



□ Rüdiger Kreuter

([ruediger.kreuter@siemens.com](mailto:ruediger.kreuter@siemens.com))

leitet in der „Siemens Software Initiative“ die Entwicklung des System- und Software-Curriculum-Programms und koordiniert dieses Curriculum mit weiteren rollenbezogenen Ausbildungsprogrammen der Siemens AG. Er ist für die Entwicklung der Rolle des „System Architekten“ im Curriculum-Programm verantwortlich.



□ Dr. Frances Paulisch

([frances.paulisch@siemens.com](mailto:frances.paulisch@siemens.com))

ist Leiterin der „Siemens Software Initiative“, deren Aufgabe der konzernweite Einsatz von „Best-Practices“ im System- und Software-Engineering ist. Darüber hinaus engagiert sie sich in der internationalen Software-Engineering-Community, insbesondere in den Bereichen Architektur, Prozesse und Management.



□ Prof. Dr. Michael Stal

([michael.stal@siemens.com](mailto:michael.stal@siemens.com))

leitet als „Principal Engineer“ bei Siemens AG Corporate Technology in München Forschungs- und Kundenprojekte. Er hält Vorträge auf internationalen Konferenzen, ist Autor zahlreicher Fachartikel und Chefredakteur der Zeitschrift JavaSPEKTRUM. Michael Stal ist für die Entwicklung der Rolle „Softwarearchitektur“ im Siemens System- und Software-Curriculum verantwortlich.



□ Peter Zimmerer

([peter.zimmerer@siemens.com](mailto:peter.zimmerer@siemens.com))

ist „Principal Engineer“ bei Siemens AG Corporate Technology und im Siemens System- und Software-Curriculum für das Thema „Testen“ verantwortlich. Er ist zertifizierter ISTQB-Tester (Full Advanced Level) und häufiger Redner auf internationalen Konferenzen.

## System und Software: Architekturausbildung bei Siemens

Dieser Artikel gibt einen Überblick über ein praxisorientiertes System- und Softwarearchitekten-Curriculum und die Erfahrungen, die wir seit 2006 bei Siemens damit gemacht haben. Der Begriff System beschränkt sich hier nicht auf IT-Systeme, sondern auf alle technischen Systeme im Fokus, z. B. auch Steuerungstechnik, Medizintechnik, Züge oder Kraftwerke. Mit dem Curriculum verfolgt Siemens zwei wesentliche Ziele: Zum einen soll die Qualifikation erfahrener System- und Softwarearchitekten weiter verbessert werden, zum anderen soll durch ein Pre-Assessment und eine Zertifizierung sichergestellt werden, dass die System- und Softwarearchitekten über die technischen und sozialen Fähigkeiten verfügen, die sie benötigen, um Systeme oder Software mit besonders kritischen oder innovativen Anforderungen entwickeln zu können. Dafür kombinieren wir Qualifizierungsmaßnahmen für verschiedene Themengebiete mit praktischen Übungen in den aktuellen Projekten der Teilnehmer. Insgesamt erstreckt sich ein Curriculum über einen Zeitraum von ca. einem Jahr.

Da Software in immer mehr Bereichen Einzug hält, die Systeme immer komplexer werden und immer stärker ineinander greifen, steigt der Bedarf nach guten System- und Softwareingenieuren stetig. Zusätzlich zur gestiegenen Komplexität sollen Systeme und Software immer flexibler

sein, die Produkteinführungszeiten (*Time-to-Market*) werden immer kürzer und die Entwicklungs-, Fertigungs- und Lebenszykluszeiten sollen stetig sinken.

All diese Faktoren zusammen genommen üben einen großen Druck auf die Entwicklungsteams aus. Viele aktuelle Ansätze

im Software Engineering – z. B. agile Prozesse, modellgetriebene Entwicklung (*Model-Driven Development, MDD*) – setzen sich mit diesen Herausforderungen auseinander. Auch in der Systementwicklung wird zunehmend ein iterativ-inkrementeller Ansatz durchgeführt – Multi-

Physik-Simulationen erlauben z. B. eine frühzeitige Systemintegration.

Gute (Software-)Engineering-Fertigkeiten waren schon immer wichtig, aber veränderte Anforderungen an Software und Systeme haben natürlich auch Auswirkungen auf die erforderlichen Qualifikationen von System- und Softwareingenieuren. Zum einen müssen diese heute neue technische Entwicklungen bewerten und – wenn notwendig – auch beherrschen. Zum anderen müssen sie dazu in der Lage sein, mehr als nur ein einzelnes Thema (Lösungsfindung) zu verstehen.

Das trifft insbesondere auf große Projekte mit großer geschäftlicher Bedeutung zu, die einen hohen Grad an Innovation und Komplexität aufweisen oder die auf einem Produktlinienansatz basieren. Neben technischen Fähigkeiten spielen auch die sozialen Kompetenzen (*SoftSkills*) eine entscheidende Rolle. Viele Projekte scheitern nicht aus technischen Gründen, sondern weil z. B. der Kommunikation und der Teamführung im Projekt nicht genügend Beachtung geschenkt wird. Ein besseres Geschäftsverständnis verhilft zu einer effektiveren Kommunikation mit dem Produktmanagement in strategischen Diskussionen. Das wiederum führt dazu, dass langfristige Architekturentscheidungen auf einer verlässlicheren Basis getroffen werden.

Man weiß heute, dass die Zusammenarbeit sowohl im Team als auch mit wichtigen Projekt-Stakeholdern ein Schlüsselfaktor für eine erfolgreiche Entwicklung ist. Besonders agile Methoden betonen das (vgl. [Agi]). Häufig werden aber Details aus der realen Welt (z. B., wie die verschiedenen Beteiligten in einem Entwicklungsprojekt zusammenarbeiten sollten) außer Acht gelassen. In dem hier vorge-

stellten Curriculum haben wir uns besonders mit diesen Fragen auseinandergesetzt; hierzu gehört z. B. eine aktive Beteiligung eines System- oder Softwarearchitekten bei Anforderungs- oder Testaktivitäten.

Insgesamt arbeiten wir an einer Veränderung des Mindsets. Der System- bzw. Softwarearchitekt soll umfassende Verantwortung übernehmen, bei Problemen in der Organisation aktiv zur Lösung beitragen und dazu die geeigneten Stakeholder involvieren. Dazu muss er deren Sprache sprechen.

Die Rolle des System- bzw. Softwarearchitekten ist das zentrale Thema – quasi „der rote Faden“, der sich durch das gesamte Projekt zieht – und hängt eng mit vielen anderen Aufgabengebieten zusammen, z. B. dem *Requirements-Engineering (RE)*, Softwareprozessen, Testen und Qualitätssicherung sowie dem Verständnis der Geschäftsstrategien. Im Gegensatz zu der Auffassung, dass Architektur nur eine Phase in der Produktentwicklung ist, sehen wir die Rolle des Architekten im gesamten Lebenszyklus aktiv – von der Business-Case-Analyse, über die Anforderungsanalyse, bis hin zum Ausphasen des Produkts bzw. zur Systemevolution. Die System- und Softwarearchitekten sehen wir, im engen Schulterschluss mit den Produkt- und Projektmanagern, als die treibenden Rollen in der technischen Definition und Realisierung eines (Software-) Systems. Der Hauptfokus ihrer Arbeit erstreckt sich von der Anforderungsanalyse bis zur Übergabe an die Fertigung bzw. an Vertrieb und Service.

Aber auch in den Phasen davor und danach sehen wir die Architekten aktiv wirkend. Frühe Architekturentscheidungen haben langfristige Auswirkungen auf das

Projekt – daher müssen, wenn Architekturentscheidungen getroffen werden, alle relevanten Aspekte in Betracht gezogen werden und die Entscheidungen sollten auf einem gemeinsamen Verständnis aller Beteiligten basieren. Ein Architekt muss daher nicht nur über die technische Qualifikation verfügen, um die richtigen Entscheidungen zu treffen, er muss außerdem auch dazu in der Lage sein, mit den anderen Beteiligten – wie z. B. den Produkt- und den Projektmanagern, den Anforderungsanalysten und Testern, sowie den Verantwortlichen für späte Phasen (Fertigung, Betrieb, Wartung, Service, ...) – zu kommunizieren und zusammenzuarbeiten. Darüber hinaus sollten die System- und Softwarearchitekten dem Entwicklerteam als Führungs- und Motivationsquelle dienen.

**Das Curriculum-Projekt**

Siemens beschäftigt weltweit ca. 20.000 Softwareingenieure und entwickelt Software für ein sehr breites Spektrum, in erster Linie für eingebettete, softwarebasierte Systeme. Wegen der herausragenden Bedeutung von Software für das Kerngeschäft von Siemens haben wir in 2006 ein Projekt aufgesetzt, dessen Ziel es war, ein Qualifizierungsprogramm für Schlüsselrollen in der Entwicklung zu entwickeln, das „Software-Curriculum“. Da die Softwarearchitektur einen maßgeblichen Einfluss auf den wirtschaftlichen Erfolg von großen Softwareentwicklungsprojekten hat, war die erste Rolle im Fokus die des *Senior Soft Ware Architect (SSWA)*. Dieser ist Softwarearchitekt für besonders wichtige oder umfangreiche Softwaresysteme, z. B. einer Produktlinie, mit weitreichenden Auswirkungen auf das Geschäft.

In der Folge wurden weitere Qualifizierungsprogramme für den *Software Architect (SWA)* und den *System Architect (SyA)* entwickelt; **Abbildung 1** zeigt einen Überblick über die Qualifizierungs-Pyramide für unsere Architekten. Die meisten Systeme, die bei Siemens entwickelt werden, sind Software-basiert. Es gibt jedoch auch viele Systeme, die nach wie vor Hardware-dominiert sind. Bei solchen Systemen überwiegen in der Regel Fertigungs-, Wartungs-, Energie-, Verbrauchsmaterial- und Verschrottungskosten bei weitem die Entwicklungskosten des Systems. Um auch die Architekten solcher Systeme mit in das Curriculum-Projekt einzubeziehen, wurde das „Software-Curriculum“ in ein „System- und Software-Curriculum“ erweitert. Das hat Auswir-

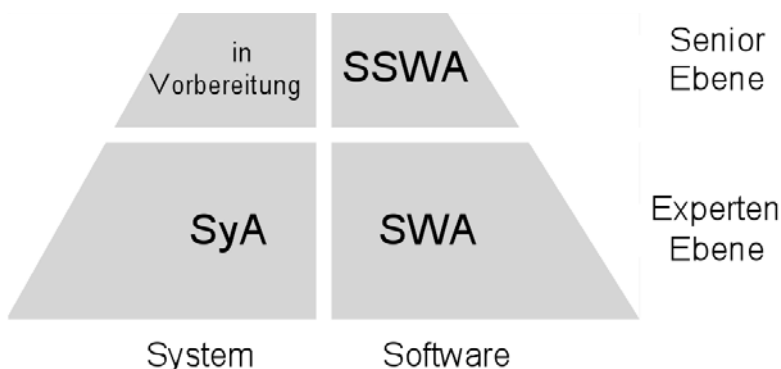


Abb. 1: Architekten-Qualifizierungsprogramme im Überblick

kungen auf alle Kursanteile; Testbarkeit im System bedeutet z. B. nicht nur Testbarkeit während der Entwicklung oder im Betrieb, sondern auch während der Fertigung und der Installation. Ein Curriculum für Senior System-Architekten wird derzeit konzipiert.

Vor der Entwicklung eigener Qualifizierungsprogramme haben wir zunächst analysiert, ob es bereits adäquate Qualifizierungsprogramme im freien Markt gibt, die unseren Anforderungen entsprechen (Stichwort „make or buy“). Von einigen rollenspezifischen Qualifizierungsprogrammen, die wir gefunden haben, z. B. dem umfangreichen Requirements-Engineering-Training von Intel (vgl. [Sim05]), waren wir sehr beeindruckt. Für den SyA haben wir beispielsweise auch das Zertifizierungsprogramm der INCOSE (vgl. [INCOSE], [GfSE]) und verschiedene Master-Aufbaustudiengänge für Systems Engineering angesehen. Wir haben aber kein Programm gefunden, das bzgl. des Aufwands angemessen ist, alle Verantwortlichkeiten, die wir bei System- oder Softwarearchitekten sehen, umfasst und welches die Umsetzung des gelernten Wissens in den Projekten der Teilnehmer enthält. Die letzten beiden Punkte sind unserer Meinung nach grundlegend für die Wirksamkeit eines solchen Qualifizierungsprogramms für (Senior-) Experten.

### Das Curriculum-Konzept

Alle unsere Architektenprogramme haben einen Eingangsprozess, in dem die Kursteilnehmer ausgewählt werden, einen mehrteiligen Kurs, gemischt mit Projektarbeit, und eine Zertifizierung.

#### Eingangsprozess

Die Kursteilnehmer müssen eine Reihe von Eingangskriterien erfüllen, um an einem der Programme teilnehmen zu können.

Für das *Experten-Programm* umfasst der Eingangsprozess eine fragebogengestützte Selbsteinschätzung, eine Einschätzung der Führungskraft und der Personalabteilung, sowie letztlich ein Interview mit einem Senior-Architekten (i. d. R. einer der Kurstrainer).

Die wesentlichen Aufnahmekriterien sind folgende:

- Die Kandidaten arbeiten aktuell als Architekten in ihrer Organisationseinheit und sind als solche in ein laufendes Projekt involviert.

- Sie haben mindestens 2-3 Jahre Berufserfahrung als Architekt oder in einer ähnlichen Rolle.
- Sie verfügen über die erforderlichen fachlichen Eingangsqualifikationen.

Für das *Senior-Programm* erfolgt die Überprüfung, ob die Eingangskriterien erfüllt sind, d. h. ob sie unserem Softwarearchitekten-Profil entsprechen, im Rahmen einer Bewerbung. Diese besteht aus einem Nominierungsbrief von der Unternehmensspitze und einem Vorstellungsgespräch mit den Kandidaten, um die für das Ausbildungsprogramm erforderlichen homogenen, hohen Ausgangsvoraussetzungen sicherzustellen.

Die wesentlichen Aufnahmekriterien sind hier:

- Die Teilnehmer sind leitende Architekten (*Key Architects*) in ihrer Organisationseinheit.
- Sie haben mindestens fünf Jahre Berufserfahrung als Architekt.
- Sie arbeiten aktuell in einem entsprechenden produktlinienbasierten Projekt mit hoher geschäftlicher Bedeutung für die Organisationseinheit.
- Sie verfügen über die erforderlichen grundlegenden Eingangsqualifikationen.

#### Kurs

Wenn ein Architekt in das Ausbildungsprogramm aufgenommen wird, absolviert er ein ca. neunmonatiges Programm, das aus einer Mischung von mehrtägigen Trainingseinheiten und mehrmonatigen Projektphasen besteht. Die Schulungsinhalte setzen sich aus 5 Themengebieten zusammen:

- Geschäftsverständnis
- Anforderungsmanagement
- Architektur / Lösungsfindung
- Testen und Qualität
- Soziale Kompetenz

In allen Workshop-Sessions gilt, dass der (inter-)aktiven Zusammenarbeit der Vorzug vor Vorträgen gegeben wird. Im Wechselspiel zwischen Wissenspräsentation und Anwendung oder Erfahrungsaustausch in Gruppenarbeiten werden die Kursinhalte intensiv eingepreßt. In den Projektphasen wird das erlernte Wissen mittels Hausaufgaben in den eigenen Projekten angewendet.

#### Zertifizierung

Für die Programme auf *Expertenebene* (SWA / SyA) gibt es mehrere Monate nach der letzten Trainingseinheit eine Zertifizierung, in Form eines Interviews mit versierten unabhängigen Assessoren. Im Zertifizierungsinterview liegt der Schwerpunkt darauf, zu analysieren, wie gut der Kandidat das erlernte Wissen in der Praxis umsetzen bzw. anwenden kann.

Für die Programme auf *Seniorebene* ist das Zertifizierungsverfahren aufwendiger. In einer Reihe von themenspezifischen *Knowledge Gates* muss jeder Teilnehmer zeigen, dass er das Gelernte in seinen Projekten auch anwenden kann. In einem sogenannten *Capability Gate* muss der Teilnehmer in einem Interview mit Psychologen und Entwicklungsleitern seine soziale Kompetenz (*Soft Skills*) und seine Teamführungsfähigkeiten unter Beweis stellen. Erst wenn er beide Gates (*Knowledge* und *Capability*) passiert hat, erhält er sein SSWA-Zertifikat. Um die Bedeutung des Ausbildungsprogramms zu unterstreichen, wird das Zertifikat vom CEO der Organisationseinheit, in der der Teilnehmer tätig ist, unterzeichnet.

Für den Begriff „Zertifizierung“ gibt es viele verschiedene Interpretationen. Wir achten sehr darauf, dass die Zertifizierung von Rollen bei Siemens auf wirklichen Projekten basiert, dass sie in den Senior-Programmen über einen langen Zeitraum erfolgt und verschiedene Fähigkeiten, einschließlich *Soft Skills*, umfasst. Hierdurch unterscheidet sich das Siemens-Programm von vielen anderen externen Programmen.

#### Zeitlicher Verlauf

**Abbildung 2a** gibt einen Überblick über den zeitlichen Verlauf der Experten-Programme. **Abbildung 2b** gibt einen Überblick über den zeitlichen Verlauf des Senior-Programms.

Die Kursteilnehmer arbeiten in den 3-4-monatigen Projektphasen zwischen den 3-5-tägigen Kurs-Workshops in ihrem aktuellen Projekt weiter. Über Hausaufgaben wird das erlernte Wissen im Kontext des eigenen Projekts angewendet. Teilnehmer von Senior-Programmen bekommen hierbei einen erfahrenen Architekten zur Seite gestellt, der als Mentor fungiert. Wie die Erfahrung gezeigt hat, führt gerade diese Konstellation dazu, dass Kollegen und Vorgesetzte bereits während des Kurses eine positive Veränderung des Arbeitsverhaltens eines Kursteilnehmers wahrnehmen.

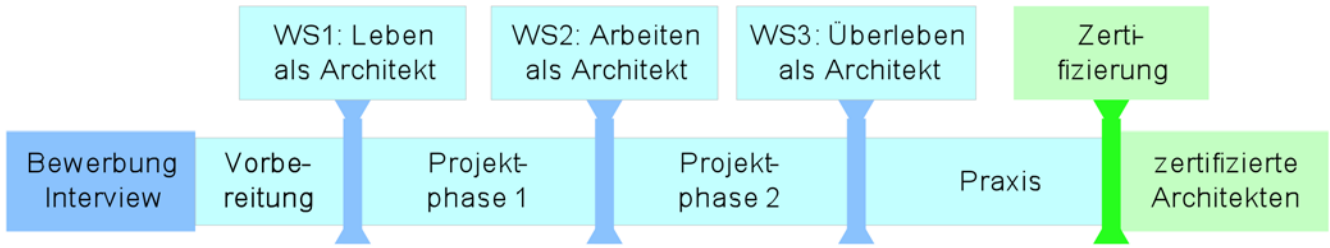


Abb. 2a: Ablauf der Qualifizierungsprogramme für System- und Softwarearchitekten

**Architekten-Netzwerk**

Einige Aktivitäten in den Projektphasen werden im Team – gemeinsam mit den anderen Teilnehmern des Programms – durchgeführt. Das fördert den Aufbau eines Netzwerks – sowohl zwischen den aktuellen Teilnehmern, als auch zu den früheren Teilnehmern des Qualifizierungsprogramms. Durch jährlich durchgeführte *Alumni-Treffen* wird dieses Netzwerk auch langfristig gefördert.

*State-of-the-Art-Qualifizierungsprogramme* und der Aufbau eines organisationsübergreifenden Netzwerks bildenden Schlüssel, um ein gemeinsames Verständnis über eine moderne System- und Softwareentwicklung bei Siemens weltweit zu etablieren.

**Guiding Principles und R&D Manager-Workshop**

Wir sehen die Architekten in einer führenden Rolle, wenn es um die Entwicklung und Umsetzung des technischen Konzepts eines neuen (Software-)Systems geht. Damit die Architekten auch die dafür notwendigen Freiheiten und Rechte bekommen, wurde ein eintägiger Workshop für Führungskräfte (z. B. Entwicklungs- und Geschäftsfeldleiter) entwickelt. In diesem werden die Curriculum-Programme vorgestellt und die Rolle, die die Architekten in ihren Organisationen spielen wird mit

den Führungskräften diskutiert. Ziel ist es, die Rolle des System- und Softwarearchitekten in der Siemens AG zu etablieren und zu stärken. Als weitere Unterstützung für die Architekten haben wir 12 *Guiding Principles* für System- und Softwareentwicklung aufgestellt, die auf eine professionelle, systematische und agile Entwicklung von (Software-) Systemen abzielen. Diese dienen auch als Leitfaden für die Entwicklung der Kursinhalte.

**Inhalt der Architektenprogramme im Detail**

Der von uns verfolgte holistische Ansatz, verschiedene Themen in das Training einzubeziehen, ist unserer Meinung nach für erfolgreiche Projekte sehr wichtig. Für die im Folgenden näher vorgestellten Hauptthemenbereiche wurde im Vorfeld genau definiert, über welchen Kompetenzlevel die jeweiligen Architekten verfügen sollten. Daraus wurde die Intensität abgeleitet, mit der die Themen in den einzelnen Qualifizierungsprogrammen behandelt werden.

Zum Kursbeginn werden die persönlichen Herausforderungen der Teilnehmer abgefragt (Was geht typischerweise in den Projekten schief?). Auch Gespräche über persönliche Fehler in eigenen Projekten, um aus den Fehlern lernen können, gehören zur Seniorität und dienen häufig als Motivation

für neue Ansätze zu bekannten Themen. Im weiteren Verlauf wird durch die Trainer immer wieder darauf Bezug genommen (Wie lassen sich solche Probleme vermeiden, wie können die Dinge aus einer anderen Perspektive betrachtet werden, wo gibt es „blinde Flecken“?). Schließlich findet ein Erfahrungsaustausch zwischen den Kursteilnehmern statt. Auch in den Projektphasen tauschen sich die Teilnehmer mit anderen Teilnehmern und Architekten aus und machen *Peer Sparring* miteinander. *Peer Sparring* ist eine Mischung aus Coaching und einem gegenseitigem Challenge bezüglich der eigenen Ideen und Lösungsansätze. Nach diesen Erfahrungen sind die Teilnehmer wieder aufnahmebereit für neue Ideen im nächsten Workshop.

**Geschäftsverständnis**

Der Architekt muss nicht nur die ökonomische Bedeutung der Architektur verstehen, sondern auch die zugrundeliegende Geschäftsstrategie. Insbesondere in den Senior-Programmen sind auch die Zusammenhänge zwischen den Produkten ein wichtiger Aspekt, um zu beurteilen, ob eine Produktlinienentwicklung aus geschäftlicher und aus technischer Perspektive sinnvoll ist oder nicht. Nur wenn der Architekt in enger Zusammenarbeit mit dem Produktmanager die Kundenwünsche, die geplanten Features und langfris-

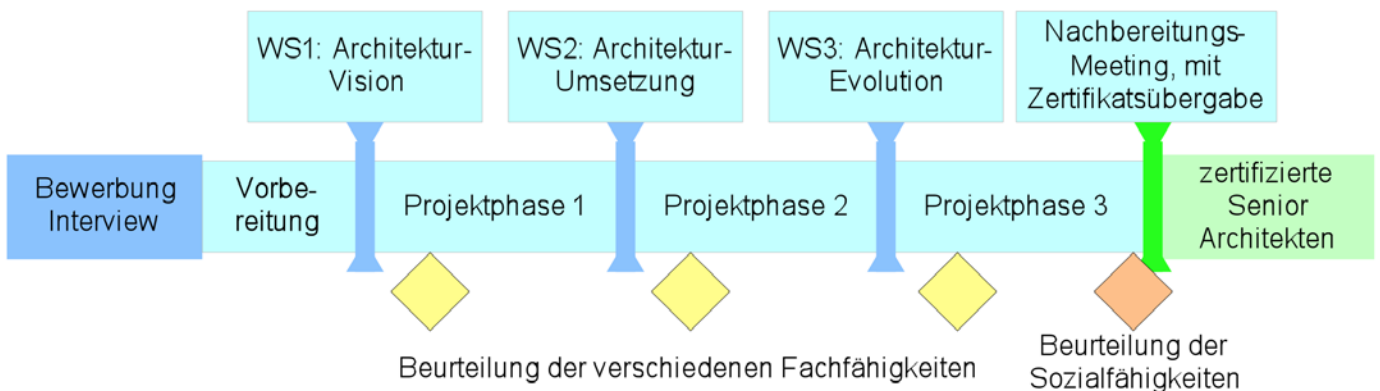


Abb. 2b: Ablauf der Qualifizierungsprogramme für Senior-Programm

tigen Trends kennt, kann er seine Architektur langfristig danach ausrichten.

### Requirements-Engineering (RE)

Beim Thema RE geht es vor allem darum, Brücken zwischen normalerweise eher getrennten Kulturen zu schlagen. Produktmanager und Requirements-Ingenieure sind oft der Meinung, dass es zwischen ihrer Arbeit und architektonischen Belangen kaum Zusammenhänge gibt. Architekten müssen daher auf ihre primär kundenorientierten Kollegen zugehen, die unter Umständen eine einseitige Sichtweise auf mögliche Lösungstechnologien haben. Die Aufgabe des Architekten besteht hier vor allem darin, ein klares Verständnis der Kunden- oder Marktanforderungen herzustellen und (frühzeitig) ein Feedback bezüglich technischer und ökonomischer Machbarkeit zu geben. Häufig werden initial firmeninterne Anforderungen „übersehen“, die aber häufig gravierende Auswirkungen auf die Architektur haben, die nicht-funktionalen Anforderungen (NFRs), wie z. B. Performance, Verfügbarkeit, Sicherheit, Skalierbarkeit, Fertigbarkeit, Installierbarkeit, oder Servicebarkeit um nur einige zu nennen.

Es ist wichtig, dass Architekten die Komplexität der Anwendungsdomäne des Kunden und die damit zusammenhängende Notwendigkeit eines Anforderungsmanagements gut verstehen. Schließlich sollten sich Architekten bewusst sein, dass die effektive Konsolidierung und Selektion der Anforderungen eine Schlüsselaufgabe ist.

### Systemarchitektur (nur SyA)

Im Systemarchitekturprogramm geht es, wie eingangs beschrieben, um *Embedded Systems* oder *hardware-dominierte Systeme*, nicht (explizit) um IT-Systeme. Kerninhalte sind die Bedeutung der Rolle des Systemarchitekten in der Systementwicklung sowie wichtige Systementwicklungskonzepte (z. B. V-Modell, iterativ inkrementelle Entwicklung, Lean Product Development).

Im *ersten Workshop* wird die Architekturentwicklung, neben RE und Testen, in zwei Schritten behandelt, der Situationsanalyse und dem systematischen strategischen Architektur-Design. Kern der Situationsanalyse (vgl. [Hab]) sind der Systemkontext, die Abgrenzung und die Problemanalyse. Im Architektur-Design geht es um die Lösungsfindung, um die Entwicklung verschiedener Varianten und der systematischen Auswahl, z. B. mit Me-

thoden wie Design Space Exploration. Letztlich gehört dazu auch die Diskussion, wie eine Architektur in die Entwicklungsmannschaft transferiert und dort am Leben erhalten wird.

Im *zweiten Workshop* geht es um hilfreiche Methoden, wie Machbarkeitsanalysen (technisch und ökonomisch), die Nutzung vorhandenen Wissens, Strukturoptimierungsmethoden (z. B. Design-Struktur-Matrizen), Modellierung und Simulation, Design-Taktiken und letztlich die Dokumentation der Architektur als Kommunikationsmittel.

Der *dritte Workshop* fokussiert Probleme aus späten Produktlebenszyklusphasen, die der Architekt minimieren kann, sowie den Umgang mit Unwägbarkeiten und Architektur-Qualitätsprobleme.

### Softwarearchitektur (nur SWA / SSWA)

Es gibt viele Bücher und Seminare zum Thema Softwarearchitektur, aber keines behandelt wirklich alle relevanten Fragen, z. B., wie man eine Softwarearchitektur systematisch erstellen kann oder wie man vorhandene Systeme beurteilen, prüfen oder refaktorisieren kann. Zumeist befassen sich die Bücher und Seminare mit vielen Themen, die für einen Softwarearchitekten mit mehreren Jahren Berufserfahrung keinerlei Mehrwert liefern. Hierzu gehören beispielsweise die Unified Modeling Language (UML) und bestimmte technische Erfahrungen, wie z. B. die Verteilung von Middleware. Deshalb war es eine große Herausforderung, adäquate Inhalte für erfahrene Softwareingenieure anzubieten. Für die im Rahmen des Curriculums durchgeführten Workshops haben wir die folgenden Themen identifiziert.

Zunächst wird auf die Rolle und Verantwortung des Softwarearchitekten eingegangen. Insbesondere darauf, welche unterschiedlichen Qualifikationen in den Entwicklungsprozessen, im RE und im Geschäftsverständnis gefordert sind, womit diese zusammenhängen und wo diese benötigt werden. In einem umfangreichen Teil des Workshops erhalten die Teilnehmer ein Framework, mit dem sie eine Softwarearchitektur systematisch entwickeln können. Im Senior-Programm steht hier die Plattform- und Produktlinienentwicklung im Fokus: Wie sollten Architekten den Geltungsbereich der Softwarearchitektur festlegen? Welche Aktivitäten sind notwendig, um eine Softwarearchitektur agil, anforderungs-, geschäfts- und risikogetrieben zu erstellen? Beispielsweise müs-

sen Architekten andere von ihren Ideen überzeugen und sie für ihre Sicht der Dinge gewinnen. Sie müssen sicherstellen, dass die Entwickler die Architekturvision verstehen, sich aktiv an der Softwareentwicklung beteiligen und von den anderen Entwicklern Feedback einfordern.

Ein weiterer Schwerpunkt besteht darin, die Teilnehmer anzuleiten, wie man die Architektur tatsächlich realisieren kann. Wie würden Architekten jene Teile des Systems analysieren, die Veränderungen unterliegen? Der Abschnitt „Entwurfsstrategien“ (*Design Tactics*) in dem Workshop umfasst Muster (Patterns), d. h. Methoden und Praktiken, um qualitative Attribute und nicht-funktionale Aspekte einer Architektur zu ermitteln. Abschließend lernen die Teilnehmer, wie man ein Architektur-Review vornimmt. Dabei erfahren sie, wie sie vorhandene Softwarearchitekturen systematisch bewerten können.

In einem dritten Themenblock geht es um Nachhaltigkeit. Die Teilnehmer lernen, wann und wie ein Architektur-Refactoring durchgeführt werden sollte, und erhalten Information über die Durchführbarkeit und Durchführung eines Re-Engineering. Außerdem werden beide Methoden erklärt und einander gegenübergestellt. Den Abschluss bilden Überlegungen zur Qualität von Design und Architektur, insbesondere Ausdrucksstärke, Symmetrie und Emergenz.

### Testen

Der Bereich „Testen“ umfasst eine Reihe von Themen, die für Architekten von Bedeutung sind und die klassische Fragestellungen im Projektalltag darstellen. Einige der Testthemen sind eng miteinander verknüpft und können von einer Beteiligung der Architekten nur profitieren.

Zunächst ist es wichtig, die Mission, Motivation und den (geschäftlichen) Wert des Testens zu verstehen – und dieses Verständnis dann innerhalb des Entwicklungsteams auch vermitteln und fördern zu können.

Darauf aufbauend werden die Grundzüge einer risikobasierten Teststrategie vermittelt. Hierbei sollte der Architekt als wichtiger Stakeholder aktiv an Produkt- und Risikoanalyse-Workshops teilnehmen und anschließend die identifizierten und priorisierten architektonischen Risiken auf entsprechende Teststufen abbilden. Die daraus abgeleiteten Architekturtests sind dann auf unterschiedlichen Ebenen – und natürlich möglichst früh – im Produktle-

benszyklus durchzuführen (vgl. „Architecture Testing“ in [Goo08]).

Eine wichtige Teststufe für den Architekten ist hierbei der Integrationstest – hier zeigt sich, ob die Architektur tragfähig ist oder nicht. Testgetriebene Entwicklung (*Test Driven Design, TDD*) wird heutzutage in der Regel bei Unit- und Akzeptanz-Tests eingesetzt. Für den Architekten ist insbesondere das zugrunde liegende Prinzip des präventiven Testens (*Preventive Testing*, vgl. [Gel88]) interessant, das die testgetriebene Erstellung von Artefakten z. B. auf Architektur- und Designebene beinhaltet – mit allen Vorteilen, die eine testgetriebene Denkweise mit sich bringt.

Für Testaktivitäten wird ein beträchtlicher Teil des Projektbudgets aufgewendet. Wird ehrlich gemessen, dann kommt man schnell auf 50 % (bei Wartungsprojekten häufig noch mehr). Ein entscheidender Hebel zur Beeinflussung dieser Kosten ist die Testbarkeit des Systems: Testbarkeit muss als nicht-funktionale Anforderung von Anfang an betrachtet werden und sie muss durch den Architekten (!) in die Architektur und das Design eingebaut werden. Insbesondere die Kosten und der Zeitaufwand für die Automatisierung von Tests und für die Fehlerdiagnose sind extrem von der zugrunde liegenden Testbarkeit des Systems abhängig.

Große Systeme bedürfen hierzu auch einer geeigneten Testarchitektur, um tragfähige und wartbare Testsysteme zu erstellen: Hier wäre es reine Verschwendung, die Architekturkompetenz und das Know-how der Softwarearchitekten nicht zu nutzen, um die Architektur des Testsystems (Testarchitektur) zu unterstützen und zu verbessern.

Je größer ein System wird, desto schwieriger und aufwendiger wird der Regressionstest. Ein Problem ist die effektive und effiziente Auswahl der Testfälle. Hier sollte man das Wissen des Architekten zu Systemarchitektur und -design bei der Analyse von Änderungen und deren Auswirkungen nutzen, um die notwendigen Regressionstestfälle – basierend auf Priorität, Kritikalität, Risiken und der Häufigkeit ihrer Benutzung – effektiv auszuwählen.

In dem Ausbildungsprogramm liegt ein besonderer Schwerpunkt sowohl auf der Zusammenarbeit der Architekten mit den Testingenieuren und -managern als auch auf ihrer aktiven Beteiligung an testbezogenen Aktivitäten.

### Soft Skills

In unserem Qualifizierungsprogramm (Die Programme nutzen jeweils einen Teil, der im Folgenden gelisteten Themen. Die SyAs und SSWAs brauchen z. B. mehr Unterstützung für den Umgang mit interkulturellen Teams, als die SWAs) geht es um die Verbesserung der folgenden Bereiche sozialer Kompetenzen (vgl. auch [Cre08]):

- **Kommunikation:** Aktives Zuhören, Feedback geben, bedarfsgerecht (rolle-spezifisch) kommunizieren, usw.
- **Interkulturelle Kompetenz:** Mit unterschiedlichen Wertevorstellungen (z. B. Kulturen verschiedener Staaten, Unternehmen und Funktionen) umgehen können.
- **Verhandlungsgeschick:** Offenheit, unterschiedliche Sichtweisen einzunehmen, und mit mitunter schwierigen Gefühlen zurechtzukommen.
- **Situativer Führungsstil:** Andere Menschen authentisch und passend zur jeweiligen Situation zu führen und zu motivieren.
- **Durchsetzungsvermögen in kritischen Situationen:** Bewusstsein über die eigenen Emotionen und eigenen Denkmuster, insbesondere in schwierigen Situationen.
- **Persönlicher Stil:** Die Fähigkeit, eigene typische Wahrnehmungs- und Verhaltensmuster zu reflektieren.
- **Konfliktmanagement:** Konflikte mit sich selbst und anderen zu erkennen und zu lösen.

So müssen die Teilnehmer beispielsweise ihre Projekte verschiedenen Stakeholdern präsentieren, um zu zeigen, dass sie die Sprache, Sichtweise und Interessen anderer Rollen verstehen.

Nach den drei Trainingseinheiten verfügen die Teilnehmer sowohl über ein solides Verständnis und konkrete Methoden, um große, qualitativ hochwertige Systeme zu entwickeln, als auch über die erforderlichen Kommunikationsfähigkeiten, um ein Projekt erfolgreich zu beenden.

Die Workshops sind keine Präsentationen, bei denen die Teilnehmer in erster Linie zuhören. Vielmehr wird von ihnen verlangt, sich aktiv zu beteiligen und ihre Erfahrungen und ihr Wissen mit den anderen Teilnehmern zu teilen. Das fördert die Bildung eines Netzwerks unter den Teilnehmern und hilft, die Lernunterlagen an den konkreten Projektkontext anzupassen.

### Erfahrungen

Inzwischen haben sechs Gruppen von SSWAs das Qualifizierungsprogramm absolviert, über 15 SWA-Gruppen und eine SyA-Gruppe. Das Feedback der Organisationseinheiten und steigende Nominierungszahlen zeigen deutlich, dass das Programm sowohl für die Architekten, als auch für ihre Projekte, für ihre Organisationseinheit und für Siemens insgesamt sehr erfolgreich ist.

Die Architekten haben aufgrund der technischen und sozialen Kompetenzen, die sie in dem Curriculum erworben haben, sowie wegen der Anerkennung, die sie von anderen Kollegen und vom Management erfahren, nun einen wesentlich stärkeren Einfluss auf wichtige Entscheidungen mit großer Tragweite als vor ihrer Teilnahme. Einige der SSWAs haben uns berichtet, dass sie inzwischen regelmäßig in strategische, technische Entscheidungen miteinbezogen werden. Weiterhin sind alle Zertifikate wichtiger Bestandteil der Personalakte und damit wertvoll für die weitere Karriere.

Ein anderer Architekt berichtet über die Vorteile, die seine Organisationseinheit aus dem Curriculum zieht: „Weil ich das Produktmanagement nun besser verstehen und besser mit diesem kommunizieren kann, werde ich nicht nur häufiger und früher in wichtige Entscheidungen miteinbezogen – auch der Umfang und die Qualität der Informationen, die ich vom Produktmanagement erhalte, sind besser geworden.“ Manche inhaltlichen Methoden etablieren sich auf diesem Wege sichtbar in den Prozessen der Organisationen, z. B. der Ansatz auch NFRs durch quantifizierte und messbare Szenarien zu beschreiben.

Zwischen den Architekten gibt es ein wachsendes Netzwerk – so profitieren auch die Organisationseinheiten von den neuen Ideen, die diese von gemeinsamen „Hausaufgaben“ mit Kollegen aus anderen Organisationseinheiten zurückbringen. Diese gegenseitige „Befruchtung“ ist besonders wertvoll, weil sie organisationsübergreifend und weltweit erfolgt. Durch dieses Netzwerk – mit kontinuierlich steigenden Teilnehmerzahlen – wächst bei Siemens weltweit das gemeinsame Verständnis über moderne System- und Software-Engineering-Ansätze, wie sie im Curriculum vermittelt werden.

Viele Erfahrungen, die wir mit diesem Ansatz gemacht haben, können für andere

Organisationseinheiten und die Leser von OBJEKTSpektrum von Nutzen sein. Ganz besonders möchten wir Ihnen ans Herz legen, darauf zu achten, dass die System- und Softwareingenieure in Ihren Teams alle Prozessschritte und Rollen gut verstehen und nicht nur *eine* Kompetenz besitzen. Außerdem ist die Kombination von Schulungen mit praktischer Anwendung sehr effektiv und führt zu einem besseren Verständnis der Projekte und des Einsatzes von Techniken. Die Verbesserung technischer und sozialer Kompetenzen in dem Bereich „Weiterbildung von Entwicklungsteams“ ist ein weiterer wichtiger und nicht zuletzt motivierender Faktor.

### Zusammenfassung und Ausblick

Ein System- oder Software-Architektenprogramm ist also eine Kombination aus:

- *Pre-Assessment*: Voraussetzung, um an dem Programm teilzunehmen.
- *Mehrstufiger Kursdurchführung*, mit unmittelbarer Umsetzung des Gelernten in den eigenen Projekten.
- *Zertifizierung*: einer Abschluss-Zertifizierung für die Expertenprogramme oder mehrere Knowledge Gates – während des Trainings und einem Capability Gate am Ende des Programms für die Senior-Programme.

In Summe stellen wir dadurch sicher, dass eine Person sowohl über das Wissen und die Erfahrungen, als auch über die Fähigkeiten verfügt, um die Rolle eines zertifizierten System- oder Softwarearchitekten oder gar eines Senior-Architekten zu übernehmen. Das Zertifikat bedeutet auch, dass siemensweit anerkannt ist, dass sie spezielle Fähigkeiten erworben haben, dass sie wissen, wie diese ineinander greifen und gemeinsam anzuwenden sind, und dass sie in der Praxis bewiesen haben, dass sie diese erfolgreich anwenden können.

Derzeit wird das Konzept für eine Senior-Qualifizierung für Systemarchitekten

erarbeitet. Weiterhin wird an einem ähnlichen Konzept für eine Testrolle gearbeitet. Es gibt zwar einige etablierte Zertifizierungsstandards für Testingenieure und -manager, z. B. den „ISTQB Certified Tester“ (vgl. [ISTQB]), den ISEB (vgl. [ISEB]) oder auch die Zertifizierungen vom QAI Global Institute (vgl. [QAI]) und vom International Institute for Software Testing (vgl. [IIST]). Diese verlangen bei der Zertifizierung jedoch lediglich eine Multiple-Choice-Prüfung. Auch fehlt hier die praktische Projektarbeit, anhand derer ein

Teilnehmer die Lerninhalte konkret anwenden und umsetzen kann – einschließlich des Feedbacks im Verlauf der Workshops durch Kollegen, Trainer oder Mentoren.

Parallele Projekte haben in ähnlicher Form Siemens-interne Qualifizierungsprogramme für weitere Schlüsselrollen, z. B. Projektleiter oder Produktmanager, entwickelt oder sind dabei. Ein aktuelles Projekt treibt die Architektur dieser Qualifizierungsprogramme, d. h. den Abgleich der Rollen und ihrer Trainings(-inhalte). ■

### Literatur & Links

**[Agi]** Agiles Manifest, siehe: [www.agilemanifesto.org](http://www.agilemanifesto.org)

**[Cre08]** O. Creighton, M. Singer, Who Leads Our Future Leaders? On the Rising Relevance of Social Competence in Software Development, in: Proc. of ICSE Workshop on Leadership and Management in Software Architecture, Leipzig 2008

**[Gel88]** D. Gelperin, B. Hetzel, The Growth of Software Testing, in: Comm. of the ACM, Vol. 31, Issue 6, June 1988, S. 687-695

**[GfSE]** Gesellschaft für Systems Engineering/German Chapter of INCOSE, siehe: [www.gfse.de](http://www.gfse.de)

**[Goo08]** Google-Suche nach „architecturetesting“, siehe: <http://www.google.de/search?hl=de&q=architecture+testing>

**[Hab]** Haberfellner et. al., Systems Engineering, Verlag: Orell Füssli, Zürich, 2012, ISBN-13: 9783280040683

**[INCOSE]** International Council on Systems Engineering, siehe: [www.incose.org](http://www.incose.org)

**[IREB]** International Requirements Engineering Board (IREB e. V.), siehe: [www.certified-re.de](http://www.certified-re.de)

**[IIST]** International Institute for Software Testing (IIST), siehe: [www.testinginstitute.com/](http://www.testinginstitute.com/)

**[ISEB]** Information Systems Examinations

Board (ISEB), siehe: [www.bcs.org/server.php?show=nav.9278](http://www.bcs.org/server.php?show=nav.9278)

**[ISTQB]** International Software Testing Qualifications Board (ISTQB), Homepage, siehe: [www.istqb.org/](http://www.istqb.org/)

**[QAI]** QAI Global Institute, siehe: <http://www.qaiworldwide.org/>

**[SEI]** Software Engineering Institute (SEI) der Carnegie Mellon University, Software Architecture Curriculum, siehe: <http://www.sei.cmu.edu/go/architecture-curriculum/?location=secondary-nav&source=715351>

**[Sim05]** E. Simmons, Lessons Learned in Five Years of Requirements Engineering Improvement, Requirements Engineering Conference, 2005, siehe <http://requirements-engineering.org/REnotDonePanelRE05/Simmons.pdf>

---

*Der Beitrag wurde aktualisiert und ist in der Ursprungsversion in der Printausgabe 04/2009 des OBJEKTSpektrums (seinerzeit ohne den Autor Rüdiger Kreuter) veröffentlicht worden.*