How hardware-in-the-loop testbenches accelerate embedded systems development

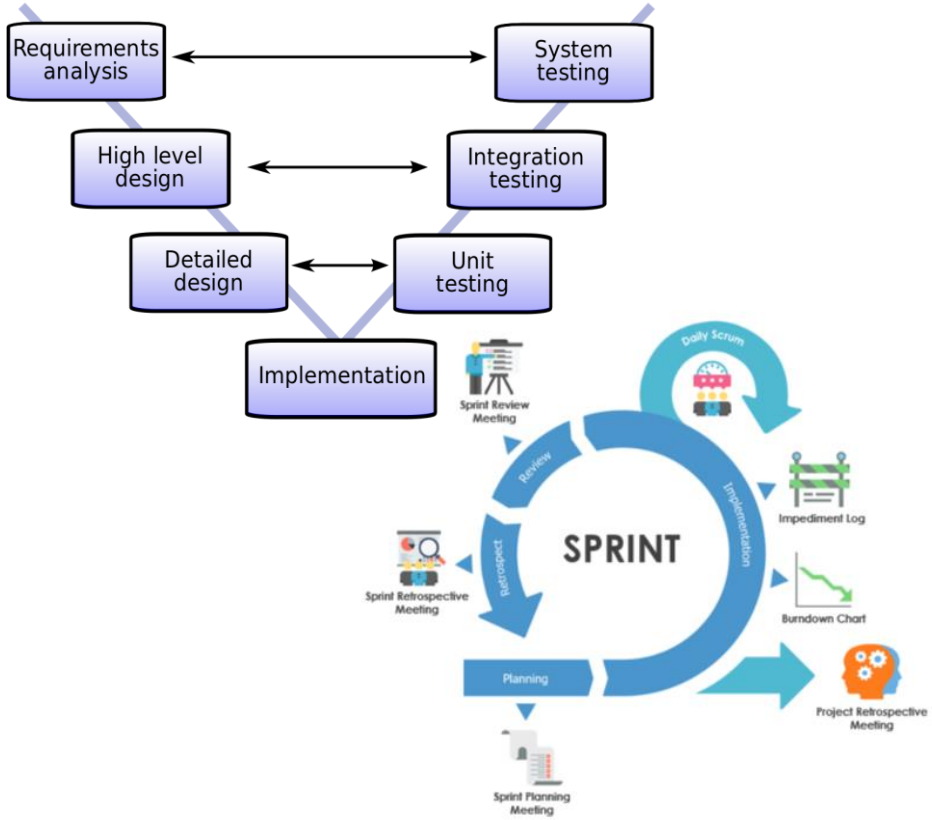
SWISSED22

Ivo Locher

Sept 12, 2022

konplan
developing together

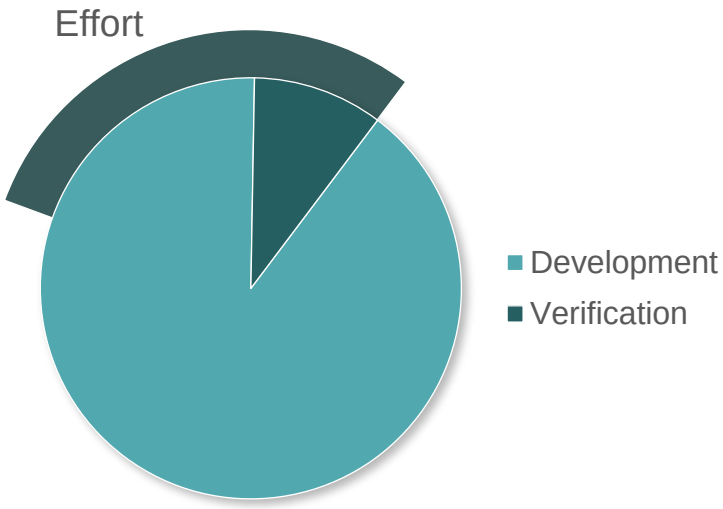
What are the everlasting trends in product development?



Drivers and Trends

- **Faster development cycles (Market pressure)**
- **Reliable and robust designs (somewhat depending on the market branch)**
- **Life cycle extension of product by adding new features over time**
- **Agile development (e.g. in sprints)**

What is often the attitude when starting a development project…?



Attitude

- **Focus is on definition and implementation**
- **Verification is considered simple**
- **Verification is done properly only once**

Reality

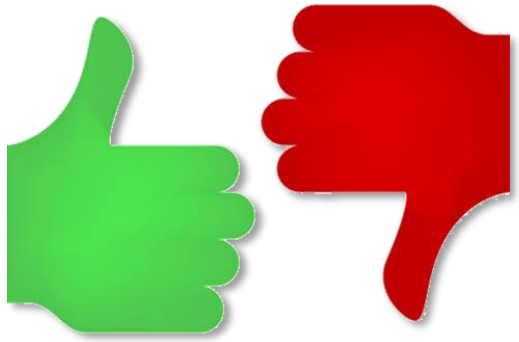
- **More verification iterations required than expected. Delayed verification**
- **Hesitation to change or add features due to the regression testing effort**

- **How can we overcome this issue in embedded systems development?**

Proposed approach: Hardware-in-the loop testbenches

Pros:

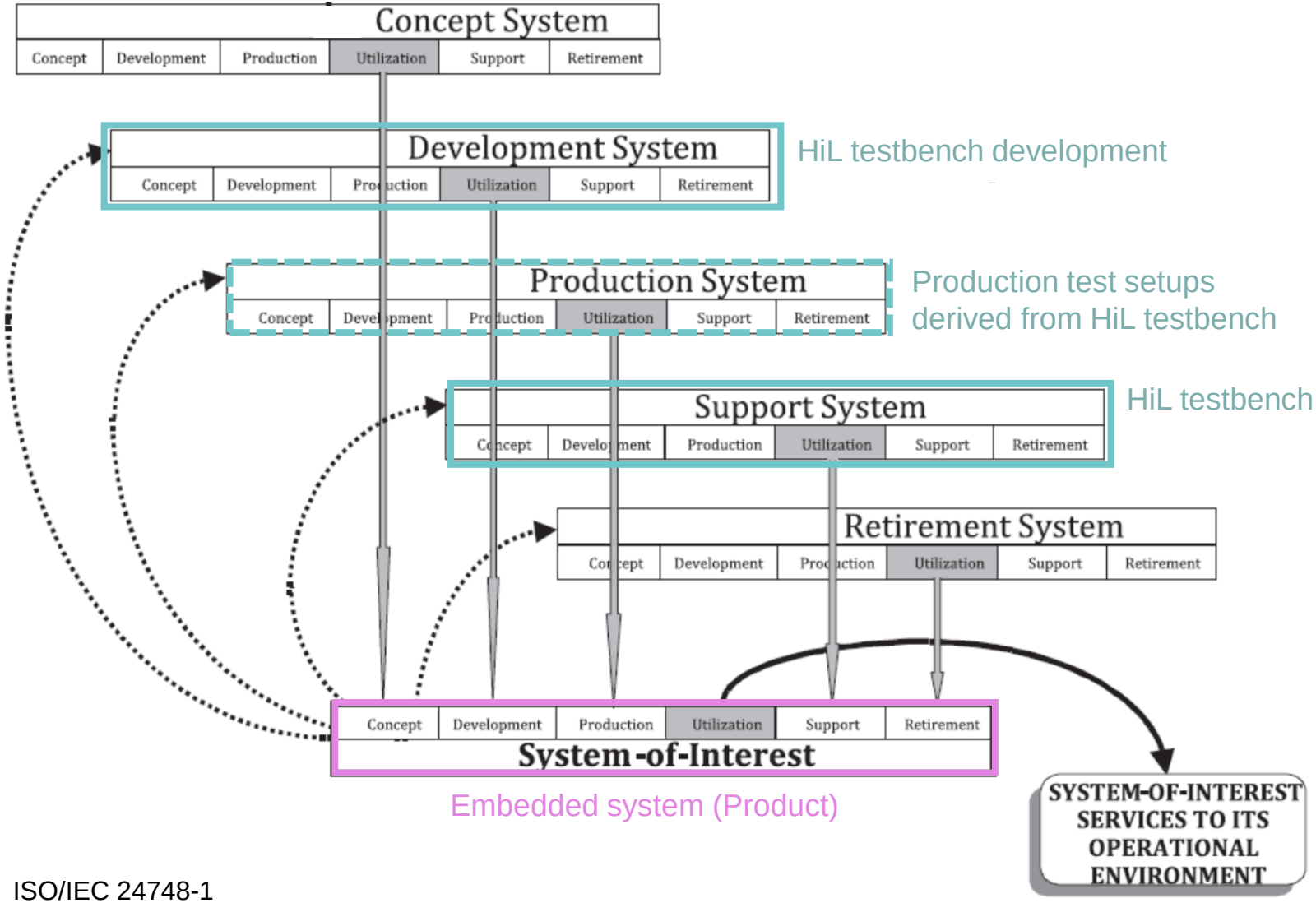
- Accelerated firmware testing by emulating the environment
- Automated firmware testing; independent of interpretation
- Increased firmware testing coverage
- Allowing integration into the CI pipeline
- Relieve verification engineer from repetitive (testing) work
- Low bar for new features (Innovation!)
- Life cycle management support



Cons:

- Effort to build the testing platform
  Effort to write test cases
  (short term view)
- Firmware architecture required to be designed (and implemented) for Testability
- Requires engineers with “Design for Test” skill sets / mindsets

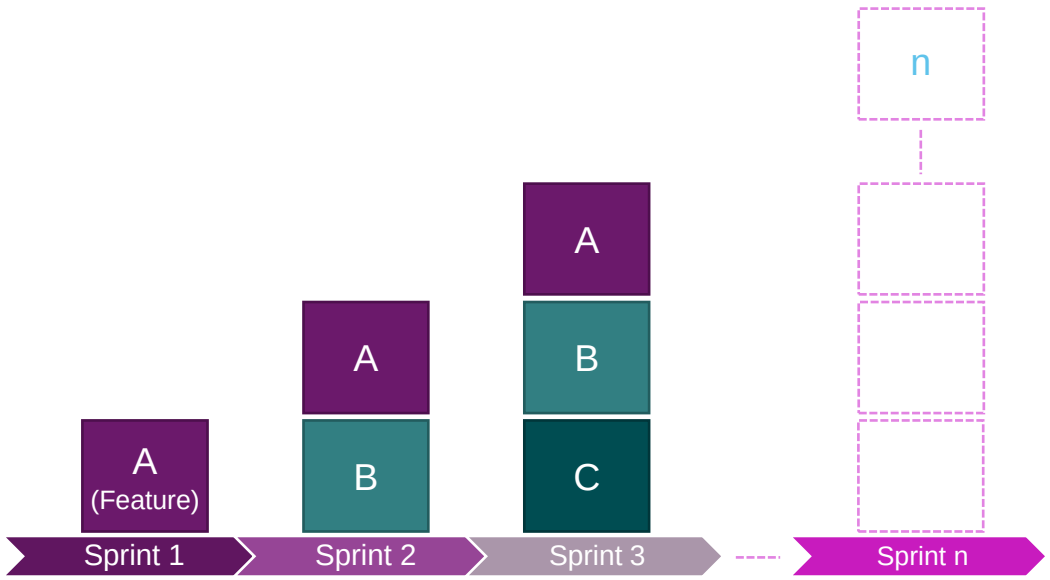
Synergies by using a HiL testbench



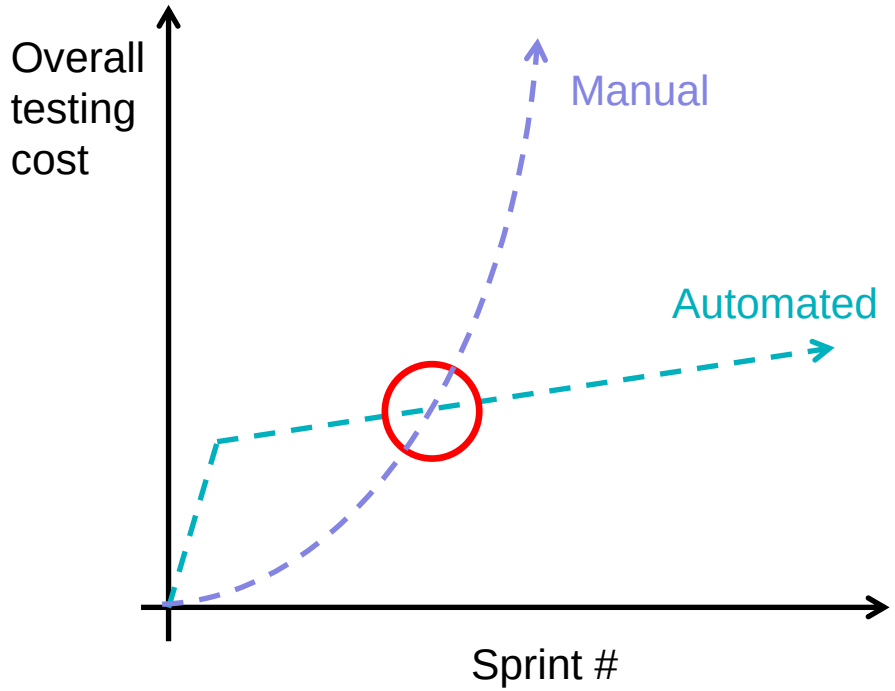
ISO/IEC 24748-1

Sweet spot for automated testing in agile development

Manual testing cost evolution

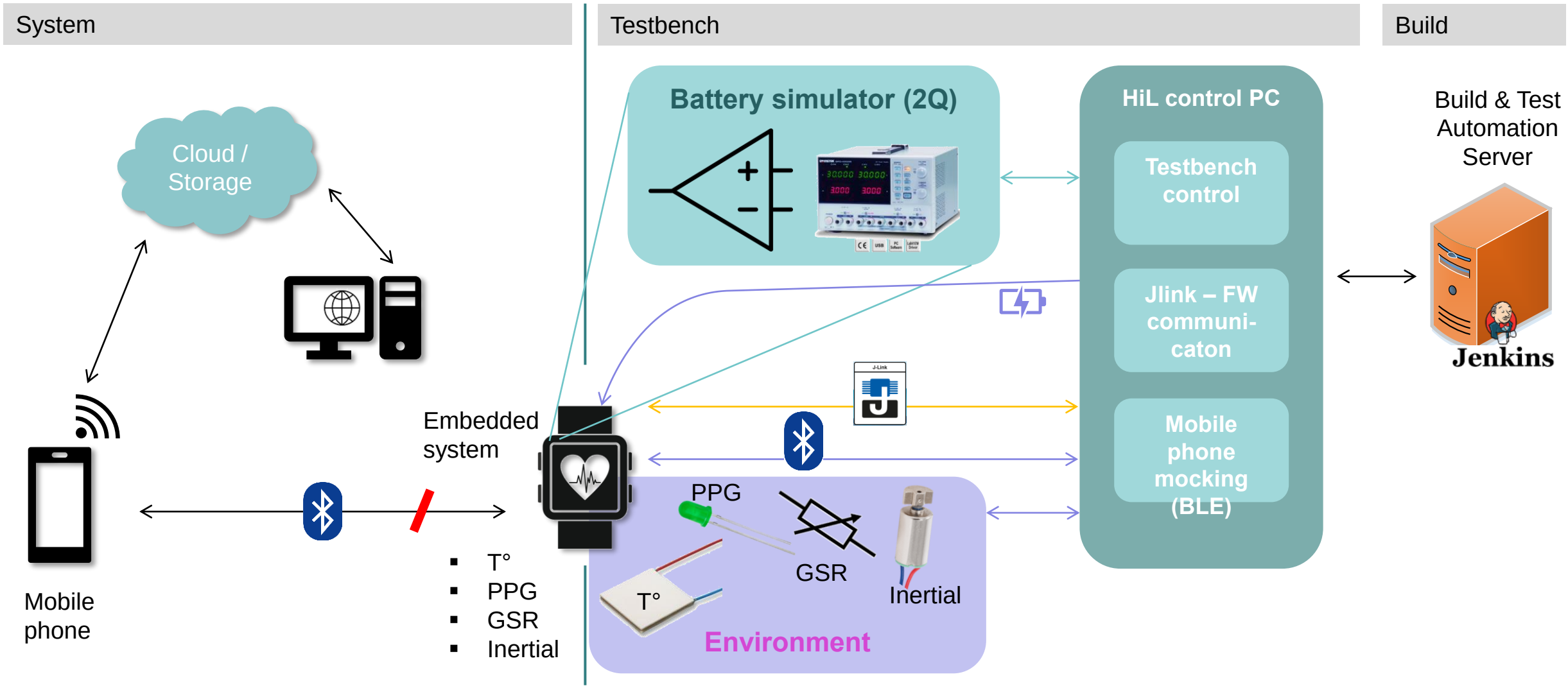


$$TotalTestingCost = \frac{n(n+1)}{2} \Rightarrow O(n^2)$$

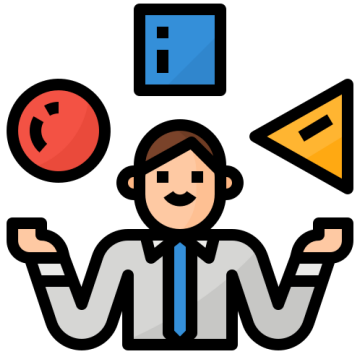


Sweet spot depending on kind of project!

HiL testbench conceptional overview for a Wearable



Capabilities of the HiL testbench



- **Manipulate (emulate) device environment / sequencing of device environment**
  - E.g. resulting in device wake up (based on combination of sensor signals)
  - E.g. fault generation ➔ triggering safety functions
- **Monitor device reaction & characterization**
  - E.g. BLE
- **Monitor and trigger internal device states. Cycling of states.**
  - E.g. thru Jlink
- **Executing test cases on device from remote**
- **Allow writing of test cases on PC in high-level languages**

Tangible benefits of the particular HiL testbench

- **BLE “real life” testing**
  - Flaws in communication discovered
- **Specific error generation / safety verification**
  - over temperature, battery overvoltage, firmware update reliability
- **Systematic testing of battery management**
  - Verification and reliability: charging / discharging cycling
- **Measurement accuracy verification of GSR**
  - Characterized by reading out from firmware (by parsing MAP file for measurement value address and accessing thru Jlink)
- **Potential for Fuzz testing**
  - i.e. generating random inputs at BLE interface to embedded system
- **Firmware bug fixing while integrating HiL testbench!**

Achievements and Considerations

- **Automation of > 80% of the firmware tests**

- **Increased reliability (by higher coverage and by having deeper insights)**
  - By emulating a defined environment
  - By higher testing coverage
  - By having deeper insight into FW at runtime
  - By systematics (boundary, fault injection)

- **HiL API abstraction**
  - test cases written by a software engineer with high-level programming knowledge

- **Determine sweet spot for HiL testbench**

  Drivers:
  - Complexity
  - Required reliability
  - Intended coverage
  - Life of product (line)
  - vs. Investment for HiL testbench

- **Start HiL test bench development early**

- **Integrate in CI pipeline**

- **Define architecture of embedded system compatible for (automated) HiL testing**

Please visit us at the booth outside

Ivo Locher, PhD, EMBA, PMP
Program Manager at konplan

https://www.linkedin.com/in/ilocher/

+41 41 799 30 10

ivo.locher@konplan.com

konplan

konplan systemhaus ag
Suurstoffi 2
CH-6343 Rotkreuz