[Developers] Survey on runtime verification and field-based testing of robotic systems.

This survey is part of an exploratory study on **guidelines** for runtime verification and field-based testing of robotic systems. We build guidelines taking inspiration from both academics and practitioners, looking into scientific papers and both proof-of-concept implementations and established tools from github. The long-term goal is to build better methods and tools for gathering confidence in robotic systems, which requires proper infrastructure. The guidelines aim to assist both developers and QA teams to perform runtime verification (RV) and field-based testing (FT) in the context of ROS. Now, we need to understand whether the guidelines are aligned with best practices by surveying experts in the domain.

Instructions: The questionnaire consists of three parts.
1st. Familiarize yourself with an overview of the guidelines (Link: http://bit.ly/3FE7vXN).
2nd. Provide your opinion on usefulness, clarity, and applicability of the guidelines. We also ask for suggestions to improve methods and tools for runtime verification and field-based testing of ROS-based robotic systems.
3rd. We briefly ask about your experience and background to put your responses into context.
* For a deeper understanding of our stance on **field-based testing**, check doi: 10.1145/3447240
* For a deeper understanding of our stance on **runtime verification**, check doi: 10.1007/s10009-021-00609-z

The survey takes approximately 15 minutes to complete. We will keep your responses confidential and only store them for analysis. The report will only contain aggregated summaries of the responses and anonymized quotes.

Please let us know if you have any questions. You can contact directly <ricardo.caldas@chalmers.se>.

Thank you in advance for your valuable contribution. Your participation will significantly contribute to advancing the field of testing for robotic systems.

Sincerely,

Ricardo Caldas (Chalmers University of Technology, Sweden)
Juan Antonio Piñera García (Gran Sasso Science Institute, Italy)
Matei Schiopu (Chalmers University of Technology, Sweden)
Genaína Nunes Rodrigues (Universidade de Brasília, Brazil)
Patrizio Pelliccione (Gran Sasso Science Institute, Italy)
Thorsten Berger (Ruhr University Bochum, Germany and Chalmers|University of Gothenburg, Sweden)

* Indicates required question

1. Informed Consent *

Tick all that apply.

- [ ] I hereby consent to participate in this experiment. I acknowledge that participation is voluntary and I can leave at any time throughout the experiment.
- [ ] I have read the instruction materials and I have access to the guidelines' overview.

2. Have you ever worked with robotics software? (e.g., development, testing, research) *

Mark only one oval.

- ◯ Yes
- ◯ No

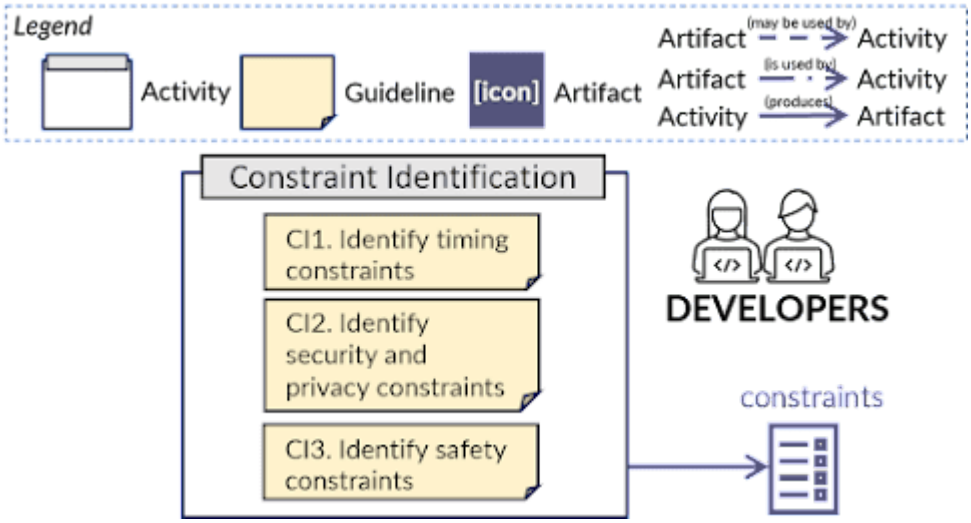
Constraint Identification

This section queries about our guidelines on constraint identification for preparing the system to runtime verification and field-based testing.

Definition: Constraint Identification consists of eliciting non-functional requirements that may be critical or non-negotiable to the system under design and test. Preparing or running test scenarios to attest correct behavior of the system should not violate the elicited constraints.

Please analyze the graphical representation of the constraint identification activity and answer the following questions about the individual guidelines CI1 - CI3.

Constraint Identification activity and guidelines



3. **Guideline CI1**. *The development team should identify timing constraints to ensure that no real-time requirements will be neglected during the system testing. For instance, Autoware_Perf (git: azu-lab/ROS2-E2E-Evaluation) allows for the calculation of response time and latency in ROS 2 applications; such measurements may be used as hard constraints to testing the system. Also, one may identify real-time constraints with respect to synchronization between robots, sensor data processing time, or fault detection and recovery.* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.				
Clearly formulated.				
Applicable to ROS-based systems that you've worked with.				

4. **Guideline CI2**. *The development team should identify security or privacy vulnerabilities that may pose a risk to participants' integrity, confidentiality, and availability, that may be caused by the testing and runtime verification activities. The ros-security workgroup, in collaboration with Alias Robotics, maintains SROS (git: ros2/sros2) for the analysis of such vulnerabilities and fixes. In addition, Alias Robotics maintains a database of vulnerabilities detected in ROS applications (git: aliasrobotics/RVD) as a means of awareness.* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

5. **Guideline CI3**. *The developers should identify the boundaries of a safe behavior (i.e., safety constraints) to enable the use of mechanisms for preventing the robot from hurting operators or property during testing and verification activities. For example, the developers can identify constraints like speed limit, distance to obstacle, or conditions for emergency stop, which can be used by monitoring tools like ROSMonitoring (git: autonomy-and-verification-uol/ROSMonitoring), or used by an independent safety controller in a separate ROS node to prevent the robot from falling from a cliff, e.g., the Kobuki robot (git: yujinrobot/kobuki).* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

6. Do we miss any guideline or do you have any suggestion for improvement? Please elaborate.

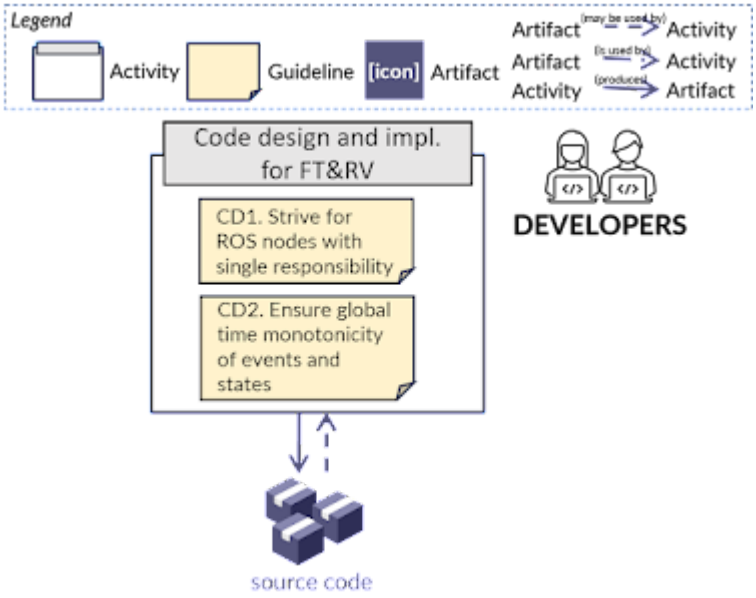
Code design and implementation

This section queries about the guidelines on code design and implementation for preparing the system to
runtime verification and field-based testing.

Definition:
Code design and implementation describes a set of principles that should be followed when designing and implementing the robotic software in order to prepare the robotics software under scrutiny to testing and verification activities.

Please analyze the graphical representation of the code design and implementation activity and respond to the following questions.

Code design and implementation activity and guidelines



7. **Guideline CD1.** *To facilitate assessment during the robot operation and enable fine-grained observation and control, the developers should implement ROS nodes following the single responsibility principle (i.e., each node should implement a single feature and different nodes can be combined to perform a complex task). For example, developers should implement independent nodes for path planning and reactive manoeuvring, or independent nodes for defining primitive skills like grasping an object or simultaneous localization and mapping (SLAM).* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

8. **Guideline CD2.** *The development team should ensure global time monotonicity of events and states to avoid potential non-determinism in scheduling. Such non-determinism is a threat to getting confidence in the system since repeated tests under the same conditions may turn into different results. A technique that can be used to ensure determinism is annotating messages and requests with timestamps and implementing a logical time synchronizer, similar to what is done by MAVROS (git: mavlink/mavros). Also, the Time Synchronizer message filter (wiki: message_filters) may be used for this purpose.* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.				
Clearly formulated.				
Applicable to ROS-based systems that you've worked with.				

9. Do we miss any guideline or do you have any suggestion for improvement? Please elaborate.

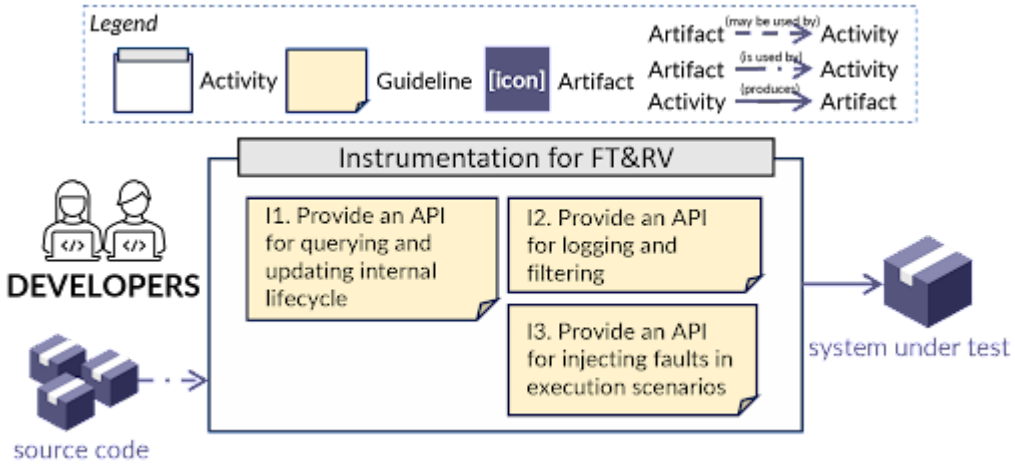
Instrumentation

This section queries about the guidelines on instrumentation for runtime verification and field-based testing.

Definition: With the source code in hands, either developers or the QA team may perform code instrumentation, which consists of modifying the source code to enabling interactions between internal variables with monitors.

Please analyze the graphical representation of the instrumentation activity and respond to the following questions.

Instrumentation activity and guidelines



10. **Guideline I1.** *To facilitate field-based testing, the development team should properly manage the ROS nodes' lifecycle and prepare APIs for querying and updating the internal nodes' life-cycle, e.g., to ensure that nodes are in the right state for testing. For example, developers can use modes (git:* ros2/demos*) for lifecycle management.* In the context of *Micro-ROS (git: micro-ROS), developers can define custom lifecycle modes (git: micro-ROS/system_modes) like sleep, power saving, starting, processing, and ending modes, and use separate extra nodes for mode monitoring and mode update.* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

11. **Guideline I2.** *The development team should provide an API for logging and filtering data to enable access to valuable runtime data which should be used for both runtime verification and field-based testing. The standard approach to logging and filtering is rosbag (wiki: rosbag). Though, in addition, AWS CloudWatch (git: aws-robotics/cloudwatchlogs-ros2) collects data from the rosout topic and provides a filter for eliminating noise from the logged events. Another example is the Robotic Black Box (git: ropod-project/black-box) which allows for listening to data traffic from distinct sources and logging the messages using MongoDB.* *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful.	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

12. **Guideline I3**. To enable runtime verification and field-based testing, developers should provide an API for injecting faults or emulating runtime errors. For instance, to emulate the consequences of software faults, ros1-fuzzer (git: aliasrobotics/ros1_fuzzer) provides an API for ROS messages fuzzing. As another example, to imitate the mistakes of programmers, IM-FIT (git: cembglm/imfit) is a tool for injecting faults in ROS applications. *

To which extent do you agree that this guideline is ...?

Mark only one oval per row.

	Fully disagree	Disagree	Agree	Fully agree
Useful	◯	◯	◯	◯
Clearly formulated.	◯	◯	◯	◯
Applicable to ROS-based systems that you've worked with.	◯	◯	◯	◯

13. Do we miss any guideline or do you have any suggestion for improvement? Please elaborate.

Personal Background

To put your response into context, please tell us briefly about your background.

*Reminder:
We will keep your responses confidential and only store them for analysis. The report will only contain aggregated summaries of the responses, and anonymized quotes.

14. How long have you worked with robotics software? *

Mark only one oval.

- ◯ < 1 year
- ◯ 1 -- 3 years
- ◯ 3 -- 5 years
- ◯ 5 -- 10 years
- ◯ > 10 years

15. How does your work in robotics software relate to ROS (Robot Operating System)? *

Mark only one oval.

- ◯ I have contributed to ROS package(s) (e.g., development, testing)
- ◯ I have worked on robotics applications that use ROS packages.
- ◯ I have worked on robotics applications, but I am unsure whether they use ROS packages.
- ◯ I haven't heard of ROS before.
- ◯ I am part of the ROS development team.
- ◯ Other: _____

16. Which type of organizations have you worked with robotics? *

Tick all that apply.

- [ ] Academia
- [ ] Industry
- [ ] Government
- [ ] Independent groups (e.g., Open Robotics, hobbists)
- [ ] Individually
- [ ] Other: _____

17. What robotics domains have you worked with? *

Tick all that apply.

- [ ] Service robotics
- [ ] Marine robotics
- [ ] Healthcare robotics (e.g., prosthetics, surgery)
- [ ] Industrial automation
- [ ] Construction
- [ ] General-purpose
- [ ] Other: _____

18. To receive the results of the study, please enter your email address.

_____

19. Could we contact you in case of follow-up questions?

Mark only one oval.

- ◯ Yes, via my email address above
- ◯ No

Thanks a lot for your time and contribution! Your responses are very valuable to the community. Please click **Submit** to finish the survey.

Note: After **submitting** the questionnaire, please consider answering also the questionnaire that targets runtime verification and field-based testing of ROS-based systems from the **QA team's perspective**,
if you haven't yet. Link: https://forms.gle/HffF8QLrpMpvtjaJ9.