

Return on Investment bei Tools

Den Nutzen des Einsatzes modellbasierter Werkzeuge individuell abschätzen

„Warum es sich lohnt, frühzeitig in modellbasierte Designlösungen für robuste und effiziente ECU-Architekturen zu investieren“ wäre eigentlich die bessere Überschrift für diesen Beitrag, aber leider ist sie viel zu lang. Der Tool-Anbieter Inchron erklärt hier, warum die rechtzeitige modellbasierte Analyse des dynamischen Echtzeitverhaltens auch aus kaufmännischer Sicht so wichtig ist. Mit einer kleinen Web-Lösung lässt sich der monetäre Nutzen des Einsatzes entsprechender modellbasierter Entwicklungslösungen sehr schnell abschätzen.

Autor: Uwe Brodtmann

Im Geschäftsleben stehen wir alle regelmäßig vor der Frage, ob sich eine Investition lohnt, oder neudeutsch, ob es einen positiven Return on Investment gibt. Der Frage „Lohnt sich ein Investment in modellbasierte Designlösungen?“ sind wir von Inchron im letzten Jahr gemeinsam mit Kunden systematisch auf den Grund gegangen. Wir wussten rein qualitativ um den Nutzen. Wie aber konnte man den Nutzen auch in Geld quantifizieren?

Die Antwort auf diese Frage resultiert aus zwei Perspektiven. Bei der einen Perspektive geht es um die Vermeidung von Redesigns durch frühzeitiges modellbasiertes Design, bei der zweiten Perspektive um die Vermeidung von überflüssigen Iterationen nach Change-Requests.

Frühzeitiger Einsatz von modellbasiertem Design

Zunächst geht es hier um die erste Perspektive. Mit dem Einsatz der Lösungen von Inchron lässt sich in einer frühen Projektphase, am besten schon in der Konzeptphase, feststellen, wie robust, skalierbar und effizient (im Sinne von Hardware-Ressourcenverbrauch) verschiedene Alternativen der Softwarearchitektur sind. Damit sinkt das Risiko, in einer späten Projektphase ein Redesign der Software-Architektur vornehmen zu müssen erheblich.

Soweit die qualitative Perspektive. Wie hoch sind aber die Kosten dieses Risikos? Den Ansatz einer Antwort auf diese Frage haben wir in einer Studie von VDC aus dem Jahr 2007 gefunden. Laut dieser Studie über Embedded-

Systems-Entwicklungsprojekte über verschiedene Branchen werden knapp 40 % der Projekte nur mit Verzögerung fertig, 10 % sogar ganz abgesagt.

Bei den Projekten, die mit Verzögerung fertig werden, beträgt die durchschnittliche Verzögerungszeit 4 Monate. Überträgt man nun diese Erkenntnis auf ein Projekt mit einer Laufzeit von 24 Monaten und einem Software-Entwicklungsbudget von 3 Millionen Euro, so ergeben sich folgende Zahlen: Wenn 24 Monate 3 Millionen Euro kosten, dann kostet 1 Monat 125.000 €. Bei 4 Monaten Verzögerung macht das 500.000 €. Diese Kosten entstehen aber in einem Projekt mit durchschnittlichem Risiko nur mit einer Wahrscheinlichkeit von 40 %. Das Kostenrisiko für das einzelne durchschnittliche Projekt beträgt somit 40 % von 500.000 €, mithin 200.000 €. Diese Größenordnungen wurden uns von all unseren Kunden mehr oder weniger bestätigt – allerdings mit einer Präzisierung. In einem Steuergeräteprojekt bedeutet Verzögerung natürlich nicht Verschiebung des SOP, sondern Mehraufwand, um den SOP zu halten; im schlimmsten Fall heißt es dann, mit reduziertem Funktionsumfang zu starten und nach SOP den vollen Umfang liefern. Auf jeden Fall entstehen Kosten, wenn etwas schief geht, und Projekte bergen immer ein mehr oder weniger großes Risiko.

Wie hoch ist das Risiko?

Welche Parameter bestimmen die Höhe des Risikos. In Diskussionen mit unseren



Bild: Eder von Rabenstein - Fotolia.com

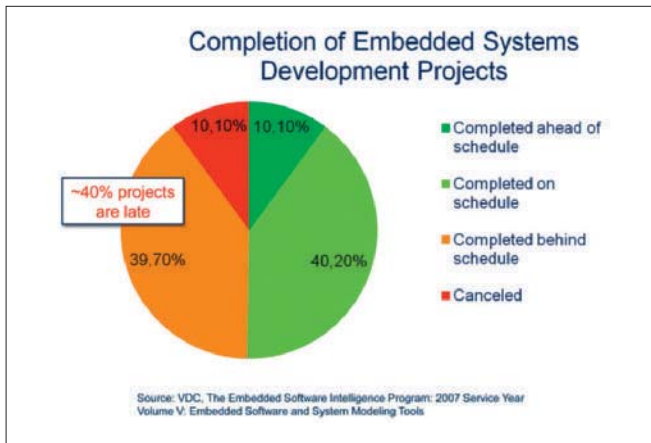


Bild 1: Wahrscheinlichkeit von Verzögerungen in Embedded-Systems-Projekten.

Kunden sind wir auf zwei Parameter gestoßen, deren Kombination das Risiko eines Projekts wesentlich bestimmen.

- Der erste Parameter ist die Komplexität des Projekts. Die Komplexität des Projekts ist eine Kombination aus der technischen Komplexität des zu entwickelnden Systems und der Komplexität der Organisation, die daran arbeitet. Letzteres gewinnt in der Automobilindustrie immer mehr an Bedeutung, weil zunehmend mehrere Parteien einschließlich des OEM Softwaremodule liefern, die dann integriert werden müssen.
- Der zweite Parameter ist der Neuheitsgrad, den ein Projekt einer gegebenen Komplexitätsstufe für die beteiligten Mitarbeiter hat. Dabei geht es nicht um Berufserfahrung im Allgemeinen, sondern um die Frage, wie oft eine Entwicklungsmannschaft ein solches Projekt schon einmal gemacht hat. Aus diesen beiden Parametern ergibt sich eine Bewertung, die aus der Matrix in Bild 2 ersichtlich ist.

Scoring-Modell hilft bei der Risiko-Einstufung komplexer Projekte

Die Frage, in welcher Risikoklasse sie ihr eigenes Projekt einstufen, können die meisten erfahrenen Projektmanager ziemlich schnell intuitiv beantworten. Es gibt aber auch ein Scoring-Modell, anhand dessen die Projektmanager Schritt für Schritt mit der Beantwortung von Fragen die Risikoklasse eines Projektes ableiten können. Dieses Scoring-Modell steht auf der Website von Inchron zur Verfügung. Stark vereinfacht gesagt birgt ein Projekt geringer Komplexität, das die Mannschaft schon mehrfach gemacht hat, nur sehr geringe Risiken. Mit steigender Komplexität entstehen Risiken allerdings auch, wenn bereits Erfahrungen vorliegen, und wenn man sich in einem komplexen Projekt auf Neuland begibt, steigen die Risiken deutlich. Dabei hat ein steigendes Risiko in diesem Konzept-Umfeld zwei Auswirkungen: So steigt die Wahrscheinlichkeit, dass etwas schiefgeht – und diese Wahrscheinlichkeit beträgt im Durchschnitt 40 % – und gleichzeitig steigen die Kosten, wenn etwas schiefgeht.

Für das eingangs genutzte Beispiel eines Drei-Millionen-Euro-Budgets können die Kostenrisiken von durchschnittlichen 200.000 € auf ein Mehrfaches wachsen, wenn das Projekt in einer entsprechend höheren Risikoklasse angesiedelt ist. In der Praxis bedeutet dies: Wenn ein Projekt in große Probleme läuft, wird schnell auch das Doppelte des Budgets oder noch mehr fällig, dies passiert aber Gott sein Dank nur in einer begrenzten Anzahl an Projekten.

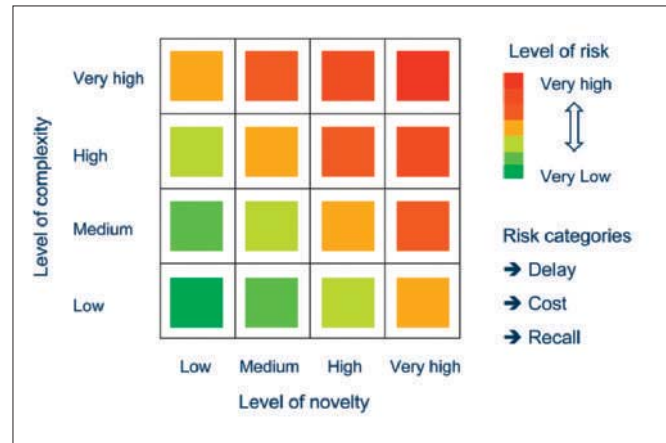


Bild 2: Risikoklassen in Abhängigkeit von Projektkomplexität und Neuheitsgrad.

Wenn man nun die Kostenrisiken eines Projekts insgesamt abgeleitet hat, stellt sich die Frage, in welcher Größenordnung der Einfluss eines Teilrisikos liegt. In diesem Fall ist eine fehleranfällige dynamische Softwarearchitektur das Teilrisiko. Auch dabei sind Expertenschätzungen ein pragmatischer Weg. Wir haben die Verantwortlichen befragt, um das Teilrisiko der dynamischen Softwarearchitektur zu ermitteln: In über 30 Interviews ergaben sich in Abhängigkeit vom jeweiligen Projekt Teilrisiko-Einschätzungen zwischen 10 % (Erweiterung bestehende Architektur) und 75 % (neues Multicore-Fahrerassistenzsystem), wobei die häufigsten Werte zwischen 40 und 50 % lagen. Damit betragen die Einsparungen in jedem Fall über 50.000 € pro Projekt, wenn nur die Perspektive 1 – frühzeitige Vermeidung von Fehlern, die später viel Geld kosten – in die Betrachtung einfließt. Allerdings gibt es auch noch die andere Perspektive.

Vermeidung von überflüssigen Iterationen nach Change-Requests

Die zweite Perspektive beleuchtet, was passiert, wenn es in einem laufenden Projekt Änderungswünsche (Change Requests) gibt, sei es durch den OEM, oder weil man beim Zulieferer eine vermeintlich bessere Lösung für eine Aufgabe hat. Welchen Nutzen bietet eine modellbasierte Betrachtung von Alternativen in diesem Fall? Wir haben uns zunächst gefragt: Welche Arbeitsschritte notwendig sind, um eine Änderung zu implementieren und wie hoch der Zeitbedarf für die Abarbeitung dieser Schritte ist. Dabei haben wir in Abstimmung mit den interviewten Kunden die folgenden fünf Schritte unterschieden:

- Überarbeitung/Anpassung der Anforderungen
- Code-Änderungen (egal ob beispielsweise manuell, modellbasiert oder Third-Party), Änderung der Autosar-Konfiguration (zum Beispiel Änderung der Scheduling-Parameter von Tasks oder Mapping von Runnables auf Tasks) und Verifikation (QAC-Check, virtuelles Target)
- Integration (Regressionstests, Feature-Tests, funktionale Stress-tests)
- Validierung
- Abstimmung OEM/Zulieferer

Für die einzelnen Arbeitsschritte haben wir nun in Interviews die jeweils von Kunden geschätzten Zeiten für die einzelnen Arbeitsschritte abgefragt. Die Antworten lagen in der Summe zwischen 10 und 19 Tagen mit einem Durchschnittswert von 12,7 Tagen für alle 5 Arbeitsschritte.

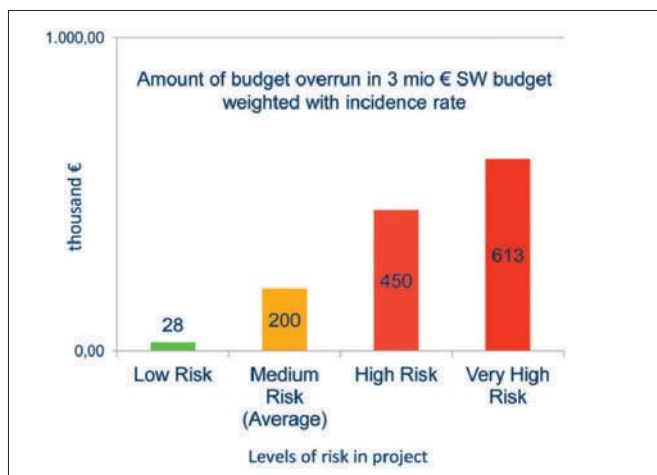


Bild 3: Kostenrisiken in Abhängigkeit von Risikoklassen eines Projekts.

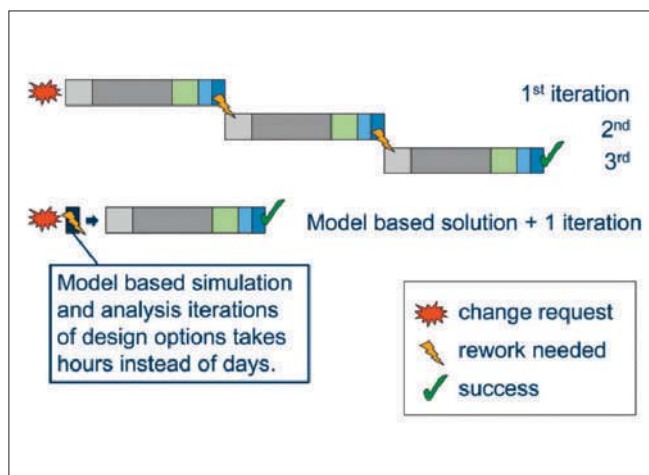


Bild 4: Iterationen bei konventioneller Vorgehensweise und bei modellbasierter Vorgehensweise.

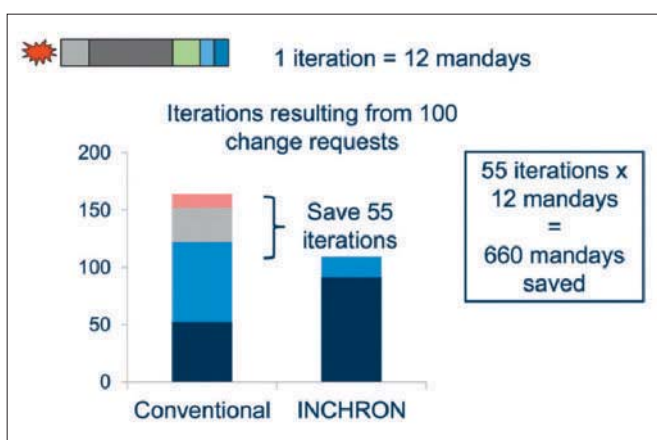


Bild 5: Einsparungen von Iterationen und Effekte auf Manntage.

Nun ist es aber in der Praxis leider so, dass nicht immer die erste Lösung funktioniert. Die nächste Frage lautete daher, in wie viel Prozent der Änderungen die implementierte Lösung erfolgreich ist, und in wie viel Prozent der Fälle einmalige oder mehrfache erforderlich war. Dem gegenüber steht der Vorteil der modellbasierten Simulation und Analyse von Lösungen, die in der Regel nur Stunden dauert. Erst wenn modellbasiert eine funktionstüchtige Lösung gefunden ist, beginnen die Verantwortlichen mit dem eigentlichen Implementierungsprozess, der dann wieder Tage dauert. Damit verringert sich im Vergleich zur konventionellen Vorgehensweise die Anzahl der Iterationen, die für Nacharbeiten notwendig sind. Aus den Interviews mit Kunden ergab sich für die Anzahl der erforderlichen Nacharbeits-Iterationen folgende Verteilung: In 52 % der Fälle funktioniert die erste Lösung. In 35 % der Fälle ist einmalige Nacharbeit erforderlich. Bei angenommenen 100 Änderungen sind somit 2×35 Iterationen = 70 Iterationen erforderlich. In 10 % der Fälle muss zweimal nachgearbeitet werden, und in seltenen 3 % der Fälle ist sogar dreimal Nacharbeit erforderlich.

Auch beim modellbasierten Ansatz kommt es hin und wieder zu Nacharbeiten – allerdings nur im Bereich von 10 % der Fälle. Es ergibt sich somit eine erhebliche Einsparung an Iterationen und damit an Manntagen. Genauso wie bei der Betrachtung der Projektrisiken galt es nun noch zu klären, in wie vielen Fällen Änderungswünsche (Change Requests) auch signifikante Auswirkungen auf das dynamische Verhalten eines Systems, also beispielsweise die Ausführungszeit eines Runnables in einer Task oder deren Aktivie-

rungsrate haben. Diese Begrenzung ist auch hier notwendig, weil die Lösungen von Inchron auf diese Perspektive fokussiert sind. Im Ergebnis der Interviews lag der Durchschnittswert bei 85 Änderungen mit Auswirkungen auf das dynamische Verhalten. In Verbindung mit den durchschnittlichen Einsparungen bei Iterationen ergab sich daraus eine Einsparung von 600 Manntagen. Geht man von einem internen Kostensatz pro Mitarbeiter von 500 € pro Tag aus, ergeben sich Einsparungen in Höhe von 300.000 Euro.

Vernünftiger Ansatz

Unser Fazit: Auch wenn die hier gewählten Perspektiven 1 (Vermeidung von Designrisiken) und 2 (Vermeidung von Iterationen bei Änderungswünschen) in der Praxis gewisse Überschneidungen aufweisen waren sich doch alle Kunden, die ihre Projekte mit uns durchgerechnet haben, darüber einig, dass der Ansatz vernünftig ist. Die Schätzungen kamen immer von langjährigen Fachleuten, von Projekt- und Teamleitern bis hin zum Hauptabteilungsleiter oder VP Entwicklung von Divisionen. Diese Schätzungen waren heruntergebrochen auf Einzelelemente, für die gute Erfahrungswerte vorliegen. Je nach Art des Projekts lagen die Einsparungen aus der Kombination von Risikovermeidung und Vermeidung von Iterationen nach Änderungen zwischen 150.000 Euro und 750.000 Euro. Selbst wenn man, was einige Kunden am Ende gemacht haben, noch einmal einen Sicherheitsabschlag von 20 % einrechnet, liegen die Einsparungen immer noch im sechsstelligen Bereich pro Projekt. (av) ■

Der Autor: Uwe Brodtmann ist CEO der Inchron GmbH.

Auf einen Blick

Dynamisches Echtzeitverhalten untersuchen?

Es gibt immer wieder ECU-Entwicklungsprojekte, die am Ende ihr Budget bei weitem überschreiten. Eine Quelle für Risiken liegt im dynamischen Echtzeitverhalten des Systems. Inzwischen ist in der Praxis mehrfach nachgewiesen, dass modellbasierte Designlösungen dabei helfen, hunderte an Manntagen und hunderttausende an Euro zu sparen. Ein einfaches Scoring-Modell hilft dabei, ein Gefühl für die Risiken eines Projektes zu bekommen. Auf Basis dieser Risikoeinschätzung lassen sich Einsparpotenziale relativ genau beziffern und Investitionsentscheidungen schlüssig begründen.

infoDIREKT www.all-electronics.de

338AEL0313