build history (last 5 of 7 builds):

✓ #7	22-May-2018 01:22	52 / 52
✓ #6	21-May-2018 19:22	52 / 52
✓ #5	20-May-2018 03:22	52 / 52

Package Summary

roscpp is a C++ programming library that eases the most widespread library for ROS. It provides a high-level interface for interfacing tasks, and [Param](#) for the high-level

Build history (last 5 of 34 builds):

✓ #34	23-May-2018 22:40	1367 / 1368
✓ #33	22-May-2018 03:15	1367 / 1368
✓ #32	21-May-2018 17:15	1367 / 1368