

Basiswissen Softwarearchitektur

Verstehen, entwerfen, wiederverwenden

von

Torsten Posch, Klaus Birken, Michael Gerdorf

3., akt. u. erw. Aufl.

Basiswissen Softwarearchitektur – Posch / Birken / Gerdorf

schnell und portofrei erhältlich bei beck-shop.de DIE FACHBUCHHANDLUNG

Thematische Gliederung:

[Software Engineering](#)

dpunkt.verlag 2011

Verlag C.H. Beck im Internet:

www.beck.de

ISBN 978 3 89864 736 6

3 Vorgehen

Der künstlerische Arbeitsprozess kommt mir vor wie eine Transsubstantiation, wie eine Verwandlung, wobei das Ich sich wie ein Beobachter fühlt und das Werk ihr Zeuge zu sein scheint.

Sofia Kouldakidou, Herdecke/Ruhr

Können Sie sich vorstellen, dass der Potsdamer Platz in Berlin entstanden ist, indem sich alle Handwerker und Bauarbeiter am Tag des Projektbeginns treffen, ein paar grobe Absprachen treffen, wie z.B. wir wollen eine herausragende Architektur, Platz für 1000 Büros, drei Kinos, fünf Hotels und ein Theater? Zudem müssen wir in einem Jahr fertig sein, und es darf nicht mehr als 30 Millionen Euro kosten. Darauf beginnt jeder mit seinen Arbeiten. Während die Gebäude errichtet werden, sprechen sich die Beteiligten immer wieder ab, sobald irgendwelche Fragen aufkommen. Malen Sie sich aus, wie und ob überhaupt das Bauvorhaben in diesem Fall abgeschlossen worden wäre!

Die letzten beiden Kapitel haben behandelt, was Softwarearchitektur ist, welche Aufgaben sie hat und wie sie im organisatorischen Umfeld eingebunden ist. Im zweiten Teil dieses Buches wollen wir Ihnen aufzeigen, wie Softwarearchitektur erstellt wird. Der eigentliche Entwurf der Architektur ist dabei nur ein Schritt. Wichtig ist es auch, die Architektur auf den richtigen Annahmen aufzubauen und sie angemessen zu dokumentieren. Ebenso müssen Sie bewerten können, ob Sie auf die richtige Architektur setzen, bevor in die Implementierung übergegangen wird.

Dieses Kapitel gibt einen Überblick über das gesamte Vorgehen der Architekturerstellung. Die einzelnen Schritte und Bausteine werden in den anschließenden Kapiteln dann noch ausführlicher beschrieben.

3.1 Überblick

Das Erstellen einer Softwarearchitektur ist mehr als der alleinige Entwurf eines Architekten im stillen Kämmerlein. Gerade ein solches Vorgehen ist bei komplexeren Projekten zum Scheitern verurteilt. Die Kommunikation zwischen allen Beteiligten ist ein wesentliches Element, das ein *Vorgehen* berücksichtigen muss. Abbildung 3–1 gibt einen Überblick über die wichtigsten Schritte während der Architekturerstellung.

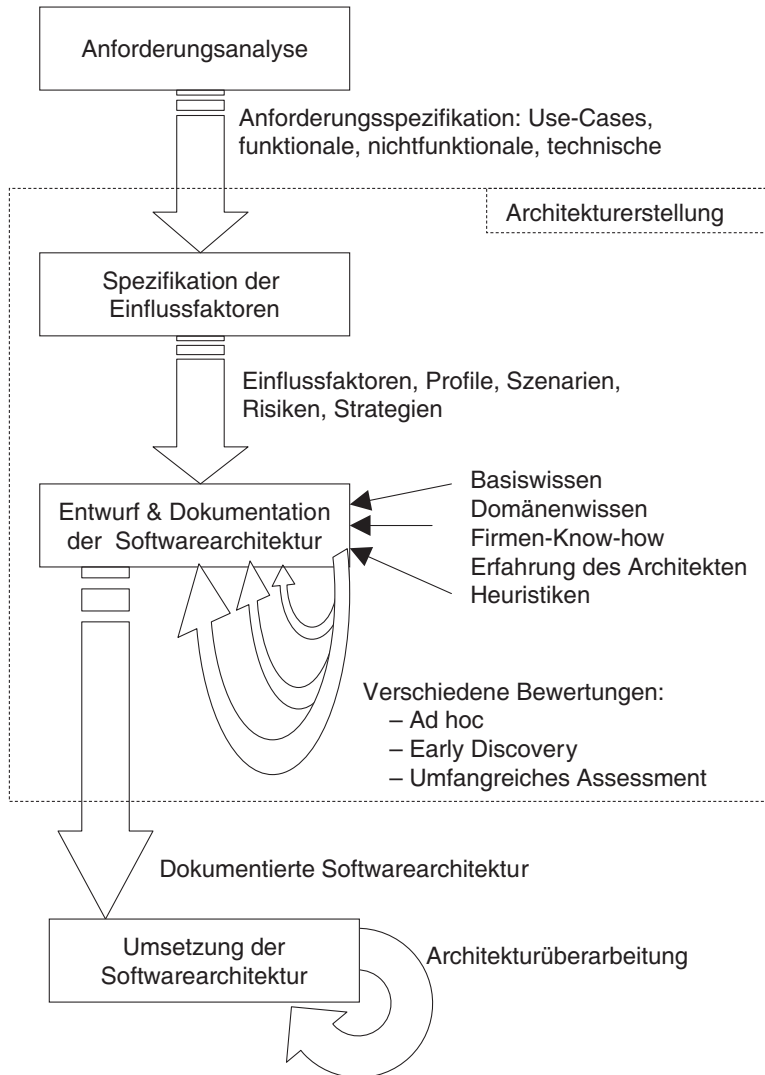


Abb. 3–1 Überblick des Vorgehens zur Erstellung einer Softwarearchitektur

Das Vorgehen kann in drei Bereiche eingeteilt werden:

- Vorbereitungen für den Entwurf
- Entwurf, Dokumentation und Bewertung
- Umsetzung der Architektur

Jeder der drei Bereiche wird im Folgenden kurz angerissen. Im weiteren Verlauf des Kapitels werden wir dann auf jeden Bereich aus der Perspektive des Vorgehens noch genauer eingehen. Darauf aufbauend werden einzelne Aspekte wie Entwurf, Dokumentation und Bewertung in späteren Kapiteln detaillierter ausgeführt.

Bevor mit der Erstellung der Architektur begonnen wird, findet die *Anforderungsanalyse* statt. Diese liefert wichtige Eingangsinformationen für die Architektur. Insbesondere ist dies die Anforderungsspezifikation, bestehend aus funktionalen Anforderungen mit Use Cases, nichtfunktionalen Anforderungen sowie technischen Anforderungen und ein zu den Anforderungen passendes fachliches Modell. Der Anforderungsanalyse schließt sich die Spezifikation der Einflussfaktoren als weiterer Vorbereitungsschritt an. Die wichtigsten Ergebnisse dieses Schrittes sind die Einflussfaktoren spezifiziert über Profile und Szenarien sowie Risiken und dazu passende Strategien. Nun sind die wichtigsten Vorbereitungen für den Entwurf der Architektur getroffen.

Vorbereitungen für den Entwurf

Der nächste Schritt ist der eigentliche Entwurf der Architektur durch den Architekten oder das Architekturteam. Gleichzeitig mit dem Entwurf wird die Softwarearchitektur angemessen dokumentiert. Einfluss auf den Entwurf nehmen, neben den Ergebnissen aus der Vorbereitung, verschiedene Formen des Wissens. Dazu zählen ein gewisses Basiswissen, Domänenwissen, Firmen-Know-how, die Erfahrung des Architekten sowie Heuristiken für den Entwurf. Die entworfene Architektur wird immer wieder bewertet. Hierfür kommen verschiedene Bewertungsmethoden, wie z.B. die Ad-hoc-Bewertung, das Early Discovery Review oder ein umfangreiches Assessment zum Einsatz. Auf Basis des Ergebnisses einer jeden Bewertung wird iterativ mit dem Entwurf der Architektur fortgefahren.

Entwurf, Dokumentation und Bewertung

Wir haben nun eine dokumentierte und bewertete Softwarearchitektur. Auf Basis dieser kann mit der Umsetzung begonnen werden. Das bedeutet, die Software wird auf Grundlage der Architektur implementiert. Im Rahmen der Umsetzung ist die Architektur jedoch nicht eingefroren, sondern wird weiter reifen und muss gepflegt werden: Eine ständige Überarbeitung der Architektur ist notwendig.

Umsetzung der Architektur

Es gibt keinen *Prozess*, der für jede Art von Projekt passend ist. Deshalb müssen Prozesse den gegebenen Umständen angepasst werden. Man spricht dann von *Prozess-Tailoring*. Jedoch gibt es grund-

Tailoring, Agilität, iterativ und inkrementell

legende Konzepte bei der Erstellung einer Softwarearchitektur, die, wenn einmal verstanden, die Basis für einen solchen Prozess legen können. Das im Folgenden noch genauer vorgestellte Vorgehen soll deshalb als Sammlung solcher grundlegenden Konzepte verstanden werden. Sie sind in einer logischen, aufeinander aufbauenden Reihenfolge beschrieben. Anfangs hilft Ihnen das, die Konzepte im Zusammenhang zu verstehen. Haben Sie deren Bedeutung erst einmal verinnerlicht, können Sie daraus jeweils Ihren angepassten und der Situation angemessenen Prozess im Sinne eines *agilen Vorgehens* gestalten. Sehen Sie den Ablauf der einzelnen Schritte auch nicht zu streng. Die Erstellung einer Softwarearchitektur ist ein *inkrementeller und iterativer Prozess*. Sie werden deshalb viele kleine Schleifen innerhalb des Vorgehens drehen, um die Architektur Stück für Stück wachsen und reifen zu lassen.

Projektarten und
Fallstudie

Die vorgestellten Konzepte eignen sich nicht nur für die Entwicklung komplett neuer Systeme. Die meisten der heutigen Entwicklungsprojekte zielen darauf ab, bestehende Produkte zu erweitern. Auch werden zum großen Teil fertige Komponenten, wie Betriebssysteme oder Datenbanken, hinzugekauft. Am Ende des zweiten Teils dieses Buches finden Sie zudem eine Fallstudie. Diese stellt sozusagen eine Instanz des Vorgehensmodells dar und veranschaulicht die wichtigsten Schritte und Aspekte nochmals an einem konkreten Beispiel.

3.2 Vorbereitungen für den Entwurf

Vorbereitend für den Entwurf finden die Anforderungsanalyse sowie die Spezifikation der Einflussfaktoren statt. Die Anforderungsanalyse ist ein sehr umfassendes Gebiet für sich und als solches nicht Bestandteil dieses Buches. Hier soll nur kurz ein Abriss des Themas gegeben werden, um ein grundlegendes Verständnis für die Basis des Architekturdesigns zu schaffen. Bei unseren Ausführungen rund um die Anforderungsanalyse beziehen wir uns weitestgehend auf die Arbeiten von Suzanne und James Robertson [Robertson99], die den *Volere-Anforderungsprozess* und die zugehörige Dokumentvorlage für die *Volere-Anforderungsspezifikation* vorstellen. Neben dem Buch von Robertson zum Volere-Ansatz gibt es noch zahlreiche weitere Literatur zum Thema Anforderungsmanagement. Sehr umfassend wird das Gebiet auch in dem deutschsprachigen Buch von Bruno Schienmann [Schienmann02] behandelt.

Die Spezifikation der Einflussfaktoren wird hier aus Prozesssicht im Überblick dargestellt. Ein eigenständiges Kapitel beschäftigt sich anschließend noch ausführlich mit diesem Thema.

3.2.1 Anforderungsanalyse

Der Volere-Prozess ist ein Leitfaden, der ausführt, wie Anforderungen ausfindig gemacht und derart beschrieben werden, dass sie in der fertigen Software auch getestet werden können. Volere ist ein sehr ausführliches Modell. Wir werden hier nur ganz kurz auf die wesentlichen Aspekte eingehen. Eine ausführliche und qualitativ hochwertige Anforderungsanalyse bildet die perfekte Basis für den Entwurf einer guten Softwarearchitektur. Je mehr Arbeit bereits im Rahmen der Anforderungsanalyse erledigt wurde, desto leichter fallen dem Architekten notwendige vorbereitende Maßnahmen für die Softwarearchitektur.

Die wesentlichen Schritte des Volere-Prozesses sind:

- Ausfindigmachen der Anforderungen
- Niederschreiben der Anforderungen
- Qualitätskontrolle

Bevor ein Team von Anforderungsanalysten ans Werk geht, findet ein Treffen für den Projektstart statt. Robertson nennt dies *Blastoff*. Daran nehmen die wichtigsten Stakeholder teil. Ziel des Treffens ist es, das Projekt vorzubereiten und die Machbarkeit sicherzustellen, noch bevor das Projekt startet. Es müssen genügend Fakten zusammengetragen werden, um deutlich zu machen, dass das Projekt lohnende Ziele hat, dass es möglich ist, diese zu erreichen, und dass genügend Rückhalt der Stakeholder vorhanden ist. Es wird ein grobes Kontextmodell entworfen und eine klare Aussage darüber formuliert, was das Produkt leisten soll. Zudem erfolgen eine erste Kostenabschätzung sowie eine erste Bewertung von Risiken. Letzten Endes ist es Ziel des Treffens, eine *Go/no go-Entscheidung* formulieren zu können.

Projekt-Blastoff

Nach Start des Projekts kann sich ein Team von Anforderungsanalysten ans Werk begeben, die konkreten Anforderungen ausfindig zu machen. Dazu müssen sie zunächst genau verstehen, was das Produkt exakt leisten soll. Soll das Produkt z.B. einen bestimmten Geschäftsprozess unterstützen, muss dieser von den Analysten im Detail verstanden sein. Die Analysten arbeiten hierfür sehr eng mit den Stakeholdern, insbesondere den zukünftigen Anwendern zusammen. Eingesetzt werden z.B. Fragetechniken, Interviews oder Use-Case-Workshops. Hängt die Anforderungsanalyse fest oder müssen Anforderungen konkreter gemacht werden, können Prototypen oder Szenarien zum Einsatz kommen, um den Anwendern eine Simulation der Anforderungen vorzulegen. Wichtig ist, dass nicht nur nach funktionalen Anforderungen gesucht wird. Auch die nichtfunktionalen Anforde-

Ausfindigmachen der Anforderungen

rungen, wie z.B. Antwortzeiten oder Benutzeroberfläche, sind von großer Bedeutung. Eine dritte Kategorie umfasst die technischen Anforderungen. Dabei handelt es sich meist um Vorgaben, welche die Freiheiten des Architekten später einschränken. Müssen Sie z.B. ein Steuergerät wie das für den Auslösemechanismus des Airbags in einem modernen Automobil entwerfen, so wird Ihnen das in der Automobilindustrie standardisierte Echtzeitbetriebssystem OSEK sehr wahrscheinlich vorgeschrieben. Diese technische Anforderung beeinflusst den Entwurf Ihrer Architektur.

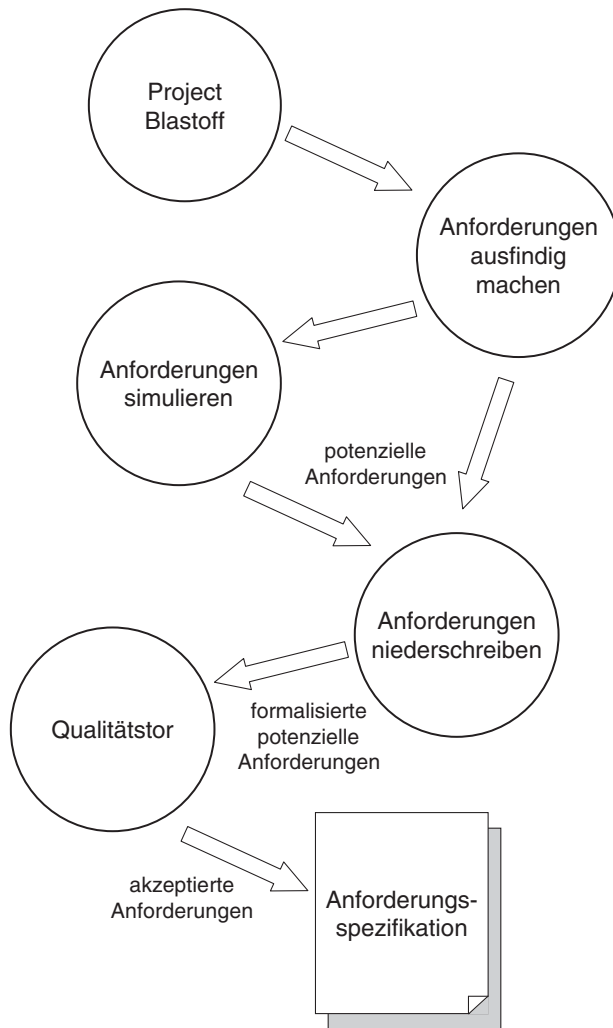


Abb. 3-2 Auszug aus dem Ablauf des Prozesses für die Volere-Anforderungsspezifikation; Quelle [Robertson99]

Die gefundenen Anforderungen müssen in einer standardisierten Form niedergeschrieben werden. Verwendung findet dabei die Sprache des Anwendungsbereiches, so dass die Anforderungen von den Anwendern verstanden werden. Um sicherzustellen, dass die Implementierung der fertigen Software auch die Anforderungen erfüllt, müssen die Anforderungen *messbar* sein. Jede Anforderung wird deshalb mit *Abnahmekriterien* versehen, dessen Aufgabe es ist, die Anforderung zu quantifizieren, so dass Tester exakt feststellen können, ob die Implementierung die Anforderung erfüllt oder nicht. Ist eine Anforderung beispielsweise, dass das System leicht portierbar sein soll, so wird ein Abnahmekriterium einen konkreten Fall aufzeigen, der in Form eines Tests durchgeführt und mit erfolgreich oder nicht erfolgreich bewertet werden kann. Im Falle unseres Steuergerätes unter OSEK könnte das Abnahmekriterium z.B. heißen, dass es möglich sein muss, das System mit einem Aufwand von maximal fünf Personentagen auf das Echtzeitbetriebssystem VxWorks zu portieren. Die Anforderungen werden in der Anforderungsspezifikation erfasst. Grundlage hierfür ist eine standardisierte Dokumentvorlage wie die Volere-Spezifikation aus [Robertson99]. Für jede einzelne Anforderung gibt es innerhalb der Spezifikation wieder eine Vorlage, die das Layout zur Erfassung der Anforderung vorgibt. Eine solche Vorlage wird als *Requirements Shell* bezeichnet.

*Niederschreiben der
Anforderungen*

Für die Qualitätskontrolle von Anforderungen beschreibt [Robertson99] das *Qualitätstor*. Jede Anforderung muss dieses virtuelle Tor passieren, um Teil der endgültigen Spezifikation zu werden. Solange eine Anforderung dieses Tor nicht passiert hat, handelt es sich nur um eine potenzielle Anforderung. Aufgabe des Qualitätstors ist es, jede einzelne Anforderung auf Eigenschaften wie Vollständigkeit, Relevanz, Zusammenhalt und Verfolgbarkeit zu überprüfen. Die vollständige Spezifikation wird dann nochmals einer umfassenden Inventur unterzogen, die feststellt, ob Anforderungen fehlen, alle Anforderungen konsistent sind und keine ungelösten Konflikte mehr existieren.

Qualitätskontrolle

Ab dem Zeitpunkt, da die gefundenen Anforderungen eine ausreichende Stabilität aufweisen, wird ein *fachliches Modell* der funktionalen Anforderungen erstellt. Typischerweise handelt es sich dabei um ein *objektorientiertes Analysemodell*, bestehend aus UML-Paket- und Klassendiagrammen für die statischen Zusammenhänge und UML-Interaktionsdiagrammen für die dynamischen Aspekte. Das Modell muss in dem Sinne vollständig sein, dass alle vom System geforderten Anwendungsfälle durch Interaktionen im Klassenmodell realisiert werden können. Die gesamte *Fachlichkeit* wird somit in einem Modell erfasst. In einem Banksystem würden als fachliche Klassen z.B. der

Fachliches Modell

Kunde oder die Überweisung existieren. Ein fachlicher Ablauf, der durch ein Interaktionsdiagramm dargestellt ist, wäre das Aufgeben einer Überweisung durch den Kunden. Das fachliche Modell ist frei von jeglicher technischer Lösung. Diese wird erst im Rahmen des Architekturentwurfs erstellt. Das fachliche Modell dient somit als erste Ausgangsbasis für die Softwarearchitektur. Teilweise wird das fachliche Modell in der Literatur auch als fachliche Architektur bezeichnet. Demgegenüber steht dann die technische Architektur. Letztere bezeichnen wir in diesem Buch als die Softwarearchitektur.

Anforderungs-
management

Der Prozess der Anforderungsanalyse ist mit dem Übergang zum Architekturdiseign nicht abgeschlossen. Ganz im Gegenteil. Anforderungen unterliegen einer permanenten Änderung und bedürfen deshalb stetiger Anpassung. Anforderungsmanagement ist somit eine der wichtigsten Aufgaben, die sich über das gesamte Projekt hinwegzieht.

3.2.2 Einflussfaktoren

Haben Sie die initiale Anforderungsanalyse abgeschlossen, beginnen Sie nicht gleich mit der Erstellung des Architekturentwurfs. Zunächst bestimmen Sie als Vorstufe zum Entwurf die Einflussfaktoren der Architektur, so dass Sie später beim eigentlichen Entwurf zielgerichtet vorgehen. Auf diese Weise stellen Sie sicher, dass die fertige Architektur die Anforderungen des Systems erfüllen kann. Das heißt, Sie legen zuerst fest, woran Sie sich beim Entwurf orientieren. Dies ist insbesondere wichtig, da ein Architekt immer wieder Abwägungen zwischen konkurrierenden Aspekten treffen muss und dafür eine solide Entscheidungsgrundlage benötigt.

Unterschieden wird zwischen drei Arten von Faktoren, die den Entwurf beeinflussen: organisatorische, technische und Produktfaktoren. Im Rahmen der Spezifikation der Einflussfaktoren werden diese genauer analysiert. Ziel der Spezifikation ist es, die wichtigsten Einflussfaktoren zu finden, deren Veränderlichkeit und Einfluss zu bestimmen sowie Risiken aufzudecken und Strategien zur Risikominimierung zu entwickeln. Bei der Spezifikation wird in drei wesentlichen Schritten vorgegangen:

■ *Identifizieren und Präzisieren der Faktoren*

Im ersten Schritt finden Sie die kritischen Einflussfaktoren aus den drei Bereichen. Da die Faktoren meist keine absoluten Größen sind und Gegenstand von Interpretation, müssen diese im Rahmen eines Kontexts noch präzisiert werden. Hierfür erhält jeder Faktor ein Profil bestehend aus mehreren Szenarien. Die Szenarien werden in zwei Schritten definiert. Zunächst wird das Profil in Kategorien

unterteilt und danach eine Menge repräsentativer Szenarien für jede Kategorie ermittelt.

■ *Analyse der Faktoren*

Der Analyseschritt untersucht die gefundenen Faktoren genauer auf ihre Flexibilität, Veränderbarkeit und deren Einfluss auf die zukünftige Architektur. Dadurch erhält der Architekt mehr Hinweise auf Gestaltungsspielräume bei der Architekturerstellung sowie die Art und Weise, wie er die Faktoren beim Entwurf berücksichtigen muss. Eine größere Flexibilität gibt mehr Verhandlungsspielräume mit den Stakeholdern, wenn Kompromisse gefunden werden müssen. Letztlich müssen der Einfluss und die Veränderbarkeit eines Faktors berücksichtigt werden, um zukünftige Anpassungen in der Architektur möglichst lokal zu begrenzen.

■ *Identifizieren von Architekturthemen und Entwickeln von Strategien*

Im letzten Spezifikationsschritt werden die bisher gewonnenen Ergebnisse nochmals konzentriert, um sie für die Arbeit des Architekten noch prägnanter zu definieren. Aus der Gesamtheit der Einflussfaktoren werden zunächst wesentliche Risiken identifiziert. Separieren Sie hierfür aus der Menge der Einflussfaktoren die für das Projekt zentralen Risiken. Danach entwickeln Sie Strategien für diese Risiken. Eine Strategie legt dabei fest, wie durch gezielte Architekturmaßnahmen dem Risiko entgegengewirkt werden kann.

Die Spezifikation der Einflussfaktoren stellt sicher, dass Sie sich beim Entwurf der Architektur auf die Aspekte konzentrieren, die für den Erfolg des Projektes entscheidend sind. Bereits beim Entwurf der Architektur fokussieren Sie auf die zentralen Risiken. In Kapitel 4 werden wir die Spezifikation der Einflussfaktoren noch ausführlich behandeln. Sie sind nun ausreichend gewappnet für das eigentliche Architekturdesign.

3.3 Iterativ, inkrementeller Entwurf, Dokumentation und Bewertung

Im Rahmen des Entwurfs greifen die Themen Design der Architektur, Dokumentation der Entwurfsentscheidungen sowie Bewertung der Architektur ineinander. Die wesentlichen Schritte beim Entwurf einer Architektur sind:

- Informationen sammeln
- Erstellung und Dokumentation des ersten Architekturentwurfs
- Bewertung des frühen Entwurfs

- Überarbeiten, Ausbauen und weiteres Dokumentieren des Architekturentwurfs
- Umfassende Bewertung der Architektur
- Abschließende Überarbeitungen und vorläufige Fertigstellung des Architekturentwurfs

Die beschriebenen Schritte stellen ein mögliches, grobes Vorgehen beim Entwurf aus der Vogelperspektive dar. In den folgenden, vertiefenden Kapiteln wird dieses Vorgehen weiter verfeinert. Für einen ersten Überblick bilden diese Schritte jedoch eine gute Orientierung. Was verbirgt sich nun hinter den einzelnen Punkten?

3.3.1 Der erste Architekturentwurf

Informationen sammeln

Bevor der Architekt den ersten Wurf seiner Architektur erstellen kann, muss er Informationen als Grundlage sammeln. Dazu gehören die bereits beschriebene Anforderungsanalyse mit dem fachlichen Modell sowie die Spezifikation der Einflussfaktoren mit den identifizierten Risiken und Strategien. Darüber hinaus wird sich der Architekt jedoch auch bei bereits bestehenden, ähnlichen Systemen umschaun, um Ideen und Lösungen zu sammeln. Dies ist nicht negativ oder einfalllos, sondern professionell! Zuletzt wird er für sich noch die *Systemidee* formulieren – ein Konzentrat der Anforderungen. Mit diesem Satz an Informationen ist er ausgerüstet für den ersten Entwurf.

Architekturwissen

Für den Erstentwurf lässt sich der Architekt von den gewonnenen Informationen leiten. Hinzu kommt, dass er eine Menge an Wissen über den Entwurf von Softwarearchitekturen einsetzt. Dazu zählt insbesondere:

- Basiswissen
- Domänenwissen und Firmen-Know-how
- Heuristiken
- Erfahrung des Architekten

Das *Basiswissen* umfasst vieles, was Sie in diesem Buch lesen können. Insbesondere zählen hierzu das Wissen über Entwurfsmethoden (z.B. Finden von Kernabstraktionen), Entwurfsprinzipien (z.B. Abstraktion und Kapselung), Architekturstile (z.B. Schichtenbildung) und Architekturmuster (z.B. solche für Kommunikationsmechanismen). Das *Domänenwissen* und *Firmen-Know-how* ist spezielles geistiges Eigentum über Softwarearchitekturen in einem bestimmten Umfeld. Ohne dieses werden Sie heute nicht mehr konkurrenzfähig sein. *Heuristiken*

sind Abstraktionen von Erfahrungen im Sinne von Prinzipien, Regeln oder Ratschlägen, die der Architekt je nach Bedarf berücksichtigt. Letzten Endes spielt auch noch die *Erfahrung* des Architekten eine entscheidende Rolle.

Die erste Architektur entsteht somit aus einem Rezept mit den Zutaten Projektinformationen und Architekturwissen, hergestellt in einem assoziativen, kreativen Prozess des Kochs. Wie bei allen Künsten gilt auch bei dieser, dass sie sich in einem großen Maße erlernen lässt! Im Wesentlichen zieht der Architekt das fachliche Modell als Basis heran und orientiert sich an den Einflussfaktoren und weiteren Entwurfskriterien, um zu einer guten technischen Lösung zu gelangen. Kapitel 5 behandelt das Thema Entwurf ausführlich.

Das Rezept

Zeitgleich mit dem Entwurf der Architektur wird diese auch dokumentiert. Dazu werden verschiedene *Modelle der Architektur* erstellt. Zum Einsatz kommt hier möglichst eine standardisierte Notationsform, wie z.B. UML. Wichtig ist, dass die Dokumentation die Perspektive der potenziellen Leser einnimmt. Da es unterschiedliche Lesergruppen mit verschiedenen Intentionen gibt, muss auch die Architekturdokumentation aus verschiedenen *Perspektiven* erfolgen. Deshalb wird die Architektur in verschiedenen Sichten dokumentiert. Diese ähneln den unterschiedlichen Konstruktionsplänen für ein Haus. Die Statik wird in einer anderen Sicht festgehalten als die Elektrik. Typische *Sichten* einer Softwarearchitektur sind z.B. die Kontextsicht, die Struktursicht oder die Verhaltenssicht. Bei der Dokumentation der Architektur achtet der Architekt darauf, seine Architekturentscheidungen explizit mit Bezug auf die Einflussfaktoren zu dokumentieren. Das heißt, er hält nicht nur fest, dass er einen bestimmten Architekturstil einsetzt, sondern auch, warum genau dieser für die Erfüllung der Anforderungen geeignet ist. Diese Form der Dokumentation unterstützt zudem die anschließende Bewertung der Architektur.

*Kontinuierliche
Dokumentation*

Wenn die erste Architektur entworfen wurde und somit die wesentlichen, tragenden Entscheidungen auf dem Tisch liegen, ist es Zeit, diese nochmals kritisch zu hinterfragen. Architekturentscheidungen sind von großer Bedeutung für das gesamte Projekt. Deshalb wäre es unverantwortlich, diese nicht ausdrücklich nochmals zu überprüfen. Eine frühe Bewertung soll sicherstellen, dass die weiteren Entwurfsentscheidungen nicht auf einer falschen Grundlage aufbauen. Geeignet hierfür ist ein *Early Discovery Review*. Dies ist eine speziell leichtgewichtige Bewertung, die sehr früh im Prozess zum Einsatz kommt.

*Frühe
Architekturbewertung*

3.3.2 Iterativ, inkrementelles Ausbauen des Entwurfs

Stabile Architektur

Nachdem der Architekt den ersten Wurf der Architektur fertiggestellt und seine Entscheidungen einer ersten Bewertung unterzogen hat, wird die Architektur Schritt für Schritt weiter ausgebaut. Zum einen wird er die Erkenntnisse aus der Bewertung einarbeiten, zum anderen aber auch die Architektur so erweitern, dass sie alle Anforderungen weitmöglichst erfüllt. Es wird dabei selten gelingen, die Architektur zu 100 % fertigzustellen. Grund hierfür ist, dass der Beginn der Implementierung nicht beliebig hinausgeschoben werden kann und auch nicht soll. Ziel muss eine *stabile Architektur* sein, die alle größeren Architekturrisiken noch vor der Implementierung beseitigt. Wichtig ist aber auch, rechtzeitig mit der Umsetzung zu beginnen, da in jedem Entwicklungsprojekt weitere Erkenntnisse mit der Implementierung einhergehen. Der Architekt geht beim weiteren Entwurf iterativ und inkrementell vor.

Bewertung, Kommunikation und Orientierungshilfen

Während des Entwurfs wird die Architektur immer wieder Bewertungen unterzogen. Dazu gehören auch Prototypen oder Simulationen, soweit diese notwendig sind, um bestimmte Risiken auszuräumen. Im Rahmen der Bewertungen wird die Architektur regelmäßig mit verschiedenen Beteiligten diskutiert. Es ist wichtig, dass die Architektur nicht im stillen Kämmerlein entsteht. Beim Entwurf selbst orientiert sich der Architekt fortwährend an den Einflussfaktoren, den Entwurfsprinzipien und Entwurfszielen. Er wendet Heuristiken an und greift in seine Toolbox, um immer wenn möglich erprobte Lösungsmuster einzusetzen. Die Toolbox des Architekten stellen wir in Kapitel 8 vor.

Stresstest

Die Dokumentation der Architektur wird stets angepasst und weiter ausgeführt. Wenn die Architektur bereits sehr deutliche Formen angenommen hat, wird nochmals eine umfassende Bewertung durchgeführt, das sogenannte umfangreiche Assessment, auf das wir in Kapitel 7, »Bewertung«, noch ausführlich eingehen. An diesem Assessment nimmt ein größerer Teil von Stakeholdern teil. Es ist sozusagen ein letzter *Stresstest*, um sicherzustellen, dass die Architektur wirklich für das Projekt tragfähig ist. Die Ergebnisse der Bewertung werden dann noch eingearbeitet und der Architektur der letzte Schliff gegeben, so dass in die Umsetzung übergegangen werden kann.

3.4 Die Umsetzung der Architektur

Ist die Architekturdesignphase abgeschlossen, geht es an die Umsetzung der Architektur. Dies bedeutet, dass ein größerer Kreis von Entwicklern mit dem Feindesign, der Implementierung und den Modultests beginnt. Bevor dies jedoch geschieht, sollte idealerweise ein *Skelett der Architektur* implementiert werden. Dieses Skelett stellt den Rahmen für die gesamte weitere Umsetzung dar. Das Skelett realisiert die Architekturbausteine als leere Hüllen. Viel wichtiger ist jedoch, dass auch die Kommunikationsbeziehungen zwischen den Bausteinen bereits realisiert werden. Dadurch existiert ein fertiger *Integrationsrahmen*, noch bevor die Entwickler damit beginnen, die einzelnen Bausteine der Architektur zu implementieren. Ein solcher Integrationsrahmen stellt in einem hohen Maße sicher, dass die Architektur bei der Implementierung eingehalten und nicht untergraben wird. Zudem ermöglicht er ein effizienteres Arbeiten, da weniger Reibungsverluste durch Integrationsschwierigkeiten entstehen. Entscheidend hierfür ist, dass die Kommunikationsschnittstellen bereits realisiert sind.

Architekturskelett

Für die weitere Implementierung stellt der Architekt soweit möglich *Schablonen* zur Verfügung, die vorgeben, wie bestimmte Strukturen und Mechanismen realisiert werden müssen. Im Idealfall definiert er ein gesamtes *Programmiermodell*. Dieses liefert Richtlinien, welche die Grundlagen für Programmierer bilden, um eine konsistente Umsetzung der Architektur zu garantieren. Wurden die Modelle der Architektur in einer standardisierten Form mit einem geeigneten CASE-Werkzeug erstellt, kann sogar eine automatische *Codegenerierung* erfolgen. Alle weiteren Aufgaben, die der Architekt während der Implementierungsphase wahrnehmen muss, wurden bereits in Kapitel 2 dieses Buches ausgeführt. Wesentlich ist, dass er als Wächter der Architektur fungiert. Dies bedeutet, er muss sicherstellen, dass die Architektur eingehalten und die konzeptuelle Integrität nicht zerstört wird. Am besten erreicht er dies, wenn er aktiv als Berater für die Entwickler tätig ist und die Schnittstellen der Architekturbausteine überwacht.

Schablonen,

Programmiermodell und

Codegenerierung

Änderungen sind ein Naturgesetz in der Softwareentwicklung. Deshalb wird die Architektur auch während der Entwicklungsphase nicht starr sein können, sondern auf Änderungen reagieren müssen. Aufgabe des Architekten ist es, diese Änderungen in einem geordneten Prozess in die Architektur einzubauen. Während der gesamten Implementierung muss er die potenziellen Risiken, die im Vorfeld identifiziert wurden, im Auge behalten. Auch muss er wachsam sein, ob neue Risiken zutage treten.

Änderungen und Risiken

3.5 Zusammenfassung

Dieses Kapitel hat einen groben Überblick über die Erstellung von Softwarearchitektur gegeben. Eingangsvoraussetzung ist eine gute Anforderungsanalyse mit einem fachlichen Modell. Die eigentliche Erstellung lässt sich dann in die drei Bereiche Vorbereitung, Entwurf und Umsetzung gliedern. Im Rahmen der Vorbereitung werden vor allem die Faktoren bestimmt, an denen sich der Entwurf orientieren muss. Der Entwurf selbst ist ein stark iterativ inkrementeller Prozess. Neben den Einflussfaktoren orientiert sich der Architekt an Entwurfsprinzipien und Entwurfszielen. Er wendet Heuristiken und spezielles Wissen über den Architekturentwurf an. Wichtig ist auch, dass der Architekt wann immer möglich auf bewährte Lösungen zurückgreift. Zeitgleich mit dem Entwurf müssen die Architekturentscheidungen dokumentiert werden. Eine standardisierte Notationsform wie UML bietet sich hierfür an. Die Dokumentation hält auch fest, aus welchen Gründen bestimmte Entscheidungen getroffen wurden. Dies unterstützt die Bewertung der Architektur. Bewertungen finden zu unterschiedlichen Zeitpunkten während des Architekturentwurfs statt und stellen sicher, dass auf die richtigen Lösungen gesetzt wird. Ist die Architektur stabil und ausgereift genug, kann mit der Umsetzung begonnen werden. Idealerweise beginnt diese, indem ein Skelett der Architektur implementiert wird. Ein solches Skelett bildet den Integrationsrahmen für die gesamte weitere Implementierungsphase.

Dieses grobe Vorgehen beinhaltet die grundlegenden Konzepte des Architekturentwurfs. Ein einheitliches Vorgehen für jede Art von Projekt gibt es jedoch nicht. Deshalb muss der Prozess immer den gegebenen Umständen angepasst werden. Ein tiefes Verständnis der Grundkonzepte ist jedoch Basis für ein gutes Prozess-Tailoring.