

INTERVIEW guidelines/questions

Demographics Information

- What is the approximate number of employees of your company? What about the size of your team?
- What is the operational domain of the team where you are employed (e.g., autonomous driving cars, robotics, drones, avionics, satellite...)?
- What are the programming languages being used in your CPS?
- Does your team rely on a CI/CD pipeline for developing CPSs?
 - If yes, when and why did your team introduce CI (i.e., approximate number of years)?
 - If not, are you planning to introduce it?
- Do you know whether in your company different teams rely on different CI/CD for CPS? If yes, can you please give us a contact of a developer working on a different team who may be available for the interview?

Background Information

- Years of development experience.
- What is your academic background?
- How many years of experience do you have in CPSs?

Role in the company

- How long have you been with this company?
- What is your role in the company?
- What is your role (if any) in the CI/CD process?

CI/CD pipeline structure

- What is in your opinion the biggest problem that needs to be addressed while considering CI/CD pipelines for CPSs? Can you rank them somehow?
- What about benefits and challenges in adopting/moving to a CI/CD pipeline strictly related to the CPS context?
- **Phases/Steps**
 - Which are the phases of your CI/CD process (e.g., compilation, testing, verification, deployment, etc..)?
 - Are there some differences in the CI/CD pipeline being used in software domains different from CPSs? Are there some specific reasons behind the above differences?
 - Are there some properties/characteristics that your pipeline has to satisfy based on the fact that you are developing CPSs? Examples of properties may be accessibility of hardware resources as well as simulators, security issues, performance, or build execution time.
- **Tools**
 - What are the technologies used in your CI/CD pipelines? Specifically for:

- versioning
 - build automation (e.g., maven, gradle)
 - CI automation server (e.g., Jenkins, Gitlab)
 - ... other tools (e.g. dockerization, Kubernetes, etc.)
- Are you customizing/integrating the steps in the CI/CD process by using shell scripts? If yes, Why?

- **Verification & Validation**

- What are the tools mainly used for static analysis?
 - How did you configure static code analysis tools being used to verify the overall software quality? Other than focusing on code styles, are there coding standards that have to be met in order to have a product that may be certified?
 - What about the usage of static code analysis tools aimed at identifying security issues, as well as vulnerabilities and possible memory leaks? Are there other factors you usually consider while integrating static code analysis tools?
 - What about the usage of static code analysis tools for identifying possible bugs inside your CPS code?
 - Other types of static analyzers?
- What type of testing frameworks and testing activities are mainly involved in your CI/CD? For instance, do you rely on unit and integration testing?
 - Are unit, integration, and system-level test cases manually written? Why? What about functional testing?
 - Are you using some specific testing frameworks for non-functional requirements?
 - Do you apply field testing? What about simulation testing? What are the challenges/barriers in applying this?
 - A key aspect in CPSs may be related to certifications and standards. Do you consider any kind of compliance testing? What are the challenges/barriers in applying this? How much of your testing is done for certification?
 - Does the pipeline test for safety, privacy and security, as well as reliability and availability requirements?
 - Finally, since using different testing activities will increase the value of your final product by increasing its cost: do you include any kind of prioritization for selecting a subset of test cases to be run?
- Do you use Model-Driven-Testing and/or Test-Driven-Development? Why? If you develop by following TDD, who is responsible for writing test cases?
- Do you have a precise expectation of how your code should behave? Possibly in terms of the order in which certain events should occur, the time they take or how the data being processed by your code looks at various points?
 - If you have such properties of your code written down, do you already have a tool that can show you when/why/how your code adhered to them?
- Do you have a way to execute code under various inputs from its input space?

- Do you use system models to support test automation?
- How do you validate your system models?
- How do you identify critical scenarios (scenarios leading to failures) for system testing?
- **Performance Analysis**
 - How are your performance requirements expressed?
 - How are the results of performance analysis presented to engineers?
 - Do you have a way to determine root causes of performance problems? If yes, how are root causes reported to engineers?
- **Deployment**
 - Does the CI/CD pipeline include Deployment? Does your CI/CD process depend on the target environment? Discuss challenges related to delivery on HiL/CPS.
 - Do you have processes and tools to observe differences on CPS behaviors between the development and the deployment environment?
 - Do you perform any runtime monitoring of the system? How does it work?
- **Inclusion of AI capabilities while developing CPSs**
 - Do you have AI capabilities included/integrated in the development of CPS? Specifically, does your code base use ML components or AI components? What types of CPS actions are controlled/decided by ML components or AI components?
 - For which development tasks do you rely more on AI/ML components? Are you using supervised or unsupervised learning approaches?
 - What are the strategies used to determine and manage data needed for MLs?
 - What about the configuration of the models being adopted? Have you experienced cases where there was a need to retrain the model on a new/different set of data points? What about parameters tuning?
 - What about the monitoring of the deployed model? Which data are you collecting to determine whether or not there may exist errors during real-world execution?
 - Are you using specific testing strategies aimed at identifying bugs in ML applications since that MLs bugs may differ from traditional software bugs?
 -
- **Inclusion of hardware components and/or simulators usage**
 - Do you attach real hardware components (e.g, sensors and actuators) to the CI/CD pipeline? Do you use final versions of the hardware devices or do you rely on simplified HW testbeds?
 - Do you make use of any simulator in the pipeline? Why do you rely on simulators instead of considering real hardware devices? What types of Simulators?
 - If you use simulators, how did you decide to trust a given simulator? Did you assess the validity and accuracy of simulated operations?
 - Can you please give us your thoughts about the possible tradeoffs you consider when choosing between using real HW devices or simulators? E.g.,

do you prioritize them depending on the features and/or changes applied to the CPS? Do you perform testing directly on the hardware?

- **Configuration**

- Did you customize the CI/CD pipeline based on the project's characteristics? Why?
- Do you schedule different build types (e.g., do you rely on nightly builds)?
- Do you have any performance (Memory and CPU) requirements? where do the performance requirements come from (e.g., usage of sensor data, etc.)?
- Can you provide us information about build triggering policies being adopted in the CI/CD pipeline for CPS?
 - Build triggering? Any change? Only specific changes? Daily builds? Nightly builds? Rebuild everything each time? Incremental build?

CI/CD Practices: Challenges and effort concerning the Pipeline:

- What is the average build execution time? Has the build execution time changed over time? If yes, do you know why?
- What is your maximum acceptable CI build execution time? What are possible barriers in CPSs that prevent you from having the desirable execution time?
- Have you ever experienced that the growing build execution time was due to some bugs in the build tools being adopted? Can you explain more about that specific situation and give us information on how you have fixed the problem?
- In terms of build execution time, how do you deal with balancing test execution speed and certainty, especially considering the safety characteristics of CPS in real word scenarios?
- What are the major costs when configuring the pipeline?
 - Do you have to cope with scarce HiL simulators resources and their limited realism? What about accessibility of the above resources?
 - Since it is important to increase security together with the ability to access and modify the CI system once needed: What do you think about having only a few people with the privileges of modifying the CI/CD configuration?
 - What are the challenges in assembling/configuring the pipeline (build, pipeline configuration scripts)? What customization did you apply to traditional CI/CD configuration steps?
 - What about the cost of testing features on the pipeline?
 - What do you think is the most important bottleneck in the way of writing/executing/automating tests in CPSs?
- How often do you have to modify your CI configuration?
 - What are the reasons for the modifications (e.g., dependencies installation or change versions for dependencies)?
 - Would it be useful to have a (semi) automatic recommender suggesting how to adapt the pipeline configuration?
- CI/CD Flakiness
 - Have you experienced flakiness in the CI/CD process? What are the main reasons behind the flaky behaviour? Is the flakiness in the hardware being

included in the pipeline such as sensors (e.g., on the quality of data) and actuators? How did you deal with the flakiness from the CPS?

- Where does the flaky behaviour manifest (e.g., testing failure)? Have you experienced problems due to lack of observability of the flakiness? Any specific method in place to deal with flakiness problems? Moreover, have you experienced problems about the distance between failure and fault of the flaky behaviour?
- What about the coordination and communication between software and hardware components? Did you experience any problems due to lack of proper documentation or lack of documentation at all?
- Do you monitor your CI/CD process (e.g., by relying on previous build history)? In other words, do you collect detailed logs during operation for debugging and monitoring?
 - How difficult is it for you to determine/locate the reason for the failure starting from the build log?
- Have you experienced specific problems while setting up a CI/CD pipeline for CPS systems (e.g., the inclusion of hardware components)?
- What is in your opinion the biggest problem that needs to be addressed while considering CI/CD pipelines for CPSs? Can you rank them somehow?
- What about benefits and challenges in adopting/moving to a CI/CD pipeline strictly related to the CPS context?
- Do you have needs about auto-configuring the CI/CD pipeline to address portability, compatibility, deployability and dependencies management requirements?