

Electrolux RVC

From a CM point of view

The robot system in brief.

- Three embedded systems communicating with each other and the cloud.
- Customer upgradeable FW.
- Continual FW improvement iterations.
- Continual HW cost reduction and product care.
- Smartphone app and cloud SW.



From one model to several...



SW for different models

- SW consists of binaries for each subsystem packed into a single archive.
- New models contain different HW and need different binaries.
- Require way to identify which type of robot: "platform".
- Platform checked at install time.
- End-of-life policy needed- how long do we support a platform?

Variants on a model

- Latest platform introduces variants.
- Same binary, different behaviour.
- Configuration variables, set at production or in field.
- Read at run time.

Producing FW with new features

- Most high-level behaviour is shared across platforms.
- Low-level behaviour might not be.
- Aim to produce with as few active code branches as possible- may not need new branch for each platform. Important question.
- Don't have to have same branch strategy for each subsystem.
- Build system must be able to simply produce binaries for any platform.

FW version naming

- Version numbers refer to system's high level capability. Works as long as we aim to maintain similar functionality across platforms.
- Give same version number to each platform for a given release.
- Platform number included in name to differentiate install binaries.

What about changes to a platform

- If no longer binary compatible- new platform.
- Can use variants to cover simple cases.
- Aim to keep changes binary-neutral or backward compatible.
- Example- removing pressure switch.

Production SW

- Some new features or platforms require changes to PT SW.
- Example: introduction of variants.
- Need to keep track of required PT version for each given FW release.

Connectivity

- Team originally part of robotics, system release included app/cloud.
- Moved to new organisation- how to handle release notes?
- App version updated periodically.
- Certain new robot FW features need matching app.
- Cannot guarantee that users update both at same time.
- Need to handle incompatible combinations at run time, also suggest matching upgrades at install time. Push notices etc.

Documentation

- Originally one platform, one set of documents.
- New platform substantially similar to first one- what to do?
 - Completely new set of documents, largely duplicated? Hard to maintain.
 - One set of documents with loads if "if/then"? Hard to read- scales badly.
 - Hybrid. Messy but only reasonable way. Case by case judgement. Also has scaling problems.
- Release notes for robot FW and App/cloud. One document or separate?