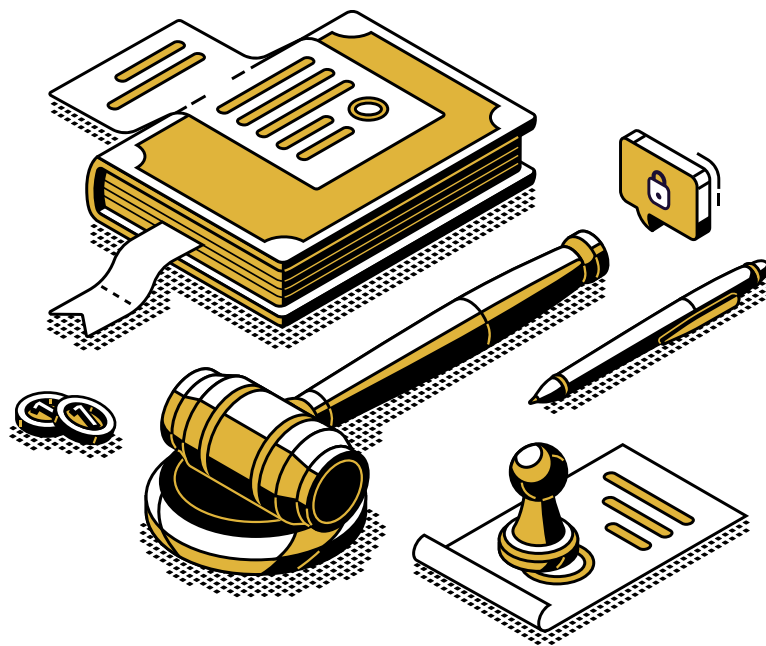


DAS ARCHITEKTURREVIEW
WORKBOOK

Nutzungsbedingungen

- Dieses Buch kann **kostenlos genutzt werden**. Dennoch sind alle **Rechte an diesem Werk vorbehalten**. Texte, Bilder und Grafiken unterliegen dem **Urheberrecht**.
- Etwaige in diesem Buch verwendeten **Markennamen** der jeweiligen Firmen unterliegen dem **Schutz der allgemeinen Warenzeichen-, Marken- oder Patentrechten der Firmen**.
- Alle Rechte, die Vervielfältigung und Verteilung von Inhalten – auch auszugsweise – sind **ausdrücklich vorbehalten**.
- Falls Sie Zitate mit der üblichen Zeilenanzahl verwenden wollen, nennen Sie bitte folgenden Quellangabe:
Architekturreview-Workbook, Christian Albrecht
- Wenn Sie dieses Buch oder Informationen daraus weiterverteilen wollen, teilen Sie bitte statt des Dokuments diesen Download-Link: <https://architekturreview.com/>
- Wenn Sie dieses Buch in gedruckter Form erwerben oder eine Lizenz zum Druck dieses PDF-Dokuments oder zum Nachdruck des Buchs erwerben wollen, kontaktieren Sie uns bitte via feedback@architekturreview.com.
- Mit der Verwendung dieses Buchs erklären Sie sich mit diesen hier aufgeführten Nutzungsbedingungen einverstanden.





Hallo und herzlich willkommen!

Mit diesem Workbook erhältst du ein hilfreiches Werkzeug mit Fragestellungen rund um Softwarearchitektur. Es soll dir dabei helfen, einerseits ein klareres Bild von deinem Vorhaben zu bekommen und andererseits ein Architekturreview strukturiert vorzubereiten. Wir wissen, dass die Informationsflut, der alle Beteiligten während der Softwareerstellung gegenüberstehen, in den meisten Fällen immens ist. Umso wichtiger ist es, ein Review gut vorzubereiten, um dem Reviewteam zu helfen, die richtigen Fragen zu stellen um passende Ergebnisse zu liefern.

Im Rahmen eines Kick-Offs kannst du mit deinem Reviewteam deine Notizen aus diesem Workbook erörtern und dann je nach Zielsetzung entscheiden, wie im Detail weiter vorgegangen werden soll und welche Aspekte dir als Auftraggeber wichtig sind. Dazu stehen dem Reviewteam verschiedene Formate zur Wahl, wie zum Beispiel Interviews mit allen/ausgewählten Beteiligten, Workshops zu ausgewählten Themen, Code Walks oder Bewertungsmethoden wie z.B. ATAM oder DCAR.



INHALT

Warum Reviews?	06
Wann macht ein Review Sinn?	07
Das große Ganze (Die Architektur ist nicht alles)	08
Das Workbook – bereite hier Dein Review vor!	09
One more thing	09

Reviewziele

Was sind deine Gründe und erwarteten Ziele für das Review?	11
Was ist der Reviewgegenstand?	11
Was sind die derzeit größten Herausforderungen?	12

Unternehmen

Wie hat sich dein Unternehmen am Markt positioniert?	14
Welchen Stellenwert hat die IT im Gesamtunternehmen?	14
Gibt es einen Bebauungsplan für die IT-Landschaft?	15
Gibt es Produkt- bzw. Projektportfolios?	15
Wer sind deine Wettbewerber und wie sind sie aufgestellt? ...	16
Wer sind deine Kunden oder Systemnutzer?	16
Wohin entwickelt sich dein Markt?	17

Organisation

Wie ist deine IT-Organisation insgesamt aufgestellt?	20
Wie ist die Sourcing-Strategie?	21
Wer sind die Hauptstakeholder?	21
Bedürfnisse	23
Wie sieht dein Anforderungsmanagement (AM) aus?	24
Welche Eigenschaften des Systems führst du auf Besonderheiten im AM zurück?	25
Stimmen die dokumentierten mit den realen Anforderungen überein?	25
Welche Stakeholder wurden im AM besonders berücksichtigt, welche weniger?	26
Sind besondere IT-Sicherheitsanforderungen für dich relevant?	26
Kennst du die Bedrohungsszenarien für dein System?	27
Welche Gesetzesvorgaben und Compliance-Anforderungen existieren?	28

Welche Qualitätsziele existieren?	28
Softwarelebenszyklus	30
Wie werden Entwicklungsprojekte aufgesetzt?	31
Wie sind aus deiner Sicht die Entscheidungsprozesse in der Softwareentwicklung?	31
Wie viele Softwaresysteme werden derzeit betrieben?	32
Wie ist die Entwicklung der Systeme organisiert?	32
Mit welchen Vorgehensmodellen wird entwickelt?	33
Wie oft im Jahr wird ausgeliefert?	33
Wie lange benötigt die Umsetzung eines Kundenwunsches bis zur Auslieferung?	34
Wie werden Entwicklungsprojekte dem Betrieb/der Wartung übergeben?	34
Wie werden Entwicklungsprojekte beendet bzw. das System abgeschaltet?	35

Architektur

Wie wird aus deiner Sicht Architektur gelebt?	38
Wie wird aus deiner Sicht Architektur gearbeitet?	38
Welche Architekturparadigmen verfolgt dein Unternehmen? Welches dein System?	39
Wie stark werden Cloud Services verwendet?	39
Werden IT-Sicherheitsanforderungen in der Architekturarbeit berücksichtigt?	40
An welcher Stelle gehst du Kompromisse ein?	40

Softwareentwicklung

Welche Programmiersprachen kommen zum Einsatz?	43
Welche Plattformen und Datenbanken werden verwendet?	43
Welche Versionskontrollsysteme (VCS) werden eingesetzt?	44
Welche Werkzeuge werden zur Qualitätssicherung eingesetzt?	44
Wie wird getestet?	45
Welchen Stellenwert haben Tests?	45

Dokumentation

Welche technischen Dokumentationen gibt es für Entwickler und Architekten? Wer ist für die Erstellung zuständig?	48
Welche Handbücher für den Betrieb der Software gibt es?	49
Welche Anforderungsdokumentationen gibt es?	49
Wie stellt dein Unternehmen die Dokumentation zur Verfügung?	49
Wie aktuell sind die Dokumentationen und wie aktiv werden diese gepflegt/genutzt/geändert?	50
Wie werden Änderungen an der	50
Dokumentation kommuniziert?	50

Betrieb

Welche Betriebsinfrastruktur, IT-Systeme und Betriebssysteme werden genutzt?	53
Wie geschieht die Übergabe von Entwicklung in Betrieb?	54
Wie wird aus deiner Sicht die Abstimmung zwischen Entwicklung und Betrieb verwirklicht/umgesetzt?	54
Wie werden die Anforderungen des Betriebs an Architektur und Implementierung gestellt und berücksichtigt?	55
Welche Aspekte des Systems fallen im Betrieb besonders positiv oder negativ auf, auch im Vergleich zu anderen Systemen?	55
Und jetzt? - Get going!	57
Über Inspired Consulting	57
Credits	57

Bonus: Techn. Fragen

Wie ist der Kontext des zu reviewenden Systems?	59
Welche Konzepte / Technologien liegen den einzelnen Programmteilen zugrunde?	60
In einem Code Walk interessiert uns	61
Wie kommt die Anwendung in Produktion?	63
Wie ist das Entwicklungsvorgehen in der Praxis?	63
Verweise/Material/Further reading	65



Warum Reviews?

Reviews dienen dazu, ausgewählte bzw. auszuwählende Details in einem definierten zeitlichen und fachlichen Rahmen kritisch zu betrachten und zu bewerten. Die Bewertungskriterien können sich auf funktionale, fachliche oder qualitative Aspekte des Reviewgegenstands beziehen – oder sogar auf das gesamte Projekt selbst.

Anders ausgedrückt kannst du mit einem Review Folgendes erreichen:



Status Quo, Auftrag und Ziele ermitteln

Sollte nicht sichergestellt sein, dass alle Beteiligten das gleiche Ziel verfolgen, nimm die Erhebung und Verständigung der Qualitätsziele mit in das Review auf.



Lösung bewerten und einordnen

Bewerte deine aktuelle Lösung anhand von nachvollziehbaren Fragestellungen und stelle die Weichen für eine nachhaltige Weiterentwicklung.



Entwurfsentscheidungen hinterfragen

Mithilfe eines Reviews können Schwachstellen und Konsequenzen szenariobasiert aufgedeckt werden.



Weiterentwicklung sichern

Passe deine Prozesse wieder den tatsächlichen Projektbegebenheiten an und übernimm wieder die Steuerung.

Auf der folgenden Seite findest du einige Fragen, die dir helfen können herauszufinden, ob du ein Architekturreview durchführen möchtest.



Wann macht ein Review Sinn?

Anhand der folgenden Fragen kannst du für dich abstecken, ob ein Review gerade Sinn macht. Geh' sie einfach durch und falls ein oder mehrere Punkte zutreffen, solltest du vermutlich bald ein Review planen.

Möchtest du

herausfinden, ob die Qualitätsziele und die Anforderungen perfekt beherrscht und von allen gleich verstanden sind?

die Konsequenzen deiner Entscheidung analysieren und verstehen?

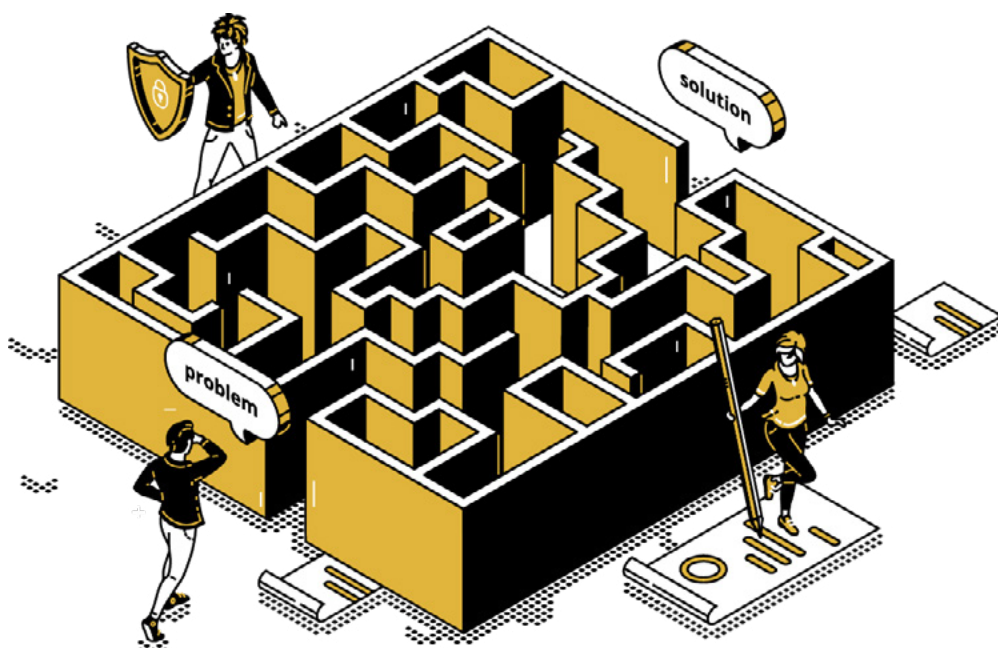
sicherstellen, dass dein Entwurf keine unerwünschten Nebeneffekte mit sich bringt?

eine taktische Entscheidung treffen, die dir möglichst viele strategische Optionen offen hält?

unter Berücksichtigung der bisherigen Strukturen oder Implementierung eine technische Renovierung durchführen und sicherstellen?

sicherstellen, dass deine Organisation eine technische Modernisierung durchhält?

überprüfen, ob es Ziel gefährdende Annahmen oder Auffälligkeiten in der Dokumentation oder Umsetzung gibt?



Das große Ganze (Die Architektur ist nicht alles)

Jede Organisation benötigt für ein Software-Projekt ein solides Fundament, um erfolgreich zu werden. Dieses ruht auf folgenden Säulen:

Projektmanagement

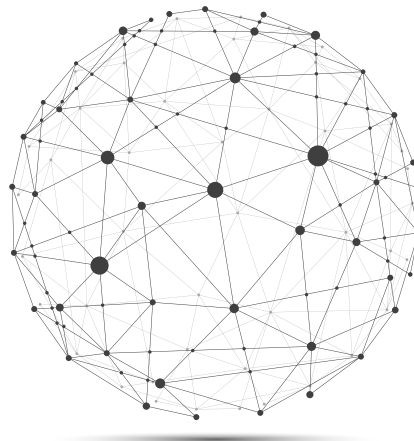
Projektplanung
Stakeholdermanagement
Suppliermanagement

Produktmanagement

Produktvision
Zielgruppenkenntnisse
Strategie

Architektur

Technische Verantwortung
Machbarkeitsstudie und
Spezifikationen
Suppliermanagement



Betrieb

Go-To-Market
Roll-out
Maintenance

Entwicklung

Hardware
Software
Testing

Das Workbook – bereite hier Dein Review vor!

Jetzt werden wir praktisch, hol' deinen Bleistift, Smart Pen oder Kugelschreiber heraus. Dieses Buch ist ein Workbook, kein Lesebuch. Als benutze es, male hinein, mache Notizen ...

Lass uns sicherstellen, dass dein Review von Anfang an gut vorbereitet ist und du zusammen mit dem Reviewteam den bestmöglichen Nutzen für deine Architektur und IT-Organisation schaffst.

Von hier aus geht es los mit einem Fragenkatalog zu den verschiedenen Aspekten deines Systems und dem Drumherum. Hier noch 3 Bearbeitungsempfehlungen:

-  Manche Antworten kannst du einfach ankreuzen, bei manchen ist Platz vorhanden, um eine Skizze zu zeichnen oder Text zu schreiben.
-  Fülle die Antworten so aus, wie es für Dein Vorhaben sinnvoll ist. Manche Frage passt eventuell nicht ganz, dann überspringe sie einfach.
-  Die Fragen nach den Reviewzielen solltest du nicht überspringen. :wink:

One more thing...

An einem Review sind typischerweise Personen aus unterschiedlichen Bereichen der Anwendungsentwicklung beteiligt, um ein möglichst umfassendes Bild zu bekommen und möglichst viele Meinungen zu hören. Allerdings ist eventuell nicht jede unten aufgeführte Frage für jede Personengruppe gleich interessant oder wichtig. Als Orientierungshilfe gibts für jede Gruppe ein Symbol:

-  Du bist der Auftraggeber dieses Reviews, Projektleiter, Entwicklungsleiter oder hast eine fachliche Rolle wie z.B. Product Owner
-  Du gehörst zum Architekturteam oder nimmst regelmäßig Architekturaufgaben wahr
-  Du gehörst zum Entwicklungs- oder Operationsteam

Diese Symbole findest du insbesondere an den Fragen, die deine Gruppe bei der Vorbereitung des Reviews nicht überspringen sollten. Alle anderen "dürfen" sich natürlich auch damit befassen.

Und nun, viel Spaß bei Vorbereiten deines Reviews. Los geht's!

Reviewziele



**“A goal properly set is
halfway reached.”**

Zig Ziglar

Was sind deine Gründe und erwarteten Ziele für das Review?

Es verbergen sich viele verschiedene Dinge hinter dem Begriff Architekturreview.
Was verstehst du darunter? Was möchtest du erreichen?

IST-Zustand eines Systems erfassen

Verifikation bestehender Annahmen

Einholen einer zweiten Meinung

Auswirkung veränderter Anforderungen auf bestehende Software-Architektur

Schwachstellen im System oder der Architektur aufdecken

Spezielle Probleme analysieren

Optimierungs- oder Modernisierungspotenzial erkennen

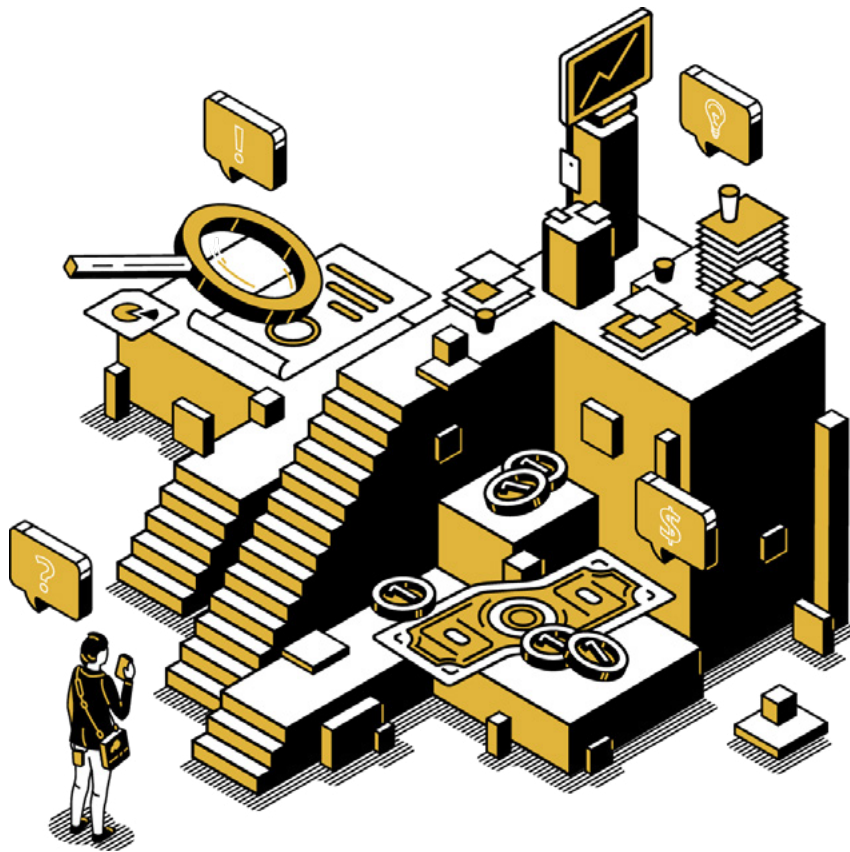
Abgleich des Systems zum aktuellen Einsatzbereich

Eignung der Architektur für das geplante Ziel feststellen

Etwas anderes:

Was ist der Reviewgegenstand?

Benenne konkret das System, um das es geht.
Grenze eventuell auch ab, um welche Systeme es gerade nicht geht.



Was sind die derzeit größten Herausforderungen?

Gibt es einen Hinweis auf den besten Einstiegspunkt?

Modernisierung nötig

Fachkenntnisse fehlen

Spezielles technisches Problem

Änderung der Strategie

Vendor Lock-in

System ist an seiner Grenze angekommen

Workshops zur Vermittlung der notwendigen Fähigkeiten notwendig

Go-Live-Termin

Unternehmen

Gib den Reviewenden einen Überblick zu deinem Unternehmen, seiner IT und dem Kontext, in dem ihr euch befindet. Das hilft bei der Einordnung.



“Unless structure follows strategy, inefficiency results.”

Alfred D. Chandler

Wie hat sich dein Unternehmen am Markt positioniert?

Entwickelt ihr ein Hauptprodukt? Entwickelt ihr mehrere Produkte? Vielleicht sogar gleichzeitig?
In welchen Märkten sind eure Produkte vertreten?

Ein-Produkt Strategie

Vertrieb in DACH-Märkten

Multi-Produkt Strategie

Internationaler Vertrieb

Welchen Stellenwert hat die IT im Gesamtunternehmen?

Wie wichtig ist das System für dich oder die Kunden?

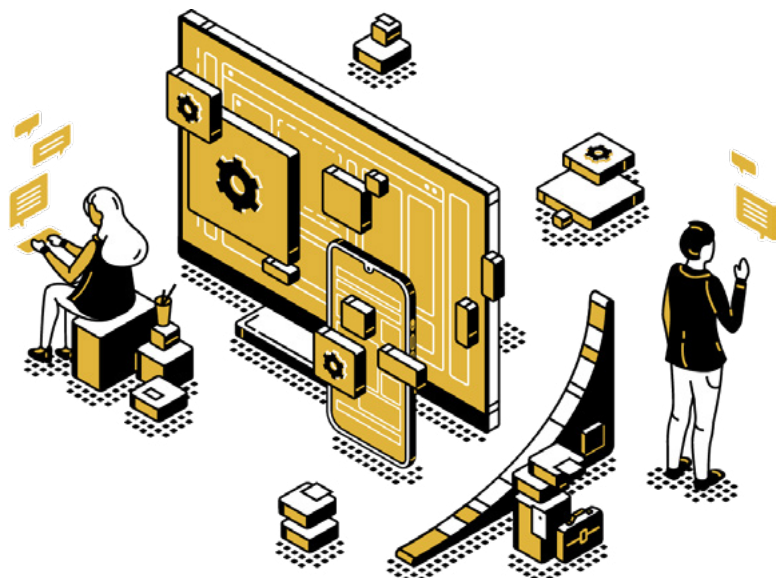
Geschäftskritisch

Nischenanwendung

Signifikanter Umsatz

Proof of Concept

Lösung für ein Teilproblem



Gibt es einen Bebauungsplan für die IT-Landschaft?

An welchen Stellen ist deine IT nicht so leistungsfähig, wie du es dir wünschst?
Welche kurz-, mittel- und langfristigen Ziele verfolgst du?

Gibt es Produkt- bzw. Projektportfolios?

Welche Systeme werden auf welche Weise kurz- bis mittelfristig gebaut oder angeschafft?
Wie entscheidest du, in welchem Projekt oder Produkt neue Features verankert werden?

Wer sind deine Wettbewerber und wie sind sie aufgestellt?

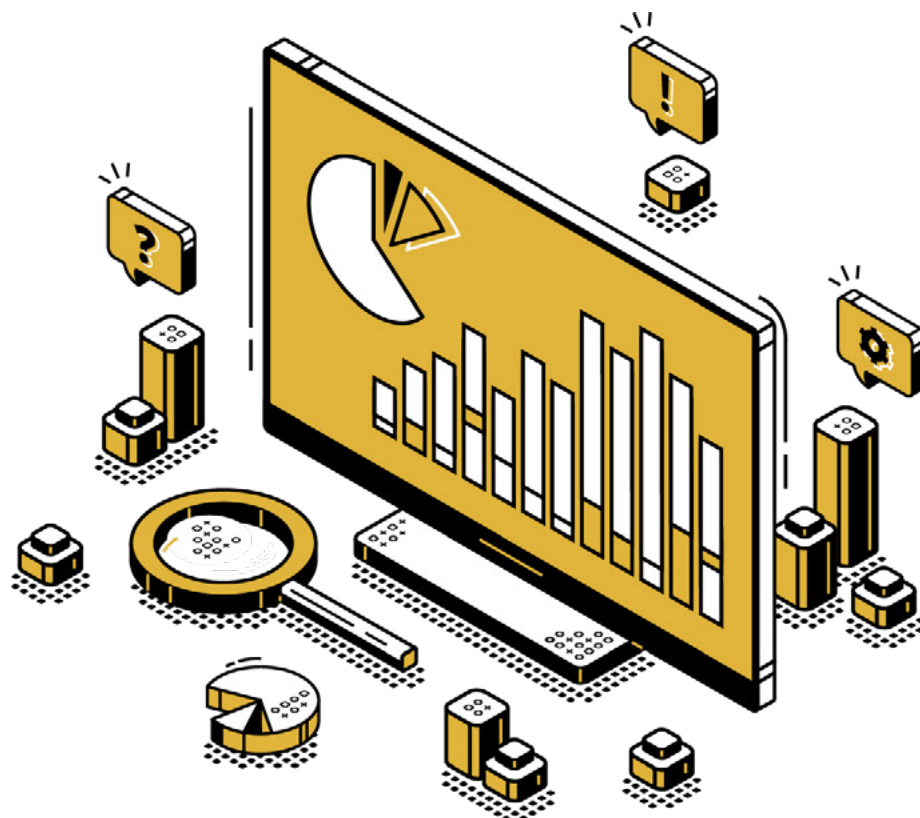
Umfeldanalyse und Ermittlung von Handlungsfeldern:

Wer sind deine Kunden oder Systemnutzer?

Umfeldanalyse und Ermittlung von Handlungsfeldern:

Wohin entwickelt sich dein Markt?

Wo siehst du dein Produkt? Wo deine Wettbewerber? Wo siehst du dein Unternehmen in 3 Jahren?



Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Organisation

Bringe dem Reviewteam deine Organisation nahe. Hieraus ergeben sich für die Reviewenden eventuell schon die ersten Interviewpartner.



“Analysis is the critical starting point of strategic thinking.”

Kenichi Ohmae

Wie ist deine IT-Organisation insgesamt aufgestellt?

Wie viele Personen, Teams und Abteilungen sind daran beteiligt?

Welche Rollen gibt es (Projektleiter, Scrum-Master, DevOps, IT-Security, Datenschutz etc.)?

Wie sehen die formalen Strukturen für Kommunikation aus, wer berichtet an wen?

Wie viele Abteilungen, Teams oder Personen benötigt man durchschnittlich, um ein neues Feature in Produktion zu bringen?

Wie ist die Sourcing-Strategie?

Ist klar definiert, welche Kompetenzen in-House sind und wo man externe Dienstleister dazuholt? Oder ergibt sich das eher zufällig?

Wer sind die Hauptstakeholder?

Wer stellt die Anforderungen? Wer ist "der Kunde"? Wer ist für die Durchführung verantwortlich? Wer trägt die Gesamtverantwortung? Wer berät?

Vorstand

Projektleitung

Produktmanagement

Vertrieb

Entwicklung

Betrieb

Programmleitung

Endkunde

Externe Dienstleister

Security

Datenschutz

IT

Analytics

Forschung & Entwicklung

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Bedürfnisse



**“Meeting requirements is a good job
and every time meeting requirements
with ease is a quality job.”**

Swaran Singh Jaggi

Eine brauchbare Darstellung eines Bedürfnisses nennt man Anforderung. Der Business Analysis Body of Knowledge Guide (BABOK) definiert vier Kategorien von Anforderungen oder Requirements:

- 1 **Business Requirements** – die vom Kunden geäußerten Bedürfnisse zur Erreichung ihrer Geschäftsziele.
- 2 **Stakeholders Requirements** – die Anforderungen der einzelnen Stakeholder, die erfüllt werden müssen, um die Business Requirements zu erfüllen.
- 3 **Transition Requirements** – die Übergangsanforderungen, die zur erfolgreichen Implementierung der Software erforderlich sind.
- 4 **Solution Requirements** – die erwarteten Funktionen und Eigenschaften der Software, die die Stakeholder Requirements erfüllen. Hier kommen die funktionalen und nicht-funktionalen Anforderungen ins Spiel.

Wie sieht dein Anforderungsmanagement (AM) aus?

Wer stellt die Anforderungen? Wer qualifiziert, bewertet und entscheidet über Anforderungen?
Wie ist der Prozess?

Welche Eigenschaften des Systems führst du auf Besonderheiten im AM zurück?

Nenne positive sowie negative Eigenschaften.

Stimmen die dokumentierten mit den realen Anforderungen überein ?

Spiegelt die Aktenlage die Realität bezüglich der Anforderungen wider?
Wenn nicht, wo gibt es Lücken?

Welche Stakeholder wurden im AM besonders berücksichtigt, welche weniger?

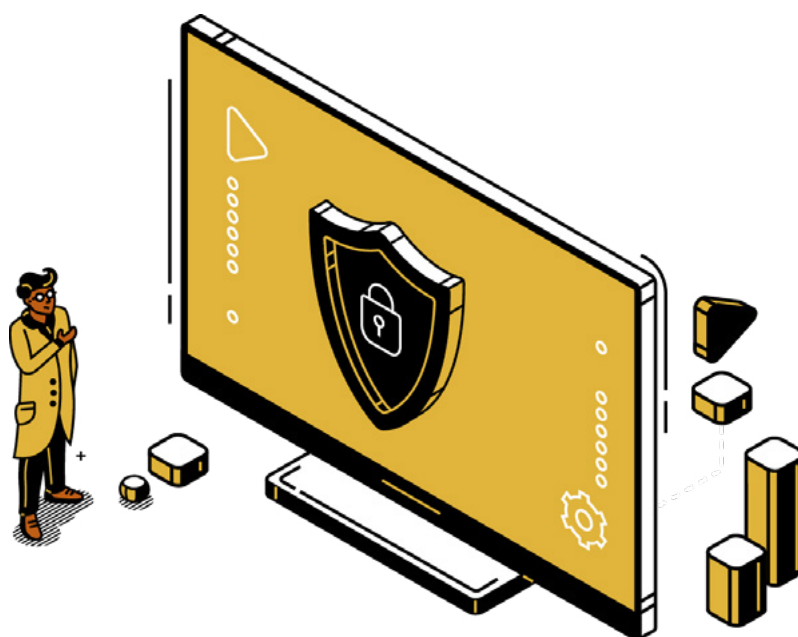
Wie ist der Prozess? Wer wird wann im Erhebungs,- und Entscheidungsprozess eingebunden?

Sind besondere IT-Sicherheitsanforderungen für dich relevant?

Je nach unternehmerischem Umfeld müssen gewisse Sicherheitsstandards beachtet werden.
Welche sind das in deinem Unternehmen? (ISO-27001/-27002, NIST CSF, OWASP Top10, SOC 2, ...)

Kennst du die Bedrohungsszenarien für dein System?

Hast du ein sogenanntes Threat Model für dein System aufgestellt und wie oft überprüfst du deine Annahmen darin? Wie werden daraus Anforderungen erstellt? Wie zuversichtlich bist du damit?



Welche Gesetzesvorgaben und Compliance-Anforderungen existieren?

Je nach unternehmerischem Umfeld müssen gewisse regulatorische Prozesse beachtet werden. Welche sind das bei deinem Unternehmen? (Bafin, BaIT, MaRisk, KAIT, VAIT, ZAIT, ...)
Gibt es Aufsichtsbehörden, an die du berichten musst oder die dich regelmäßig prüfen?

Welche Qualitätsziele existieren?

Wurden und werden bei den Solution Requirements auch Qualitätsziele abgestimmt?
Welche übergeordneten Qualitätsziele gibt es?

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Softwarelebenszyklus

Von der ersten Idee bis zur Außerbetriebnahme kann sich der Lebenszyklus einer Software strecken. Welche Phasen kennt dein Unternehmen?



“The function of good software is to make the complex appear to be simple.”

Grady Booch

Wie werden Entwicklungsprojekte aufgesetzt?

Welche Entscheidungen müssen getroffen worden sein, damit ein Entwicklungsprojekt startet?

Wie sind aus deiner Sicht die Entscheidungsprozesse in der Softwareentwicklung?

Welcher Personenkreis trifft ganz allgemein Technologieentscheidungen, legt Rahmenbedingungen, verhandelt Prioritäten und kommuniziert Entscheidungen?

Wie viele Softwaresysteme werden derzeit betrieben?

Gibt es ein übergeordnetes Architekturdiagramm? Auch bekannt als Zielarchitektur, Architekturvision, Tapete, Spinne, ...

Wie ist die Entwicklung der Systeme organisiert?

Wie groß sind die Entwicklungsteams? Welche Rollen sind besetzt?
Welche Kommunikationswege gibt es? Wie werden Impediments aufgelöst?

Mit welchen Vorgehensmodellen wird entwickelt?

Wasserfall, Safe, Scrum, Kanban, Prince2? Welche Methodik setzt du ein? Wird diese Methode auf allen Ebenen des Unternehmens eingesetzt?

Wie oft im Jahr wird ausgeliefert?

Wie viele Produktinkremente liefert dein Unternehmen in einem Jahr an Kunden aus?

Mehrmals am Tag/täglich

Alle zwei bis vier Wochen

Quartalsweise

Ein-/zweimal im Jahr

Welche Faktoren beeinflussen die Auslieferung am meisten?

Wie lange benötigt die Umsetzung eines Kundenwunsches bis zur Auslieferung?

eher einen Tag

eher einen Sprint

eher ein Quartal

eher ein Jahr

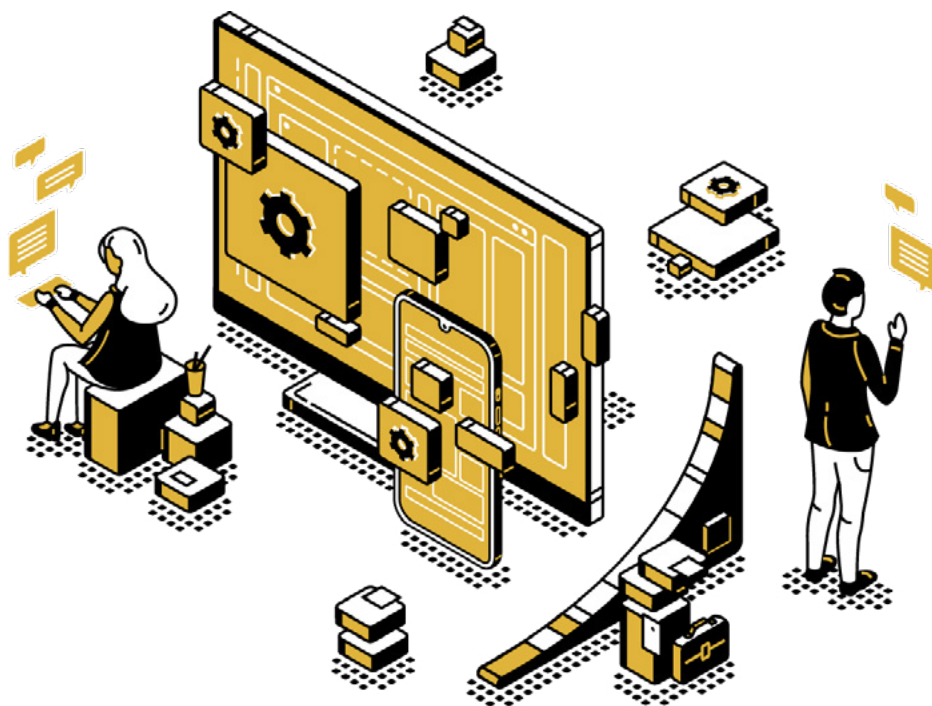
Welche Faktoren beeinflussen die Umsetzungszeit am meisten?

Wie werden Entwicklungsprojekte dem Betrieb/der Wartung übergeben?

Was passiert dabei?

Wie werden Entwicklungsprojekte beendet bzw. das System abgeschaltet ?

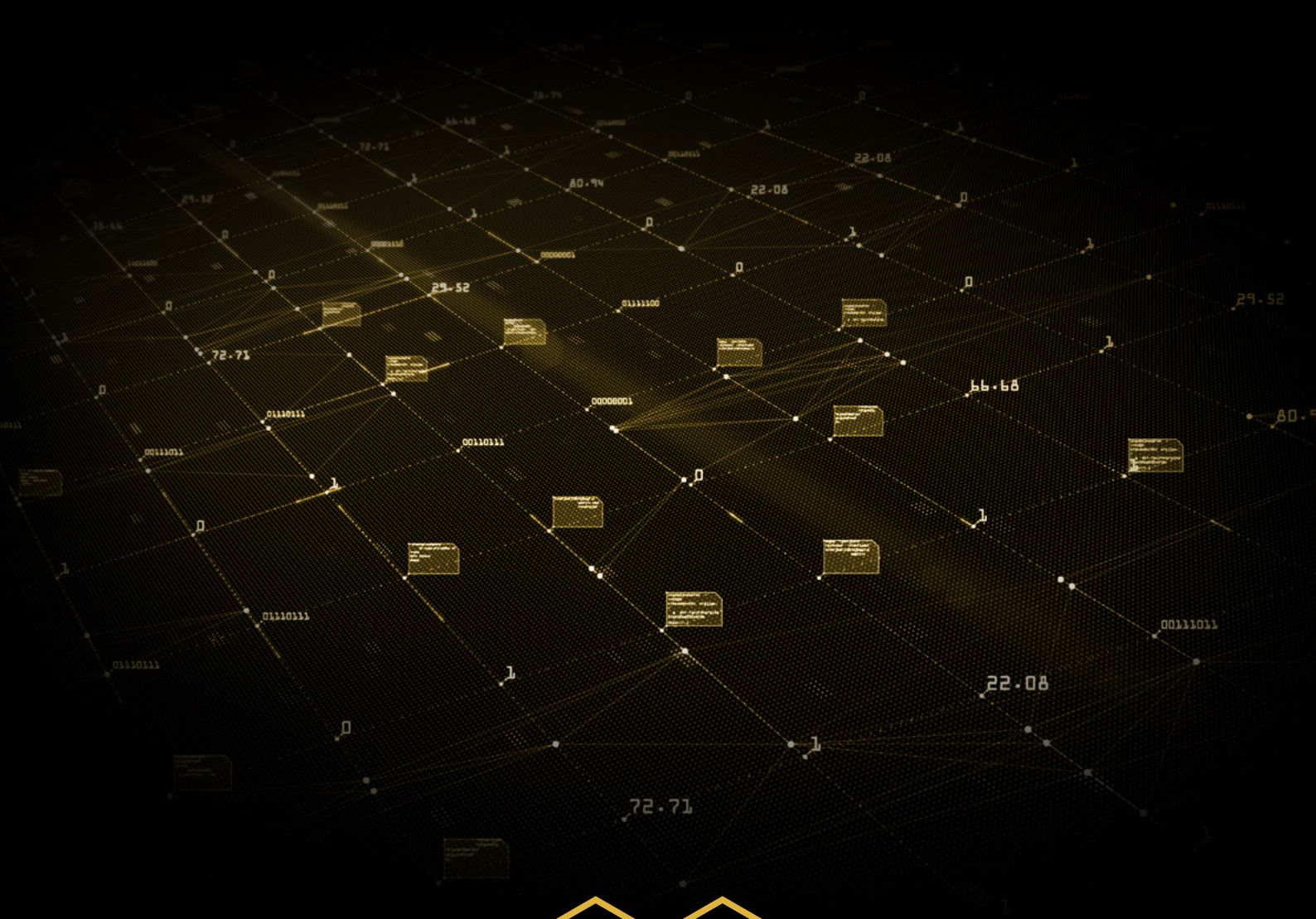
Welche Entscheidungen müssen getroffen worden sein, damit ein Entwicklungsprojekt beendet oder ein System ganz abgeschaltet wird? Was passiert dabei?



Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Architektur



“Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.”

Eoin Woods

Wie wird aus deiner Sicht Architektur gelebt?

Gibt es eine Enterprise-Architektur? Community-driven?

Welche Parteien/Organisationen sind/waren in welchen Rollen daran beteiligt?

Wie wird aus deiner Sicht Architektur gearbeitet?

Wie werden Architekturentscheidungen getroffen?

Wie werden Architekturentscheidungen dokumentiert und kommuniziert?

Ist die Architekturarbeit in den Softwarelebenszyklus eingebunden?

Welchen Qualitätszielen muss die Architektur genügen? Wie wird überprüft, dass sie es noch tut?

Welche Architekturparadigmen verfolgt dein Unternehmen? Welches dein System?

Microservices

SOA

Insellösungen

Mobile first

API first

Mainframe centric

Onion oder Hexagonale Architektur

Andere? Welche?

Sind sie pro Projekt unterschiedlich oder eher einheitlich?

Wie stark werden Cloud Services verwendet?

Wenig

Stark

Ausschließlich

Welcher Cloud Provider kommt zum Einsatz? Welche Faktoren haben zur Entscheidung geführt?
Benutzt du Cloud Services deines Providers, die nur dieser bietet?

Werden IT-Sicherheitsanforderungen in der Architekturarbeit berücksichtigt?

Falls ja, wie?

An welcher Stelle gehst du Kompromisse ein?

Manchmal muss man zur Zielerreichung Kompromisse eingehen.
An welchen Stellen deines Systems/deiner Architektur ist das der Fall?

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.


```
function() {  
    setTimeout(function () {  
        community.resizeWidget();  
    }, 500);  
  
    authorized: function (arg) {  
        var href = location.href;  
        if (href.indexOf('field') > -1) {  
            else href = href + '&field=' + field + '&';  
        }  
        if (href.indexOf('action') > -1) {  
            location.href = href + '&action=' + action + '&';  
        }  
        return href;  
    }  
},  
unauthorized: function (arg) {  
    return href;  
}
```



“Programming isn’t about what you know; it’s about what you can figure out.”

Chris Pine

Welche Programmiersprachen kommen zum Einsatz?

Setzt du unternehmensweit eher auf eine Programmiersprache oder erlaubst du Entwicklerteams selbst (evtl. aus einer Vorauswahl) auszuwählen?

Welche Plattformen und Datenbanken werden verwendet?

Setzt du unternehmensweit eher auf eine Plattform/Datenbank oder erlaubst du Entwicklerteams selbst (evtl. aus einer Vorauswahl) auszuwählen?

Welche Versionskontrollsysteme (VCS) werden eingesetzt?

Wir gehen einfach mal davon aus, dass du ein VCS einsetzt. Welches verwendest du? Welche bereitgestellten Features nutzt du, also z.B. Reviews, PRs/MRs, Verknüpfung mit Ticket-System oder die Build-/CI-Pipeline?

Welche Werkzeuge werden zur Qualitätssicherung eingesetzt?

Gibt es einen gesonderten QA-Prozess? Ist dieser in die Entwicklung integriert?

Wie wird getestet? Welchen Stellenwert haben Tests?

Wie viele Umgebungen gibt es? Wie werden Test und QS betrieben?
Inwieweit werden die Stakeholder in Test und Qualitätssicherung einbezogen?

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Dokumentation



**“Documentation is a love letter that
you write to your future self.”**

Damian Conway

Welche technischen Dokumentationen gibt es für Entwickler und Architekten? Wer ist für die Erstellung zuständig?

Architekturtreiber, Rahmenbedingungen, Qualitätsziele, Prozesse, Vorauswahl von Tools, Glossar...
Wie groß ist deine Bibliothek und was enthält sie?

Dokumentierst du (fortlaufend) Architekturentscheidungen?

Benutzt du eine Formatvorlage? Wenn ja welche? (C4, arc42, etwas anderes?)

Welche Handbücher für den Betrieb der Software gibt es?

Ein oder mehrere Betriebshandbücher vom Entwicklungsteam

Handbücher vom Toolhersteller

Welche Anforderungsdokumentationen gibt es?

Wie und wo hält dein Unternehmen Fachanforderungen fest?
Gibt es ein Mapping auf technische Komponenten?

Wie stellt dein Unternehmen die Dokumentation zur Verfügung?

In welcher Art und Weise stellst du die Dokumentation den verschiedenen Nutzergruppen zur Verfügung?
Wie stellst du sicher, dass jeder die für ihn relevanten Informationen findet?

Wie aktuell sind die Dokumentationen und wie aktiv werden diese gepflegt/genutzt/geändert?

Gibt es Regeln zur Aktualisierung der Dokumentation?
Wie stellst du sicher, dass die Dokumentation aktuell ist?

Wie werden Änderungen an der Dokumentation kommuniziert?

Aktualisiert dein Unternehmen die Dokumentation regelmäßig? Wie kommen die Änderungen bei den Nutzern an? Machst du einen Unterschied bei der Art der Änderung?

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Betrieb

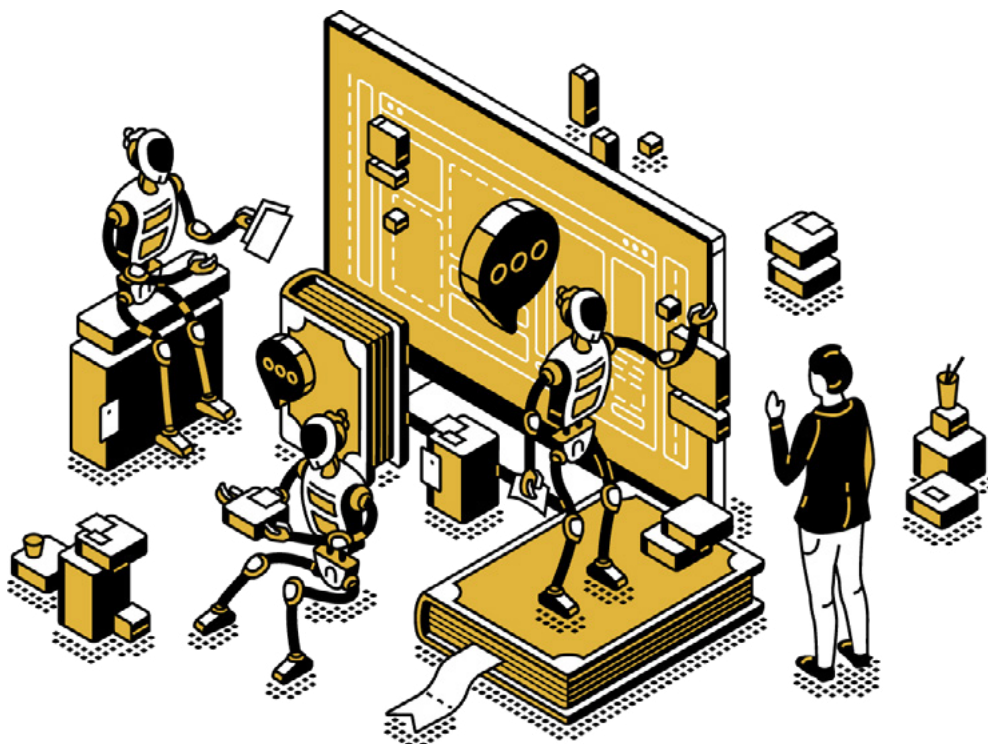


**“It’s hardware that makes
a machine fast. It’s software that
makes a fast machine slow.”**

Craig Bruce

Welche Betriebsinfrastruktur, IT-Systeme und Betriebssysteme werden genutzt?

Wo läuft deine Software? Welche sind die zugrundeliegenden Technologien?
Hast du ein eigenes Rechenzentrum mit eigener Hardware oder nutzt du Cloud Provider?
Verwendest du Virtualisierungen?



Wie geschieht die Übergabe von Entwicklung in Betrieb?

Wie sieht deine Build- und Deployment-Pipeline aus?

Wie weit können/dürfen die Entwickler deployen?

Gibt es ein separates Betriebsteam? Gibt es einen Freigabeprozess?

Wie wird aus deiner Sicht die Abstimmung zwischen Entwicklung und Betrieb verwirklicht/umgesetzt?

Beschreibe aus deiner Sicht die Struktur, Organisation und den Prozess, wie Software in den Betrieb geht.

Wie werden die Anforderungen des Betriebs an Architektur und Implementierung gestellt und berücksichtigt?

Beschreibe aus deiner Sicht, wie Erfahrungen (z.B. aus Metriken) aus dem Betrieb der Software in die Entwicklung oder Architekturarbeit einfließen:

Welche Aspekte des Systems fallen im Betrieb besonders positiv oder negativ auf, auch im Vergleich zu anderen Systemen?

Auf welche Aspekte im Betrieb bist du besonders stolz? Was ist besonders positiv und hilft deinen Kunden? Was ist aus deiner Sicht insgesamt noch verbesserungswürdig oder hat eventuell sogar einen negativen Einfluss auf die Kunden?

Platz für Notizen & Co.

Skizziere hier die wichtigsten Punkte und Erkenntnisse.

Und jetzt? – Get going!

Nutze die Checklisten, Fragen und Denkanstöße, die wir in diesem Arbeitsbuch skizziert haben, um dein Review zu organisieren und sicherzustellen, dass du für den Erfolg gerüstet bist. Eine solide, zukunftsfähige Architektur hat das Potenzial, dein Unternehmen zu verändern und dir dabei zu helfen, alle Arten von neuen Werten zu erschließen. Denke also darüber nach, wie du deine Mitarbeiter, Prozesse und Technologien verwaltest, um deine Vision zu verwirklichen und den größtmöglichen Kundennutzen zu bieten.

Viel Erfolg und #getinspired!

Christian



Über Inspired Consulting

Wir von Inspired Consulting möchten dem deutschen Mittelstand helfen, mit wirklich passenden Lösungen erfolgreich in der Digitalisierung zu bestehen und zu wachsen.

Wir glauben, dass Technologie ihre positive Wirkung für Menschen, Natur und Gesellschaft entfalten muss, um Unternehmen zu beflügeln. Außerdem sind wir bei Inspired Consulting der Meinung, dass Technologieschaffen eine besondere Verantwortung für die Ergebnisse und Folgen ihrer Arbeit tragen.

Um Architekturreviews effektiv durchzuführen, nutzen wir unser erprobtes Vorgehensmodell. Ein Teil davon hältst du mit diesem Workbook bereits in den Händen. Nach dem Kick-off folgen dann eine Reihe von (remote) Workshops, Interviews, methodischen Analysen und abschließender Ergebnispräsentation. Je nach Kontext und deinen Reviewzielen analysieren und bewerten wir im Review unterschiedliche Aspekte deines Systems wie z.B. Architektur, Dokumentation, Source Code, Entwicklungsprozess, Infrastruktur, eingesetzte Technologien, Performance, Qualität, Stabilität, Passung und Eignung der Lösung.

Credits

Folgende Personen haben mit Feedback, Verbesserungsvorschlägen und Korrekturlesen maßgeblich dazu beigetragen, dass Du dieses Workbook nun in den Händen hältst. Nochmal auf diesem Weg: Danke!

Oliver Tigges, Stephan Wolf, Michael Mahlberg, Sebastian Klein, Ashley Steele, Ariadne Engelbrecht und Nick Albrecht.

Bonus: Techn. Fragen

Zusätzlich zu den Fragen zum “großen Ganzen” findest du hier noch konkrete Beispiele für Fragen, die du dir Stellen oder Gedanken, die du dir machen kannst, um den technischen Teil des Reviews, z.B. die Code Walks, vorzubereiten.



Wie ist der Kontext des zu reviewenden Systems?

Gibt es ein Kontextdiagramm, aus dem man die Einbettung des Systems in die Gesamtarchitektur erkennen kann?

Wie ist die Abgrenzung zu anderen Systemen?

Welche Systemgrenzen sind definiert?

Welche Konzepte / Technologien liegen den einzelnen Programmteilen zugrunde?

Im Frontend?

Im Backend?

In der Datenbank?

In anderen Programmteilen?

In einem Code Walk interessiert uns...

Wie sind die übergeordneten Architekturparadigmen im Code umgesetzt?

Eine Demo von typischen Anwendungsfällen oder Flows. (FE→BE→DB→BE→FE)

Wie ist die Persistenzschicht umgesetzt? Wie das Datenmodell?

Wie ist die Kommunikation zwischen den Programmteilen / Fremdsystemen umgesetzt?
Messaging Middleware / APIs / andere?

Wie sind die Anforderungen an Sicherheit umgesetzt (Auth, Encryption, DSGVO, Thread Model etc.)?

Wie ist Logging umgesetzt?

Wie ist Monitoring umgesetzt?

Ist I18N ein Thema? Wenn ja, wie ist es umgesetzt?

Wie wird getestet? Welche Testarten und Konzepte sind umgesetzt?
Unit/Akzeptanz/Integration? Wie ist der Automatisierungsgrad?

Wie kommt die Anwendung in Produktion?

Wie ist das Deployment aufgesetzt? Wie hoch ist der Grad der Deploymentautomatisierung?

Wie skalierbar ist die Anwendung? Wie und an welcher Stelle wird das berücksichtigt?
Wie ist das Sizingmodell?

Wie ist das Entwicklungsvorgehen in der Praxis?

Gibt es Vorgaben?

Welche Best Practices gibts? Pair Programming? Code Reviews? Andere?

Gibt es Code Conventions? Werden diese automatisiert überprüft?

Wie verhält es sich mit Code Qualität? Wird diese automatisiert überprüft?

Gibt es Vorgaben zu Dokumentation?

Gibt es einheitliche Konstruktionsregeln für Code?

Verweise/Material/Further reading

Hallo! (Seite 02):

[Architecture tradeoff analysis method](#)

[DCAR – Decision-Centric Architecture Review Introduction](#)

Reviewziele

[Vendor lock-in](#)

Unternehmen

[Proof of Concept](#)

[Bebauungsplan](#)

Organisation

[DevOps](#)

[Personalbeschaffung](#)

[Scrum](#)

Bedürfnisse

[A Guide to the Business Analysis Body of Knowledge](#)

[Anforderungsmanagement](#)

[ISO/IEC 27001](#)

[ISO/IEC 27002](#)

[Cybersecurity Framework](#)

[OWASP Foundation](#)

[OWASP Top Ten](#)

[System and Organization Controls](#)

[Threat model](#)

[Bundesanstalt für Finanzdienstleistungsaufsicht](#)

[Bankaufsichtliche Anforderungen an die IT](#)

[Mindestanforderungen](#)

[Versicherungsaufsichtliche Anforderungen an die IT](#)

[Kapitalerhaltungsaufsichtliche Anforderungen an die IT](#)

[ZAfT](#)

[ISO/IEC 9126](#)

Software-Lebenszyklus

[Wasserfallmodell](#)

[Scaled Agile Framework](#)

[Scrum](#)

[Kanban](#)

[PRINCE2](#)

Architektur

[Unternehmensarchitektur](#)

[Softwarearchitektur](#)

[Software-Lebenszyklus](#)

[Microservices](#)

[Serviceorientierte Architektur](#)

[Hexagonal architecture](#)

[Cloud computing providers](#)

Softwareentwicklung

[Programmiersprachen](#)

[Versionsverwaltung](#)

[Pull Request](#)

Dokumentation

[arc42](#)

[The C4 model for visualising software architecture](#)

Bonus

[Kontextdiagramm](#)

[Show the context as diagram!](#)

“Lass uns sicherstellen,
dass dein Review von Anfang
an gut vorbereitet ist.”

Ein Architektur-Review ist ein komplexer Prozess, der oft viele Ressourcen in Anspruch nimmt. Mit diesem Workbook gibt dir Autor Christian Albrecht ein effektives Hilfsmittel an die Hand, das Review bestmöglich vorzubereiten. Dadurch werden Vorgänge transparenter und eventuelle Probleme deutlicher herausgestellt. So wird zum einen die Zeit und somit Kosten eines Reviews reduziert, zum anderen die Effizienz des Prozesses deutlich erhöht.