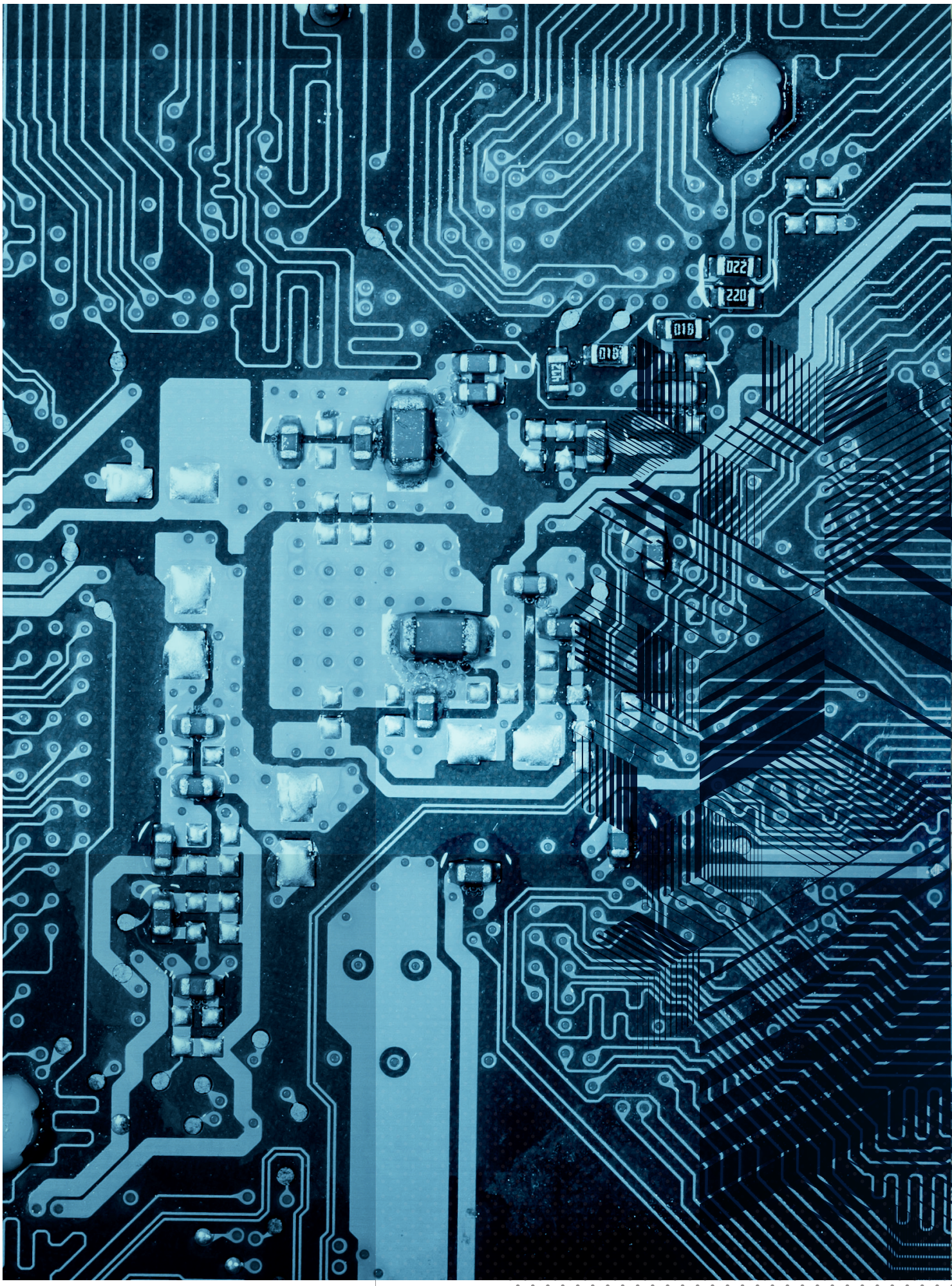

022
220
015
018
102

STEUERUNGS-MODELL FÜR DIE SMARTE PRODUKTENT-WICKLUNG

Autor: Dr. Bernhard Burger,
ROI Management Consulting GmbH

DIE ENTWICKLUNG SMART VERNETZTER PRODUKTE ERFORDERT EIN HOCHINTEGRATIVES STEUERUNGSMODELL – DOCH WIE LASSEN SICH UNTERSCHIEDLICHE RELEASE-ZYKLEN ZWISCHEN WENIGEN TAGEN UND MEHREREN MONATEN ZUSAMMENFÜHREN?

Für ein Gerät mit einem hohen Elektronik- und Softwareanteil soll im Rahmen der Serienpflege eine neue Funktionalität freigeschaltet werden. Jedoch ist die gewählte Hardware zu diesem Zeitpunkt schon unzureichend, da man bei der Erstentwicklung exakt auf die bestehenden Anforderungen optimiert hatte. Um die dafür notwendigen Speicherkapazitäten zu erhalten, beschließen die Produktentwickler Code-Zeilen für "nicht benötigte" Funktionalitäten zu löschen. Stunden später steht die Produktion still. Der Grund: Die gelöschten Funktionalitäten waren zwar für den Betrieb des Geräts nicht relevant, wurden aber für den EOL-Test benötigt. Das Beispiel zeigt, wie eng Hard- und Software-Komponenten in smarten Produkten zusammenwirken. Für die Produktentwicklung bedeutet das, dass die verschiedenen Komponenten nicht nur in der Konstruktion bzw. Entwicklung eng aufeinander abgestimmt, sondern auch integriert getestet werden müssen. Es zeigt auch, welche Auswirkungen softwarezentrische Funktionsentwicklung auf die Hardwareplanung und Selektion der Bauteile hat. In der Entwicklungspraxis ist dies allerdings oftmals nicht der Fall: Da Hardware-Komponenten häufig später bereitstehen, erfolgen Software-Tests dort oft mithilfe von Simulationen, im Normalfall jedoch nicht einmal damit. Eine Reihe von Feinabstimmungen kann erst im integrierten System vorgenommen werden. Ein integriertes Entwicklungsmodell ist daher dringend erforderlich. Doch wie kann das aussehen, wenn die beteiligten Komponenten nach völlig unterschiedlichen Methoden und Entwicklungsphasen arbeiten (siehe Abb. 1)?

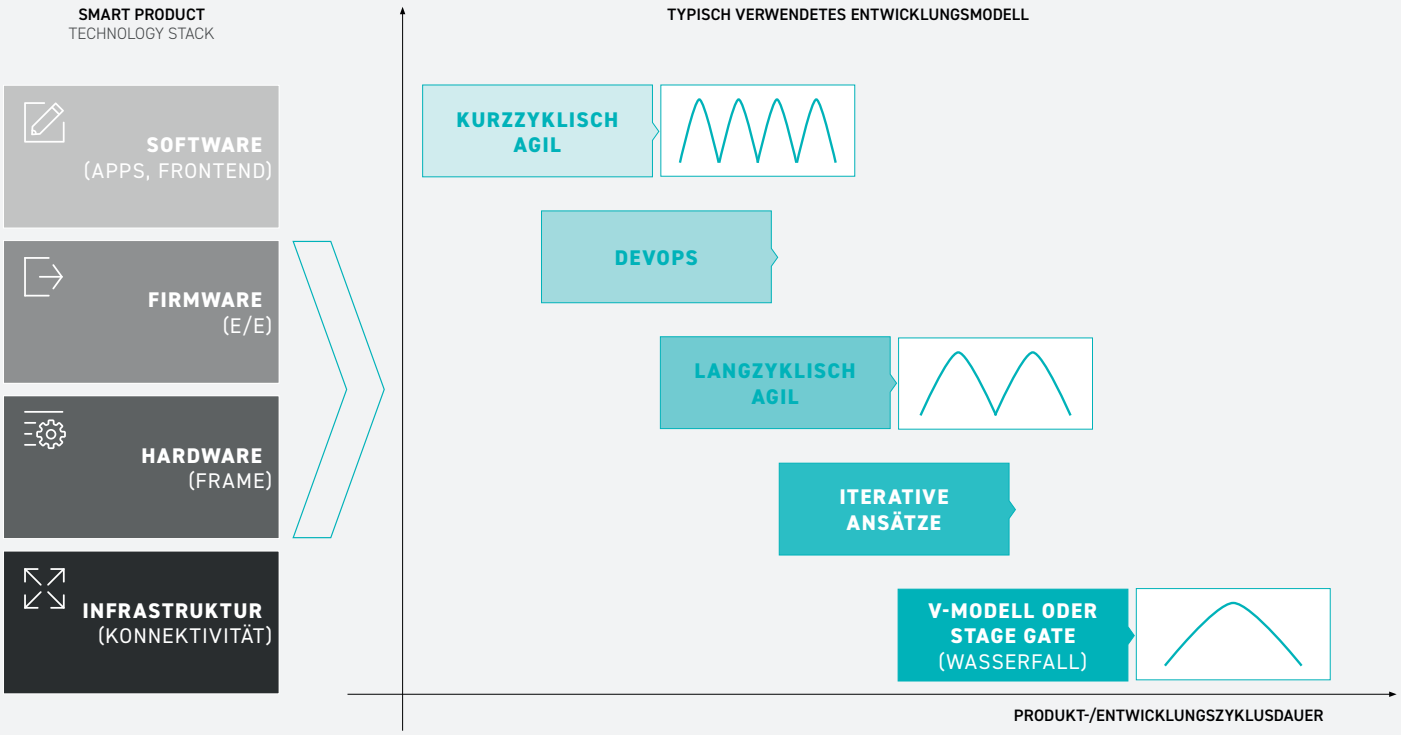


Abb. 1

INTEGRATION STATT ADAPTION

Eine reine Übertragung von Methoden der Software- auf die Hardware-Entwicklung und umgekehrt ist dabei wenig erfolgversprechend. Zu unterschiedlich sind die Methoden und Entwicklungszyklen in den jeweiligen Bereichen. Während beispielsweise ein Software- und Applikationsteam täglich oder noch kurzfristiger eine neue Version fertigstellen kann, dauert die Herstellung eines testbaren Hardware-Bauteils oft mehrere Wochen oder sogar Monate. Das soll nicht heißen, dass nicht auch in der HW Entwicklung kurzzyklischer gearbeitet werden kann, als dies heute oftmals passiert. Geschicktes Schneiden der Funktionalitäten und frühe Entwicklungsspikes einzelner Module sorgen auch hier für kürzere Zykluszeiten.

Dennoch gilt es, für die verschiedenen Komponenten die jeweils optimale Methodik zu adaptieren (siehe Abb. 1) und über ein übergreifendes Steuerungsmodell ein Zusammenspiel der einzelnen Entwicklungs-Streams sicherzustellen. Dabei sollten folgende Prämissen berücksichtigt werden:

- *HOCHINTEGRIERTE ENTWICKLUNG*
  Über regelmäßige, definierte Synchronisationspunkte , die deutlich kurzzyklischer liegen als typischer Stage Gates, muss eine möglichst frühe und häufige Integration der Hard- und Software-Entwicklung sowie weiterer Streams sichergestellt werden.
- *PERMANENTES FEEDBACK*
  Über ein inkrementelles Vorgehen und ein frühes und regelmäßiges Testing sollte das Einholen und Einsteuern von Kundenfeedback zu jedem Zeitpunkt ermöglicht werden.
- *SCHNELLE INKREMENTE*
  Das Steuerungsmodell sollte eng getaktete Releases neuer Inkremente unterstützen.
- *VOLLE TRANSPARENZ*
  Durch das Entwicklungsmodell sollte für alle am Entwicklungsprozess beteiligten Personen eine volle Transparenz über Entwicklungsumfänge, -fristen und Abhängigkeiten sichergestellt werden.

Auf Grundlage dieser Prämissen hat ROI ein Entwicklungsmodell entworfen, das ein integratives Framework für die Entwicklung smarter Produkte liefert (siehe Abb. 2 nächste Seite).

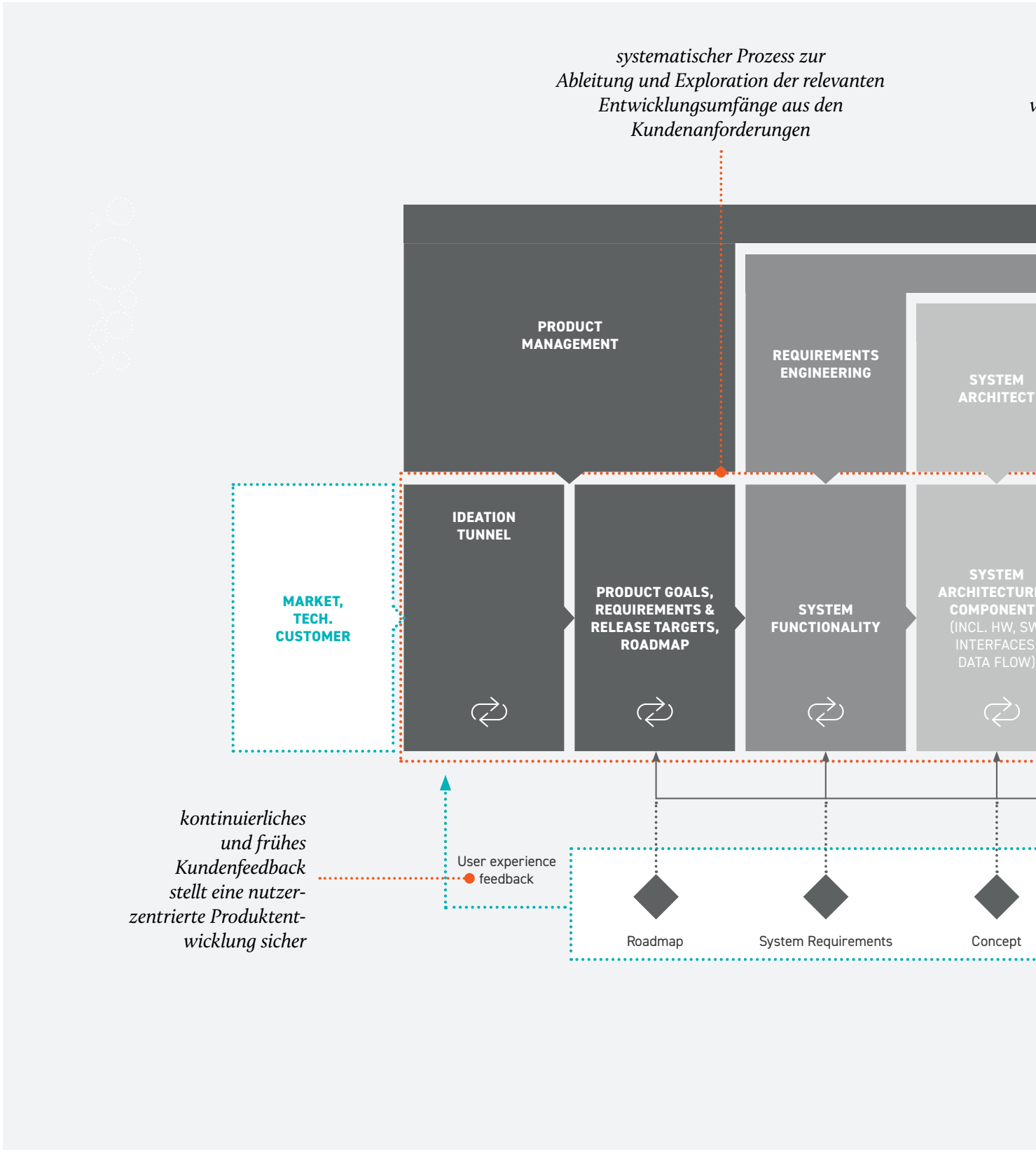
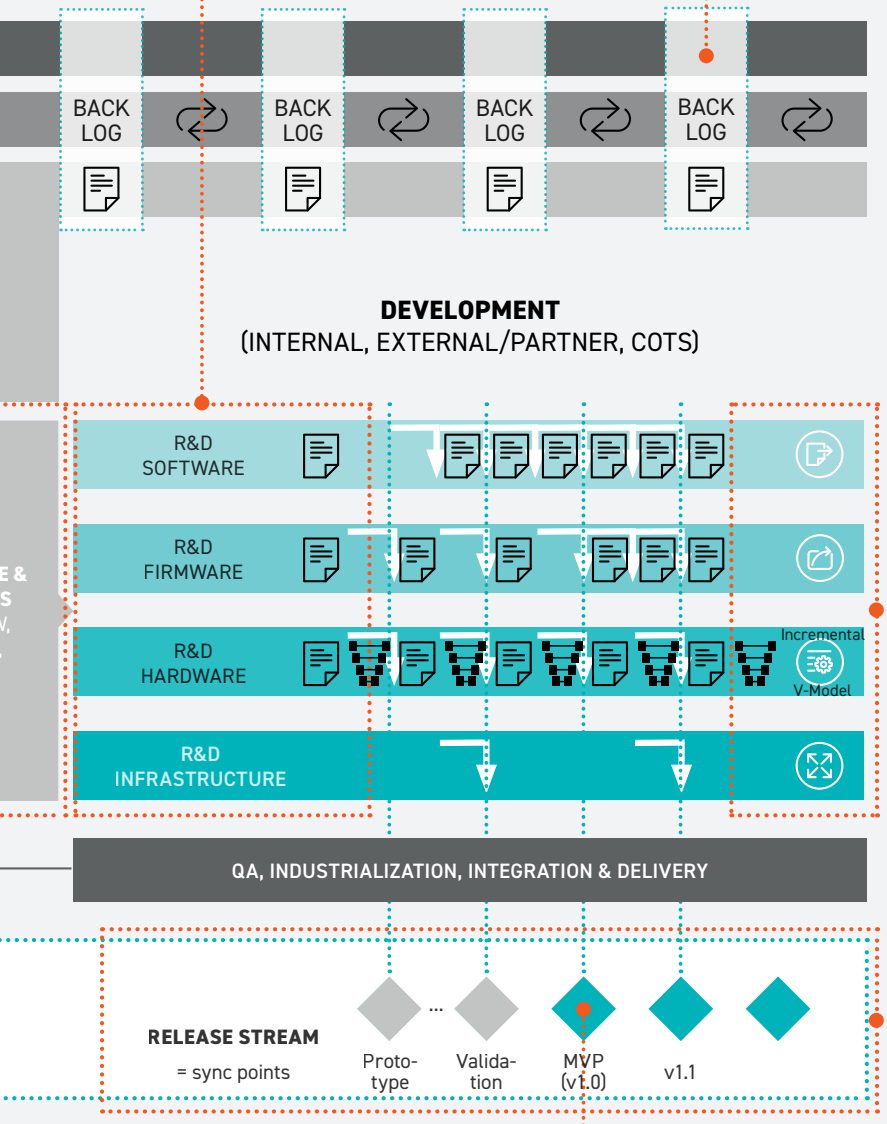

systematischer Prozess zur Ableitung und Exploration der relevanten Entwicklungsumfänge aus den Kundenanforderungen
PRODUCT MANAGEMENT
REQUIREMENTS ENGINEERING
SYSTEM ARCHITECT
MARKET, TECH. CUSTOMER
IDEATION TUNNEL
PRODUCT GOALS, REQUIREMENTS & RELEASE TARGETS, ROADMAP
SYSTEM FUNCTIONALITY
SYSTEM ARCHITECTUR COMPONENT (INCL. HW, SW INTERFACES DATA FLOW)
kontinuierliches und frühes Kundenfeedback stellt eine nutzer-zentrierte Produktent-wicklung sicher
User experience feedback
Roadmap
System Requirements
Concept

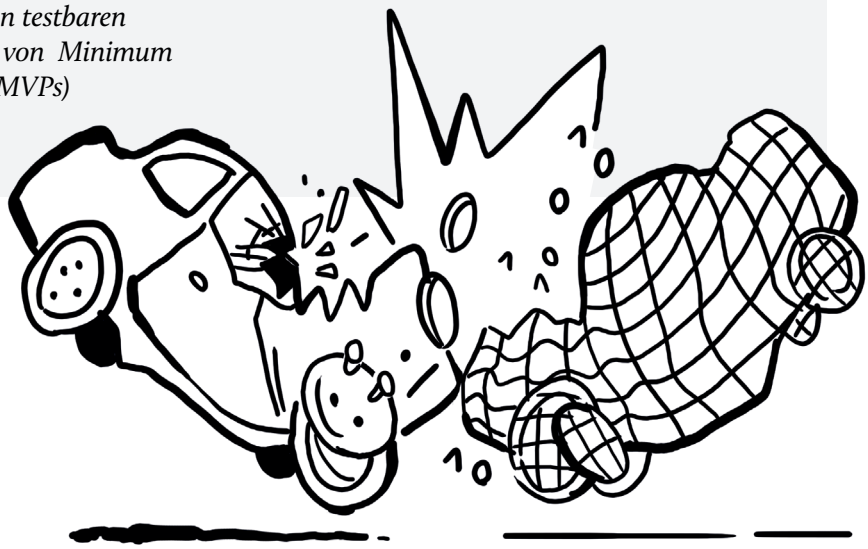
Abb. 2

gesonderte Betrachtung der verschiedenen Entwicklungs-Streams im smarten Produkt

volle Transparenz über die abgestimmten Entwicklungsumfänge in den übergreifenden Entwicklungs-Backlogs



Berücksichtigung der unterschiedlichen Steuerungsmethoden je nach Produkt-/Entwicklungszyklusdauer

regelmäßige Synchronisationspunkte für eine frühe und häufige Integration der verschiedenen Entwicklungs-Streams

frühe Bereitstellung von testbaren Produktversionen in Form von Minimum Viable Products (MVPs)