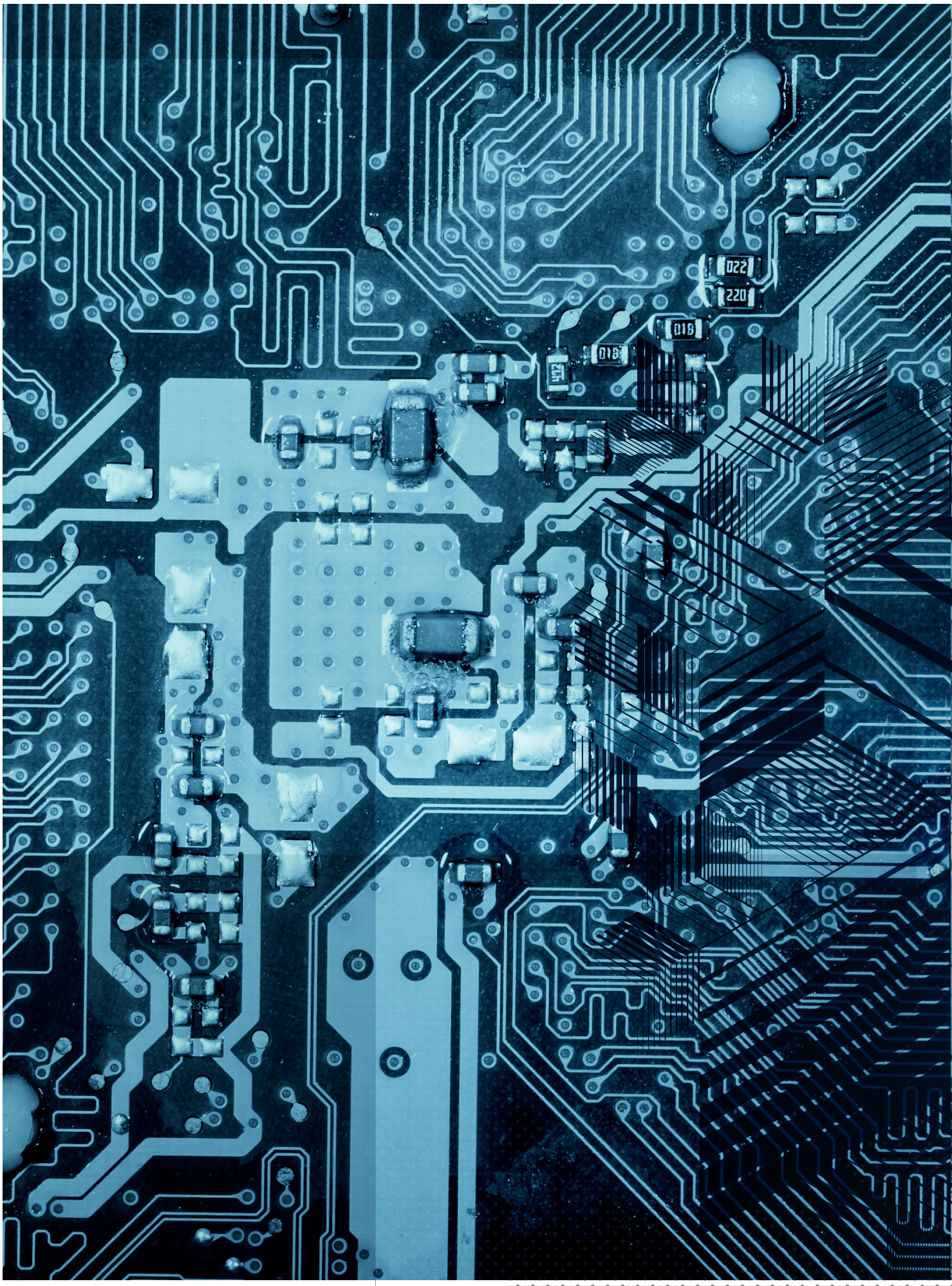

022
220
015
018
102

CONTROL MODEL FOR SMART PRODUCT DEVELOPMENT

Author: Dr. Bernhard Burger,
ROI Management Consulting GmbH

THE DEVELOPMENT OF SMART NETWORKED PRODUCTS REQUIRES A HIGHLY INTEGRATIVE CONTROL MODEL - BUT HOW CAN DIFFERENT RELEASE CYCLES BETWEEN A FEW DAYS AND SEVERAL MONTHS BE MERGED?

For a device with a high proportion of electronics and software, a new functionality is to be enabled during operation. In order to maintain the necessary memory capacity of the device, the product developers decide to delete lines of code for functions that are no longer required. Hours later, the entire work comes to a standstill. The reason: Although the deleted functionalities were not relevant for the operation of the device, they were required for the EOL test.

The example shows how closely hardware and software components interact in smart products. For product development, this means that the various components must not only be closely coordinated in design and development, but also tested in an integrated manner. It also shows what effects software-centric function development has on hardware planning and component selection. In development practice, however, this is often not the case: Since hardware components are often made available later, software tests are often carried out with the help of simulations, but normally not even with them. A number of fine tunings can only be made in the integrated system. An integrated development model is therefore urgently needed. But what can this look like if the components involved work according to completely different methods and development phases (see Fig. 1)?

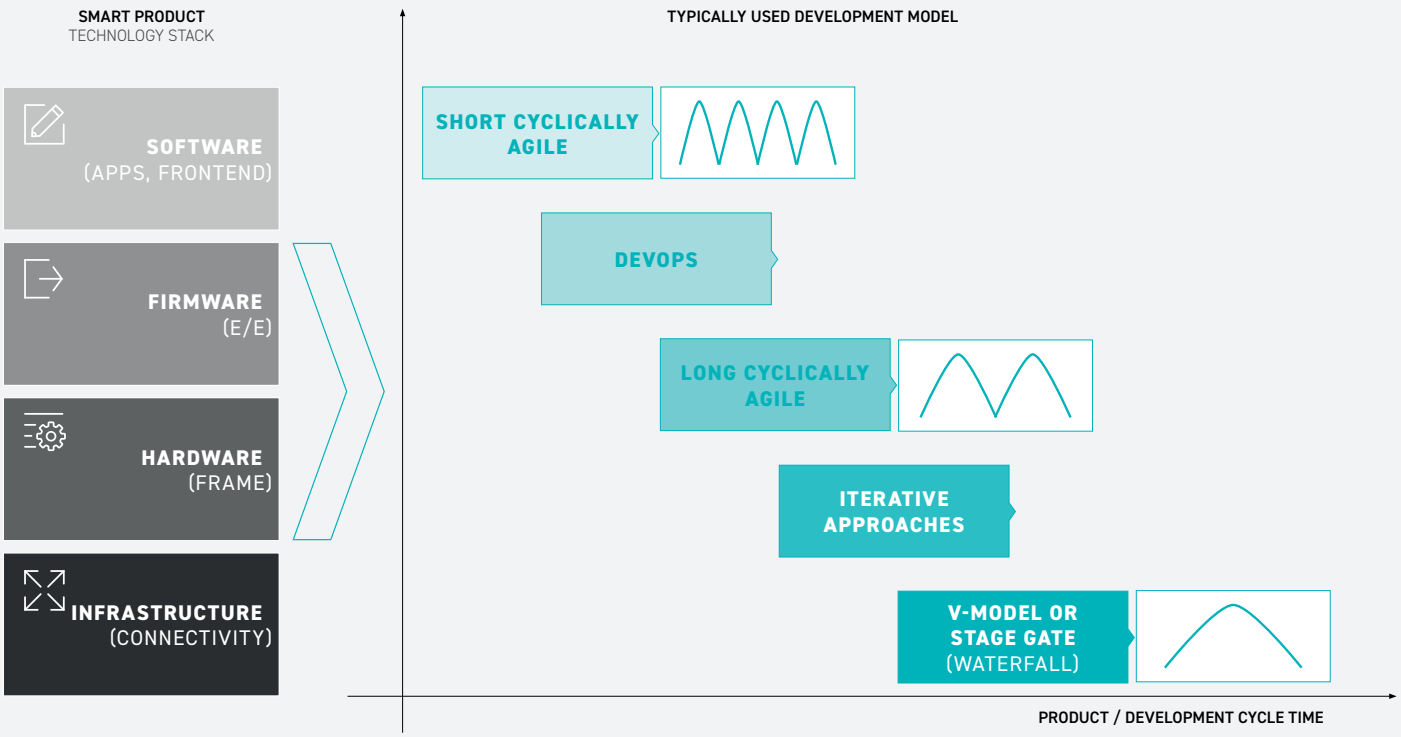


Fig. 1

INTEGRATION INSTEAD OF ADAPTATION

A pure transfer of methods from software to hardware development and vice versa is not very promising. The working methods and development cycles in the respective areas are too different. While, for example, a software and application team can complete a new version daily or even more short-term, the production of a testable hardware component often takes several weeks or even months. This does not mean that we cannot work more short-cyclically in HW development than it is often the case today. Clever cutting of functionalities and early development spikes of individual modules ensure shorter cycle times here as well. Nevertheless, it is necessary to adapt the optimal methodology for each of the various components and to ensure interaction between the individual development streams via an overarching control model. The following premises should be taken into account:

- ***HIGHLY INTEGRATED DEVELOPMENT***
  Regular, defined synchronization points, which are much shorter cyclical than typical stage gates, must ensure early and frequent integration of hardware and software development as well as other streams.
- ***PERMANENT FEEDBACK***
  An incremental approach and early and regular testing should make it possible to obtain and control customer feedback at any time.
- ***FAST INCREMENTS***
  The control model should support closely timed releases of new increments.
- ***FULL TRANSPARENCY***
  The development model should ensure full transparency for all persons involved in the development process regarding development scope, deadlines and dependencies.

Based on these premises, ROI has designed a development model that provides an integrative framework for the development of smart products (see Fig. 2 on next page).

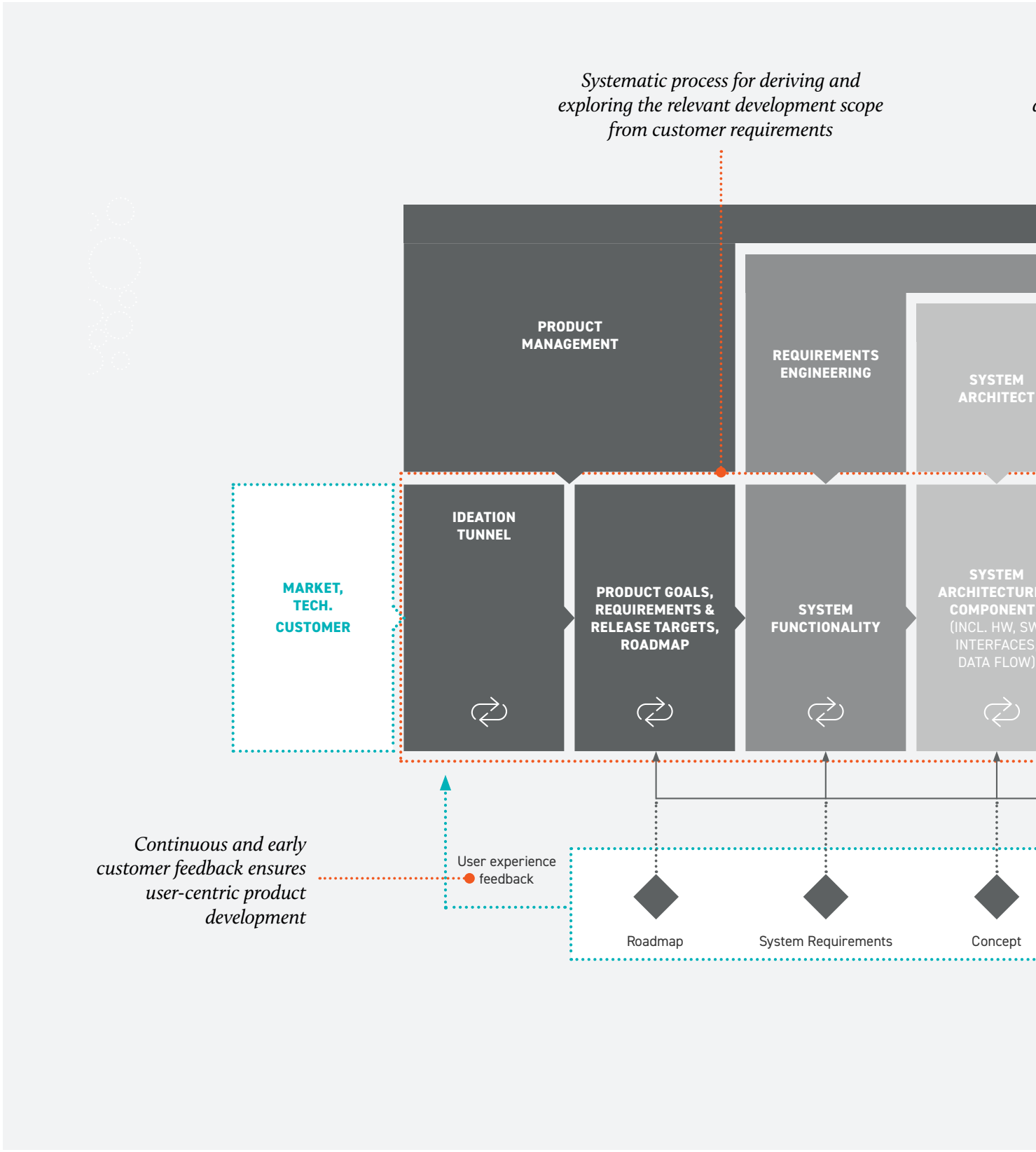
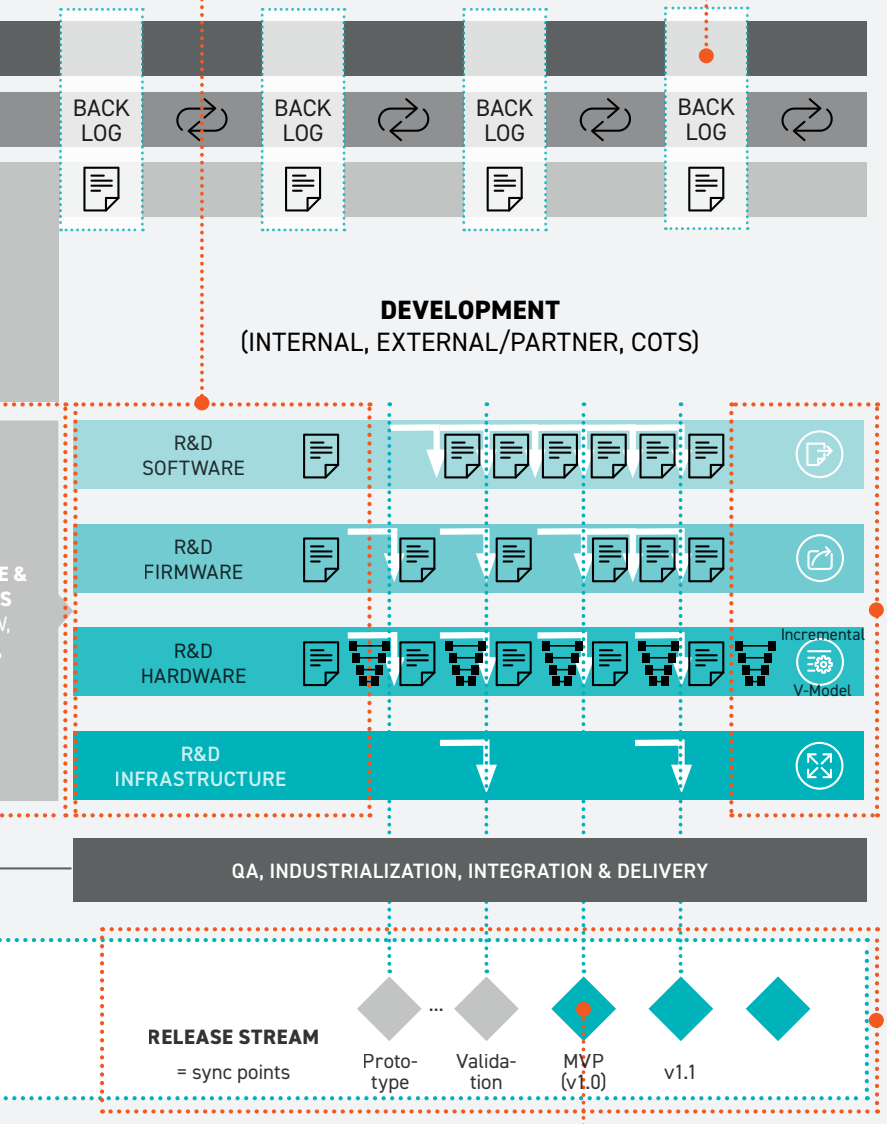

Systematic process for deriving and exploring the relevant development scope from customer requirements
PRODUCT MANAGEMENT
REQUIREMENTS ENGINEERING
SYSTEM ARCHITECT
MARKET, TECH. CUSTOMER
IDEATION TUNNEL
PRODUCT GOALS, REQUIREMENTS & RELEASE TARGETS, ROADMAP
SYSTEM FUNCTIONALITY
SYSTEM ARCHITECTUR COMPONENT (INCL. HW, SW INTERFACES DATA FLOW)
Continuous and early customer feedback ensures user-centric product development
User experience feedback
Roadmap
System Requirements
Concept

Fig. 2

Separate consideration of the different development streams in the smart product

Full transparency about the coordinated development scope in the overall development backlogs



Consideration of different control methods depending on product/development cycle duration

Regular synchronization points for early and frequent integration of different development streams

Early provision of testable product versions in the form of Minimum Viable Products (MVPs)