

Agile Softwareentwicklung im Spannungsfeld öffentlicher Institutionen

SOPHIST GmbH

Vordere Cramergasse 13

90478 Nürnberg, Deutschland

Fon: +49 (0)911 40 900 - 0

Fax: +49 (0)911 40 900 - 99

www.sophist.de

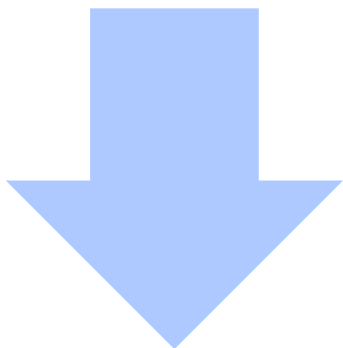
heureka@sophist.de

Das Projekt und dessen Umfeld

Eines der aktuell größten IT-Projekte der Behörde.



Agilität und auf Stabilität ausgelegte Prozesse erzeugen ein Spannungsfeld.



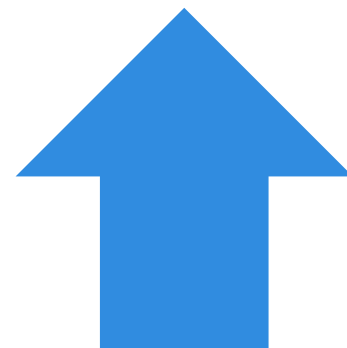
Ziele des agilen Vorgehens

- Ganzheitliches, interdisziplinäres Denken
- Ergebnisverantwortung durch Fachseite und IT
- Dynamisches Online-Umfeld beherrschen



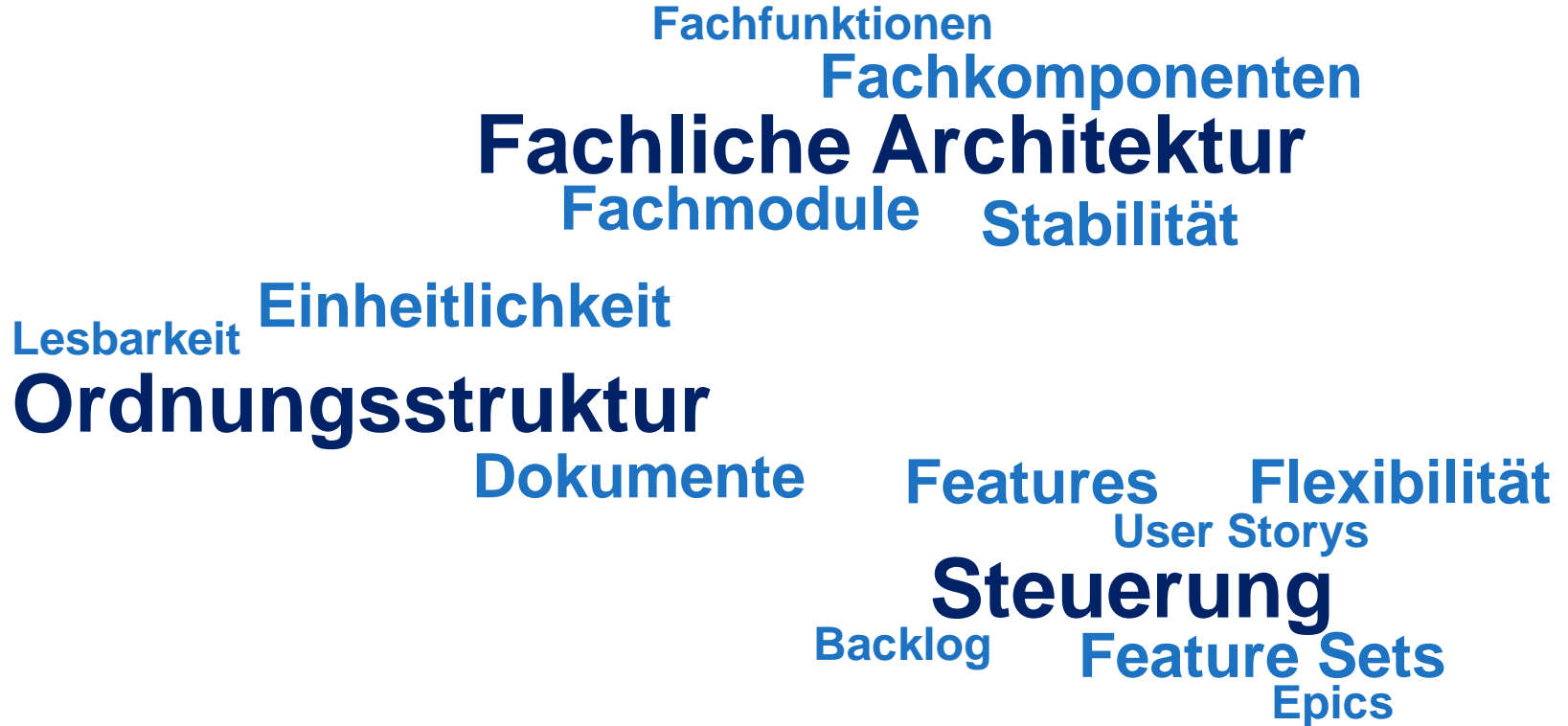
Rahmenbedingungen

- Hierarchische Aufbauorganisation
- Revision, Projektaudits, Bundesrechnungshof
- Etablierte Entwicklungs- und Betriebsprozesse



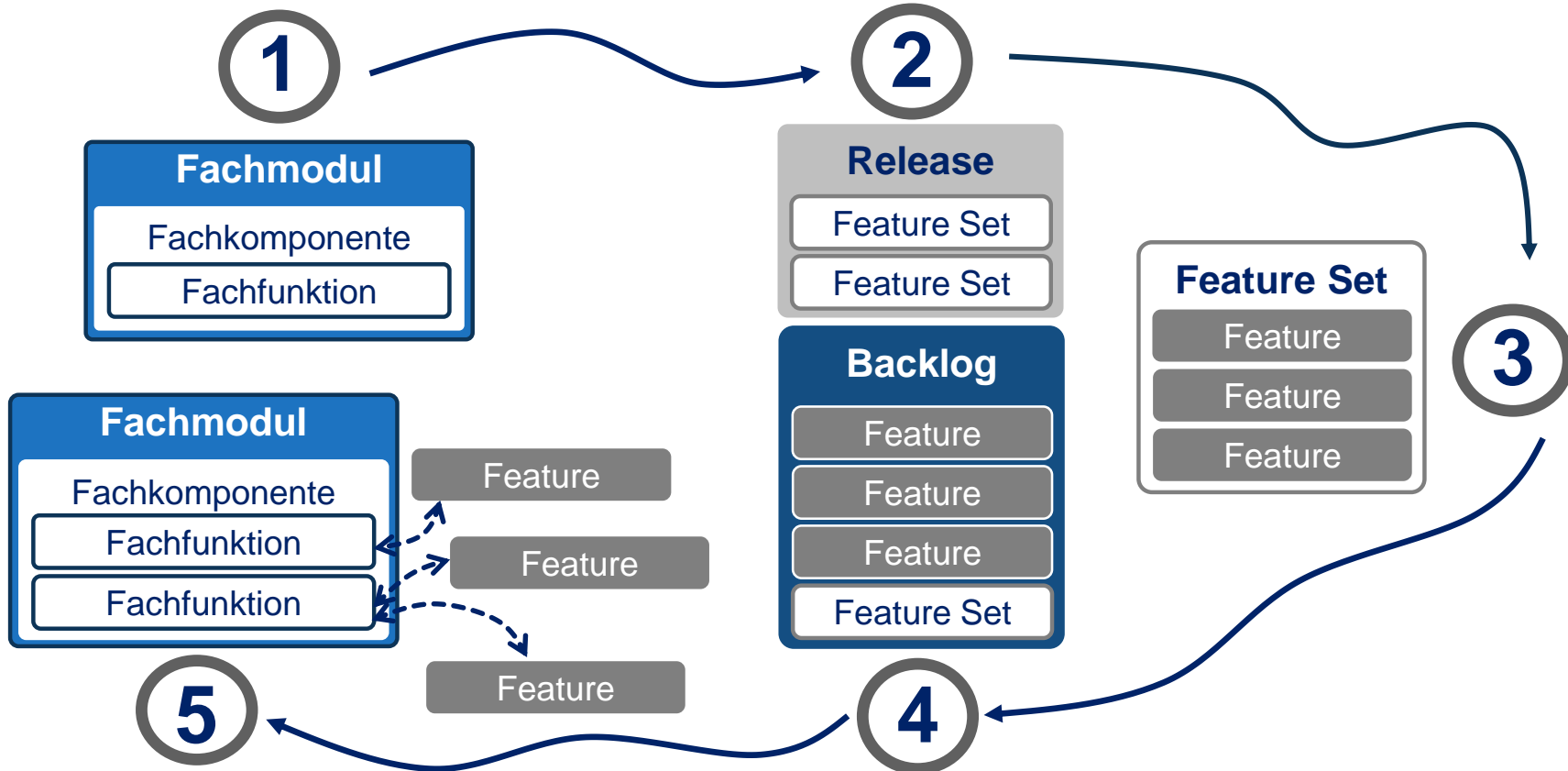
Fachliche Architektur als Ordnungsstruktur

Das Verständnis des Gesamtauftrags bildet die Grundlage agiler Arbeit.



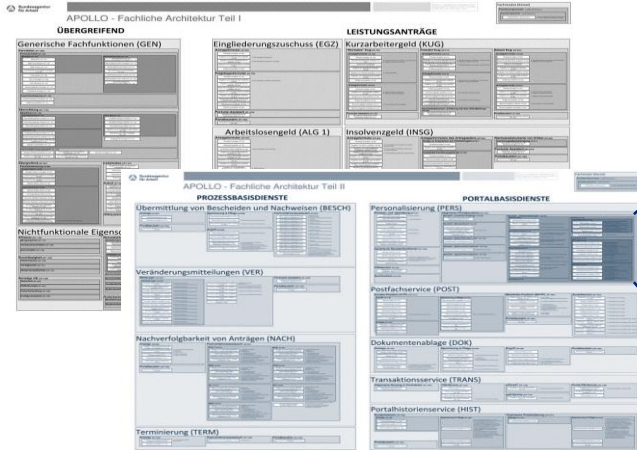
Wissen ordnen und Produktion steuern

Die fachliche Architektur ist ungeeignet um die Produktion zu steuern.



Lösungsansatz zur Stabilität

Die fachliche Architektur ist eine lebenslange Ordnungsstruktur.



Orientierungsrahmen für Schnitte

Fachmodule

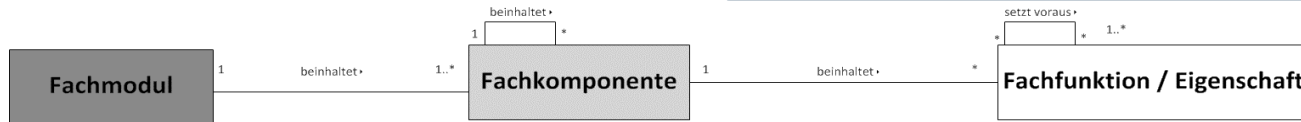
- nach Basisfachkonzepten
- mit kleineren Anpassungen

Fachkomponenten

- Fachliche Aggregation aus Fachfunktionen
- Berücksichtigung der Modulstruktur
- Ziel: Handhabbarkeit erhöhen

Fachfunktionen

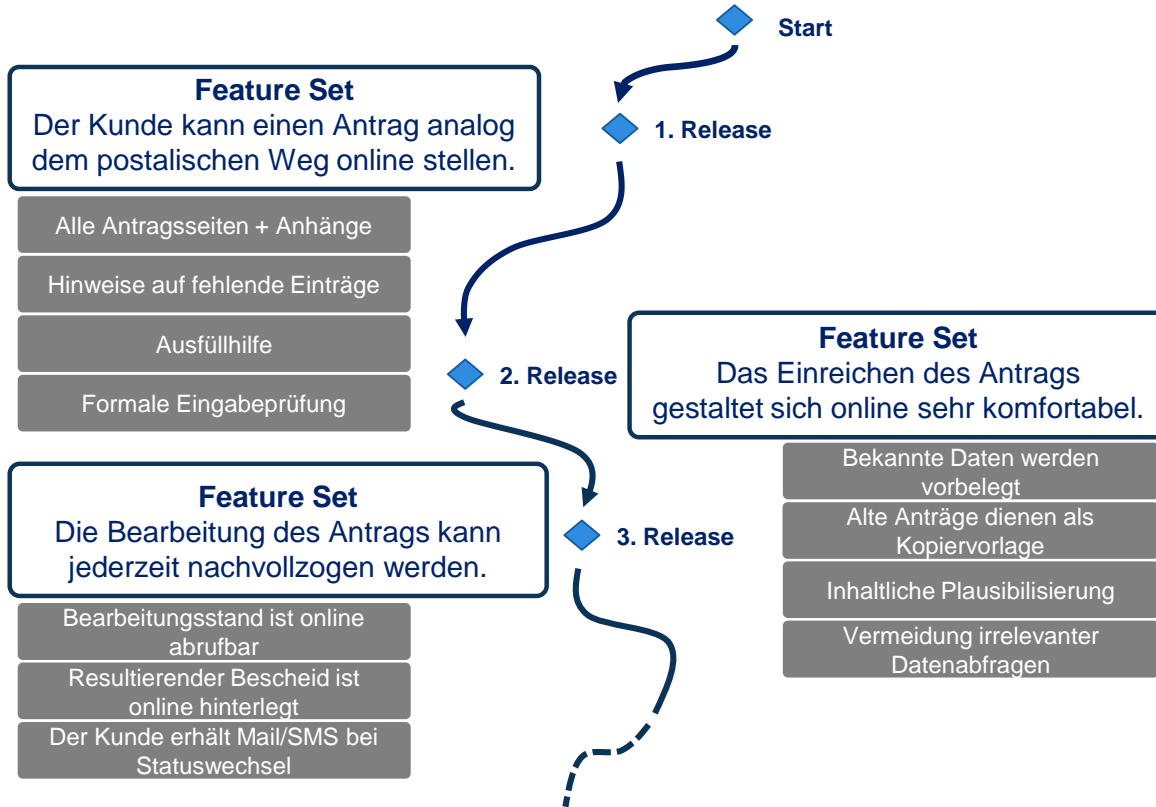
- Struktur der existierenden Geschäftsprozesse
- Kundensicht der Funktionalität
- Normalisierung
- Erfahrungsaustausch im Rahmen der Definition



Stabilität und Verständlichkeit im Fachbereich vor methodischer Perfektion

Lösungsansatz zur Agilität

Product Owner denken in Releases, Feature Sets und Features.

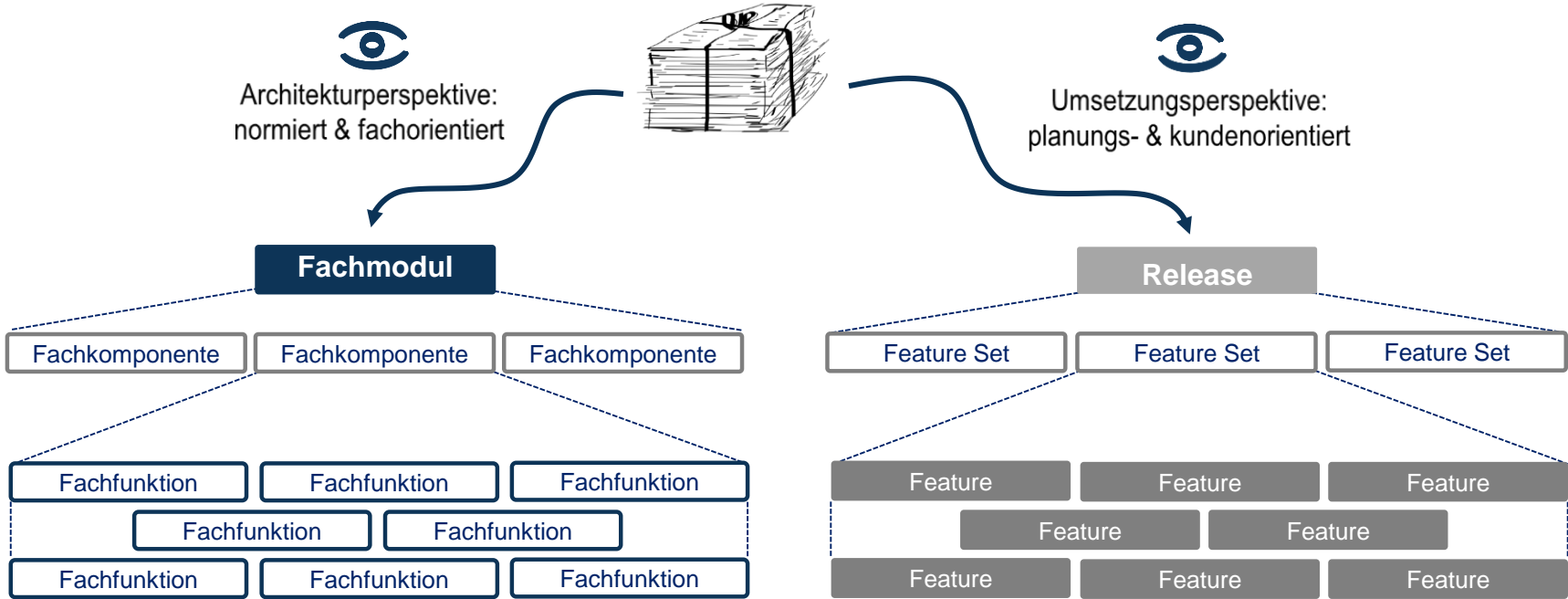


Features

- Kernwünsche der Stakeholder
- variieren in Größe und Genauigkeit
- sind überlappend und überholen sich
- entstehen auch während Entwicklung (im Sprint-Review, aus dem Team, in Gremien, aus Kundenfeedback etc.)
- werden zu Feature Sets aggregiert

Integration von Agilität und Stabilität

Vereinigung der konträren Sichten in einem Vorgehensmodell.



Schlüsselfaktor: Integration der Sichten

Aspekte des Vorgehensmodells

Das Modell bildet eine Ordnungsstruktur für mehr als 120 Projektmitarbeiter.

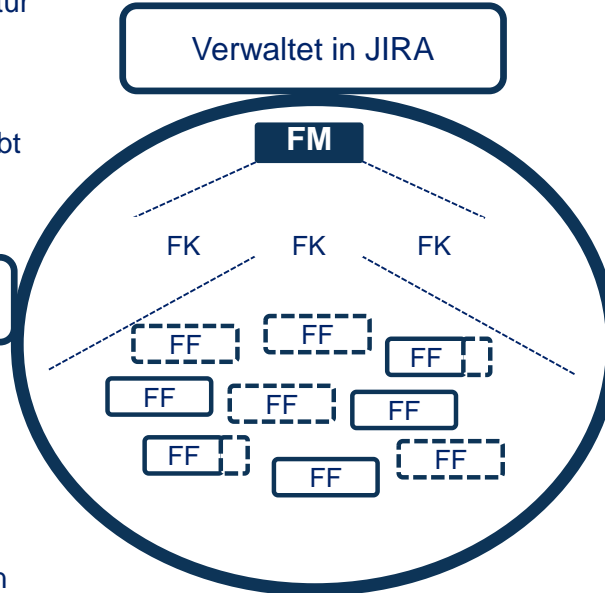
Systemdokumentation & Pflege

- Gliederung systembeschreibender Dokumentation nach fachlicher Architektur
- Einstieg der Auswirkungsanalyse (Fehlerfall, Wartung, Change Request) immer über fachliche Architektur
- Konsequente Trennung im Projekt erlaubt einfache Verteilung auf mehrere Linieneinheiten

Lebenszyklus durch Statusmodell

Systemtestfall- und Codeorganisation

- Alle Systemtestfälle werden in die fachliche Architektur einsortiert
- Ausrichtung der Codestrukturen und Softwareartefakten an den Fachmodulen oder tiefer
- Durch gemeinsame Ordnungsstruktur implizite Traceability Fach – Code – Test



Teamstruktur

- Ausrichtung der Entwicklungs-Teams an den Fachmodulen
- Falls nicht möglich: niemals arbeiten mehrere Teams gleichzeitig an der selben Fachfunktion
- Langfristiger auf Modul/Komponente bezogener Know-How-Aufbau in den Teams

Lebenslange Primärschlüssel

Berichtswesen & Erfolgskontrolle

- Schätzung und Planung auf Basis von Fachfunktionen
- Stabile Fortschrittsberichte je Fachfunktion an das Top-Management
- Kosten- und Aufwandsbuchung auf Feature Sets in den Teams
- 100% der nicht übergreifenden Arbeiten werden Feature Sets zugeordnet

Bewertung nach über 3 Jahren Entwicklungszeit

Es war die richtige Entscheidung.

Was hat sich bewährt?

- Vereint fachliche, technische und organisatorische Aspekte
- Stabil und unabhängig von Architektur-/Technologieentscheidungen
- Erscheint auch heute noch zukunftssicher
- Teamschnitt nach Fachmodulen schafft Teamunabhängigkeit

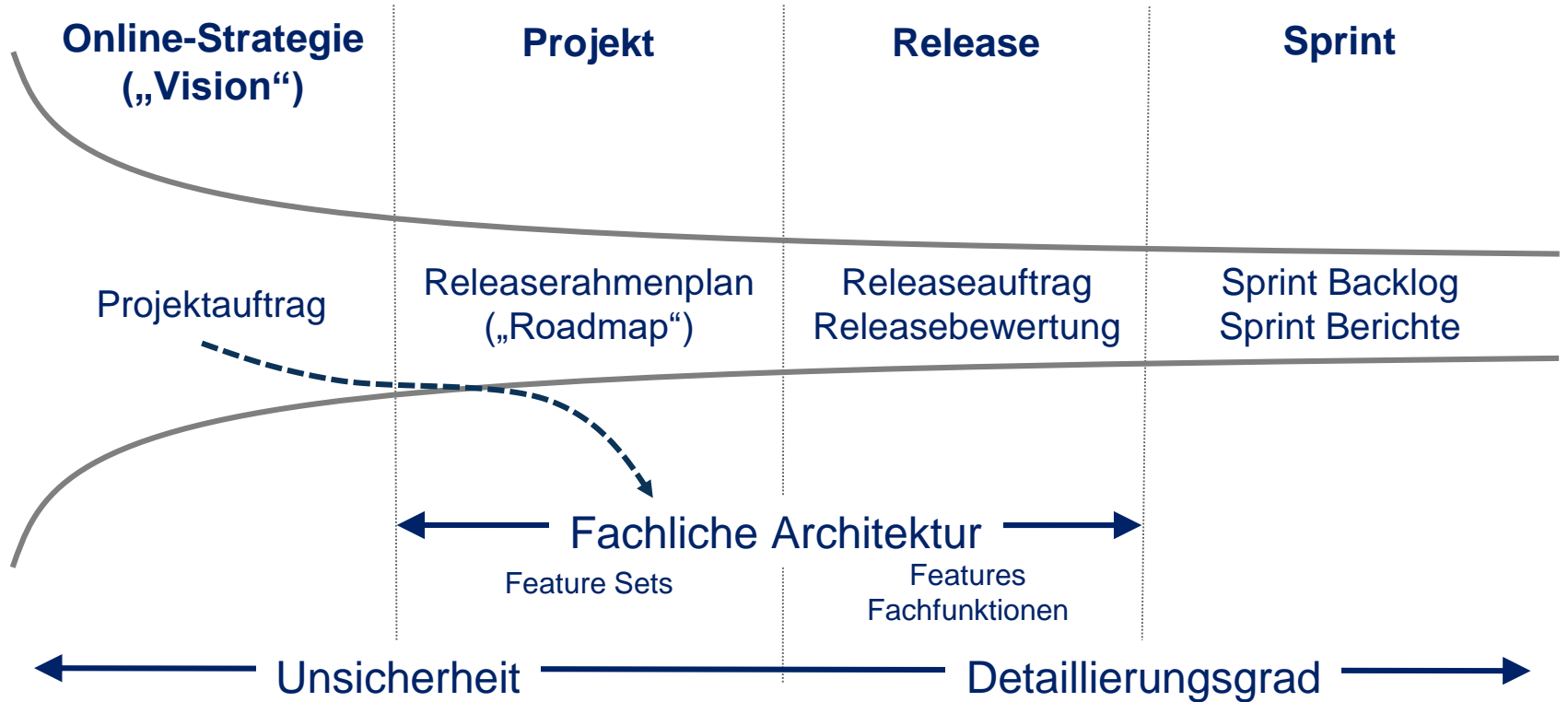
Was würden wir beim nächsten Mal anders machen?

- Schnitt der technischen Systemgrenzen nach Fachmodul
- Keine Nutzung zur Planung (v. a. bei „Wasserfall“-Erfahrung)
- Keine Normalisierung der fachlichen Architektur



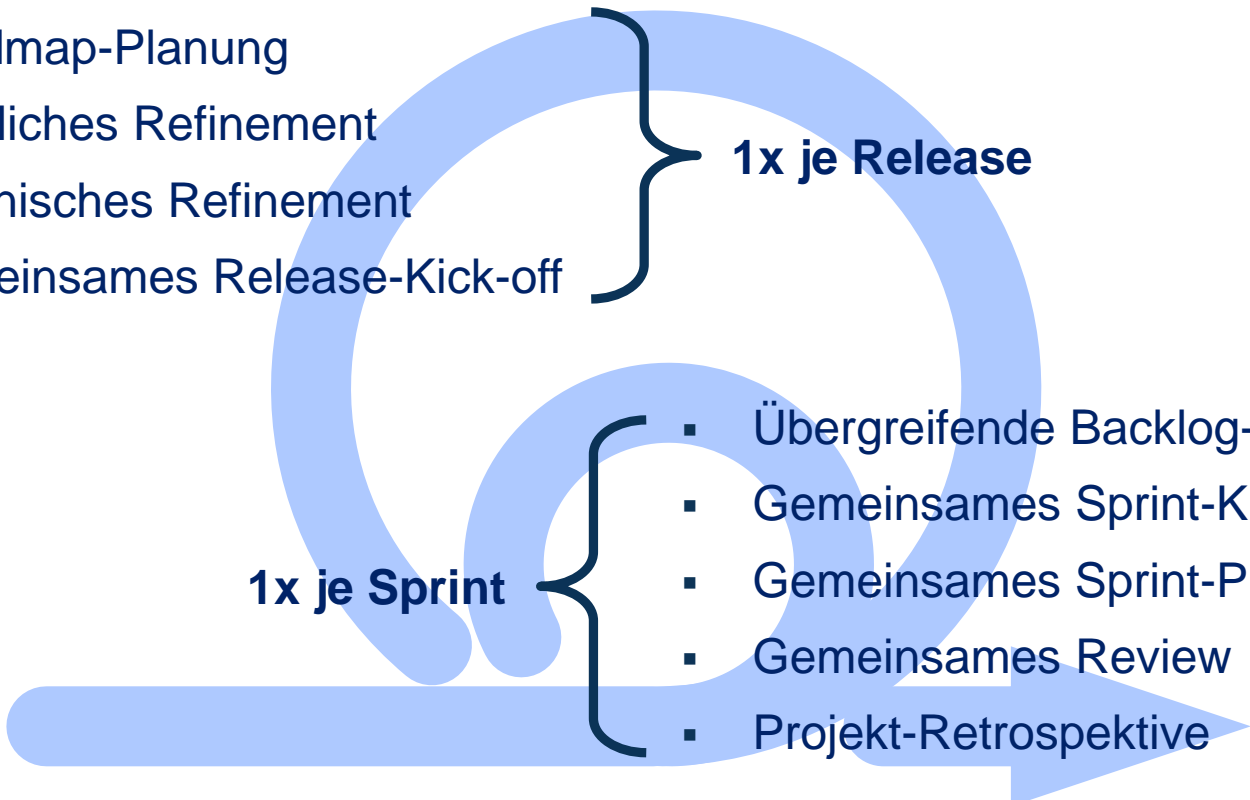
Die Fachliche Architektur im Planungsprozess

Abgeleitet aus der Online-Strategie, Grundlage der Roadmap und Releases.



Prozess der mehrstufigen Planung

Synchronisation über regelmäßige projektweite Abstimmungsformate.

- 
- Roadmap-Planung
 - Fachliches Refinement
 - Technisches Refinement
 - Gemeinsames Release-Kick-off
- 1x je Release**
- Übergreifende Backlog-Priorisierung
 - Gemeinsames Sprint-Kick-off
 - Gemeinsames Sprint-Planning I
 - Gemeinsames Review
 - Projekt-Retrospektive
- 1x je Sprint**

Erfolgsfaktoren der agilen Planung

Vertrauen bestätigen – Freiräume schaffen.

Glaubwürdigkeit gegenüber Auftraggeber und Teams

- Product Owner und Teams einbeziehen
- Umfang nicht „verordnen“
- Mit frühzeitigen Ergebnissen überzeugen
- Controlling des Fortschritts (Backlog vs. Restlaufzeit)
- Transparenz der Planung
 - „Warum sind wir überzeugt, das Ziel wirklich zu erreichen?“
 - Auch und gerade, wenn es nicht rund läuft („Bad news first!“)

Ausreichende Qualifikation und Erfahrung

- Moderation und Steuerung durch erfahrene Kollegen
- Bereitschaft der Projektleitung zur rollierenden Planung

Jede Anpassung der Software verändert bestehende Arbeitsabläufe.

Auftraggeber

Hotline

Interne Stakeholder

Sachbearbeiter

Beratungsfachkräfte

Multiplikatoren

Compliance

Externe Stakeholder

Kunden

Single Point of Contact

Product Owner

Product Manager **Linienorganisation**

Informationsdrehscheibe

Scrum Master

Kontinuität

Rollendefinition

Aufgaben- und Verantwortungsfülle im Vergleich zur Linienorganisation.

Der Product Owner als Single Point of Contact

- Einbindung interner und externer Stakeholder
- Abstimmung der veränderten Prozesse in der Organisation

Der Product Owner als Product Manager

- Planung, Entwicklung, Produkteinführung, Support
- Führungs- und Managementaufgaben
- Operative Tätigkeiten delegieren

Scrum Master – Erfolgsfaktor im Hintergrund

Scrum Master unterstützen Product Owner und agieren als Infodrehscheibe.

Verlässlichkeit in volatilem Projektumfeld

- Konstante auch bei Fluktuation im Team

Informationsdrehscheibe

- Informationsaggregation für Projektleitung
- Sorgt für Kommunikation zwischen den Entwicklungsteams

Organisatorische Voraussetzungen

- Ein Scrum Team pro Scrum Master
- Keine Personalunion mit anderen Rollen – auch nicht verdeckt
 - Der Scrum Master unterstützt den Product Owner
 - Neutralität als Voraussetzung für Arbeit als Team-Entwickler

Nicht-funktionale Anforderungen bergen das Risiko technischer Schulden.



Unterstützung der Product Owner

Delegation der Durchführung an Querschnittsrollen und Fachexperten.

Querschnittsrollen

- Architektur
- Schnittstellenmanagement
- Fachlich-technische Analyse
- Betriebsunterstützung
- Release-Management
- User Experience Experten
- IT-Sicherheitsverantwortliche / „Agile Security“

Bewertung

- Konfliktpotential „Technik“ vs. „Fachlichkeit“
- Hohes Maß an Kompromissbereitschaft notwendig

Kontinuierlicher Verbesserungsprozess (KVP)

Die „Revolution von unten“ fördert Verantwortungsübernahme.

Änderungsbedarfe aus den Entwicklungs-Teams adressieren

Wie lässt sich die „Trägheit der großen Masse“ überwinden?

Formalisierter, team-übergreifender KVP

- Anstoß durch Management, selbst aktiv zu werden
- Dedizierte Themen-Verantwortliche
- Vertreter aller Teams und der Product Owner
- Moderation durch einen Scrum Master

Entwicklungs-Teams übernehmen Verantwortung

Vorgehen für einen erfolgreichen KVP

Realistische Ziele und Erwartungen angepasst an das Machbare.

Kleine Schritte und erreichbare Ziele

- Gemeinsame, projektweite Retrospektive
- Erfolge feiern, z.B. im Review gleichwertig zu Features vorstellen

Culture Eats Strategy for Breakfast

- Offene Fehlerkultur statt Absicherungsdenken
- Bereitschaft des Managements zu Experimenten schaffen
 - Rahmen vorher klar abstecken
 - Negative Auswirkungen begrenzen
- Klare Verantwortlichkeiten für Tasks („Man müsste mal...“)
- Mitarbeiter nicht überfordern
- Unterstützung durch agilen Coach
- Viel Zeit und Beharrlichkeit

Wo stehen wir heute?

Auf dem Weg zur agilen Organisation.



Erfolgsfaktoren einer agilen Organisation

Zusammenarbeit unabhängiger, eigenverantwortlicher Teams.

Entwicklungs-Teams

- Übernahme weiterer Verantwortung und Aufgaben

Product Owner

- Ganzheitliche Verantwortung über den gesamten Lebenszyklus

Querschnittsfunktionen

- Arbeiten darauf hin, sich selbst obsolet zu machen

Management

- Beschränkt sich auf Vorgabe grober Leitlinien
- Passt Prozesse kontinuierlich an Fähigkeiten der Mitarbeiter an

Ist eine fachliche Architektur agil?

Raum für Refactorings und iteratives Vorgehen wird kaum gewährt.

Agile Erstellung
Ordnungsstruktur
Fachliche Architektur
Ex ante Stabilität

Iterationen
Agilität
Refactorings
Prohibitiv hohe Kosten





Melden Sie
sich zum
SPEED
Consulting an.

Unser Messestand

Nachhaltig – Unverpackt – Individuell



Besuchen Sie
unseren
„leckeren“
Messestand

- Titel: Rückspiegel
 - Quelle: -
 - Autor: Chris Rupp

Rainer Joppich

SOPHIST GmbH

Vordere Cramergasse 13

90478 Nürnberg

rainer.joppich@sophist.de