# Ziele

**Softwarearchitekturen beschreiben und kommunizieren**

Softwarearchitekten:

* können Architekturen stakeholdergerecht dokumentieren und kommunizieren und dadurch unterschiedliche Zielgruppen adressieren, z. B. Management, Entwicklungsteams, QS, andere Softwarearchitekten sowie möglicherweise zusätzliche Stakeholder
* sind in der Lage, die Beiträge unterschiedlicher Autorengruppen stilistisch und inhaltlich zu konsolidieren und harmonisieren
* kennen den Nutzen von Template-basierter Dokumentation

# Einfluss Stakeholder

## Stakeholder den Sichten zuordnen

**Stakeholder der Kontextsicht**

* Projektleitung
* Anforderungsanalysten
* Systemanalysten
* Fach- oder Domainenexperten
* Design und Entwickler
* Nachgelagert ggf. Administratoren und Vertrieb
* Controlling
* bei Produkten Markt und Vertrieb

**Stakeholder der Bausteinsicht**

* alle an Architektur, Entwurf, Erstellung und Test der Software beteiligten Projektmitarbeiter,
* zusätzlich die Qualitätssicherung (sofern sie nicht sowieso direkt dem Projekt zugeordnet ist),
* das Projektmanagement, dem die Bausteinsicht bei der Erstellung von Arbeitsoder Aktivitätsplänen hilft, sowie
* die Wartung, da nach Abschluss des ursprünglichen Softwareentwicklungsprojekts die Bausteinsicht auch eine effizientere Wartung der entstandenen Software ermöglicht.

**Stakeholder der Laufzeitsicht**

* die Betreiber des Softwaresystems,
* die Systemarchitekten sowie
* allgemein alle an Entwurf, Erstellung und Test der Software beteiligten Projektmitarbeiter.
* Hinzu kommt die Qualitätssicherung (sofern sie nicht sowieso direkt dem Projekt zugeordnet ist).

**Stakeholder der Verteilungssicht**

* die Betreiber des Softwaresystems als besonders wichtige Zielgruppe,
* die Systemarchitekten und Softwarearchitekten sowie
* Entwickler, damit diese wissen, in welchen Umgebungen und Netzen bzw. wie verteilt ihre Software abläuft.

Quelle Basiswissen für Softwarearchitekten, 4. Auflage, Kapitel 4.3

## Typische Fragen von Stakeholdern an der Architekturdokumentation

|  |  |
| --- | --- |
| **Stakeholder** | **Fragen** |
| Kunden und Auftraggeber | * Hat das Projektteam das Problem verstanden? * Was ist der Kontext des Systems? Mit welchen Nachbarn arbeitet das System zusammen? * Was sind die wesentlichen Entwurfsentscheidungen bzw. Lösungsstrategien? * Passen die Strukturen und Konzepte der (geplanten) Lösung zum Problem? * Ist diese Architektur für den geplanten Einsatz tragfähig hinsichtlich nichtfunktionaler Anforderungen wie Performance und Wartbarkeit? |
| Manager | * Was muss der Auftraggeber zuliefern, welche Teile müssen wir selbst entwickeln, welche Teile können wir zukaufen? |
| Fachexperten,  Entwickler und  Qualitätssicherer | * Welchen Teil des Systems entwerfe/realisiere/prüfe ich gerade und wie ordnet sich dieser in das Gesamtsystem ein? * Aus welchen Gründen wurden Entwurfsentscheidungen getroffen? |
| Neue Mitarbeiter | * Was hat das Team bisher entworfen und entwickelt? * Wie sieht das System aus der Vogelperspektive aus? Was ist die „Lösungsstrategie“ des Systems? * Welche technischen Grundlagen (Implementierungstechnologien, Frameworks, Konzepte) sind relevant? * Warum sieht der Code so aus? Welche Gründe haben dazu geführt? * Wie sehen die technischen Konzepte (etwa: für Persistenz, Logging, Sicherheit o. Ä.) aus? |
| Administrator u. Betreiber | * Wie müssen wir die Bausteine des Systems zur Laufzeit verteilen, damit alles korrekt funktioniert? * Welche Regeln und Randbedingungen müssen wir bei der Installation des Systems, im laufenden Betrieb sowie beim Austausch von Systemteilen beachten? * Wie muss die zugrunde liegende Hardware dimensioniert sein? * Welche Abhängigkeiten von anderen Softwaresystemen gibt es? |

Quelle Effektive Softwarearchitekturen 8. Auflage, Kapitel 5.2.1

# Vorteile von Templates

Gerade wenn Sie häufiger Architekturdokumente schreiben, ist deren (einheitliche) Struktur von Wichtigkeit. Sie bietet Ihren Lesern 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 der Leser 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.

Quelle Basiswissen für Softwarearchitekten, 4. Auflage, Kapitel 4.7.4