

Entwicklung von Embedded-Systemen

Wie OEM und ihre Partner Komplexität meistern

Bild: berkahlinaertf/stock.adobe.com



Die Leistungsfähigkeit und Komplexität von Embedded-Systemen steigt unaufhörlich. Zukunftssichere Controller, HMI-Systeme, IoT-Gateways und Embedded-KI-Lösungen werden für OEM und Maschinenbauer immer wichtiger – und erfordern die Zusammenarbeit mit Partnern, die Entwicklungs-Tools, Software-Engineering und Projektmanagement auf hohem Niveau beherrschen. Von Oliver Roth

Die Embedded-Entwicklung hat sich in den letzten Jahren grundlegend verändert und wird sich auch in Zukunft verändern – nicht zuletzt durch weiter steigende Anforderungen an die Cybersicherheit und KI-Fähigkeiten einer wachsenden Zahl von Geräten und Systemen. Aus Sicht von OEM und Maschinenbauern bringt dies eine ganze Reihe von Herausforderungen mit sich:

Embedded-Systeme müssen heutzutage umfangreiche Funktionen erfüllen, anspruchsvolle Kommunikationsstandards unterstützen und strengeren regulatorischen Vorgaben entsprechen,

zum Beispiel in Bezug auf Cybersecurity und Remote Updates. Die primäre Herausforderung liegt also in einer zunehmend komplexen Software. Dazu tragen auch steigende regulatorische Anforderungen ihren Teil bei: Normen wie IEC 62443 oder die MDR (Medical Device Regulation) stellen hohe Anforderungen an die Entwicklungsprozesse und erfordern ein tiefgreifendes Verständnis der relevanten Standards. Komplexe Software erfordert leistungsfähige Hardware, die in zunehmender Vielfalt verfügbar ist: Moderne Embedded-Systeme beruhen auf einer

Vielzahl von Prozessorplattformen wie STM, ESP32, i.MX8, i.MX93/95 oder Allwinner. Dies erfordert skalierbare Hardware- und Software-Konzepte, die sich flexibel an unterschiedliche Anforderungen anpassen lassen. Aus den genannten Punkten folgt, dass die Entwicklung von Embedded-Lösungen eine enge Zusammenarbeit zwischen Hardware-, Software-, Konstruktions- und Mechanik-Experten erfordert. Die steigende Variantenvielfalt und der hohe Kostendruck lassen sich nur mit interdisziplinärem Projektmanagement bewältigen.

Strukturierung von Projekten: Hard- und Software parallel entwickeln

Diese Zusammenarbeit will professionell gemanagt werden. Dazu ist eine klare Projektstruktur ebenso unerlässlich wie der Einsatz einer durchdachten Tool-Landschaft.

Die Grundlage bildet dabei eine ganzheitliche Definitionsphase, in der ein detailliertes Lastenheft erstellt werden muss. Es definiert die Anforderungen an das zu entwickelnde Embedded-System und schafft die Basis für die – elementar wichtige – Parallelisierung der Hardware- und Software-Entwicklung. Die Hardware-Entwicklung erfolgt daraufhin auf Basis eines präzisen Pflichtenhefts, während die Software-Entwicklung agilen Methoden mit Epics, Backlogs und User Stories folgt. Doch egal, wie gut das Pflichtenheft und die gewählten Methoden auch sein mögen: Erst die optimale Tool-Landschaft unterstützt die Zusammenarbeit in verteilten Teams und ermöglicht ein effizientes Projektmanagement. Eine große Bedeutung kommt zudem der Wahl der Hardware-Tools zu.

Ein weiterer entscheidender Aspekt ist die Integration von Industrial Design und UX/UI-Design: Die frühzeitige Einbindung von Designern sichert die Benutzerfreundlichkeit und die ästhetische Gestaltung des Embedded-Systems. UX/UI-Design und Industrial Design sollten eng verzahnt sein, um eine nahtlose Überführung in die Software-Frontend/GUI-Entwicklung zu gewährleisten (zum Beispiel durch die Verwendung von Figma-Dateien).

Projektmanagement in der Embedded-Entwicklung

Wie gesagt, ist ein effektives Projektmanagement entscheidend für den Erfolg von Embedded-Projekten. Der OEM muss dabei die führende Rolle übernehmen und die Zusammenarbeit aller Beteiligten koordinieren. Sein Projektmanagement muss darum unter anderem die Bereiche Anforderungsmanagement, Managementvorgabe sowie Teamstruktur und -koordinieren berücksichtigen – besonders im Bereich Software-Entwicklung.

TOOL	EINSATZBEREICH	BEISPIELE
Altium 365	Hardware-Design	PCB-Design
SolidWorks	Mechanik-Design	Gehäusekonstruktion
GitHub	Software-Versionierung	Repository-Management
Jira	Agiles Projektmanagement	Backlog und Sprint-Planung
Confluence	Dokumentation	Projekt-/Wissensmanagement
Aligned Elements	Requirement Management	MDR-konforme Spezifikationen

Erst die optimale Tool-Landschaft unterstützt die Zusammenarbeit in verteilten Teams und ermöglicht ein effizientes Projektmanagement. (Quelle: Grossenbacher Systeme)

nation berücksichtigen – besonders im Bereich Software-Entwicklung.

→ Produktmanagement und Service liefern wertvolle Informationen aus den Anwendungsfeldern, die in die Definition der Anforderungen einfließen.

→ Das Management definiert klare Innovations- und Kostenziele.

→ Ein Projektleiter koordiniert die Teilprojektleiter für Hardware und Software. In der agilen Software-Entwicklung kann ein Product Owner die Entwicklung inhaltlich steuern und für eine zügige Umsetzung sorgen.

→ Die Koordination der Teams gewinnt an Bedeutung, wenn die Software-Entwicklung an verschiedenen Standorten oder Unternehmen erfolgt, zum Beispiel wenn Anwendungs-Software und Embedded-KI-Software in separaten Containern ausgeführt werden sollen.

→ Neben der Software-Entwicklung müssen häufig auch die Prozesse und Entwicklungsaktivitäten im Bereich der Mechanikkonstruktion (Design, Werkzeuge, Varianten) berücksichtigt und koordiniert werden.

Parallele Entwicklung von Hardware und Software

Wie erwähnt, sollte – oder muss – die Entwicklung nach der Definitionsphase in klar getrennten Strängen erfolgen: Zum einen sollte die Hardware iterativ auf Basis eines detaillierten Pflichtenhefts entwickelt werden. Für die Software-Entwicklung ist diese

bewährte Vorgehensweise zu langsam und unflexibel: Grossenbacher Systeme hat in diversen Projekten die Erfahrung gemacht, dass ein agiles Vorgehen mit einzelnen »Sprints« zum Erfolg führt. Dabei sind iterative Anpassungen durch den Einsatz von Epics, Backlogs und User Stories möglich. Somit sind die Managementmethoden für die Teilprojekte Hard- und Software grundverschieden – und gerade deshalb sehr förderlich für die Funktionalität des Endprodukts und das Erreichen der Time-to-Market-Ziele.

Lifecycle-Management: nachhaltige Planung

Selbst eine rundum erfolgreiche Entwicklung endet nicht mit einem »fertigen« Produkt. Den langfristigen und effizienten Betrieb von Embedded-Systemen sichert erst ein zukunftsorientiertes Lifecycle-Management, das Langzeitunterstützung, Updates, Wartung, Zulassungen und Compliance nicht nur mitdenkt, sondern vor allem konsequent umsetzt. Dazu müssen Komponenten und Software über Jahre hinweg aktualisiert und gewartet werden können, was wiederum sichere und unkomplizierte Remote-Updates voraussetzt. Zudem muss die Einhaltung relevanter Standards wie MDR kontinuierlich gewährleistet sein.

Rollenaufteilung: Wer macht was?

Der OEM sollte sich auf die ganzheitliche Produktdefinition konzentrieren –

schließlich weiß er am besten, was er braucht, um am Markt Erfolg zu haben. Der Entwicklungspartner sollte dagegen in der Lage sein, folgende Kompetenzen einzubringen:

Besonders wenn die letzte Basis-Produktentwicklung beim OEM längere Zeit zurückliegt und sich Anforderungen, Tools und Regularien stark verändert haben, sind interdisziplinäre Expertise und Tool-Erfahrung seitens des Entwicklungspartners unentbehrlich, um den OEM kompetent zu unterstützen. Gleichzeitig ist Innovationsfähigkeit gefragt – denn letztlich ist ein moderner »Baukasten« aus skalierbarer Hardware erforderlich – ergänzt durch eine breite Software-Modulbasis (Yocto, Cybersecurity, Remote Updates) – der die Entwicklung beschleunigt und die Flexibilität steigert.

Die strategische Bedeutung von KI legt deshalb nahe, dem Entwicklungspartner auch bei der Umsetzung von Embedded-KI zu vertrauen – schließlich ist sie die entscheidende Voraussetzung für lokale Prozessoptimierungen und zukunfts-sichere Automatisierungslösungen.

Fazit: Was einen kompetenten Entwicklungspartner auszeichnet

Die Entwicklung moderner Embedded-Systeme stellt OEMs vor Herausforderungen, denen sie durch die strategische Zusammenarbeit mit einem geeigneten Entwicklungspartner effektiv begegnen können. Doch woran erkennt man die Eignung, die es OEMs ermöglicht, sich auf ihre Kernkompetenzen zu konzentrieren, innovative und zukunftssichere Embedded-Lösungen auf den Markt zu bringen und die Wettbewerbsfähigkeit dauerhaft zu stärken? Nach Erfahrung von Grossenbacher Systeme sind es vor allem folgende Aspekte:

- Zugang zu spezialisiertem Know-how: Ein geeigneter Entwicklungspartner verfügt über umfassende Expertise in den Bereichen Hardware, Software, Mechanik und den relevanten Regularien.
- Nutzung effizienter Entwicklungsprozesse: Der Einsatz bewährter Methoden und Tools kann Entwicklungszeiten verkürzen und Kosten reduzieren. Das Projektmanagement sichert dies entsprechend ab.
- Affinität für skalierbare Lösungen: Ein kompetenter Partner bietet modulare und skalierbare Lösungen an, die sich flexibel an die individuellen Anforderungen des OEM anpassen lassen.
- Interesse an modernem Vertragswesen: Ein seriöser Partner wird Wert darauf legen, ein Vertragsverhältnis einzugehen, das der skizzierten zeitgemäßen Rollenverteilung bei der Entwicklung von Embedded-Systemen gerecht wird.

ak



Oliver Roth

ist Vorsitzender des Verwaltungsrats der Grossenbacher Systeme AG, St. Gallen (CH). (Bild: Grossenbacher Systeme AG)