

Software-Entwicklung neu gedacht

Teil 1: Wie Multicore, Safety- und Security-Aspekte die Softwareprojekte von heute verändern

Embedded-Softwareentwicklung findet heute in den meisten Fällen immer noch für eine Singlecore-Umgebung statt. Doch Software steuert immer komplexere Abläufe, die nur in Multicore-Umgebungen funktionieren. Für die Entwicklung ergeben sich daraus neue Herausforderungen.

Um einen hohen Standard in der Softwarequalität zu erreichen, helfen unterschiedliche Modelle für die einzelnen Entwicklungsphasen in einem Projekt. Die üblichen Ansätze für den Ablauf in der Software-Entwicklung greifen für gewöhnlich auf das V-Modell und auf das Modell für agile Entwicklungsmethoden zurück.

Das V-Modell (XT) schreibt dabei folgende Schritte vor:

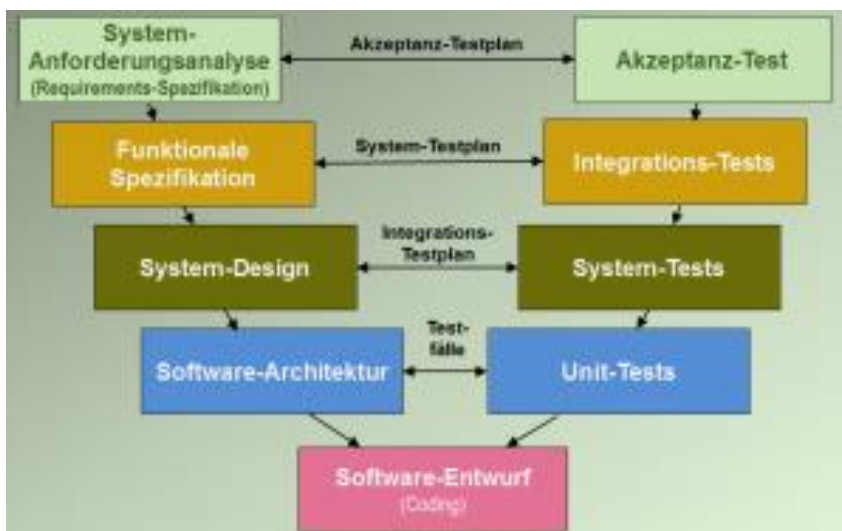


Bild 1: Projekt-Entwicklungsphasen im V-Modell

Bei der Entwicklung von Systemen in agilen Projekten sind diese Schritte verpflichtend:

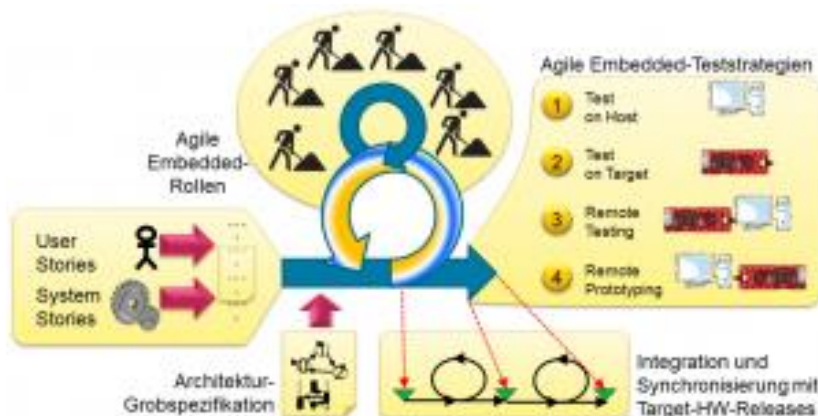


Bild 2: Entwicklungsphasen in agilen Projekten

Doch Software muss heute immer komplexere Abläufe steuern. Für die Software-Entwicklung ergeben sich daraus neue Herausforderungen, beispielsweise um die strengen Anforderungen von Safety und Security zu meistern.

Mit Multicore die Akku-Kapazität besser nutzen

Ein kurzer Blick zurück: Mit der Einführung von Notebooks und der mobilen Verfügbarkeit von Rechenleistung stieß man schnell an die Grenzen der erzielbaren Rechenleistung pro Watt, ausgedrückt in Millionen Instruktionen pro Sekunde (MIPS). Je größer die Rechenleistung, desto mehr wurde der Akku damit belastet. Der gleiche Prozess vollzog sich bei den Mobiltelefonen.

Die Rettung kam nicht nur in Gestalt riesiger Akkupacks (auch die gab es, wenn man sie tragen wollte), sondern mit der Entwicklung von Multicore-Prozessoren, die mehr Software-Pakete parallel abarbeiten konnten, ohne den Prozessor stärker zu belasten. Damit hielten die Akkus der mobilen Geräte länger und die Geräte wurden leistungsfähiger, ohne viel mehr Strom zu verbrauchen.

In der PC-Welt fand die Umstellung von Single- auf Multicore bereits Anfang des Jahrtausends statt. Für die ersten Multicore-Notebooks musste man damals viel Geld auf den Ladentisch legen. Und das, obwohl die verfügbare PC-Software – entwickelt für Singlecore-Architekturen – für diese Herausforderung noch nicht gerüstet war.

Ohne Multicore-Umgebungen sind neue Herausforderungen nicht zu bewältigen

An einem vergleichbaren Punkt stehen wir heute in der Embedded-Welt:



Bild 3: Multicore, Safety und Security und neue Herausforderungen

Während der Großteil der Embedded-Software weiterhin für Singlecore-Mikrocontroller entwickelt wird, stehen die neuen Herausforderungen bereits vor der Tür: Fahrerassistenzsysteme, autonomes Fahren, intelligente Ladekonzepte für Batterien im Auto und für Solaranlagen, sichere mobile Kommunikation und vieles mehr brauchen effiziente Software.

Ohne **Multicore-Umgebungen** sind diese neuen Herausforderungen nicht zu bewältigen. Dabei kommen zwei Aspekte besonders zum Tragen, an denen große Verantwortung seitens der Hersteller hängt:

- **Safety:** Die Systeme dürfen Anwender und ihre Umwelt nicht gefährden.
- **Security:** Applikationen und Daten sollen sicher verwahrt werden. Unzulässige und nicht gewünschte Zugriffe sollen in jedem Fall vermieden werden.

Multicore, Safety und Security bilden zusammen eine komplexe Ausgangslage für die Embedded-Programmierung. Die Suche nach klaren Antworten im Embedded-Bereich fällt dabei nicht immer zufriedenstellend aus, denn es gibt noch viel unbekanntes Terrain, wo diese drei Größen zusammentreffen.

Was müssen wir also in unseren Software-Projekten beachten und durchführen, damit am Ende qualitativ hochwertige Produkte entstehen, die in einem Multicore-System unter Berücksichtigung von Safety und Security zuverlässig abgearbeitet werden können?

1. Die **System-Anforderungen und die funktionale Spezifikation** (System Requirements, functional Specification) müssen klar getrennte Anforderungspfade für diese unterschiedlichen Herausforderungen aufweisen.
2. Die **Software-Architektur** hat entsprechend der unterschiedlichen Anforderungspfade unterschiedliche bzw. getrennte Module zu enthalten und deren logische Verbindungen aufzuzeigen. Sie muss also für die neuen komplexen Systeme Multicore, Safety und Security berücksichtigen und abbilden.
3. Der **Software-Entwurf** (sprich: die Programmierung) hat sich an die Regeln zu halten, die sich aus den Requirements ergeben. So müssen z.B. modulare/ prozedurale/ objektorientierte Programmierschritte zum Einsatz kommen. In einigen Fällen wird durch Software-Spezifikationen die Art der Programmierung genau festgelegt – Software-Entwicklung nach AUTOSAR, wie z.B. die API für Treiber nach dem AUTOSAR-Standard (MCAL-Treiber) und selbst entwickelte Complex-Treiber, die RTE-Erweiterungen für Multicore etc.
4. Der **Software-Test** (Unit-, System-Test) hat genau nach den Vorgaben der Requirements und der definierten Software-Architektur zu prüfen, ob alle definierten Qualitätsmerkmale (Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz Änderbarkeit, Übertragbarkeit usw.) eingehalten werden – insbesondere im Zusammenhang mit den strikten Realzeitanforderungen in der Embedded-Welt.

Die weiteren Teile dieser **Beitragsserie zum Thema Embedded-Software-Entwicklung unter den Aspekten Multicore, Safety und Security** beschäftigen sich näher mit den Punkten Requirements, Software-Architektur und Software-Tests. In [Teil 2](#) erfahren Sie, wie Sie dank umfassender Requirements typische Fehler vermeiden. In [Teil 3](#) geht es um das Thema Software-Architektur.

Holen Sie sich das richtige Wissen zu Embedded Software-Entwicklung, Multicore und Safety & Security. MicroConsult bietet Ihnen professionelle Trainings und Coachings rund um diese Themen an – im Live-Online- und im Präsenz-Format.

Weiterführende Infos

[MicroConsult Fachwissen: Multicore & Mikrocontroller](#)

[MicroConsult Training & Coaching: Multicore & Mikrocontroller](#)

[MicroConsult Fachwissen: Embedded SW-Entwicklung](#)

[MicroConsult Training & Coaching: Embedded SW-Entwicklung](#)

[MicroConsult Fachwissen: Safety & Security](#)

[MicroConsult Training & Coaching: Safety & Security](#)

[Alle MicroConsult Trainings & Coachings](#)

Autor

Ingo Pohle ist Mitgründer und Geschäftsführer der MicroConsult GmbH und international anerkannter Spezialist für Embedded-Lösungen, mit einem reichen Erfahrungsschatz rund um den Einsatz von Embedded-Mikrocontrollern, Bussystemen und RTOS.