INNOQ

Canvas 101

One-Pager für bessere Kommunikation

Markus Harrer • Anja Kammer • Lena Kraaz • Jörg Müller
Patrick Roos • Gernot Starke • Benjamin Wolf

Canvas 101

One-Pager für bessere Kommunikation

Markus Harrer
Anja Kammer
Lena Kraaz
Jörg Müller
Patrick Roos
Gernot Starke
Benjamin Wolf

innoQ Deutschland GmbH
Krischerstraße 100 · 40789 Monheim am Rhein · Germany
Phone +49 2173 33660 · www.INNOQ.com

Layout: Tammo van Lessen with XeLaTeX
Umschlag: Murat Akgöz
Typesetting: André Deuerling

Canvas 101 – One-Pager für bessere Kommunikation
Published by innoQ Deutschland GmbH
1. Auflage · Februar 2025

Inhaltsverzeichnis

1 Einführung 1

Empfehlungen ... 3

2 Die Canvases 5

2.1 Business Model Canvas ... 5

2.2 Architecture Inception Canvas ... 10

2.3 Architecture Communication Canvas ... 22

2.4 Tech Stack Canvas ... 33

2.5 Software Analytics Canvas ... 43

2.6 Team Communication Canvas ... 50

2.7 Meta-Canvas ... 62

3 Tools 65

Papier ... 65

Digitale Zeichenwerkzeuge ... 65

Online-Werkzeuge ... 66

PowerPoint® und Co. ... 66

4 Anhang 69

Autor:innen 71

1 Einführung

> **Das Ganze ist mehr als die Summe seiner Teile.**
>
> **– Aristoteles**

Mit Canvas bezeichnen wir die strukturierte Visualisierung von Sachverhalten - und *nicht* die Zeichenfläche der Malerei (oder der grafischen Programmierung etwa mit Java-Swing oder HTML5). Damit erreichen wir die kürzestmögliche Dokumentation, vom zugrunde liegenden Geschäftsmodell über konkrete Anforderungen, die Lösungsarchitektur, betriebliche Aspekte bis hin zur Zusammenarbeit im Team.

Wir zeigen Ihnen Struktur und Inhalte verschiedener Canvases[1], die wir in der Realität anspruchsvoller IT-Projekte zu schätzen gelernt haben (und immer wieder gerne verwenden). Einige davon haben wir selbst entwickelt und durch Feedback aus der Praxis verfeinert.

- Business Model Canvas: Sicherlich **der** Canvas schlechthin: Der klassische Steckbrief für Geschäfts- oder Business-Modelle.
- Architecture Inception Canvas: Der solide Start in IT-Vorhaben, von einem Überblick der Anforderungen zu ersten Lösungsideen.
- Architecture Communication Canvas: Architekturdokumentation in Kurzform, passend zu arc42.
- Software-Analytics Canvas: Grundlage systematischer Verbesserung.
- Tech-Stack Canvas: Die technischen Grundlagen Ihres Systems.
- Team Communication Canvas: Wie wollen wir zusammenarbeiten?

All diese *Steckbriefe* hängen inhaltlich eng zusammen:

[1] Als Mehrzahl verwenden wir *Canvases*, weil es der klassischen Pluralform der englischen Sprache entspricht.

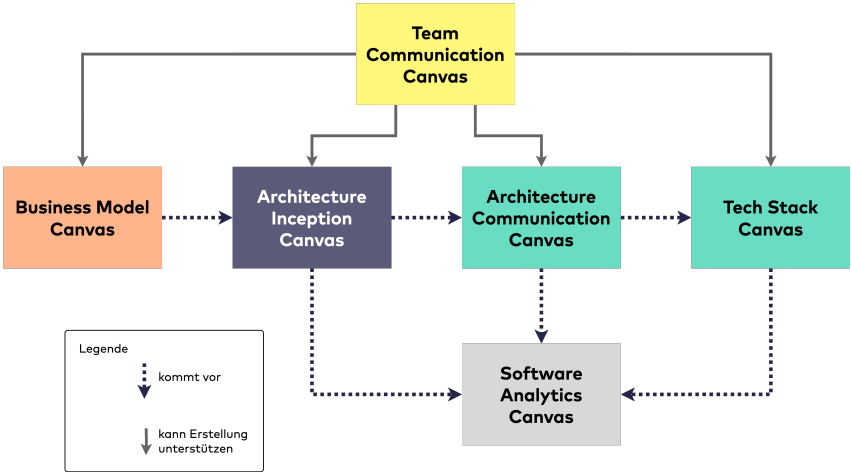


Abbildung 1.1: Inhaltlicher Zusammenhang der Canvases

Empfehlungen

- **Erstellen Sie Ihren Canvas im Team** oder zumindest einer (kleinen) Gruppe. Dadurch erhalten Sie einen breiteren Blick auf die Situation und werden mehr interessante Aspekte berücksichtigen. Die Akzeptanz steigt beträchtlich, und außerdem macht es mehr Spaß (was beim Thema Dokumentation ein gewichtiges Argument darstellt).
- **Erst die Inhalte, dann die Werkzeuge**: Statt lange über das passende Werkzeug zu diskutieren, sollten Sie sich um die Inhalte Ihres Canvas kümmern. Wir haben ein Kapitel zu **Werkzeugen** aufgenommen, ab Seite 65, und stellen einige unserer Favoriten vor.
- **Holen Sie Feedback zum Canvas ein**, am besten von unterschiedlichen Stakeholdern. Dadurch validieren Sie Hypothesen und können den Inhalt schnell verbessern.
- Arbeiten Sie **pragmatisch** mit den Canvases: Lassen Sie ruhig eine Sektion frei, oder kennzeichnen Sie sie mit *noch unklar* oder *under discussion*. (Beispiel: Im Architecture Communication Canvas nennen manche User die Sektion „Risiken“ in „Technische Schulden“ um, weil ihnen diese Information wichtiger scheint.)

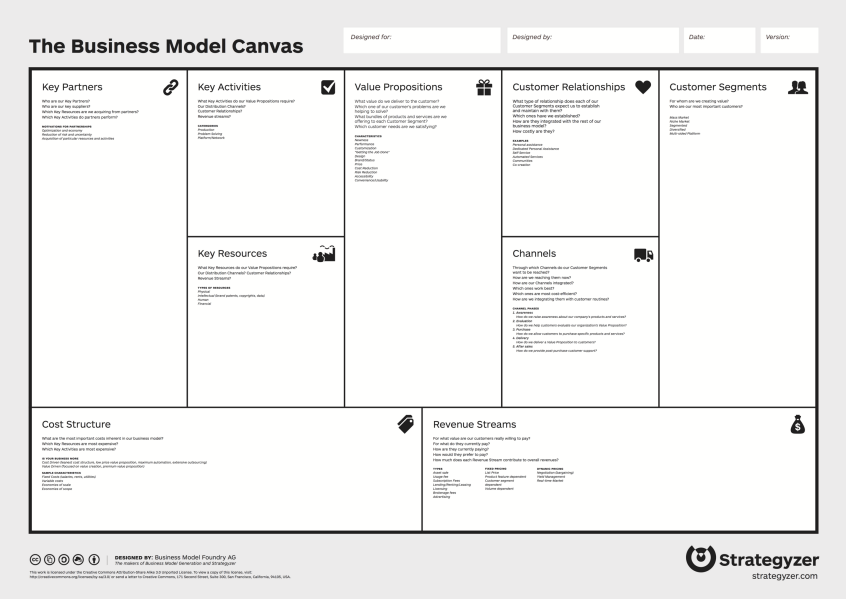
2 Die Canvases

2.1 Business Model Canvas

Von Gernot Starke.

Das Geschäftsmodell ist das Herzstück jeder erfolgreichen Innovation."

Henry Chesbrough[1]



Motivation

Der Business Model Canvas (kurz: BMC) von Alex Osterwalder und Jaques Pigneur stellt *Geschäftsmodelle* übersichtlich auf einer einzigen Seite dar. Er fördert

[1] https://en.wikipedia.org/wiki/Henry_Chesbrough

gemeinsames Verständnis, erleichtert Diskussionen und unterstützt die kontinuierliche Verbesserung von Geschäftsideen.

In IT-Projekten kann der BMC als Einstieg in die Entwicklung von Systemen sowie das Requirements-Engineering dienen. Zwar leitet der Architecture Inception Canvas noch IT-spezifischer auf konkrete Anforderungen hin, dafür genießt der BMC jedoch einen immens hohen Bekanntheits- und Verbreitungsgrad.

Zielgruppe

Vor allem Gründer:innen, Manager:innen, Berater:innen, Produkt- und Strategieteams profitieren vom Business Model Canvas. Auch Investor:innen, Innovationsabteilungen und Lehrkräfte nutzen ihn, um Geschäftsmodelle rasch zu erfassen und zu vermitteln.

Inhalte

Der BMC gliedert Geschäftsmodelle in neun Bausteine:

1. Kundensegmente: Wer sind die wichtigsten Kundengruppen?
2. Wertangebote: Welche Produkte oder Dienstleistungen erfüllen Kundenbedürfnisse?
3. Kanäle: Über welche Wege erreichen Angebote die Kunden?
4. Kundenbeziehungen: Wie pflegt das Unternehmen die Kundenkontakte?
5. Einnahmequellen: Wie erzielt das Geschäftsmodell Umsätze und Erträge??
6. Schlüsselressourcen: Welche Ressourcen sind entscheidend für den Betrieb?
7. Schlüsselaktivitäten: Welche zentralen Aufgaben sichern den Erfolg?
8. Schlüsselpartner: Wer unterstützt das Geschäftsmodell effektiv?
9. Kostenstruktur: Welche Kosten entstehen?

Wichtig zu bemerken: Lesen und bearbeiten Sie den BMC von links nach rechts: Von den wesentlichen Partnern geht es über Aktivitäten und Ressourcen zum Wertversprechen, und von dort über Kanäle und Beziehungen zu den Kundensegmenten. Diese logische Reihenfolge bildet das Rückgrat des BMC und macht eine wesentliche Stärke aus.

Vorgehen

Teams entwickeln den Canvas oft iterativ. Sie starten z. B. beim Wertangebot und Kundensegmenten, fügen anschließend weitere Bausteine hinzu und prüfen immer wieder die Zusammenhänge.

Quellen

- Die Erfinder des BMC bieten bei Strategyzer[2] Downloads, Videos und weitere Infos rund um diese *mother of all canvases* an.
- Das Buch schlechthin zum Thema BMC ist „Business Model Generation“ von Osterwalder/Pigneur ([1]). Neben dem großartigen Inhalt brilliert dieses Buch auch durch das innovative Layout und die grafische Gestaltung.

[2] https://www.strategyzer.com/library/the-business-model-canvas

Beispiele

Zoom Inc: Video-Meetings

Business Model Canvas — Entwickelt für: zoom Meetings — Entwickelt von: Miro Inc. / Gernot Starke — Januar 2025

Schlüsselpartner

- Cloud-Hosting-Anbieter (z. B. AWS, Microsoft Azure).
- Tech-Unternehmen für Integrationen (z. B. Microsoft Teams, Slack, Box, Miro etc.).
- Reseller und Vertriebspartner.
- Bildungseinrichtungen und Unternehmen für Großkundenverträge.
- Anbieter von Kalender- und ToDo-Apps

Schlüsselaktivitäten

- Entwicklung und Pflege der Softwareplattform.
- Verbesserung von Audio und Videofunktionen.
- Marketing und Kundengewinnung.
- Skalierung der Cloud-Infrastruktur.
- Kundenbetreuung und Schulungsangebote.

Schlüsselressourcen

- Backend-Infrastruktur für Video Kommunikation.
- Software-Teams für Produktverbesserung.
- Server- und Cloudkapazitäten für die Bereitstellung der Dienste.
- Markenname.

Wertversprechen

Hochwertige, benutzerfreundliche Videokonferenz Lösungen für Unternehmen und Privatpersonen:

- **Einfach** zu bedienen und zuverlässig.
- **Hochwertige** Audio- und Videoqualität mit geringen Latenzen.
- **Skalierung** von Einzelgesprächen bis zu großen virtuellen Events.
- Funktionen wie **Bildschirmfreigabe, Breakout-Räume**, Webinar-Hosting und Chat.
- Hohe **Sicherheit** und **Datenschutz**

Kundenbeziehungen

- Selbstbedienung durch intuitive Oberfläche.
- Kundensupport mit Chatbots und FAQs.
- Personalisierter Support für Premium-Kunden.
- Community-Plattform.
- Schulungsangebote für optimale Nutzung der Software

Kanäle

- Eigene Website für Direktverkauf und Information.
- App-Stores.
- Vertriebspartner und Reseller.
- Werbung auf sozialen Medien.
- Partnerschaften mit IT- und Telko-Dienstleistern.

Kundensegmente

- **Unternehmen** (klein, mittel, groß) für Teamkommunikation und virtuelle Meetings.
- **Bildungseinrichtungen** (Universitäten, Schulen) für Online-Unterricht.
- **Selbstständige** und Freiberufler, die professionelle Kommunikation benötigen.
- **Veranstalter** für virtuelle Events und Webinare.
- **Privatpersonen** für persönliche virtuelle Kommunikation

Kostenstruktur

- Betrieb und Wartung der Server- und Cloudinfrastruktur.
- Entwicklungskosten für Software sowie Sicherheitsupdates.
- Marketing- und Vertriebskosten.
- Support- und Schulungskosten.

Einnahmequellen

- **Abonnements**: Verschiedene Preismodelle (Basic, Pro, Business, Enterprise).
- **Einmalige Gebühren** für Webinare und virtuelle Events.
- **Kostenpflichtige Add-ons** wie mehr Speicherplatz oder Premium-Funktionen.
- Partnerschaften und Lizenzeinnahmen (z. B. Integration in andere Software).

AirBnb

Business Model Canvas

Entwickelt von: Gernot Starke | Januar 2025

Schlüsselpartner

- **Gastgeber** (Immobilienbesitzer)
- **Zahlungsdienstleister** (z. B. Stripe, PayPal)
- Lokale **Regulierungsbehörden**
- **Versicherungsanbieter**
- **Marketing-** und Werbepartner

Schlüsselaktivitäten

- Entwicklung und Pflege der Softwareplattform.
- Akquise von Gastgebern und Gästen
- Kundenservice /Support
- Internationaler Markenaufbau

Schlüsselressourcen

- **Plattform** und **Infrastruktur**.
- **Markenname** und Vertrauen der Community.
- **Datenbank von Unterkünften** und Nutzerdaten.

Wertversprechen

Für **Reisende**:

- Vielfalt & Authentizität: Große Auswahl an Unterkünften weltweit,.
- Kosteneffizienz: Oft günstiger als Hotels.
- Einzigartige Erlebnisse: Lokale Aktivitäten und sichere Buchung.

Für **Gastgeber**:

- Einfache Vermietung: Flexible Vermietung über intuitive Plattform.
- Unterstützung: Versicherung, Kundenservice und Tools zur Preis- und Kalenderverwaltung.

Kundenbeziehungen

- **Automatisierte Buchung** und Kommunikation.
- **Kundenservice** und Support.
- **Community-Bildung** durch Bewertungen und Empfehlungen.

Kanäle

- Eigene Website für Direktverkauf und Information.
- Mobile-Apps in bekannten App-Stores.
- Werbung auf sozialen Medien.

Kundensegmente

- **Reisende** (Urlauber, Geschäftsreisende) suchen flexible und authentische Unterkünfte.
- **Gastgeber** (Privatpersonen, Immobilienbesitzer) möchten Einkünfte durch Vermietung erzielen.

Kostenstruktur

- Betrieb und Wartung der Server- und Cloudinfrastruktur.
- Technologie- und Plattformentwicklung.
- Marketing und Werbung.
- Kundenservice und Community-Management.
- Support und Versicherungen.

Einnahmequellen

- **Vermittlungsgebühren von Reisenden** (Buchungsgebühren).
- **Gebühren von Gastgebern** (Provision).

2.2 Architecture Inception Canvas

Von Patrick Roos.

> Der kollaborative Kickstart für Ihre Softwarearchitektur.

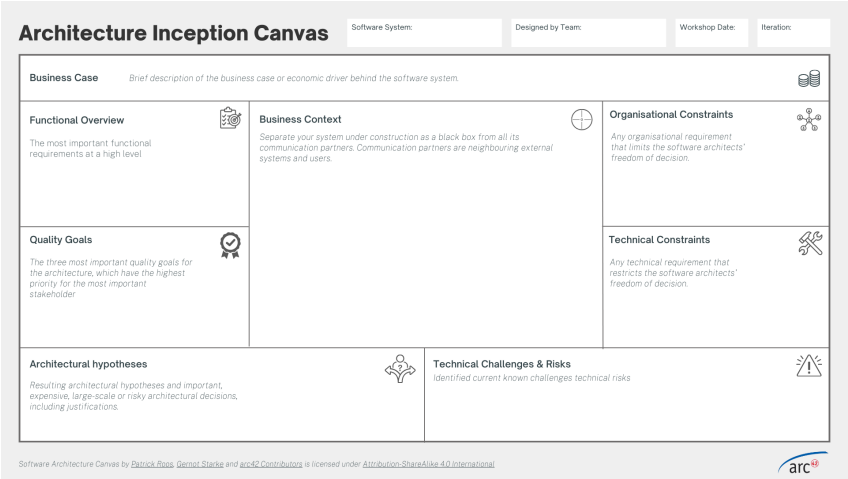


Abbildung 2.1: Architecture Inception Canvas

Motivation

Das Szenario, welches wahrscheinlich jede:r Entwickler:in liebt: Der Start einer neuen Softwareinitiative und die damit verbundenen technischen Fragestellungen:

- Sollen wir einen Microservices-Ansatz verfolgen?
- Funktional oder objektorientiert?
- Folgen wir dem hexagonalen Architekturansatz?
- Machen wir ein Event-driven Design mit CQRS?
- Oder probieren wir mal einen Serverless-Ansatz?
- Java, F#, Go oder Rust?

- Azure, AWS oder GCP?
- NoSQL vs. SQL? MongoDB vs. PostgreSQL?
- Angular vs. React vs. Solid?
- Docker vs. Podman?
- Continuous Integration und Deployment Pipelines? GitOps?

Das sind ganz viele spannende technische Fragestellungen für Technologie-Enthusiasten wie uns.

Aber Moment – sind diese Fragen überhaupt relevant am Anfang einer Softwareinitiative?

Entwicklungsteams geraten zu Beginn einer Softwareinitiative leider oft in die gleiche typische Falle: der starke Drang, sich ausführlich mit den faszinierenden, technologisch getriebenen Detailfragen auseinanderzusetzen.

Natürlich diskutieren wir gerne über Technologien und trendige architektonische Ansätze. Aber diese Diskussionen sind zu Beginn einer neuen Softwareinitiative nutzlos, weil man zuerst sein „Architektur- und Technologiespielfeld" definieren muss.

Wenn man als Team sein „Spielfeld" nicht kennt, hat man keine Grundlage, um technische Entscheidungen zu fällen und entsprechend zu begründen.

Aus der Erfahrung zeigt sich, dass Produktteams stets dieselben grundlegenden Fragen klären müssen, bevor die Diskussion technischer Themen sinnvoll ist.

Deshalb hat Patrick Roos siehe [2] den Architecture Inception Canvas (AIC) entwickelt, der Produktteams dabei hilft, die Grundlage für Softwarearchitekturarbeit effizient und kollaborativ zu erarbeiten.

Zielgruppe

Der Architecture Inception Canvas (AIC) lädt das ganze Produkt- oder Projekt-Team dazu ein, gemeinsam die Grundlage für die Softwarearchitektur kollaborativ zu erarbeiten.

Der AIC ist für alle geeignet, die an Softwarearchitektur arbeiten, beispielsweise:

- Product Owner:innen
- Projektmanager:innen
- Business Analyst:innen
- UX Designer:innen
- Softwareentwickler:innen
- Softwarearchitekt:innen
- Tester:innen
- ...

Inhalte

Der Architecture Inception Canvas besteht aus den folgenden drei Hauptbereichen:

- **Ziel**: Was soll die Software leisten?
- **Lösung**: Wie können wir das erreichen?
- **Bewertung**: Wie beurteilen wir die aktuelle Situation hinsichtlich Herausforderungen und Risiken?

Business Case - das Warum und Wozu

Wenn man als Unternehmen Geld verdienen muss, gibt es hinter jeder Softwareinitiative einen wirtschaftlichen Treiber bzw. einen Business Case. Wenn nicht, handelt es sich um ein Hobby- oder NPO-Projekt.

Ein häufiger Fehler ist es, dass das Entwicklungsteam – oder schlimmer noch, das gesamte Produktteam – sich des spezifischen wirtschaftlichen Treibers oder des Business Cases hinter der Softwareinitiative nicht bewusst ist.

Wenn Sie den wirtschaftlichen Treiber oder den Business Case hinter der Softwareinitiative nicht kennen oder erklären können, sollten Sie nicht weitermachen. Sie sind als Produktteam so nicht in der Lage, einen sinnvollen Rahmen für

Abbildung 2.2: Die drei Hauptbereiche des AIC

Ihre Softwarearchitektur zu definieren, weil Sie nicht genau wissen, welche High-Level-Treiber Ihre Softwarearchitektur maßgeblich bestimmen.

Deshalb ist es essenziell, dass jedes Teammitglied den wirtschaftlichen Treiber kennt und den Business Case hinter der Softwareinitiative erklären kann.

Mit dem Abschnitt „Business Case“ des Architecture Inception Canvas setzen Sie jedem Teammitglied den unternehmerischen Hut auf und befähigen so Ihr Team bzw. jedes einzelne Teammitglied, unternehmerisch zu denken.

Oft stärkt dieses Mitdenken die intrinsische Motivation des Teams und unterstützt Sie sowohl bei den täglichen Mikro- als auch Makro-Entscheidungen während der Umsetzung des Softwareprodukts.

Functional Overview - Was auf der Produktbox Ihrer Software stehen sollte

Während Sie als Produktteam den funktionalen Überblick Ihrer Softwareinitiative erarbeiten, hilft die Vorstellung, Sie gestalten eine Produktbox Ihrer Software (sicher kennen Sie die Produktboxen noch von früher?).

Was würde auf der Produktbox Ihrer Software Initiative stehen?

An die Produktbox zu denken, hilft Ihnen, an mögliche Antworten der folgenden Fragestellungen zu kommen:

- Welchen Geschäftswert liefert Ihre Software?
- Welche Vorteile und Mehrwerte bringt sie Ihrem Unternehmen oder Ihren Kunden?
- Was sind die übergeordneten Geschäfts- oder Hauptfunktionen der Software?
- Welche zentralen Fähigkeiten und Funktionen bietet die Software an?

Mit den resultierenden Antworten auf diese Fragen können Sie die Schlüsselanforderungen Ihrer Software beschreiben.

Diese Beschreibung hilft Ihnen, den Fokus auf die wichtigsten funktionalen Anforderungen zu legen und das Gesamtbild Ihrer Software besser zu verstehen.

Quality Goals - die zentralen Treiber der Softwarearchitektur

Hier erarbeiten Sie die drei wichtigsten Qualitätsziele der Stakeholder für das Softwaresystem.

Es ist wichtig, dass diese Qualitätsziele jedem bzw. jeder bekannt sind, da sie die Architektur maßgeblich prägen.

Hilfestellung: Der *Standard ISO/IEC 25010* bietet einen Überblick über mögliche passende Qualitätsattribute, während das moderne *arc42-Qualitätsmodell* (Q42) Ihnen mehr als 150 Qualitätsattribute mit ausführlichen Erklärungen und Beispielen sammelt.

Business Context - den Scope entdecken

Der Business Context betrachtet Ihr System als eine Blackbox. Der Fokus liegt auf der Umgebung des Systems und darauf, wer das System benutzt und wie das System mit den externen Systemen interagiert. Mit dem Business Context können Sie die Kommunikationswege identifizieren und dadurch potenzielle Risiken aufdecken. Er dient zudem als wertvolles Werkzeug, um mit verschiedenen

Stakeholdern den Umfang des Systems sowie die Kommunikationswege zu externen Systemen zu diskutieren. Beim Erarbeiten des Business Contexts sollten Sie versuchen, die Nutzer Ihres Systems sowie alle angrenzenden Systeme zu identifizieren.

In diesem Abschnitt erstellen Sie eine geschäftsorientierte Kontextansicht, um eine gemeinsame Sicht auf den geschäftlichen Umfang des Systems zu erhalten. Der Fokus liegt hier auf einem gemeinsamen Verständnis. Sie können den Business Context nutzen, um verschiedenen Stakeholdern das System mit den Aktoren und Umsystemen einfach zu kommunizieren. Der Business Context bildet die Grundlage für einen späteren technischen Kontext. Es ist der Ausgangspunkt für vertiefte Architekturdiskussionen und Identifikation von Risiken.

Bis hierhin waren alle Informationen im AIC textuell - beim *Business Context* können Sie gerne auch ein Diagramm zur Beschreibung verwenden.

Constraints - die Begrenzung der Entscheidungsfreiheit

Organisatorische und technische Einschränkungen bestimmen den Grad Ihrer Entscheidungsfreiheit als Produktteam.

In diesem Abschnitt erarbeiten Sie alle Rahmenbedingungen, die Ihre Freiheit als Produktteam bei der Entwicklung des Softwareprodukts einschränken.

Organisational Constraints

Organisatorische Einschränkungen umfassen häufig Zeit und Budget. Zusätzlich gibt es oft Vorgaben zu den Methoden und Vorgehensweisen, die verwendet werden sollen (z.B. Scrum), oder zu den Techniken, die angewandt werden dürfen bzw. sollen.

Technical Constraints

In größeren Unternehmen gibt es oft zahlreiche technische Einschränkungen. Häufig existiert ein übergeordnetes Architekturkomitee oder Architekturboard, das Standards, Rahmenbedingungen und Prinzipien für die Technologieauswahl setzt.

Ein Beispiel: Wenn Sie sich dazu entscheiden, ein Web-Frontend zu entwickeln, kann das Architekturkomitee festlegen, dass Angular als Frontend-Framework innerhalb des Unternehmens oder eines bestimmten Geschäftsbereichs genutzt wird, um Synergien im technischen Fachwissen zwischen den einzelnen Teams zu schaffen bzw. fördern.

Architectural Hypotheses - die ersten Architekturannahmen treffen

Gratuliere – Sie haben es geschafft!

Sie haben eine erste Iteration durchlaufen und so die erste Grundlage für erste Architekturentscheide ihres zukünftigen Softwareprodukts erarbeitet. Sie können nun auf dieser Basis Ihre ersten Architekturannahmen treffen.

Der Fokus liegt in diesem Abschnitt auf übergeordneten architektonischen Annahmen („Makro-Architektur Hypothesen"), die das Fundament Ihrer Softwarearchitektur bilden, nicht auf detaillierten Design-Fragen.

Idealerweise nutzen Sie diese ersten Architekturannahmen, um erste Entwürfe von Architecture Decision Records (ADR)[3] zu erstellen. So können Sie die initialen Architekturentscheidungen Ihres Softwareprodukts dokumentieren und im Team finalisieren.

Technical Challenges & Risks

Auf Basis des geschaffenen architektonischen Spielfelds – wie Business Context und Umfang, Qualitätsziele, Constraints und erste Architekturannahmen – können Sie die kommenden Herausforderungen und die daraus resultierenden Risiken ableiten bzw. identifizieren.

So können Sie vorbereitend erste risikominimierende Maßnahmen einleiten, wie z.B. Einplanung von Pufferzeiten, Erstellung von Prototypen oder Proof-of-Concepts.

[3] https://www.cognitect.com/blog/2011/11/15/documenting-architecture-decisions

Vorgehen

Der Architecture Inception Canvas (AIC) ist ein kollaboratives Werkzeug, das das gesamte Produktteam dazu einlädt, gemeinsam iterativ die Grundlage für die Softwarearchitektur zu erarbeiten.

Folgendes Vorgehen hat sich bewährt:

- **Vorbereitung für einen ersten on-site Workshop**
    - Laden Sie das AIC-Template als PNG oder PDF herunter und drucken Sie es groß aus oder stellen Sie die einzelnen Abschnitte des AIC analog in guter Größe zur Verfügung.
        - Stellen Sie sicher, dass alle notwendigen Materialien (u. a. Klebezettel, Stifte) vorhanden sind.
            - Es empfiehlt sich, für die drei Bereiche (Ziel, Lösung, Bewertung) unterschiedliche Farben der Klebezettel zu verwenden.
        - Planen Sie mindestens 60 Minuten für die erste Durchführung des AIC ein.
            - Es empfiehlt sich, die Teilnehmer:innen anzuweisen, die Abschnitte des AIC für sich als Vorbereitung auszufüllen, bevor sie die Ergebnisse mit dem Team teilen.
    - Laden Sie alle relevanten Stakeholder ein, mind. Product Owner:in, die Entwickler:innen und Softwarearchitekt:innen des Teams.

- ***Vorbereitung für einen ersten Remote-Workshop***
    - Nutzen Sie das AIC-Template in einem digitalen Kollaborations-Tool wie z.B. Miro oder Mural
    - Es empfiehlt sich, für die drei Bereiche (Ziel, Lösung, Bewertung) unterschiedliche Farben der Sticky Notes zu verwenden.
    - Planen Sie mindestens 60 Minuten für die erste Durchführung des AIC ein.
        - Es empfiehlt sich, die Teilnehmer:innen anzuweisen, die Abschnitte des AIC für sich als Vorbereitung auszufüllen, bevor sie die Ergebnisse mit dem Team teilen.

    - Stellen Sie jeder Workshop-Teilnehmer:in ein dediziertes AIC-Board für die Vorbereitung zur Verfügung.
  - Laden Sie alle relevanten Stakeholder ein, mind. Product Owner:in und die Entwickler:innen und Softwarearchitekt:innen des Teams und stellen Sie sicher, dass alle Zugriff auf das digitale Kollaborations-Tool haben.

- **Durchführung des Workshops**
  - Beginnen Sie mit einer kurzen Einführung in den AIC und erläutern Sie kurz die einzelnen Abschnitte.
  - Es empfiehlt sich folgendes Vorgehen pro Abschnitt:
    - Jedes Team-Mitglied präsentiert die Vorbereitung pro Abschnitt.
    - Die Punkte werden anschließend geclustert und diskutiert.
    - Die konsolidierten Ergebnisse bleiben anschließend auf dem AIC stehen.
  - Versuchen Sie die Timebox von 60 Minuten einzuhalten. Falls es sich abzeichnet, dass Sie mehr Zeit benötigen, ist es besser, in einer weiteren Iteration die Ergebnisse zu vertiefen.

- **Nachbereitung des Workshops**
  - Die Ergebnisse können anschließend sehr einfach in eine arc42-basierte Architekturdokumentation überführt werden. Der AIC ist vollständig kompatibel mit dem arc42-Template.
    - Das kann auch ein guter Zeitpunkt sein, die Architekturdokumentation (im arc42-Format) zu starten. Definieren Sie eine:n Verantwortliche:n, der die Ergebnisse in die Architekturdokumentation überführt.
  - Es können zusätzliche erste Architecture Decision Records auf Basis der Architekturannahmen erstellt werden.
  - Es muss überprüft werden, ob ggf. noch eine weitere Iteration für die Erarbeitung der Softwarearchitektur-Grundlagen notwendig ist.

Empfehlungen

AIC als Kickstart für die Dokumentation Ihrer Softwarearchitektur

Der AIC eignet sich perfekt als Start für die Dokumentation Ihrer Softwarearchitektur. Der AIC ist vollständig kompatibel mit dem arc42-Template[4]. Sie können die Ergebnisse des AIC sehr einfach in eine arc42-basierte Architekturdokumentation überführen.

Die folgende Abbildung zeigt Ihnen die Links auf die Kapitel des arc42-Templates[5].

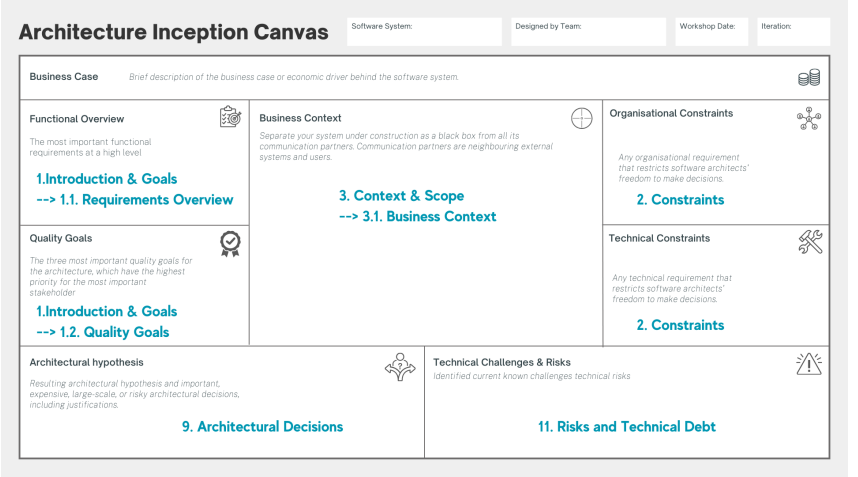


Abbildung 2.3: Kapitel des arc42-Templates im AIC

[4] https://arc42.org/overview
[5] https://arc42.org/overview

Der AIC ist auch der Start für Ihr C4 Model

Sofern Sie planen, mit dem C4-Model[6] zu arbeiten, ist der AIC auch der Start für Ihr C4 Model. Das C4 Model ist ein einfaches Modell, das Ihnen hilft, die Softwarearchitektur auf unterschiedlichen Abstraktionsebenen zu beschreiben. Der Business Context im AIC entspricht nämlich dem System Context Diagram (Level 1)[7] des C4-Models.

Möglichst Technologie-agnostisch arbeiten

Versuchen Sie in einem ersten Schritt möglichst Technologie agnostisch zu arbeiten (insb. alle Ziel relevanten Abschnitte) und sich auf die Eingrenzung bzw. Erarbeitung des „Architekturspielfeldes“ zu fokussieren. Versuchen Sie dies in einem ersten Schritt auch zu forcieren.

[6] https://c4model.com/

[7] https://c4model.com/diagrams/system-context

Beispiele

Auf der offiziellen Website zum AIC[8] finden Sie verschiedene Beispiele, wie der AIC in der Praxis aussehen könnte.

Modernisierung eines Webshop-Systems

Die folgende Abbildung zeigt einen Ausschnitt, wie der AIC in einer ersten Iteration zur Modernisierung eines Webshop-Systems verwendet wurde.

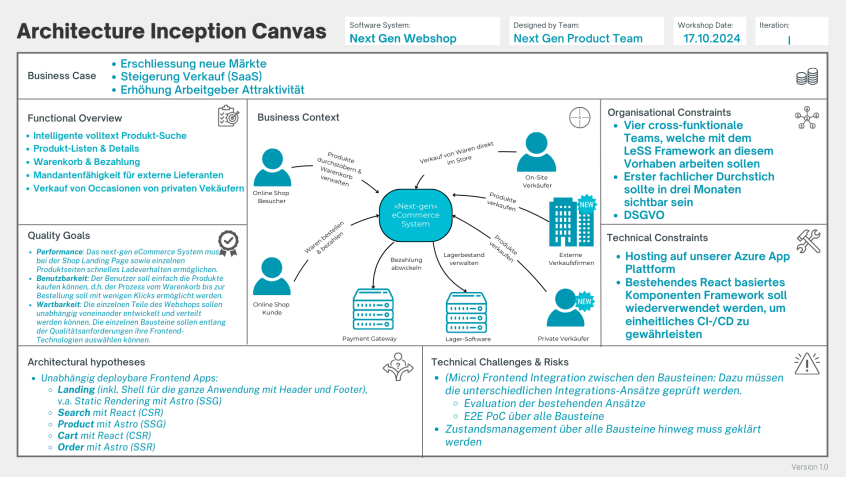


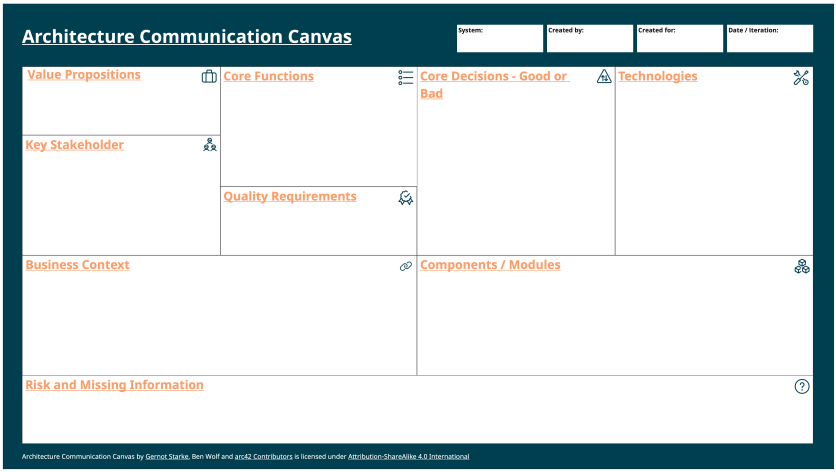
Abbildung 2.4: Ausschnitt einer ersten Iteration mit dem AIC

[8] https://canvas.arc42.org/architecture-inception-canvas

2.3 Architecture Communication Canvas

Von Benjamin Wolf und Gernot Starke.

> Architekturdokumentation in weniger als 60 Minuten.



Der Architecture Communication Canvas (im Folgenden kurz ACC) ist die kompakteste Art der Architekturdokumentation von IT-Systemen. Glauben Sie uns nicht? Dann lassen Sie sich in diesem Kapitel davon überzeugen!

Motivation

Gernot und Ben verwenden seit Jahr(zehnt)en arc42[9] zur Dokumentation von Softwarearchitekturen. Wir wissen, dass wir damit die vielfältigen Interessen von Stakeholdern bezüglich Kommunikation, Dokumentation und Argumentation erfüllen können. Allerdings sind uns immer wieder Situationen begegnet, in denen eine noch einfachere und schnellere Form der Dokumentation notwendig war. Sei es, weil Zeit für Dokumentation fehlte – in unseren Augen eine Ausrede oder ein

[9] https://arc42.org

falsch gelebter Entwicklungsprozess –, oder weil niemand wusste, wie schlanke Dokumentation genau aussehen sollte.

Aus dem ACC können Sie Ergebnisse *direkt* in arc42 überführen. Mehr dazu finden Sie im Abschnitt **Empfehlungen** auf Seite 28.

Mit dem ACC bekommen Sie ein kompaktes Architekturdokument, das inkrementell mit dem Projektfortschritt wachsen kann.

Darüber hinaus soll Sie der ACC anregen, miteinander über das System zu reden. Mehr dazu im Abschnitt **Anwendung des ACC** auf Seite 27.

Zielgruppe

Der ACC kann einige typische Fragen von Architekt:innen, Entwickler:innen, POs und Teamleiter:innen beantworten, analog zu arc42. Durch seine kompakte Form adressiert der ACC jedoch noch weitere Stakeholder, wie z. B. Personen aus dem Fachbereich oder Management, die sich einen kompakten Überblick über ein System verschaffen wollen.

Inhalte

Analog zu arc42 reicht das Spektrum der ACC-Inhalte von Anforderungen zu Lösungs- und sogar Management-Aspekten. Dadurch ergeben sich wertvolle Beziehungen zu einigen anderen Canvases:

- Die übergreifende Zielsetzung (*Value Proposition*) klärt der Business Model Canvas deutlich genauer.
- Details zu Anforderungen finden Sie im **Architecture Inception Canvas** auf Seite 10.
- Falls Sie den ACC begleitend zu einem Review oder Audit einsetzen, beachten Sie unbedingt den **Software Analytics Canvas** auf Seite 43.
- Schließlich gibt der **Tech-Stack Canvas** ab Seite 33 einen detaillierten Überblick der eingesetzten Technologien.

Value Proposition

Welchen Mehrwert bietet das System? Wozu existiert es, was ist der wirtschaftliche oder fachliche Wert, den es generiert? Versuchen Sie, Ihre Antwort auf 140 Zeichen oder 2-3 Halbsätze zu beschränken.

Key Stakeholder

Welche Personen oder Organisationen üben prägenden Einfluss auf das System und die zugehörigen Prozesse aus? Welche haben ein großes Interesse am System, oder dessen Mehrwert (siehe *Value Proposition*) Beschränken Sie sich auf die fünf wichtigsten dieser Stakeholder.

Core Functions

Erklären Sie in Stichworten die wesentlichen Funktionen, Aufgaben oder Prozesse des Systems. Beschreiben Sie die Kernfunktionen. Zeigen Sie insbesondere auf, welche Funktionen den Mehrwert des Systems (siehe *Value Proposition*) ausmachen.

Zusätzlich können Sie erwähnen, welche Funktionen/Aufgaben des Systems besondere Risiken enthalten, oder besonders schwierig zu entwickeln/betreiben sind.

Quality Attributes

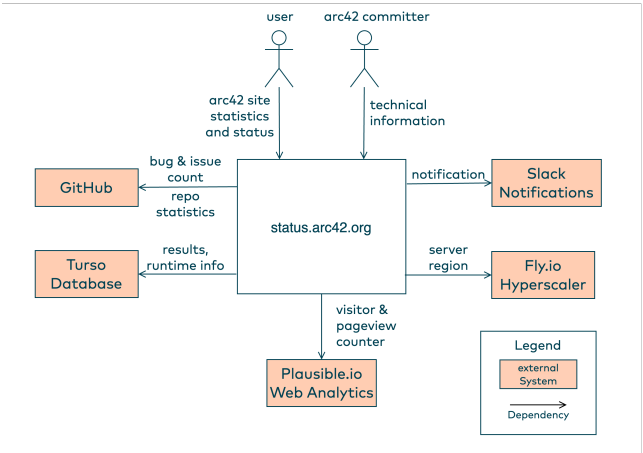
Neben den Kernfunktionen bilden die Qualitätseigenschaften (oder -anforderungen) ein Kernthema der Architektur. Zeigen Sie die 3-5 wichtigsten dieser Qualitätseigenschaften, am besten inklusive der geforderten Ausprägung oder Metriken.

Sie können dazu verkürzte Qualitätsszenarien verwenden – viele praktische Beispiele finden Sie unter https://quality.arc42.org

Business Context

Die Kontextabgrenzung, also die Beschreibung der wesentlichen externen Schnittstellen oder Nachbarsysteme. Bei größeren Systemen beschränken Sie sich auf die wesentlichen 3-5 externen Nachbarn. Häufig stehen diese in direktem Zusammenhang zu den Kernfunktionen (siehe *Core Functions*) des Systems.

Bis hierhin waren alle Informationen im ACC textuell – beim *Business Context* können Sie gerne auch ein Diagramm zur Beschreibung verwenden.

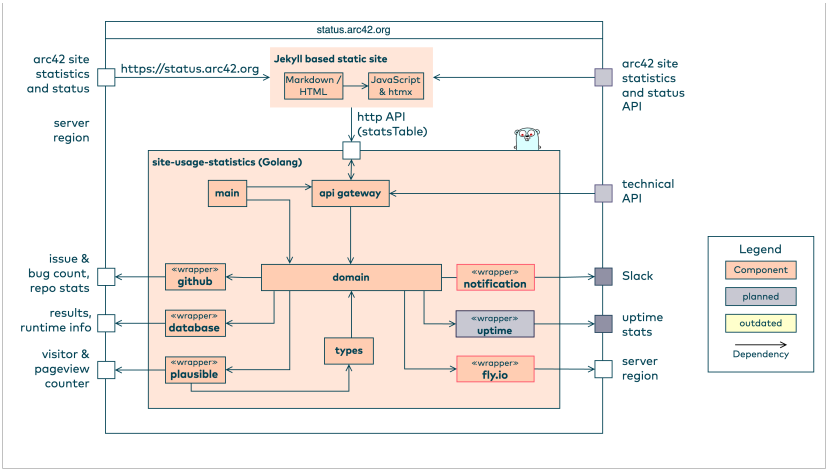


Components / Modules

Der *Grundriss* des Systems, dessen Aufbau aus Komponenten, Modulen, Services oder wie auch immer Sie die einzelnen Teile Ihres Systems so nennen. Dazu kommen deren wesentlichen gegenseitigen Abhängigkeiten, also die internen Schnittstellen des Systems.

In arc42[10] entspricht das der (möglicherweise etwas verkürzten) Bausteinsicht Ebene 1. Verkürzt deswegen, weil Sie sich auch hier auf die wesentlichen Teile beschränken sollten. Nur einen Teil der vorhandenen Subsysteme zu erwähnen geht völlig in Ordnung, sofern Sie dann ein „Sonstiges“ dazufügen.

[10] https://arc42.org



Wie bereits beim Kontext mit den externen Schnittstellen können Sie auch hier sowohl eine Aufzählung (*bullet list*) verwenden als auch ein Diagramm. In den **Empfehlungen zum ACC** auf Seite 28 finden Sie ein paar Tipps, wie Sie Diagramme für den Canvas etwas vereinfachen können.

Core Decisions – Good or Bad

Auf welche Entscheidungen blicken Sie mit Stolz zurück, und welche finden Sie aus heutiger Sicht fürchterlich? In großen IT-Systemen finden sich meist beide Aspekte – dieser Teil des Canvas soll genau das sichtbar machen. Beschränken Sie sich auf maximal 8-10, und bleiben Sie bei deren Formulierung auf Stichwort-Niveau. Falls Sie im Entwicklungsprozess ADRs[11] verwenden, dürfen Sie natürlich gerne auf die jeweils zutreffenden ADRs verweisen.

Technologies

Zeigen Sie in Stichworten das technische Fundament Ihres Systems: Welche Technologien, Frameworks, Bibliotheken oder Produkte bilden dessen Grundlage oder besitzen besondere Bedeutung? Neben den Programmiersprachen zählen hierzu beispielsweise Frameworks für grafische Oberflächen (QT, GTK, Electron, React,

[11] https://www.cognitect.com/blog/2011/11/15/documenting-architecture-decisions

JavaFX o. Ä.), Kommunikation/Integration (REST, Apache-Kafka, gRPC, TCP- or Websockets o. Ä.) oder Persistenz (PostgreSQL, SQLite, MariaDB, CouchDB o. Ä.). Solche Technologien haben oftmals prägenden Einfluss auf die Entwicklung und den Betrieb von Systemen. Eine detaillierte Darstellung bietet der Tech-Stack Canvas auf Seite 33.

Risks and Missing Information

Schon bei den Entscheidungen im Canvas (siehe oben, *Decisions – Good or Bad*) konnten Sie problematische Entscheidungen als solche kennzeichnen. Hier gehen Sie noch einen Schritt weiter und führen wesentliche Risiken auf, die beim System drohen. Die können sich auf alle beteiligten Stakeholder oder Prozesse beziehen, angefangen von den Anforderungen, über Entwurfs- und Architekturthemen, bis hin zu Test, Deployment, Betrieb, Support oder auch Management des Systems. Denken Sie bei der Betrachtung der Risiken in die Breite und führen Sie auch Risiken auf, die außerhalb von Code und Technologie liegen.

Anwendung des ACC

Wir verwenden den ACC zu ganz unterschiedlichen Zwecken:

- Als Einstieg in die *Nachdokumentation* bestehender Systeme. In diesem Fall verschafft sich das Team durch den ACC einen Architekturüberblick und entscheidet auf dieser Grundlage, welche Teile von arc42 noch weiter ausgeführt werden sollen.
- Als Einstieg in Reviews von Systemen: Hier bekommen die Reviewer einen Überblick relevanter Entscheidungen und können die Schwerpunkte des Reviews damit besser festlegen. Beachten Sie für diesen Fall unbedingt unseren Software-Analytics Canvas auf Seite 43.
- Als Steckbrief fürs Management und Management-Gremien (z.B. die berühmten Lenkungskreis-Meetings): Markieren Sie in diesem Fall die wesentlichen Änderungen seit dem letzten dieser Meetings. Nutzen Sie dafür die Powerpoint-Version vom Canvas, dann können Sie im Meeting im Präsentationsmodus ganz einfach vorher-nachher zeigen.

- Zur Vorstellung Ihres Systems. Im Canvas dokumentieren Sie Ihr System auf einem Level, das für viele Zielgruppen verständlich ist. Sie können in größeren Runden Ihr System anhand des Canvas vorstellen, in dem Sie die einzelnen Abschnitte der Reihe nach durchgehen.

Empfehlungen

ACC als Einstieg in umfangreichere Dokumentation

Betrachten Sie den ACC als die Zusammenfassung oder Kurzfassung der gesamten (Architektur-)Dokumentation, und verweisen Sie aus dem ACC auf die Verfeinerung oder Details der *restlichen* arc42-Dokumentation. Schematisch erkennen Sie das in Abbildung 2.5. Der Canvas gibt Interessierten einen ersten Überblick. Weitere Details und vertiefende Informationen liegen im *gesamten* arc42-Dokument.

> Wir haben gute Erfahrung damit gemacht, auf Basis einer bestehenden arc42-Dokumentation erhebliche Teile des ACC per LLM erzeugen zu lassen. Voraussetzung ist die inhaltliche Korrektheit und Aktualität der bestehenden Dokumentation.

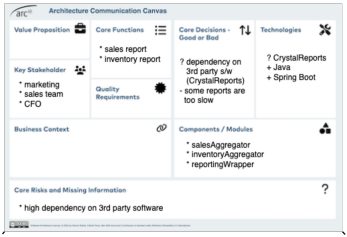
ACC zur Dokumentation von Microservices

Aus unserer Sicht ein *Gamechanger* im Umgang mit der Dokumentation von Microservices: Spendieren Sie jedem (relevanten) Service einen eigenen Canvas. Zusätzlich erstellen und pflegen Sie einen übergeordneten Canvas, der neben den Abhängigkeiten und Schnittstellen der Services zueinander die wesentlichen übergreifenden Entscheidungen und Technologien aufführt. Abbildung 2.6 skizziert dieses Vorgehen beispielhaft.

Nutzen Sie Diagramme für Kontext und Komponenten

Wir mögen Diagramme, weil Sie unserer Erfahrung nach mehr Übersicht geben als eine reine Aufzählung (*bullet list*) von Komponenten oder auch Nachbarsystemen. Daher bevorzugen wir im Normalfall Diagramme.

Architecture Communication Canvas



"gesamtes" arc42

1. Einführung und Ziele	7. Verteilungssicht
2. Randbedingungen	8. Querschnittliche Konzepte
3. Kontextabgrenzung	
4. Lösungsstrategie	
5. Bausteinsicht	9. Architekturentscheidungen
	10. Qualitätsszenarien
6. Laufzeitsicht	11. Risiken & technische Schulden
	12. Glossar

Abbildung 2.5: ACC als Einstieg in umfangreichere arc42 Dokumentation

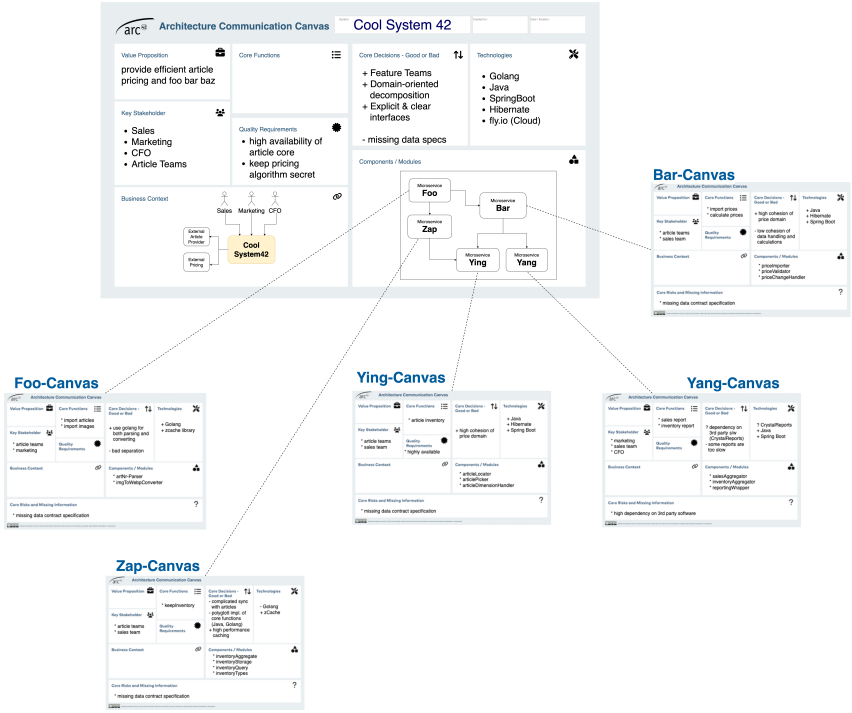


Abbildung 2.6: ACC für Microservices

Was aber, wenn ein solches Diagramm zu groß für den verfügbaren Platz auf dem Canvas ist?

In solchen Fällen schlagen wir den pragmatischen Ausweg eines „Sonstiges"-Symbols vor, was Sie in Abbildung 2.7 am Beispiel sehen können. Alle *wichtigen* Komponenten bekommen ihr eigenes Symbol, und die weniger wichtigen subsumieren Sie durch „Sonstiges".

Sie können darüber hinaus aber noch ein paar Vereinfachungen an diesen Diagrammen vornehmen:

- Eliminieren Sie detaillierte Informationen wie Stereotypen oder ausführliche Beschriftungen.

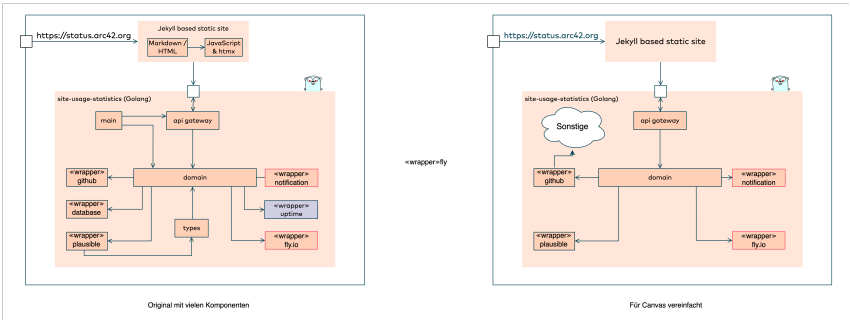


Abbildung 2.7: „Sonstiges“ als Platzhalter im Diagramm

- Beschränken Sie sich auf einige wesentliche Schnittstellen, statt alle möglichen Beziehungen zwischen Komponenten zu zeigen.
- Verwenden Sie Abstraktionen statt Details. In Abbildung 2.7 sehen Sie auf der linken Seite die `jekyll static site` als Whitebox mitsamt den enthaltenen Elementen. Auf der rechten Seite gibt es diese nur noch als Blackbox, d.h. ohne *Innenleben*.

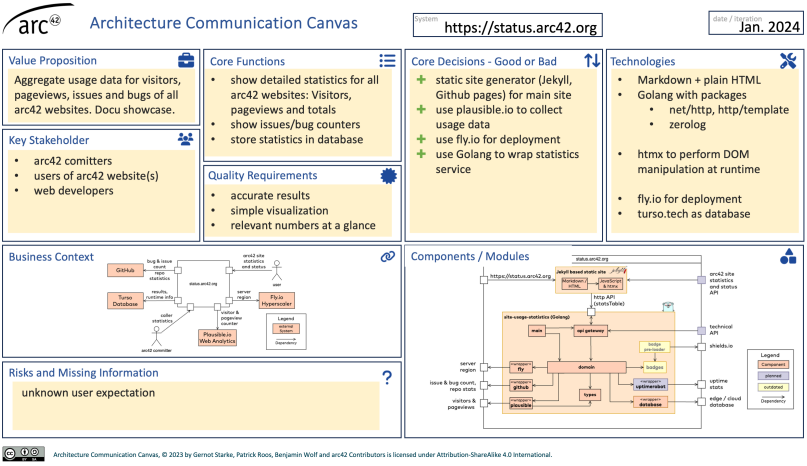
Beispiele

Wir haben für Sie ein Beispiel mitgebracht. Weitere Beispiele zum ACC finden Sie auf der zugehörigen Website[12].

status.arc42.org

Die kleine Anwendung bei arc42, die Besucherzähler sowie offene Bugs und Issues der arc42-Websites aggregiert.

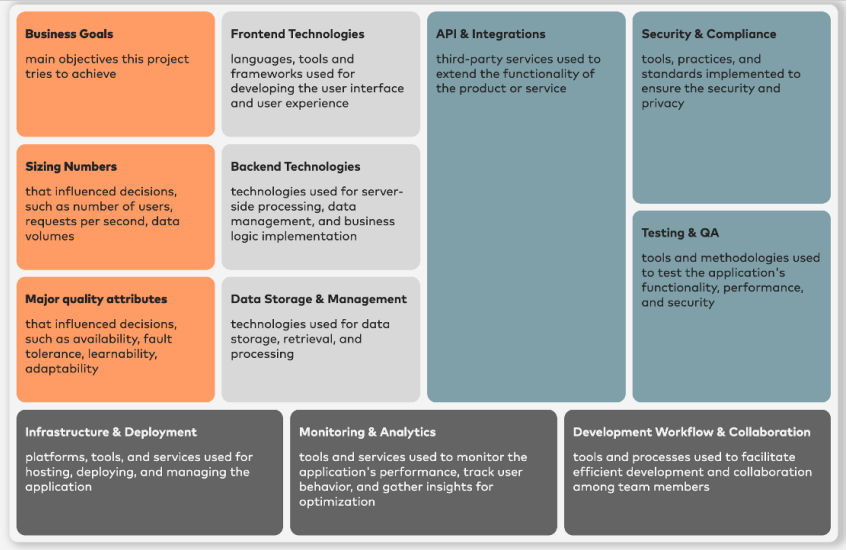
Obwohl die Aufgaben dieses Systems fast trivial anmuten, haben wir es immerhin mit ein paar sehr unterschiedlichen externen Schnittstellen zu tun. Weiterhin besteht die Herausforderung, den (Cloud-)Betrieb des Systems möglichst kostengünstig respektive kostenfrei zu gestalten, weil das Ganze ja als Open-Source ohne Monetarisierung daherkommt.



[12] https://canvas.arc42.org/examples

2.4 Tech Stack Canvas

Von Jörg Müller.



Motivation

In jedem Softwareprojekt gibt es Situationen, in denen das Team entscheiden muss, welche technischen Komponenten zum Einsatz kommen. Dabei stehen nicht selbst entwickelte Komponenten im Fokus, sondern bestehende Technologien und Angebote, die die Grundlage für eigene Entwicklungen bilden.

Eine typische Situation ist der Start einer neuen Kollegin oder eines Kollegen und die Frage, wo eventuell Einarbeitung stattfinden muss. Oft werden diese Informationen auch bereits bei der Erstellung der Stellenanzeige benötigt. Häufig improvisieren Verantwortliche solche Details und riskieren dadurch, wichtige Komponenten zu vergessen oder veraltete Informationen zu nutzen. Manchmal enthalten Dokumentationen oder das Wiki diese Daten, jedoch verstreut und oft nicht aktuell.

Auch beim Start eines neuen Projekts ist entscheidend, auf welchen technologischen Komponenten das Team seine Arbeit aufbaut. Eine unstrukturierte Sammlung führt schnell dazu, dass das Team wichtige Aspekte übersieht. Obwohl umfangreiche Dokumententemplates hilfreich sein können, eignen sie sich schlecht für die gemeinsame Erarbeitung im Team.

Noch komplexer wird es, wenn mehrere Teams mit unterschiedlichen Tech Stacks arbeiten. Wie sollten Sie die eingesetzten Technologien erfassen und systematisch vergleichen? Ein einheitlicher Überblick liefert wertvolle Erkenntnisse und zeigt Optimierungspotenziale auf.

Der Tech Stack Canvas wurde entwickelt, um genau diese Herausforderungen in der Softwareentwicklung zu adressieren. Er bietet eine strukturierte Methode, um die wesentlichen Aspekte eines Tech Stacks auf einer einzigen Seite zu erfassen. Damit unterstützt er das Team dabei, den Tech Stack zu konzipieren, zu dokumentieren und klar an alle Beteiligten zu kommunizieren.

Zielgruppe

Der Tech Stack Canvas ist für alle Projektbeteiligten relevant.

Für Softwareentwickler:innen bietet er einen klaren Überblick, welches Wissen zu welchen Technologien notwendig ist. Das erleichtert die Einarbeitung und den Wissensaufbau. Auch die Suche nach spezifischen Lösungen oder Frameworks gestaltet sich durch diese Übersicht einfacher.

Softwarearchitekt:innen erhalten einen Überblick über die Komponenten und können leicht erkennen, wenn mehrere Technologien dasselbe Problem lösen sollen. Auch Technologien, die nicht gut zusammenpassen, fallen schnell auf. Zudem erleichtert der Canvas den Abgleich mit projektübergreifenden Vorgaben zu Technologien. Gleichzeitig verbessert er die Kommunikation des Tech Stacks sowohl innerhalb als auch außerhalb des Projekts.

Projektmanager:innen oder Produktowner:innen treffen oft finanzielle Entscheidungen zum Einsatz externer Technologien. Ein vollständiges Bild aller eingesetzten Technologien hilft dabei, den Nutzen einzelner Komponenten besser zu beurteilen.

Ein Überblick über die eingesetzten Technologien ist auch für das Management und die Personalentwicklung entscheidend. Dadurch lassen sich Maßnahmen effektiver planen.

Die Erstellung des Tech Stack Canvas ist in der Regel eine Teamaufgabe. Unterschiedliche Aspekte fließen in die Entscheidung ein, welche Technologien eingesetzt werden. Das Vorgehen zur Erstellung wird im Folgenden genauer beschrieben.

Inhalte

Der Tech Stack Canvas unterteilt sich in vier grobe Abschnitte, die Sie an den unterschiedlichen Farben der Elemente erkennen:

- **Produktkontext** (Business Goals, Sizing Numbers, Major Quality Attributes)
- **Kerntechnologien** (Frontend, Backend, Database Technologies)
- **Unterstützende Technologien** (API & Integrations, Security & Compliance, Testing & QA)
- **Infrastruktur** (Infrastructure & Deployment, Monitoring & Analytics, Development Workflow & Collaboration)

Produktkontext

Der Produktkontext ermöglicht den Nutzer:innen des Canvas, die Basis für die Entscheidungen in den anderen Abschnitten zu verstehen. Eine Entscheidung für eine bestimmte Technologie lässt sich nur sinnvoll bewerten, wenn klar ist, wofür sie eingesetzt wird. Daher beschreibt dieser Abschnitt die wesentlichen Rahmenbedingungen, damit der Canvas auch ohne weitere Dokumente verständlich bleibt. Es ist jedoch wichtig zu beachten, dass hier kein Anspruch auf Vollständigkeit besteht.

Zum Kontext gehört zunächst eine grobe Beschreibung des Produktziels, im Canvas als **Business Goal** bezeichnet. Welche fachlichen Ziele soll der aufgeführte Tech Stack unterstützen? In diesem Bereich fassen Sie in ein paar Stichpunkten zusammen, warum dieses Produkt existiert und was es erreichen soll.

Anschließend folgt der Bereich **Sizing Numbers**. Wird das System eher von 10 oder von 10.000 Nutzer:innen verwendet? Welche Datenmengen müssen verarbeitet werden? Zu welchen Zeiten muss das System verfügbar sein? Diese und ähnliche Fragen haben großen Einfluss auf die Wahl der Technologien. Legen Sie hier den Fokus auf die Zahlen, die die Entscheidungen wesentlich beeinflusst haben.

Im Abschnitt **Major Quality Attributes** beschreiben Sie die Qualitätsziele, die die Technologieentscheidungen geprägt haben. Verwenden Sie hierfür Stichpunkte. Hilfreich ist ein Blick auf die Kriterien der ISO 25010 oder das arc42-Qualitätsmodell[13]. Ein Produkt mit Fokus auf Benutzerfreundlichkeit erfordert möglicherweise andere Technologien als ein System, bei dem die Performance an erster Stelle steht.

Kerntechnologien

Der Bereich **Kerntechnologien** umfasst die zentralen Elemente, die Sie vermutlich zuerst nennen würden, wenn jemand nach dem Tech Stack Ihres Produkts fragt. Welche Programmiersprachen und wesentlichen Frameworks nutzen Sie für die Entwicklung? Wie speichern Sie die Daten? Der Canvas teilt diese Technologien in drei Bereiche, die auch in einer klassischen 3-Schichten-Architektur üblich sind:

- **Frontend-Technologien**
- **Backend-Technologien**
- **Datenbank**

Diese Unterteilung ist keine Architekturvorgabe, sondern eine Struktur, die für die meisten Produkte passt. Bei einer klassischen Webanwendung bildet das Frontend den Teil, der im Browser läuft, während das Backend die Serverseite darstellt. In einer mobilen Anwendung bezeichnet das Frontend den Teil der App, der auf dem Mobilgerät ausgeführt wird, und das Backend die zentralisierte Infrastruktur. Die meisten Anwendungen enthalten zudem eine Form der Datenhaltung, die in der Regel über Standardsysteme erfolgt. Diese Systeme sollten Sie in diesem Bereich festhalten.

[13] https://quality.arc42.org/

Auch in diesem Abschnitt liegt der Fokus auf den wesentlichen Technologien. So ist es beispielsweise wichtig, zu erwähnen, wenn Sie im Frontend **TypeScript** mit **React** verwenden. Weniger relevante Details, wie etwa die verwendete Bibliothek zum Rendern von Buttons, können Sie weglassen.

Unterstützende Technologien

Der Bereich **unterstützende Technologien** konzentriert sich auf Technologien, die neben den klassischen Kernkomponenten eingesetzt werden. Er umfasst verschiedene Elemente, die wichtige Bestandteile des Tech Stacks darstellen und nicht vergessen werden sollten. Dieser Abschnitt enthält drei Elemente:

- **APIs & Integrations**
- **Security & Compliance**
- **Testing & QA**

Keine moderne Anwendung kommt ohne die **Integration** mit anderen Anwendungen oder Services aus. Dies gilt sowohl für unternehmensinterne Systeme als auch für externe Dienste. Typische Beispiele externer Dienste sind Services zum Versenden von Kurznachrichten an Mobiltelefone oder Plattformen, die externe Datenquellen bereitstellen. Auch interne Integrationen, wie etwa die Anbindung an ein Buchhaltungssystem, gehören in diesen Bereich. Dieses Element ist bewusst größer angelegt, da hier häufig eine längere Liste zu erwarten ist.

Das nächste Element **Security & Compliance** listet sowohl relevante Technologien als auch Vorschriften auf. Im Bereich Technologien stehen hier häufig die Authentifizierung und Autorisierung im Fokus. Gleichzeitig werden auch Vorschriften, wie beispielsweise die ISO 27001 oder branchenspezifische Richtlinien, aufgeführt. Dies gibt den Nutzer:innen einen klaren Überblick über die regulatorischen Anforderungen, die für das Produkt wichtig sind.

Der Abschnitt **Testing & QA** beschreibt alle Methoden und Technologien, die zur Sicherstellung der Qualität des Produkts verwendet werden. Dazu gehören wesentliche Elemente der Testpyramide, wie Frameworks für Unit-Tests und Integrationstests. Werden spezifische Testmethoden eingesetzt, können diese ebenfalls in diesem Bereich dokumentiert werden.

Infrastruktur

An der Basis des Canvas befindet sich der Bereich für die Infrastruktur. Dieser gliedert sich in drei verschiedene Elemente:

- **Infrastructure & Deployment**
- **Monitoring & Analytics**
- **Development Workflow & Collaboration**

Das erste Element, **Infrastructure & Deployment**, wird selten vergessen, wenn es um den Tech Stack eines Projekts geht. Hier beantworten Sie die Frage, wo das Produkt betrieben wird. Läuft es in der Cloud oder On-Premise? Welcher Cloud-Anbieter wird genutzt? Zudem sollten Sie beschreiben, welche wesentlichen Services des Anbieters eingesetzt werden. Dieser Abschnitt enthält auch Informationen darüber, wie das Produkt gebaut und ausgerollt wird.

Das zweite Element, **Monitoring & Analytics**, wird beim Tech Stack oft übersehen, ist jedoch ebenfalls essenziell. Es beschreibt, wie die Anwendung überwacht wird – sowohl technisch als auch fachlich. Beispielsweise ist es wichtig zu dokumentieren, ob ein Werkzeug wie Google Analytics verwendet wird.

Das dritte Element, **Development Workflow & Collaboration**, umfasst Werkzeuge, die indirekt zur Erstellung der Anwendung beitragen. Dazu gehören die Entwicklungsumgebung (IDE), das Versionskontrollsystem und das Ticketmanagementsystem. Diese Tools sind ein wesentlicher Bestandteil des Tech Stacks.

Sie kennen jetzt alle wichtigen Elemente des Tech Stack Canvas. Doch wie erstellen Sie ein solches Canvas effektiv?

Vorgehen

Das Vorgehen zur Erstellung eines Tech Stack Canvas unterscheidet sich je nach Situation, in der sich das Projekt befindet. Es gibt drei typische Anwendungsfälle:

- **Dokumentation eines bestehenden Tech Stacks**

- **Erarbeitung eines groben Tech Stacks im Rahmen eines Scoping-Workshops**
- **Entwurf eines vollständigen und detaillierten Tech Stacks zu Projektbeginn**

Wenn Ihr Produkt und der dazugehörige Tech Stack **bereits existieren** und Sie diesen nur mithilfe des Canvas dokumentieren möchten, empfiehlt sich folgendes Vorgehen: Organisieren Sie einen Workshop mit den wichtigsten Know-how-Träger:innen aus dem Team. Für jeden Bereich des Canvas sollte eine Expertin oder ein Experte anwesend sein. Das bedeutet, dass Sie in der Regel mindestens Personen mit folgenden Rollen einbeziehen sollten: jemand mit Produktwissen, eine Softwareentwicklerin oder ein Softwareentwickler, eine Testverantwortliche oder ein Testverantwortlicher sowie eine Person, die für den Betrieb des Produkts zuständig ist. Die genaue Anzahl der Teilnehmenden hängt von Ihrem Produkt und Ihrem Team ab.

Diese Kolleg:innen verwenden eine gedruckte oder elektronische Version des Canvas (siehe unten für Templates) und füllen diese nach bestem Wissen aus. Eine Erarbeitung in Gruppen ist dabei sinnvoll und effizient. Erfahrungsgemäß dauert ein solcher Workshop nicht länger als eine Stunde. Halten Sie den Bereich des **Produktkontextes** knapp und beschränken Sie sich auf Informationen, die den Tech Stack tatsächlich beeinflussen.

Es hat sich bewährt, den ausgefüllten Canvas anschließend auszudrucken und in den Räumlichkeiten des Teams aufzuhängen. So werden eventuelle Informationslücken innerhalb der nächsten Wochen sicher entdeckt und können ergänzt werden.

Ein weiterer typischer Anwendungsfall ist der **Produkt Scoping Workshop**. Dieser Workshop findet oft in einer sehr frühen Phase der Produktkonzeption statt. Dabei stehen in der Regel die Beschreibung der notwendigen Funktionalitäten und die Erstellung einer groben Roadmap im Vordergrund. Es kann jedoch sehr hilfreich sein, bereits in diesem frühen Stadium über den Tech Stack zu sprechen. In dieser Phase geht es vor allem um grobe Beschreibungen.

Der erste Schritt besteht häufig darin, bestehende Einschränkungen oder bereits vorgegebene Technologien im Produktkontext zu identifizieren. Diese Informa-

tionen können direkt im Canvas festgehalten werden und bieten so einen ersten Überblick über mögliche Lösungsoptionen. Der Canvas dient hier auch als Checkliste, um sicherzustellen, dass relevante Themen besprochen werden. Typische Beispiele, die oft übersehen werden, sind **Security** oder **Testing**.

Findet der Workshop vor Ort statt, hat es sich bewährt, die Struktur des Canvas auf eine Moderationswand zu zeichnen. Beschreibungen lassen sich dann mithilfe von Post-it-Notes hinzufügen und während der Diskussion flexibel anpassen. Ein grob ausgearbeiteter Canvas bietet im Anschluss eine wertvolle Grundlage für die weitere Produktplanung. Für diesen Teil des Scoping Workshops sollten Sie 30 bis 60 Minuten einplanen.

Die dritte und umfangreichste Variante, einen Canvas zu erstellen, ist ein Workshop zu Beginn eines Projekts, mit dem Ziel, einen **detaillierten Tech Stack** zu entwerfen. Ein solcher Workshop dauert in der Regel einen ganzen Tag. Das detaillierte Vorgehen finden Sie auf der Webseite techstackcanvas.io[14]. Der Workshop besteht aus fünf Schritten:

1. **Einführung in den Canvas**
2. **Diskussion der Projektziele**
3. **Größenordnungen und Qualitätsattribute umreißen**
4. **Alle technischen Elemente des Canvas im Detail diskutieren**
5. **Prüfung des Gesamtbildes**

Die Teilnehmer:innen des Workshops entsprechen im Wesentlichen denjenigen, die auch bei der Dokumentation eines bestehenden Tech Stacks beteiligt sind, wie weiter oben beschrieben.

In den Schritten 1 bis 3 werden die Rahmenbedingungen ausführlicher besprochen. Voraussetzung dafür ist, dass diese bereits relativ gut bekannt sind. Es ist wichtig, dass alle Teilnehmer:innen einen klaren Überblick über die Anforderungen besitzen. Diese Anforderungen werden zunächst grob dokumentiert. Im weiteren Verlauf des Workshops können sie präzisiert werden, sobald deutlich wird, welche Anforderungen besonders relevant für die Entscheidungen zu Technologien sind.

[14] https://techstackcanvas.io

Im vierten Schritt wiederholen Sie das gleiche Muster für alle Elemente des Canvas. Beginnen Sie mit den externen Einschränkungen. Eine Technologieauswahl ist nur im richtigen Kontext sinnvoll. Es wäre Zeitverschwendung, über Optionen zu diskutieren, die aus unterschiedlichen Gründen in Ihrem Unternehmen nicht einsetzbar sind. Häufig wird der Lösungsraum dadurch stark eingegrenzt. Sobald die möglichen Optionen bekannt sind, können Sie im Detail deren Eignung für die Projektanforderungen sowie die Konsequenzen ihrer Nutzung besprechen. Das Ergebnis dieser Diskussion sollte abschließend dokumentiert werden. Ein Format, das sich an **Architecture Decision Records** (**ADR**) orientiert, ist hier empfehlenswert.

Nachdem alle Elemente des Canvas auf diese Weise ausgefüllt wurden, erfolgt im letzten Schritt eine Überprüfung, ob der Tech Stack als Ganzes stimmig ist. Passt die Frontend-Technologie zum Backend? Sind die automatisierten Tests mit dem geplanten Deployment-Mechanismus umsetzbar? Widersprüche auf dem Canvas lassen sich für erfahrene Architekt:innen oft leicht erkennen. Es kann sinnvoll sein, Dritte um Feedback zu bitten.

Die meisten Methoden, um einen Tech Stack Canvas zu erstellen, sind schnell und leicht zu erlernen. Probieren Sie es aus! Als Unterstützung stehen Ihnen verschiedene Vorlagen zur Verfügung, die im folgenden Abschnitt beschrieben werden.

Tools, Vorlagen, Quellen

Der erste Anlaufpunkt für den Tech Stack Canvas ist die Webseite techstackcanvas.io[15]. Dort erhalten Sie eine Übersicht über den Aufbau des Canvas sowie detaillierte Informationen zu den einzelnen Elementen. Durch Klicken auf die Elemente können Sie zusätzliche Erläuterungen und Beispiele abrufen, die zeigen, welche Informationen dort eingetragen werden können. Zudem bietet die Webseite eine ausführliche Anleitung für die Erstellung eines Tech Stack Canvas.

Darüber hinaus finden Sie auf der Webseite Vorlagen in verschiedenen Formaten, mit denen Sie direkt starten können. Aktuell stehen folgende Vorlagen zur Verfügung:

[15] https://techstackcanvas.io

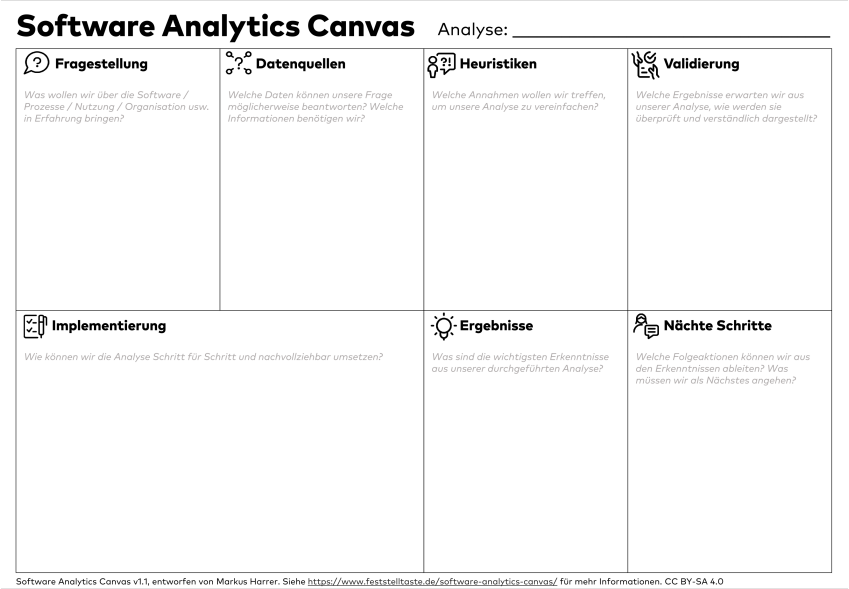
- **Miro Template**
- **PDF**
- **Apple Keynote**
- **PowerPoint**

Die Community trägt kontinuierlich weitere Vorlagen bei, die bei Verfügbarkeit ebenfalls auf der Webseite ergänzt werden. Derzeit befindet sich beispielsweise eine Vorlage für **AsciiDoc** in Vorbereitung. Es lohnt sich daher, regelmäßig auf der Webseite vorbeizuschauen.

2.5 Software Analytics Canvas

Von Markus Harrer.

Nachvollziehbare, datengetriebene Softwareanalysen



Software Analytics Canvas Analyse: _____

Fragestellung	Datenquellen	Heuristiken	Validierung
Was wollen wir über die Software / Prozesse / Nutzung / Organisation usw. in Erfahrung bringen?	*Welche Daten können unsere Frage möglicherweise beantworten? Welche Informationen benötigen wir?*	*Welche Annahmen wollen wir treffen, um unsere Analyse zu vereinfachen?*	*Welche Ergebnisse erwarten wir aus unserer Analyse, wie werden sie überprüft und verständlich dargestellt?*
Implementierung		**Ergebnisse**	**Nächte Schritte**
Wie können wir die Analyse Schritt für Schritt und nachvollziehbar umsetzen?		*Was sind die wichtigsten Erkenntnisse aus unserer durchgeführten Analyse?*	*Welche Folgeaktionen können wir aus den Erkenntnissen ableiten? Was müssen wir als Nächstes angehen?*

Software Analytics Canvas v1.1, entworfen von Markus Harrer. Siehe https://www.feststelltaste.de/software-analytics-canvas/ für mehr Informationen. CC BY-SA 4.0

Motivation

Tieferliegende Probleme in komplexen Softwaresystemen lassen sich aufgrund der Menge an verstricktem Code nicht mehr einfach per Hand durchführen. Klassische Codeanalysewerkzeuge liefern zudem in diesen Situationen oftmals nur Erkenntnisse, die den Entwicklungsteams ohnehin bekannt sind, oder liefern Antworten auf Fragen, die nicht für die gegebene Situation relevant sind.

Hier kommt die Softwaredatenanalyse, auch „Software Analytics“, ins Spiel, wo es darum geht, anhand von Daten aus der Softwareentwicklung (wie Code-Historie, Log-Files oder dem Code selbst) durch Analysen nachvollziehbar und fundiert

handlungsorientierte Erkenntnisse zu gewinnen. Diese Analysen sind jedoch anfangs schwierig umzusetzen und die Gefahr ist groß, sich bei aufwendigeren Analysen im Detail zu verlieren.

Der Software Analytics Canvas hilft, eine Softwaredatenanalyse gezielt und strukturiert durchzuführen. Der Canvas stellt dafür wichtige Eckpunkte einer Analyse explizit dar. Er dient auch dazu, Denkfehler und falsche Vorstellungen zu vermeiden und regt Nutzende dazu an, vorab auch noch einmal über den Sinn und Zweck sowie den Aufwänden einer geplanten Analyse nachzudenken.

Die Ergebnisse aus den Analysen dienen den Canvas-Nutzenden dazu, weniger zu raten, was die eigentlichen Probleme eines Softwaresystems sind, sondern Daten und Fakten zu liefern, die zeigen, woran es genau hakt, um darauf aufbauend gewinnbringende Verbesserungen vorzunehmen. Der Canvas eignet sich auch ideal als Einstieg in die Welt von Software Analytics, um sich bei eigenen Analysen am Anfang weniger zu verzetteln und direkt von Beginn an, robuste, reproduzierbare Analysen und Ergebnisse zu liefern.

Weniger raten, mehr analytisch vorgehen, denn „Without data, you're just another person with an opinion", wie Edward W. Deming bereits zu sagen pflegte!

Zielgruppe

Die Zielgruppe des Software Analytics Canvas sind alle wissbegierigen Menschen, welche mithilfe von klassischen Datenanalysewerkzeugen wie Python/Pandas oder auch Excel systemische Probleme in großen und mit der Zeit gewachsenen Softwaresystemen explizit sichtbar machen möchten.

Inhalte

Canvas-typisch besteht auch der Software Analytics Canvas aus mehreren Sektionen. Klassischerweise wird er Schritt für Schritt von der oberen Reihe von links nach rechts und nachfolgend in der unteren Reihe ebenfalls in dieser Reihenfolge durchgearbeitet. Aber er bietet auch unstrukturierten Ideen einen Platz und kann dadurch auch zuerst skizzenhaft in beliebiger Reihenfolge verwendet werden, um Analysen später auszuarbeiten.

Fragestellung

Was wollen wir über die Software / Prozesse / Nutzung / Organisation usw. in Erfahrung bringen?

Jede gute Analyse beginnt mit einer guten, interessanten oder relevanten Frage.

In diesem ersten Abschnitt kannst du deine konkrete Fragestellung formulieren:

- Was genau möchtest du untersuchen?
- Handelt es sich um ein echtes Problem oder nur um ein Symptom?
- Wer hat ein Interesse an der Fragestellung und wird den (vielleicht unangenehmen) Antworten Gehör schenken?
- Und vor allem: Was würde sich ändern, wenn du das Ergebnis kennst?

Analysen brauchen auch Zeit. Mit diesem Abschnitt kannst du direkt bewerten, ob es sich rentiert, eine datengetriebene Analyse überhaupt anzufangen.

Datenquellen

Welche Daten können unsere Frage möglicherweise beantworten? Welche Informationen benötigen wir?

Es gibt viele potenzielle Datenquellen, die für deine Analyse relevant sein könnten.

Frage dich:

- Welche Datenquellen stehen dir zur Verfügung? Welche Art von Informationen liefern sie?
- Fehlen dir vielleicht bestimmte Daten, die du für deine Analyse brauchst?
- Müsstest du gegebenenfalls manuell Daten in deinem Team erheben?

Die Auswahl der passenden Daten ist von Beginn an entscheidend, um später falsche Schlussfolgerungen zu vermeiden. Wenn du Daten aus einem System extrahieren musst, dokumentiere hier auch grob das Vorgehen, wie du diesen Export vorgenommen hast, damit andere deine Analysen nachvollziehen oder sogar selbst reproduzieren können.

Heuristiken

Welche Annahmen wollen wir treffen, um unsere Analyse zu vereinfachen?

Nicht alle Daten aus vorhandenen Systemen eignen sich für eine Softwaredatenanalyse. Oft kannst du auch nicht davon ausgehen, dass die verfügbaren Daten deine eigentliche Frage exakt beantworten. Wir brauchen also Annäherungen auf Basis der vorhandenen Daten. Lege offen, welche Informationen und Aussagemöglichkeiten du in den Daten siehst, und dokumentiere die vereinfachenden Annahmen, die du zu den Daten triffst. Dadurch kannst du deine Analyse schneller durchführen und andere auf die Schwachpunkte deiner Analyse hinweisen.

Validierung

Welche Ergebnisse erwarten wir aus unserer Analyse, wie werden sie überprüft und verständlich dargestellt?

Ein zentraler Erfolgsfaktor bei datengetriebenen Softwareanalysen ist, die Resultate so aufzubereiten und zu präsentieren, dass sie für Entscheider*innen (mit Budget) verständlich sind. Wie lässt sich das sicherstellen? Zudem gilt es zu klären, woran du erkennst, dass du deine Frage wirklich beantwortet hast. Was brauchst du, um dem analysierten Problem wirklich auf den Grund gegangen zu sein?

Diese Sektion hilft dir bei dem Sanity Check und dabei, dass du später nicht in die falsche Richtung abbiegst.

Implementierung

Wie können wir die Analyse Schritt für Schritt nachvollziehbar umsetzen?

Jetzt geht es ans Eingemachte. Bevor du jedoch loslegst, lohnt es sich, ein bis zwei Minuten über einen möglichen, geschmeidigen Implementierungsweg nachzudenken.

Wenn der Programmcode für die Analyse zu kompliziert wird, versteht ihn niemand mehr – und das kann zu Zweifeln an deinen Ergebnissen führen. Skizziere

daher grob die Implementierungsidee und die Tools, die du verwenden möchtest, damit du dich später nicht im Code der Analyse verirrst.

Ergebnisse

Was sind die wichtigsten Erkenntnisse aus unserer durchgeführten Analyse?

Hier kannst du nach der Umsetzung die zentralen Punkte auflisten, die du mit Hilfe deiner Analyse herausgefunden hast. Vor allem: Welche Erkenntnisse hast du gewonnen, die direkt zu Handlungen führen können? Wenn dies nicht offensichtlich ist, gerne auch, was nicht wie geplant lief oder was die Resultate zu unsicher macht.

Falls überhaupt keine relevanten Ergebnisse zustande kamen, dokumentiere das dennoch. Das ist auch kein Problem! Du hast wertvolle Erfahrung in der Datenanalyse im Softwarebereich gesammelt. Beim nächsten Mal kannst du es anders angehen!

Nächste Schritte

Welche Folgeaktionen können wir aus den Erkenntnissen ableiten? Was müssen wir als Nächstes angehen?

Eine Softwaredatenanalyse-Aufgabe ist nicht abgeschlossen, wenn sich daraus keine konkreten nächsten Schritte ergeben, die die aktuelle Situation verbessern.

Also:

- Welche Entscheidungen lassen sich nun auf Basis der Analyseergebnisse treffen oder vielleicht auch widerlegen?
- Was sind die Stellen, wo die ein oder andere Hand angelegt werden soll?

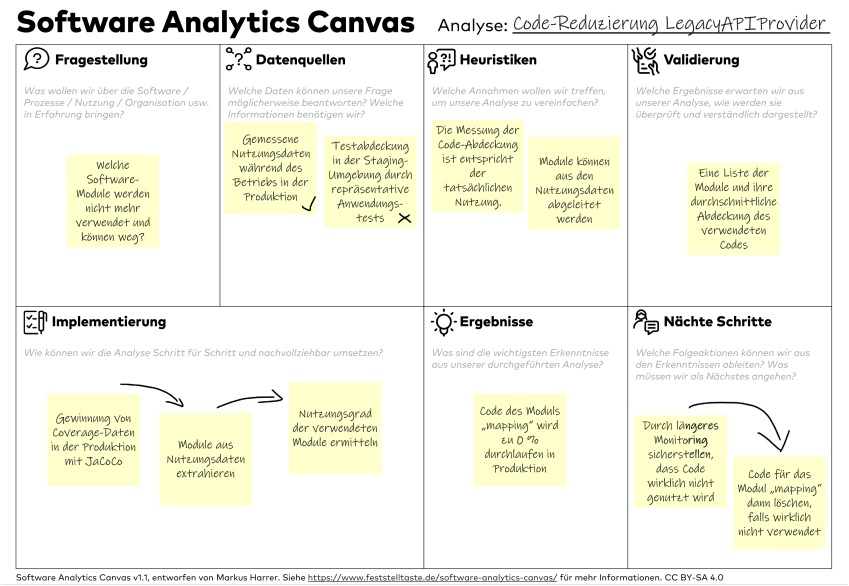
Oder auch:

- Welche weiteren Experimente können wir auf Grundlage der nun verfügbaren Fakten durchführen?

Notiere dir dazu eine To-do-Liste oder Ideen für weitere Analysen, um neu gewonnene Einsichten vertiefend zu untersuchen.

Beispiel

Hier findest du ein Beispiel für einen Software Analytics Canvas. Für das schon etwas ältere Softwaresystem wurde vermutet, dass große Teile des Codes bis zu ganzen Modulen des Systems nicht mehr benötigt werden würden. Eine maßgeschneiderte Analyse, welche über den Software Analytics Canvas festgehalten wurde, lieferte Tatsachen und brachte das Team ein Stück weiter in ihrer Arbeit.



Software Analytics Canvas v1.1, entworfen von Markus Harrer. Siehe https://www.feststelltaste.de/software-analytics-canvas/ für mehr Informationen. CC BY-SA 4.0

Bonus

Wenn du diesen Software Analytics Canvas für deine datengetriebenen Analysen von Softwaresystemen ausgefüllt hast, erhältst du automatisch eine Art Dokumentation deiner Analyse. Diese kannst du mit anderen besprechen und später wieder

zur Hand nehmen, um bereits durchgeführte Untersuchungen noch einmal nachzuvollziehen. So holst du andere Mitwirkende deines Softwareentwicklungsprojekts schnell ab und musst selbst nicht immer das Rad neu erfinden bei deinen weiteren Analysen.

Viel Spaß bei der Erstellung deines ersten Software Analytics Canvas!

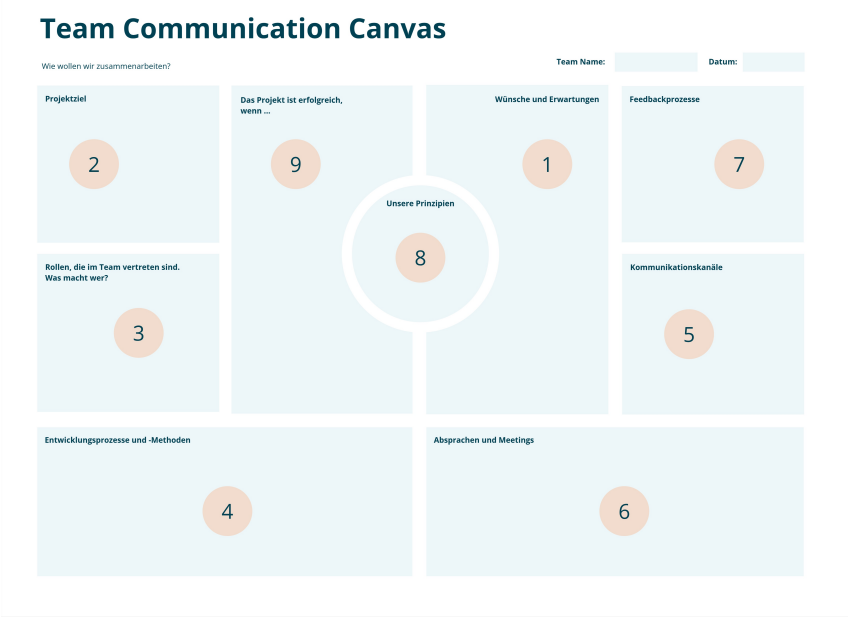
Weitere Informationen

- Initiale Vorstellung des Canvas: https://www.feststelltaste.de/software-analytics-canvas/
- Microsite zum Thema „Software Analytics“: https://softwareanalytics.de/
- Sammlung an interessanten Ressourcen rund um das Thema „Software Analytics“: https://github.com/feststelltaste/awesome-software-analytics

2.6 Team Communication Canvas

Von Anja Kammer und Lena Kraaz.

Struktur für erfolgreiche Zusammenarbeit.



2.6.1 Motivation

Der Team Communication Canvas dient primär als Dokumentation eines gemeinsamen Workshop-Prozesses. Der eigentliche Mehrwert entsteht jedoch während des Workshops selbst: Durch den Austausch und die Reflexion der Teammitglieder werden Missverständnisse und Konfliktpotenziale frühzeitig sichtbar.

Um dies zu erreichen, lädt der Canvas zum gemeinsamen Gespräch ein. Er unterstützt dabei, Missverständnisse und Konflikte in der Zusammenarbeit zu vermeiden, indem er Teamarbeit und Kommunikation bewusst in den Fokus rückt. Ein häufiges Problem sind unklare oder unausgesprochene Erwartungen, die zu Spannungen führen können – etwa bei der Aufgabenpriorisierung, im Umgang

mit Feedback oder bei unterschiedlichen Auffassungen von Höflichkeit. Oft wird fälschlicherweise angenommen, dass alle im Team dasselbe Verständnis von guter Zusammenarbeit haben. Der Canvas schafft hier Klarheit, indem er gezielt Fragen zu Erwartungen, Arbeitsweisen und Kommunikationsstilen stellt und so hilft, Konflikte frühzeitig zu erkennen und zu vermeiden.

Ein Beispiel: Wenn eine Person sich wünscht, dass am Vormittag alle Teammitglieder jederzeit erreichbar sind, während eine andere flexibel die eigene Zeit plant, sind Spannungen programmiert – es sei denn, man hat schon zu Beginn offen darüber gesprochen. Teams, die über ihre Erwartungen sprechen, können Missverständnisse früh identifizieren und gezielt gegensteuern. So entsteht ein konstruktiver Raum, in dem man Konflikte gar nicht erst eskalieren lässt, sondern sie als „abweichende Erwartungen" gemeinsam reflektiert.

Der Canvas macht Erwartungen und Arbeitsweisen transparent und bringt sie in eine Struktur. Teams erkennen so unklare Prozesse und können ihre Zusammenarbeit bewusst reflektieren. Das ist besonders wertvoll in dynamischen Projektteams, die sich häufig neu zusammensetzen, sodass sich die Konstellation der Beteiligten ändert. Gemeinsame Reflexion reduziert Missverständnisse, stärkt den Fokus auf gemeinsame Ziele und minimiert Konfliktpotenziale. Das Ergebnis: transparenter, respektvoller Austausch im Team, der langfristig die Produktivität und Zufriedenheit steigert.

2.6.2 Zielgruppe

Der Team Communication Canvas lädt das ganze Produkt- oder Projektteam dazu ein, kollaborativ zu erarbeiten, wie man die gemeinsame Zusammenarbeit gestalten möchte.

Der Canvas eignet sich für Teams, die gemeinsam in die Teamarbeit starten oder in denen sich die Zusammensetzung verändert – etwa, wenn neue Kolleg:innen hinzukommen. Entscheidend ist, dass alle festen Teammitglieder, die täglich operativ (egal ob in Präsenz, hybrid oder komplett remote) zusammenarbeiten, an dem Workshop teilnehmen.

Das sind in der Regel:

- UX Designer:innen
- Softwareentwickler:innen
- Softwarearchitekt:innen
- Tester:innen
- Product Owner:innen
- Projektmanager:innen
- Scrum Master/Agile Coach

2.6.3 Inhalte

Der Canvas umfasst 9 Abschnitte:

- Wünsche und Erwartungen: was sind die Wünsche und Erwartungen, die ich an meine Kolleg:innen habe?
- Projektziel: was ist das Ziel des Projekts (oder Abteilung oder des Produktes)?
- Rollen: welche Rollen und Tätigkeiten gibt es in unserem Team?
- Entwicklungsprozesse und Methoden: was möchten wir anwenden?
- Kommunikationskanäle: welche nutzen wir?
- Absprachen und Meetings: wann finden Meetings statt? Gibt es meetingfreie Zeiten?
- Feedbackprozesse: wie wollen wir uns untereinander Feedback geben?
- Unsere Prinzipien: welche Leitplanken/Entscheidungshilfen haben wir in unserem Team?
- Das Projekt ist erfolgreich, wenn ...: welche Kriterien gibt es, damit es für das Team als erfolgreiches Projekt empfunden wird?

Wünsche und Erwartungen

Im Abschnitt „Wünsche und Erwartungen“ des Team Communication Canvas werden persönliche Präferenzen, Arbeitsstile und Erwartungen der Teammitglieder ausgetauscht. Dabei steht nicht im Vordergrund, den Canvas auszufüllen. Es geht darum, sich gegenseitig im Arbeitskontext besser kennenzulernen und zu verstehen, ohne dass eine Einigung auf gemeinsame Regeln zwingend notwendig ist. Wenn sich das Team auf Inhalte einigt (Bsp: „Wir wollen Konflikte offen ansprechen.“) sollte diese aber auf dem Canvas festgehalten werden. Die

Inhalte umfassen unter anderem bevorzugte Arbeitszeiten, Kommunikationsstile, Arbeitspräferenzen, Themenbereiche, in denen Unterstützung angeboten wird, und individuelle Werte wie Pünktlichkeit oder Etikette. Ziel dieses X ist es, gegenseitiges Verständnis zu fördern und Missverständnisse im Alltag zu vermeiden.

Projektziel

Im Abschnitt „Projektziel“ des Team Communication Canvas werden Ziele und Absichten dokumentiert, die das Team gemeinsam verfolgt. Der Begriff Projektziel ist hierbei austauschbar, da nicht immer ein „Projekt“ im eigentlichen Sinne des Wortes der Kontext ist. Auch in Abteilungen, langfristigen Zusammenarbeitskontexten oder bei der Einführung neuer Prozesse ist es sinnvoll, über Ziele zu sprechen.

Hier können individuelle Perspektiven zu den übergeordneten Absichten festgehalten werden. Diese können konkret (z.B. Kosten einsparen oder Ablösung eines Altsystems) oder auch vage formuliert sein. Wenn kein gemeinsames Ziel erkennbar ist, ist das in diesem Schritt ebenfalls akzeptabel. In solchen Fällen kann es sinnvoll sein, die Klärung der Zielsetzung auszulagern, um den Rahmen des Meetings nicht zu sprengen.

Rollen

Im Abschnitt „Rollen“ des Team Communication Canvas werden die verschiedenen Tätigkeiten und Verantwortlichkeiten im Team dokumentiert. Dabei geht es nicht um formale Jobtitel, sondern um die tatsächlich ausgeübten Aufgaben. Rollen wie Infrastruktur-Expert:in, UX Designer:in oder Architekt:in beschreiben, welche Aufgaben und Schwerpunkte die einzelnen Teammitglieder übernehmen.

Ziel ist es, Klarheit darüber zu schaffen, wer welche Tätigkeiten übernimmt und wo es möglicherweise Überschneidungen oder Lücken gibt. Das heißt, es kann sichtbar gemacht werden, wo es gemeinsame Zuständigkeiten gibt (Beispiel: Tickets schreiben, Architekturarbeit).

Entwicklungsprozesse und Methoden

Im Abschnitt „Entwicklungsprozesse und Methoden“ des Team Communication Canvas werden Arbeitsweisen, technische Praktiken und Teamregeln im Entwicklungsprozess dokumentiert. Es geht darum, individuelle Ansichten und Präferenzen zu Themen wie Dokumentation, Code Reviews, Fehlerkultur oder Pair-Programming zu besprechen.

Der Abschnitt ermöglicht es, unterschiedliche Erwartungen im Team offenzulegen – etwa, wie viel dokumentiert werden soll, ob Pair-Programming gewünscht ist oder wie Code Reviews zeitlich organisiert werden. Dabei werden sowohl persönliche Vorlieben als auch bestehende oder gewünschte Teampraktiken besprochen.

Kommunikationskanäle

Im Abschnitt „Kommunikationskanäle“ des Team Communication Canvas werden bevorzugte Wege und Tools zur Kommunikation für das Team dokumentiert. Dabei geht es um die Klärung, welche Kanäle für welche Art von Kommunikation genutzt werden sollen – beispielsweise Chat für schnelle Abstimmungen oder E-Mails für formellere Anfragen.

Ziel des Abschnitts in diesem Workshop ist, Transparenz über individuelle Vorlieben und bestehende Praktiken zu schaffen, etwa, ob jemand bevorzugt über Telefon kommuniziert oder wie Bugmeldungen dokumentiert werden sollten. Gleichzeitig werden mögliche Unklarheiten wie der Umgang mit mehreren Kommunikationskanälen angesprochen, um Überkommunikation oder das Verpassen relevanter Informationen zu vermeiden.

Absprachen und Meetings

Hier geht es darum, Transparenz über bevorzugte Meeting-Zeiten, -Formate und -Frequenzen zu schaffen, beispielsweise der Wunsch nach meetingfreien Tagen oder fokussierter Arbeitszeit ohne Unterbrechungen.

Hier können auch individuelle Vorlieben sichtbar werden, etwa ob jemand schriftliche Dailys bevorzugt oder Meetings erst ab einer bestimmten Uhrzeit sinnvoll

erscheinen. Ziel ist es, Missverständnisse über die Verfügbarkeit und den Umgang mit Besprechungen zu vermeiden, ohne dass zwingend alle Präferenzen sofort vereinheitlicht werden müssen.

Die Diskussion im Team hilft, gemeinsame Richtlinien zu entwickeln, die sowohl den Bedarf an Abstimmung als auch den Wunsch nach ungestörter Arbeitszeit berücksichtigen.

Feedbackprozesse

Im Abschnitt „Feedbackprozesse“ des Team Communication Canvas werden Erwartungen und Bedürfnisse rund um den Umgang mit Feedback im Team besprochen, wie z.B. der Wunsch nach regelmäßigem, direktem Feedback oder der Vorschlag eines festen Feedback-Tages.

Das Ziel ist es, eine gemeinsame Feedbackkultur zu fördern, indem besprochen wird, in welcher Form Feedback gegeben werden soll – ob mündlich, schriftlich oder in anderen Formaten. Auch grundlegende Prinzipien, wie der respektvolle Umgang mit kritischem Feedback, werden hier thematisiert.

Der Abschnitt schafft Klarheit darüber, was die Teammitglieder benötigen, um Feedback als konstruktiv und hilfreich wahrzunehmen, und kann so helfen, Missverständnisse zu vermeiden.

Unsere Prinzipien

Der Abschnitt „Unsere Prinzipien“ bündelt die Leitlinien, die für das Team besonders wichtig sind. Prinzipien sind kurze, prägnante Sätze, die wie eine Art Kompass wirken, wenn schnelle Entscheidungen getroffen werden müssen. Beispiele könnten sein „Kommuniziere lieber einmal zu viel als einmal zu wenig“, „Lieber konsistent statt perfekt“ oder „Qualität statt Quantität“.

Eine Person, die unsicher ist, ob sie ein Problem sofort im Teamchat ansprechen soll, wird unweigerlich an das gemeinsame Prinzip „Kommuniziere lieber einmal zu viel als einmal zu wenig“ denken und die Information zügig teilen.

Die letzten beiden Prinzipien – „Lieber konsistent statt perfekt“ und „Qualität statt Quantität“ – können dabei helfen, beim Programmieren, etwa bei Refactorings oder beim Schreiben von Tests, ein einheitliches Vorgehen zu finden, ohne für jeden Einzelfall eine Teamdiskussion führen zu müssen.

Ziel ist es, einen klaren Orientierungsrahmen zu schaffen, an dem sich alle im Team bei Unsicherheiten ausrichten können. Wichtig ist, dass es sich nicht um starre Regeln handelt, sondern um Grundsätze, die das tägliche Handeln leiten sollen.

Das Projekt ist erfolgreich, wenn ...

Im Abschnitt „Das Projekt ist erfolgreich, wenn ...“ des Team Communication Canvas werden Kriterien dokumentiert, die den Erfolg aus Sicht des Teams definieren. Dabei geht es nicht nur um Projektziele, sondern um eine umfassendere Perspektive auf den gemeinsamen Erfolg. Die Frage kann auch allgemein verstanden werden als „Wann sind wir als Team erfolgreich?“

Neben messbaren Faktoren wie Kundenzufriedenheit, stabile Software oder Budgetausnutzung können hier auch individuelle oder teambezogene Aspekte thematisiert werden – etwa Lernfortschritte, Spaß an der Arbeit oder eine gute Zusammenarbeit auf Augenhöhe.

Dieser Abschnitt bietet Platz für verschiedene Sichten auf „Erfolg“ und schafft Klarheit darüber, worauf das Team hinarbeitet. So können unterschiedliche Erwartungen besprochen und ein gemeinsames Verständnis für Erfolg entwickelt werden.

2.6.4 Vorgehen

Das gemeinsame Bearbeiten des Canvas wird in einem Workshop-Format durchgeführt. Dabei ist es unerheblich, ob dieser Workshop vor-Ort oder in einem Remote-Setup durchgeführt wird. Wichtig ist lediglich genug Fläche um virtuelle oder reale Zettel kleben zu können und genug Freiraum für eine stille Einzelarbeit.

Vorkenntnisse

Es benötigt keine besonderen Vorkenntnisse. Wichtiger als Vorwissen ist die Bereitschaft, sich offen über eigene Bedürfnisse, Arbeitsweisen und Erwartungen auszutauschen. Daher hilft es, wenn das Team bereits ein grundlegendes Verständnis darüber hat, warum es überhaupt über Kommunikation und Zusammenarbeit reden sollte. Es sollte also vorher kurz erklärt werden, welchen Nutzen ein offener Dialog über Erwartungen haben kann.

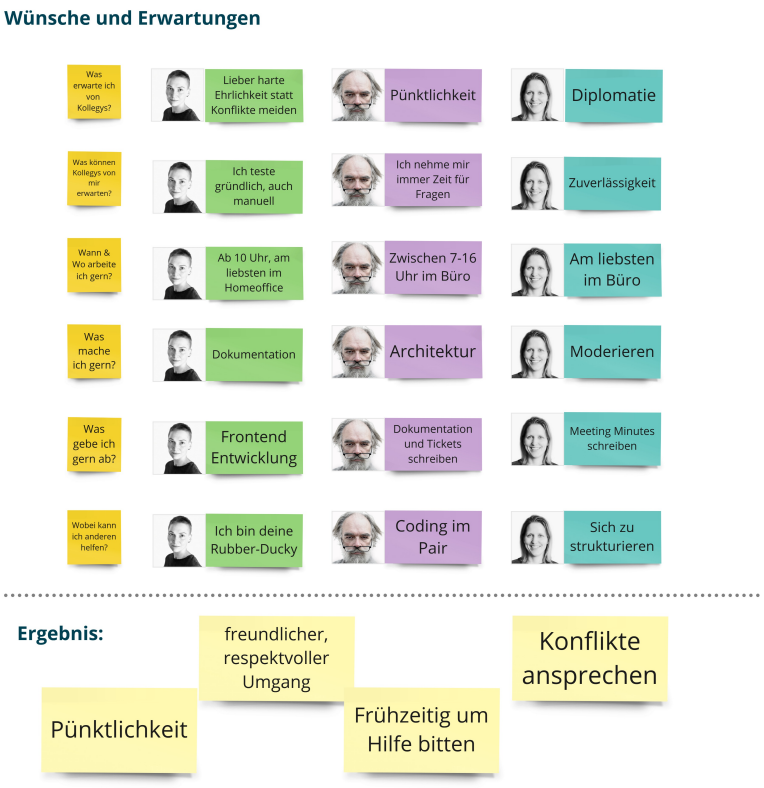
Reihenfolge

Wir empfehlen eine bestimmte Reihenfolge, da Abschnitte aufeinander aufbauen. Gerade der Beginn mit den „Wünschen und Erwartungen“ eröffnet eine persönliche Art der Kommunikation und des Verständnisses:

1. Wünsche und Erwartungen
2. Projektziel
3. Rollen
4. Entwicklungsprozesse und Methoden
5. Kommunikationskanäle
6. Absprachen und Meetings
7. Feedbackprozesse
8. Unsere Prinzipien
9. Das Projekt ist erfolgreich, wenn...

Einzelarbeit

Jeder Abschnitt wird in der oben genannten Reihenfolge im Team zunächst in Einzelarbeit bearbeitet. Jede Person notiert ihre Sichtweisen auf den aktuellen Abschnitt (z. B. „Wünsche und Erwartungen“).



Diskussion

Das Team tauscht sich über Gemeinsamkeiten und Unterschiede zu jedem Abschnitt aus. Dabei ist es hilfreich, sich gegenseitig von Erfahrungen aus anderen Projekten zu berichten, um den eigenen Standpunkt verständlich zu machen.

Ziel ist es, Transparenz über die individuellen Sichtweisen zu schaffen und gemeinsam zu diskutieren, welche Methoden und Praktiken sinnvoll sind, um sie für das gesamte Team zu etablieren.

Dokumentation

Die Absprachen, die in Zukunft für das ganze Team gelten sollen, werden in knappen Stichpunkten im Canvas notiert. Dabei braucht es keine zwingende oder vollständige Einigung. Unterschiedliche Standpunkte können deutlich werden und Zweifelsfall auch später weiter diskutiert werden. Lücken sind erlaubt und schaffen Freiräume.

2.6.5 Empfehlungen

Folgende Tipps helfen bei der Durchführung des Workshops:

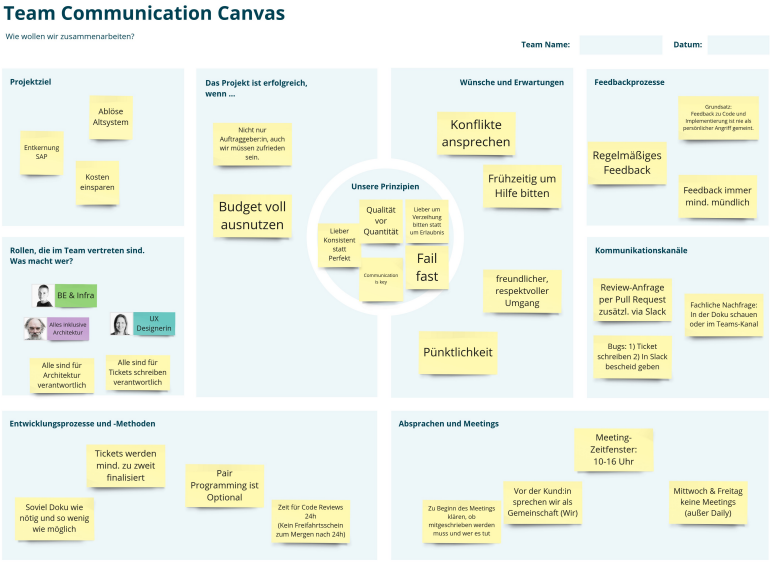
- Schafft eine vertrauensvolle Atmosphäre: Alle sollen das Gefühl haben, frei über ihre Erwartungen und Bedürfnisse sprechen zu können.
- Seid offen für unterschiedliche Perspektiven und versucht, konstruktiv Kompromisse zu finden oder lasst bewusst Lücken, um Freiräume zu schaffen.
- Dokumentiert die Ergebnisse so, dass sie leicht für die Beteiligten auffindbar sind (z.B. einsehbar im Team-Wiki).
- Plant genug Zeit ein. Wir empfehlen 2-3 Stunden.

- Vereinbart, den Workshop zu wiederholen, sobald sich Teamkonstellationen ändern, und aktualisiert den Canvas entsprechend.
- Holt euch bei Bedarf eine „neutrale“ Moderation dazu.

Folgende Stolpersteine können auftreten:

- **Zu hohe Erwartungen an „Endgültigkeit“**: Der Canvas ist kein fixes Regelwerk für alle Ewigkeit. Er lebt davon, dass man ihn anpasst, wenn sich das Team weiterentwickelt.
- **Zu wenig Offenheit**: Wenn Teammitglieder sich nicht trauen, ihre echten Bedürfnisse auszusprechen, bleibt das Potenzial ungenutzt.
- **Erstellen und Vergessen**: Einfach nur „ausfüllen und abheften“ reicht nicht. Macht euch die gemeinsamen Absprachen immer wieder im Alltag bewusst und integriert diese. In Retrospektiven können der Abgleich des Projektziels und das Erinnern an die Erfolgsfaktoren eine notwendige Kurskorrektur motivieren.
- **Bewertung durch Außenstehende**: Der Canvas gehört dem Team. Er ist nicht dazu gedacht, Teams von außen zu bewerten oder zu vergleichen.

2.6.6 Beispiele

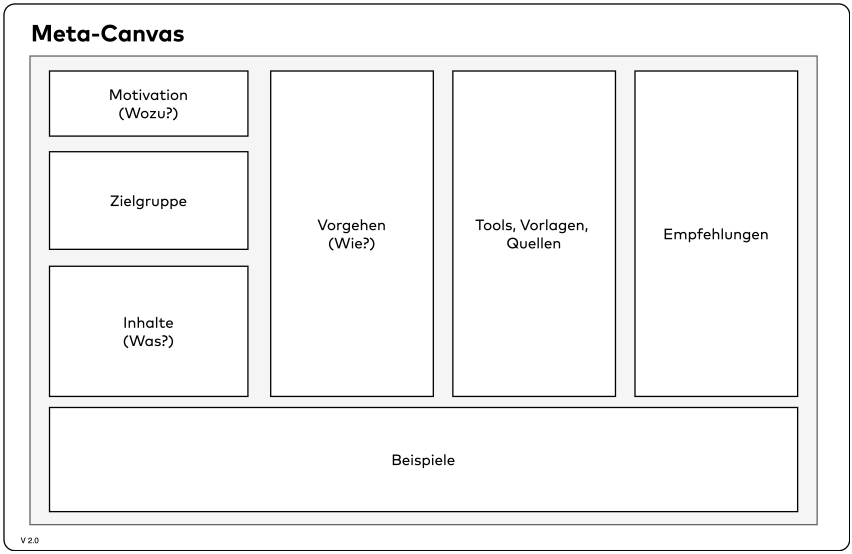


Gerade in Beratungsteams, die immer wieder in neuen Konstellationen starten, hat sich dieses Vorgehen bewährt. Es wird ein neuer Canvas zu Projektbeginn ausgefüllt, sodass alle Beteiligten gleich zu Anfang wissen, wie sie miteinander umgehen wollen. Auch in langlaufenden Projekten kann ein Canvas-Workshop helfen, wenn sich die Teamzusammensetzung stark ändert oder neue Personen hinzukommen, die andere Perspektiven mitbringen. Der Canvas erleichtert es, Anregungen einzubringen, ohne dass sie im „blinden Fleck“ der eingespielten Teamrituale untergehen.

2.7 Meta-Canvas

Von Benjamin Wolf und Gernot Starke.

> We drink our own champagne.



Eigentlich ist dieser Canvas aus einem Scherz heraus entstanden. Als wir uns Gedanken machten, wie wir die einzelnen Canvases möglichst einheitlich beschreiben sollten, schlug Ben als Witz einen Canvas zur Beschreibung der Canvases vor. Gar nicht so abwegig, oder? Im Folgenden sind kurz die einzelnen Felder beschrieben, die Sie im Canvas oben sehen können. Vielleicht ist dieser Canvas auch für Sie nützlich, um Themen strukturiert zusammenzufassen – es muss ja nicht zwingend ein Canvas sein.

Motivation (Das *Wozu*)

Dieses Feld beschreibt relativ knapp, welchen Zweck der beschriebene Canvas erfüllt. Hier können Sie erklären, welchen Mehrwert er bietet, welche Probleme Sie damit genau lösen wollen, oder welche speziellen Fragestellungen Sie damit beantworten möchten.

Zielgruppe

Ein wichtiger Aspekt bei jedem Canvas stellt – wie bei jeder Form von Dokumentation – die Zielgruppe dar, die Sie ansprechen möchten. Richtet sich Ihr Canvas an Fachbereiche, Product Owner oder das Management? Möchten Sie damit Strategieteams oder Softwareteams adressieren? Die Zielgruppe bestimmt, auf welchem Vorwissen, welcher Sprache und welchen Perspektiven der Canvas aufbauen kann.

Inhalte (Das *Was*)

Mit einer der interessantesten Gesichtspunkte eines Canvas ist dessen Inhalt. Welche Bereiche, Themenfelder oder Dimensionen decken Sie mit dem Canvas ab? Beschreiben Sie hier die einzelnen Bausteine des Canvas und welche Informationen die jeweiligen Felder enthalten.

Vorgehen (Das *Wie*)

Hier beschreiben Sie, wie der Canvas am besten eingesetzt und befüllt werden soll. Stellen Sie sich die Fragen, ob Sie spezielle Methoden zur Erstellung des Canvas verwenden möchten, und ob der Canvas eher von einer Person alleine oder von einer Gruppe befüllt werden soll, z. B. in speziellen Workshop-Formaten.

Tools, Vorlagen, Quellen

Bestimmt gibt es bereits eine Reihe von Vorlagen (druckbar oder digital), welche die Nutzung des Canvas unterstützen. Diese listen Sie hier auf und fügen Literaturhinweise, Websites oder andere Ressourcen hinzu.

Empfehlungen

Wenn Sie bereits mit Ihrem Canvas gearbeitet haben, gibt es sicher einige positive wie negative Erfahrungen, die Sie dabei gemacht haben. Leiten Sie daraus Empfehlungen und Tipps zur Anwendung ab. Ebenfalls hilfreich sind Hinweise,

wie die Ergebnisse im größeren Kontext funktionieren und mit welchen anderen Methoden oder Canvases sie zusammenhängen.

Beispiele

Konkrete Anwendungsfälle oder Fallstudien zeigen den Mehrwert und erleichtern das Verständnis abstrakter Konzepte.

3 Tools

A fool with a tool is still a fool.

Grady Booch[1]

Die hier vorgestellten Werkzeuge respektive Werkzeugkategorien können Sie für alle Arten von Canvases verwenden.

Papier

Eines unserer favorisierten Werkzeuge zum informellen Arbeiten an einem Canvas. Papier hat diverse Nachteile (schlecht änderbar, nicht automatisch durchsuchbar, begrenzte Fläche, geringe Haltbarkeit, empfindlich gegenüber Kaffeeflecken ...), jedoch auch bestechende Vorteile:

- Keine Einstiegshürde, extrem einfach.
- Als *haptisches* Medium fühlt es sich wirklich wie *gemeinsame* Arbeit an, einen Canvas auf einem Papier-Flipchart zu zeichnen und mit Inhalt zu befüllen.
- Der Fokus liegt automatisch mehr auf dem Inhalt als auf dem Werkzeug.
- Alle Beteiligten können sich aktiv beteiligen und kommen ins Gespräch.

Besonders gut finden wir Papier in frühen Phasen, beim Brainstorming oder der Ideenfindung beim Canvas. Später, wenn die Inhalte sich gesetzt haben oder abgestimmt sind, können Sie auf digitale Werkzeuge wechseln.

Digitale Zeichenwerkzeuge

Falls Sie partout nichts mit Papier anfangen können, sind vielleicht digitale Zeichenwerkzeuge die richtige Wahl für Sie. Die Gruppe diskutiert, eine Person schreibt / zeichnet in den Canvas.

[1] https://research.ibm.com/people/grady-booch

- draw.io[2] ist vermutlich das bekannteste Zeichenwerkzeug heutzutage. Sie finden für einige der hier vorgestellten Canvases bereits Templates für draw.io.
- Visio, OmniGraffle oder Photoshop, letztlich können Sie jedes beliebige Zeichenwerkzeug verwenden, mit dem Sie zurechtkommen. Sie müssen nur darauf achten, dass alle Beteiligten die jeweiligen Werkzeuge zur Verfügung haben.

Online-Werkzeuge

Browser-basierte Online-Werkzeuge eignen sich gut für die gemeinsame Arbeit am Canvas. Sie können Ergebnisse gut aufheben, weitergeben, archivieren oder in andere Formate exportieren. Leider müssen Sie bei allen der genannten Werkzeuge für das Thema „Versionierung" eine eigene Lösung finden.

- Canvanizer[3], ein Start-up, das kollaborative Erstellung und Pflege von Canvases erlaubt. Einige Features können Sie in einer freien Version verwenden, für fortgeschrittene Aufgaben benötigen Sie ein Abo.
- Conceptboard®[4], Miro®[5] oder Mural®[6] sind kollaborative Zeichenwerkzeuge, mit denen Sie in einer Gruppe online an Ihrem Canvas arbeiten können.
- draw.io[7]-Plugin für Confluence[8]. So haben Sie die Möglichkeit, in einem Ihnen bekannten Zeichenwerkzeug kollaborativ zu arbeiten – eine ideale Lösung, wenn Sie bereits Confluence in Ihrer Firma einsetzen.

PowerPoint® und Co.

Ja, Sie lesen richtig: PowerPoint®. Für komplett andere Anwendungszwecke gedacht, halten wir Präsentationswerkzeuge wie PowerPoint® für den Einsatz bei Canvases für hilfreich:

---

[2] https://www.drawio.com
[3] https://canvanizer.com/
[4] https://conceptboard.com/
[5] https://miro.com/de/
[6] https://www.mural.co/
[7] https://www.drawio.com
[8] https://www.atlassian.com/software/confluence

- Die Struktur des Canvas ist als Folienvorlage (in PowerPoint®-Terminologie: *Master*) gespeichert.
- Die Inhalte liegen dann als einfache Textfelder vor.

Sehr charmant finden wir die Evolutionsfähigkeit dieser Art von Dokumentation: Einfaches Copy+Paste einer Canvas-Folie schafft eine Replik, in der Sie dann lediglich die Änderungen zur Vorgängerversion übernehmen müssen. Damit können Sie sehr einfach den Fortschritt der Arbeit darstellen.

4 Anhang

Wir beraten ehrlich, denken innovativ und entwickeln leidenschaftlich gern. Das Ergebnis: Erfolgreiche Softwarelösungen, Infrastrukturen und Geschäftsmodelle.

Als Technologieunternehmen fokussieren wir uns auf Strategie- und Technologieberatung, Softwarearchitektur und -entwicklung, Methoden- und Technologietraining sowie Plattform-Infrastrukturen.

Wir unterstützen mit über 160 Mitarbeiter:innen an Standorten in Deutschland und der Schweiz Unternehmen und Organisationen bei Konzeption und Umsetzung komplexer Vorhaben und der Verbesserung bestehender Softwaresysteme.

Wir engagieren uns in Open-Source-Projekten sowie dem iSAQB® e.V., und geben Wissen und Erfahrungen auf Konferenzen und Meetups sowie in zahlreichen Büchern und Fachartikeln weiter.

Besuchen Sie uns: **www.innoq.com**

Literatur

[1] A. Osterwalder und Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, 2010, S. 278.

[2] P. Roos, „The Software Architecture Canvas: An Efficient and Collaborative Way to Define Your Software Architecture Playground“. https://www.workingsoftware.dev/software-architecture-canvas/, 2023.

[3] H. D. Special, „Episode 97: Architecture Canvas (2): Einstieg in die Lösung“. Podcast, https://www.heise.de/blog/Episode-97-Architecture-Canvas-2-Einstieg-in-die-Loesung-9697394.html, 2024.

[4] G. Starke und B. Wolf, „Der Architecture Communication Canvas“. https://www.informatik-aktuell.de/entwicklung/methoden/der-architecture-communication-canvas.html, 2024.

[5] canvas.arc42.org, „Architecture Communication Canvas“. https://canvas.arc42.org/architecture-communication-canvas, 2024.

Autor:innen

Markus Harrer

Markus Harrer ist Principal Consultant bei INNOQ. Er unterstützt Unternehmen bei der Modernisierung von Legacy-Systemen und der Einführung von effektiven Engineering-Praktiken.

Anja Kammer

Anja Kammer ist Senior Consultant bei INNOQ und begleitet Unternehmen auf ihrem Weg in die Cloud. Neben der Beratung zu Entwicklungsprozessen und -plattformen entwickelt sie Cloud-native Webanwendungen in cross-funktionalen Teams. Zudem ist sie akkreditierte Trainerin und Co-Kuratorin für das iSAQB Advanced-Level-Modul CLOUDINFRA.

Lena Kraaz

Lena Kraaz arbeitet als Senior Consultant bei INNOQ. Ihre Schwerpunkte liegen in der Arbeit mit Teams, Prozessen und Product Ownership. Sie unterstützt Unternehmen als Facilitator bei der Umsetzung von Projekten und agilen Prozessen.

Jörg Müller

Jörg Müller ist Principal Consultant bei INNOQ. Er verfügt über mehr als 25 Jahre Erfahrung in der Softwarebranche. In dieser Zeit hatte er viele Engagements in der Beratung, der Entwicklung von Softwareprodukten und der Leitung von Teams. Seine Kernkompetenzen liegen in den Bereichen Softwarearchitektur und DevOps. Neben dieser Arbeit engagiert sich Jörg für die Community. Er organisiert User Groups und Konferenzen und hält dort Vorträge.

Patrick Roos

Patrick ist als Tech Lead, Softwarearchitekt und -entwickler im Bereich von SaaS-Lösungen bei der Fellow GmbH tätig. Er engagiert sich auch als nebenamtlicher Dozent an der Hochschule Luzern für Themen rund um Frontend-Architekturen, Webtechnologien und Software-Architektur. Auf seinem Blog workingsoftware.dev teilt er Einblicke und Erfahrungen aus der Praxis.

Gernot Starke

Dr. Gernot Starke (INNOQ Fellow) arbeitet seit über 25 Jahren in der Softwareentwicklung, als Softwarearchitekt, Coach, Consultant und Trainer.
2005 hat Gernot gemeinsam mit Dr. Peter Hruschka das arc42-Template veröffentlicht und es seitdem in zahlreichen Entwicklungsprojekten eingesetzt und mehrere größere Systeme damit dokumentiert. Gernot ist Gründungsmitglied des iSAQB e. V., leitet die Arbeitsgruppe Foundation Level und kuratiert die Lehrpläne IMPROVE, REQ4ARC und ADOC.

Benjamin Wolf

Ben ist Senior Consultant bei INNOQ mit den Schwerpunkten Modernisieren von Legacy-Systemen, Architekturdokumentation sowie Architekturberatung und -entwicklung. Dabei richtet er ein besonderes Augenmerk auf die Entwicklungsprozesse und die Einstellung zu Softwarequalität in Teams.