4.6 Übliche Dokumenttypen für Softwarearchitekturen

Zur Beschreibung der in den vorangegangenen Abschnitten dargestellten Architekturinformationen kommt üblicherweise eine Reihe verschiedener Dokumente zum Einsatz. Dieser Abschnitt gibt hierüber einen ersten Überblick.

4.6.1 Zentrale Architekturbeschreibung

Die zentrale Architekturbeschreibung ist (sinnvollerweise) das Kerndokument für eine Softwarearchitektur. Sie enthält soweit möglich alle architekturrelevanten Informationen, dazu gehören:

- Aufgabenstellung, Ziele (Vision), Qualitätsanforderungen und Stakeholderinnen
- Technische und organisatorische Rahmenbedingungen
- Sichten, Entscheidungen, verwendete Muster
- Technische Konzepte
- Qualitätsbewertungen
- Erkannte Risiken

Hier bietet sich der Einsatz eines passenden Schablonendokuments, also einer Art »Architektur-Dokumentations-Schablone« an. Zum Beispiel folgt die Dokumentenschablone unter [arc42] zu einem guten Teil den in den vorangegangenen Abschnitten dieses Kapitels vorgestellten Inhalten.

Eine zentrale Architekturbeschreibung kann durchaus einen größeren Umfang annehmen, sodass ihre Beschreibung (und Pflege) in einem einzelnen Dokument ggf. nur noch bedingt sinnvoll ist. Die Inhalte einer solchen Beschreibung können mit verschiedenen Werkzeugen verwaltet werden. Einige typische Optionen sind:

- **Dokumente:**
  Dokumente, die mit gängigen Textverarbeitungen erstellt werden können, sind in der Regel recht einfach zu nutzen und zu verwalten, solange sie nicht zu groß werden.
- **CASE-/MDA-/UML-Tools:**
  Modellierungswerkzeuge können mittels oft mächtiger Reportgeneratoren recht hilfreich bei der Erstellung von Dokumentationen sein (siehe auch Kap. 6). Nicht unterschätzt werden sollte allerdings (gerade bei kleineren Projekten) der Aufwand, den die projektspezifische Anfangskonfiguration solcher Werkzeuge annehmen kann. Von Vorteil ist der in der Wartung dann oft deutlich höhere Automatisierungsgrad für das wiederholte Generieren einer Architekturdokumentation, da z.B. UML-Diagramme oder gar Codestücke ohne »Copy & Paste« in eine neue Dokumentenversion eingebaut werden können.

- **HTML-Seiten oder Wikis:**
  Eventuell erlauben »potenziell etwas pragmatischere« Werkzeuge wie Wikis einen Mittelweg zwischen Dokumentation und Modellierungswerkzeugen.
- Beliebige Mischformen

In Kapitel 6 wird eine Reihe weiterer Softwarewerkzeuge behandelt, die für Softwarearchitekturen nützlich sein können.

4.6.2 Architekturüberblick

Der Architekturüberblick dient als schnell lesbare Kurzfassung (möglichst nicht mehr als 30 Seiten) der zentralen Architekturbeschreibung. Er betrachtet vergleichbare Inhalte, beschränkt sich aber auf die wesentlichen Punkte wie zentrale Sichten, Hauptqualitätsanforderungen und Kernentscheidungen.

Falls für die Erstellung einer ausführlichen zentralen Architekturbeschreibung, z.B. aus zeitlichen oder Aufwandsgründen, keine Möglichkeit besteht, kann der Architekturüberblick als Minimalersatz für eine komplette Beschreibung dienen.

4.6.3 Dokumentübersicht

Die Dokumentübersicht ist ein Verzeichnis, das (pro Projekt bzw. pro zu beschreibender Softwareanwendung) als Index für alle jeweils architekturrelevanten Dokumente dient und auch deren Abhängigkeiten nennt. Sinnvollerweise sollten organisatorische Richtlinien festgelegt werden, wie dieses Verzeichnis auszusehen hat und an welcher Stelle es zu finden ist. Zudem sollten Hinweise enthalten sein, welche Dokumente von wem, also von welcher Projektrolle, in welcher Reihenfolge zu lesen sind.

4.6.4 Übersichtspräsentation

Eine Übersichtspräsentation ist ein Foliensatz, anhand dessen die Architektur in maximal einer Stunde (technisch) präsentiert werden kann.

Eine managementtaugliche Variante davon sollte die zentralen Aussagen und insbesondere auch den Geschäftsnutzen in 10 Minuten zusammenfassen können.

4.6.5 »Architekturtapete«

Mit einer »Architekturtapete« sollen viele Architekturaspekte in einem Gesamtüberblick dargestellt werden. Praktisch ist dies oft eine Sammlung von Postern z.B. mit Sichtendarstellungen zuzüglich Verfeinerungen, Qualitätsaspekten usw. Diese werden gemeinsam an einer Wand oder auf Metaplanwänden aufgehängt und erlauben die interaktive Diskussion z.B. zwischen Architektur und Entwicklung über spezielle Themen.

Achtung: Eine »Architekturtapete« kann, als Diskussionsgrundlage eingesetzt, sehr hilfreich sein. Zu vermeiden ist hingegen, sie dogmatisch zu sehen, denn dann wirkt sie schnell abschreckend auf diejenigen, die das Softwaresystem umsetzen, testen und betreiben sollen.

4.6.6 Handbuch zur Dokumentation

Das Handbuch erläutert die Funktionsweise und die Struktur der gesamten Dokumentation. Es ist auch der geeignete Ort, um Notationen gesammelt zu erläutern.

4.6.7 Architecture Decision Record

Ein Architecture Decision Record oder auch ein »ADR« ist ein festes Format zur Dokumentation von getroffenen Architekturentscheidungen. In einem ADR wird eine Entscheidung in einem eigenen Dokument oder Abschnitt isoliert betrachtet, mit dem Ziel, eine zugänglichere Dokumentation zu erhalten. Der Fokus liegt dabei nicht nur auf dem Ergebnis der Entscheidung, sondern insbesondere auf der Dokumentation der getroffenen Annahmen, Beweggründe und Einflussfaktoren zum Zeitpunkt der Entscheidung [Ny11]. Ein ADR, wenn korrekt eingesetzt, hilft dabei, eine Dokumentation zielgruppengerecht, korrekt und nachvollziehbar zu gestalten. Die Erfindung des Formats wird Michael Nygard zugeschrieben, der die Struktur 2011 definierte. Ein ADR besteht aus den folgenden Abschnitten (siehe [Ny11]):

- **Titel:**
  Er soll eine aussagekräftige Auskunft über das Thema der Entscheidung geben.
- **Kontext:**
  Er soll die Einflüsse auf die Entscheidung beschreiben. Der Abschnitt soll eine akkurate Abbildung der Umstände sein und Randbedingungen oder vorherige Entscheidungen auflisten, die nun zur aktuellen Entscheidungsnotwendigkeit führen.
- **Entscheidung:**
  Dieser Abschnitt liefert die »Antwort« auf den vorherigen Abschnitt. Hier werden oft auch die zur Verfügung stehenden Diskussionen aufgelistet und diskutiert.
- **Status:**
  Er gibt Auskunft darüber, ob eine Entscheidung in Kraft ist, lediglich einen Vorschlag darstellt oder vielleicht schon überholt ist.
- **Konsequenzen:**
  Dieser Abschnitt soll beschreiben, wie sich der Kontext durch die Entscheidung verändert. Auch hier soll eine möglichst ehrliche Beschreibung inklusive positiver und negativer Auswirkungen der Entscheidung stattfinden, um Nachvollziehbarkeit zu gewährleisten.

4.6.8 Technische Informationen

Hierbei handelt es sich um ein oder mehrere Dokumente mit wichtigen Informationen für Projektentwickler und Tester. Darin sollten Informationen zu den Entwicklungsmethoden, Programmierrichtlinien sowie zum Bau, Start und Test des Systems hinterlegt sein (vgl. hierzu auch Abschnitt 4.5).

4.6.9 Dokumentation von externen Schnittstellen

Ein besonderes Augenmerk sollte auf die Dokumentation von insbesondere extern sichtbaren Schnittstellen des Softwaresystems gelegt werden. Diese sind für das Zusammenwirken des Gesamtsystems in seinem Kontext von zentraler Bedeutung.

Leider werden externe Schnittstellen in realen Projekten (allzu) häufig zu »Zeit fressenden Problemstellen«. Widmen Sie ihnen eher früher als später die nötige Aufmerksamkeit, denn gerade dort »liegt der Teufel oft im Detail«. Folglich macht sich ein etwas stärkerer und recht frühzeitiger Beschreibungsaufwand an dieser Stelle oft bezahlt (deutlich mehr als »ganz besonders schöne Diagramme« oder »das letzte Quäntchen syntaktischer UML-Feinheit« in einer Sichtenbeschreibung).

4.6.10 Template

Bei Schnittstellenbeschreibungen bietet es sich an, wieder einem gleichbleibenden Muster zu folgen. Tabelle 4–9 stellt eine Reihe typischer Elemente aus Schnittstellenbeschreibungen zusammen:

Überschrift	Inhalt
Identifikation	Genaue Bezeichnung und Version der Schnittstelle
Bereitgestellte Ressourcen	Welche Ressourcen stellt dieses Element bereit? ■ Syntax der Ressource: API, Methodensignaturen (z.B. OMG-IDL, WSDL) ■ Semantik der Ressource: Welche Auswirkungen hat ein Aufruf dieser Ressource? • ausgelöste Events • geänderte Daten • geänderte Zustände • sonstige wahrnehmbare Nebenwirkungen • Restriktionen bei der Benutzung der Ressource
Fehlerszenarien	Beschreibung der Fehlersituation und deren Behandlung
Variabilität und Konfigurierbarkeit	Kann das Verhalten verändert oder konfiguriert werden (beispielsweise über Konfigurationsparameter)?
Qualitätseigenschaften	Welche Qualitätseigenschaften (Verfügbarkeit, Performance, Sicherheit, Parallelisierbarkeit etc.) gelten für diese Schnittstelle?
Entwurfs-entscheidungen	Welche Gründe haben zum Entwurf dieser Schnittstelle geführt? Welche Alternativen gibt es und warum wurden sie verworfen?
Hinweise	Hinweise oder Beispiele zur Benutzung

Tab. 4–9 *Typische Elemente aus Schnittstellenbeschreibungen*

4.7 Praxisregeln zur Dokumentation

Für jedwede Art von Architekturdokumentation gibt es – wie für viele andere vorgehens- oder technisch orientierte Dokumentationen – eine Reihe von bewährten Regeln. Diese dienen vor allem der praxistauglichen Lesbarkeit und Zweckdienlichkeit solcher Dokumentationen.

4.7.1 Regel 1: »Schreiben aus der Sicht der Leserin«

Versuchen Sie Ihre Dokumentation aus der Betrachtungsweise Ihrer Leserinnen zu erstellen, denn sonst fühlen sich diese in ihren Anforderungen nicht ernst genommen. Naturgemäß werden Dokumente häufiger gelesen als geschrieben – und Dokumentation wird tatsächlich gelesen.

Versuchen Sie zu starken Fachjargon zu vermeiden (in der Praxis ist dies in Dokumenten stets eine Gratwanderung). Besonders zentrale Fachbegriffe sollten separat erläutert werden, z. B. in Form eines Glossars.

Augenmerk sollten Sie auf die Struktur von Dokumenten legen, um Ihre Gedanken und Ideen in eine zielführende Reihenfolge zu bringen. Einmal mehr ist hier die Verwendung passender Dokumentenschablonen zu empfehlen.

4.7.2 Regel 2: »Unnötige Wiederholung vermeiden«

Versuchen Sie soweit möglich Wiederholungen zu vermeiden bzw. setzen Sie sie wenn nötig nur sparsam und gezielt ein. Im Ergebnis

- vereinfachen Sie dadurch normalerweise die Verwendung der Dokumentation und
- vereinfachen deren Pflege bzw. Änderungen merklich.

Stellen Sie sich bei Wiederholungen (auch in leichter Variation) die Fragen:

- Ist die Variation Absicht?
- Ist dieser Variation Bedeutung beizumessen?

Dokumentation aus unterschiedlichen Sichtweisen ist allerdings typischerweise nicht als Wiederholung einzuordnen, sondern dient vielmehr zur Vertiefung des Verständnisses über einen Sachverhalt.

4.7.3 Regel 3: »Mehrdeutigkeit vermeiden«

Architekturdokumentation lässt häufig bewusst spätere Designentscheidungen als geplanten Handlungsspielraum offen. Zu viel Interpretationsspielraum bei Architekturvorgaben führt jedoch auch leicht zu ungeplanter Mehrdeutigkeit.

Um hier Abhilfe zu schaffen, könnten formale Beschreibungssprachen eingesetzt werden. Allerdings ist deren Einsatz in der Praxis (zu) selten.

Werden symbolische Notationen verwendet (wie z. B. UML), so sollten Sie die Bedeutung der Symbole erklären oder eine Referenz auf die Begriffserklärung einbauen.

Mehrdeutigkeit entsteht auch wenn sich mündliche und schriftliche Dokumentation nicht decken. Achten Sie daher darauf, identische Begriffe und Argumentationen zu verwenden.

4.7.4 Regel 4: »Standardisierte Organisationsstruktur bzw. Schablonen«

Gerade wenn Sie häufiger Architekturdokumente schreiben, ist deren (einheitliche) Struktur von Wichtigkeit. Sie bietet Ihren Leserinnen einen Wiedererkennungswert. Zudem vereinfacht sie die Referenzierbarkeit von Dokumentteilen (z. B. in der Form: »Siehe Bausteinsicht aus dem Kunden-Management-System«).

Wenn die Struktur festgelegt ist, sollte die Leserin darüber informiert werden. Anhand einer festen Struktur ergibt sich leicht eine Übersicht über bereits abgeschlossene und noch zu erledigende Teile der Dokumentation. Ebenfalls unterstützt wird die Qualitätssicherung der Dokumentation, da alle von Dokumenten abzudeckenden Aspekte vorab definiert sind.

4.7.5 Regel 5: »Begründen Sie wesentliche Entscheidungen schriftlich«

Um Ihre Leserinnen dabei zu unterstützen, Ihre Architekturen und Entwürfe nachzuvollziehen, ist es sehr hilfreich, wesentliche Entscheidungen kurz zu begründen. Begründungen können z. B. durch Verweise auf entsprechende Unternehmensrichtlinien erfolgen (».NET oder Java EE ist bevorzugt für Anwendungen der Art X einzusetzen« oder: »Die Datenhaltung der Kundenstammdaten erfolgt ausschließlich auf dem Mainframe in der zentralen Kundendatenbank«).

Ebenfalls von Interesse können explizit verworfene andere Optionen sein, aber auch die Diskussion von Vor- und Nachteilen einer Lösung, z. B. in der Form: »Die Nutzung eines selbst entwickelten Persistenz-Frameworks mag zwar zu höherer Flexibilität führen, rechtfertigt aber gegenüber bestehenden Frameworks wie Y nicht den resultierenden Entwicklungs- und langfristigen Wartungsaufwand.«

Im Ergebnis kann u. a. Zeit bei Diskussionen gespart werden, wenn Entscheidungen unter neuen Bedingungen überdacht werden müssen. Insgesamt helfen Begründungen somit Ihren Leserinnen, Ihre Entscheidungen zu verstehen und für sich im Nachhinein nachzuvollziehen.

4.7.6 Regel 6: »Überprüfung auf Gebrauchstauglichkeit«

Ein bedeutsamer Punkt für Dokumentation ist der tatsächliche praktische Nutzen für Ihren Leserkreis. Bevor Sie eine Dokumentation abschließend veröffentlichen, sollten Sie daher unbedingt von passenden Vertreterinnen Ihrer Zielgruppe Reviews durchführen lassen – und diese anschließend einarbeiten. Letztlich kann nur der anvisierte Nutzerkreis entscheiden, ob die richtigen Informationen in der richtigen Weise dargeboten werden.

Neben den Reviews sollte auch der Reviewprozess an sich begutachtet werden. Installieren Sie also einen Verbesserungsprozess für Dokumentation, anhand dessen auch die Dokumentationsrichtlinien selbst regelmäßig auf Schwachstellen hin überprüft werden.

4.7.7 Regel 7: »Übersichtliche Diagramme«

Eine Regel – von der es allerdings durchaus des Öfteren begründbare Abweichungen geben kann – lautet: »Vermeiden Sie zu große Diagramme.« Ergebnissen der Kognitionswissenschaft zufolge sind fünf bis neun Elemente (7±2) in Diagrammen für Menschen noch gut überschaubar.

Beachten Sie: Die Architekturtapete aus Abschnitt 4.6.5 ist eine klare Ausnahme dieser Regel.

4.7.8 Regel 8: »Regelmäßige Aktualisierungen«

Aktualisieren Sie Ihre Dokumentation bzw. etablieren Sie einen Prozess, der die Dokumentation nicht nur während der Entwicklung regelmäßig anpasst, sondern dies auch als Teil von Wartungsarbeiten durchführt.

Mangelnde Aktualisierung …

- führt zu unvollständiger Dokumentation,
- lässt Nutzen und Nutzung der Dokumentation sinken,
- lässt Dokumentation sich nicht als maßgebliche Quelle für Informationen etablieren, und es erfolgt unnötiges Nachfragen.

Noch unbeantwortete Fragen in einer Dokumentation verlangen ebenfalls nach zeitnaher Aktualisierung. Ändern sich jedoch Designentscheidungen sehr schnell, so sollten Sie auch nicht zu oft aktualisieren (sonst tun Sie bald nichts anderes mehr …), sondern ggf. warten, »bis sich der Staub etwas gelegt hat«. Hilfreich ist es, hierfür relativ fixe Synchronisationszeitpunkte festzulegen.

4.7.9 EXKURS: Regel 9: »Passen Sie die Änderbarkeit der Dokumentation an die Architektur an«

Je nach Zeitpunkt im Lebenszyklus einer Architektur werden mehr oder weniger Änderungen an ihr vorgenommen. Zum Beispiel werden die ersten Entwürfe einer Architektur sich oft und schnell ändern, wohingegen Änderungen an einer seit Jahren bestehenden Architektur eher langsam und mit Bedacht durchgeführt werden [Kee17]. Sie können den Formalisierungsgrad der Dokumentation diesen Gegebenheiten anpassen und dadurch die Geschwindigkeit maximieren oder das Risiko einer unbedachten Änderung minimieren. Dafür stehen Ihnen unterschiedliche Stellschrauben zur Verfügung, wie zum Beispiel der Detaillierungsgrad der Dokumentation, das Format, die Zugänglichkeit, die Formalität der Diagramme oder die Länge und das Vorhandensein eines Prüfungs-, Änderungs- und Freigabeprozesses.

4.8 Beispiele weiterer Architektur-Frameworks

Neben dem in Kapitel 2 genannten umfassenden Standard ISO/IEC/IEEE 42010: 2011 und dem in den vorangegangenen Abschnitten beschriebenen pragmatischen iSAQB-Ansatz gibt es eine Vielzahl weiterer Ansätze zur Beschreibung von Softwarearchitekturen bzw. Architektur-Frameworks. Um Ihnen einen Eindruck davon zu vermitteln – und ggf. Neugier für weitere Recherche zu wecken –, zeigen wir in diesem Abschnitt in Kurzform einige Beispiele, auch hier ohne Anspruch auf Vollständigkeit.

Einige bekanntere Framework-Ansätze für Softwarearchitektur oder noch weiter gehende Ansätze für Unternehmensarchitektur (Enterprise Architecture) sind u.a.:

- 4+1 (Kruchten)
- Department of (US) Defense Architectural Framework (DoDAF)
- Model Driven Architecture der Object Management Group (OMG-MDA)
- Standards und Architekturen für E-Government-Anwendungen (SAGA) als Framework der deutschen Bundesverwaltung
- SAPs Enterprise Architecture Framework
- The Open Group Architecture Framework (TOGAF®)
- Zachman Framework (IBM)

In den folgenden Abschnitten gehen wir kurz auf zwei dieser Frameworks ein.