

DEvelopers
Köln
Meetup

S&N
INVENT



Dark Side

Microservices – Was man bedenken sollte

Dr. Jürgen Lampe, Principal Consultant



Über mich

- Mathematiker
- 7 Jahre Programmierer (Prozessrechner)
- 16 Jahre TU (Assistent/Dozent)
- 20+ Jahre Berater
 - Finanzwirtschaft
 - Gesundheitswesen
 - Verkehrsunternehmen
- Interessengebiete
 - Fachsprachen (DSL)
 - Clean Code
 - Performance



Für die vorgetragenen Ansichten und Meinungen ist allein der Autor verantwortlich.

S&N Invent ist ein bundesweit aufgestelltes **IT-Beratungs-Unternehmen** mit der Dachgesellschaft S&N Group AG und den Tochtergesellschaften S&N AG, **A:gon Solutions GmbH**, ABISCON GmbH, ABISCON SAS GmbH und S&N CQM GmbH.

Branchenübergreifend bieten wir unseren Kunden ein ganzheitliches Leistungsspektrum für Ihren IT-Lifecycle mit einem soliden Fundament aus aktueller Architektur-, Technologie- und Methodenkompetenz.

Ergänzt wird dieses umfassende Leistungsportfolio um langjährige Projekterfahrung und ausgeprägte Branchenexpertise in der Finanz- und Versicherungswirtschaft und aktuelles SAP-Technologie-Know-how.

- 1 Motivation
- 2 Microservices – Quellen und Konzepte
- 3 Modularisierungserfahrungen
- 4 Verteilte Systeme
- 5 Bewertung – Fragenliste
- 6 Zum Schluss: Entscheiden

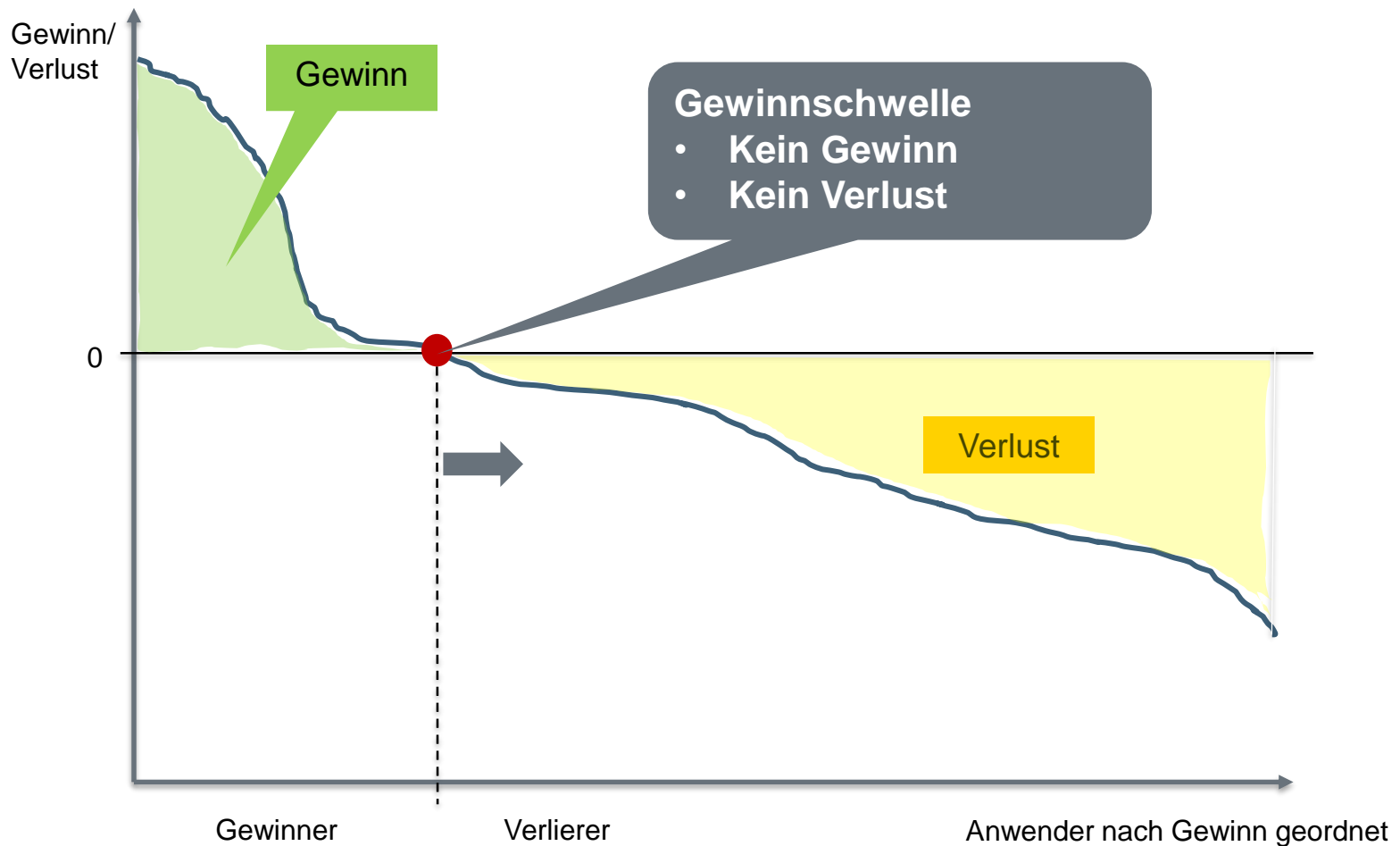
Motivation:

Herausforderung durch neue Technologie

- Vorteile – **helle Seite**
 - Propagierung notwendig, aber meist übertrieben
 - Nutzen aus Pilotanwendungen bekannt
 - - *Siehe vorangegangenen Vortrag* -
- Kosten – **dunkle Seite**
 - Immer vorhanden
 - Initial unbekannt, bestenfalls schätzbar
- Schadenspotential
 - Risiko bei Anwendung (z. B. durch Unfälle, Rechtsverstöße)
 - Unerwünschte Nebenwirkungen

(Bleibt hier unberücksichtigt)

Rentabilitätsdiagramm



Fragen vor dem Einsatz einer neuen Technologie

- Um was geht es überhaupt?
 - Oft überraschend unscharf

Microservies:

Adrian Cockcroft:

„Loosely coupled service oriented architecture with bounded contexts“

James Lewis:

„Applications that fit in your head“

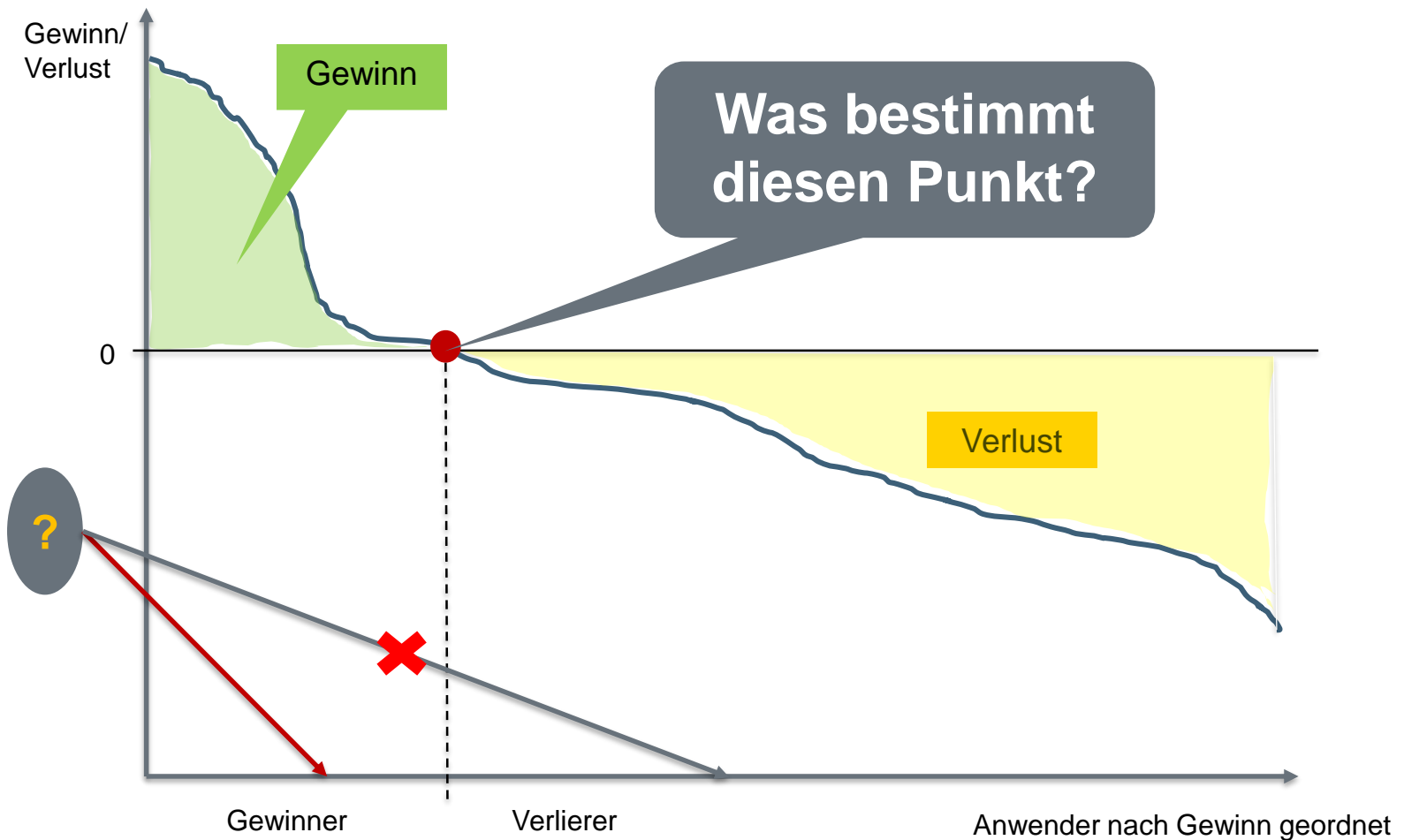
- Was ist der Nutzen?
 - Ist der Nutzen im betrachteten Einsatzgebiet realisierbar?
 - Nicht immer gut quantifizierbar
- Was sind die Kosten?
 - Welche Kosten entstehen?
 - Wovon hängt die Kostenhöhe ab?

Notwendig: Erfahrungen

Aber woher?

- Beste Quelle: gescheiterte / verzögerte Projekte
 - „*Aus Fehlern lernen*“
 - Leider existiert in IT keine *Fehlerkultur*
- Aus anderen Projekten
 - Ebenfalls schwierig
 - Externe als Vermittler (wichtig!)

Ihr Ziel: Gewinner

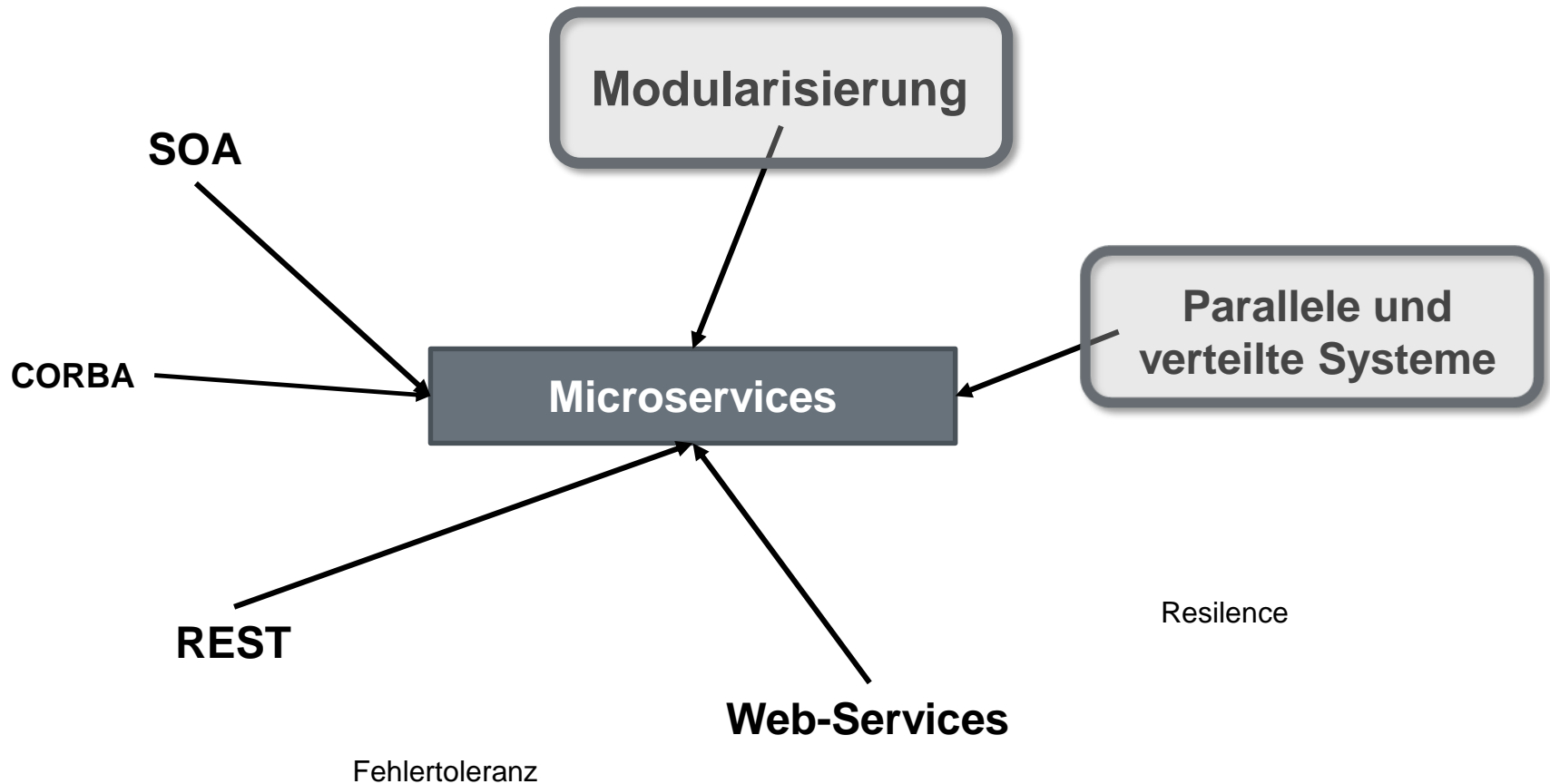


Wie den „roten Punkt“ bestimmen?

- Klarheit über Technologie gewinnen
- Identifizierung von Kernbestandteilen bzw. Konzepten
- Existieren Erfahrungen mit einzelnen Konzepten?
- Sind die Erfahrungen übertragbar?
- Welche Konsequenzen ergeben sich?
- Notwendige Bedingungen betrachten
- Einflussfaktoren bewerten

Microservices – Quellen und Konzepte

(Auswahl)



Modularisierung

- Nicht neu (1968: National Symposium on Modular Programming)
 - Trotzdem nur wenige Erfahrungsberichte
- Spezialisierungen
 - Information Hiding
 - Separation of Concerns
 - Objekt (i. S. von OO)
 - ...
- Übertragbarkeit der Erfahrungen
 - Ein Microservice ist konsequente Form der Modularisierung
 - Bisweilen findet sich sogar die (zweifelhafte) Empfehlung, durch Microservices die vollständige Unabhängigkeit der Module zu erzwingen

Modularisierungserfahrungen

Zerlegungsillusion

Illusion:

„Komplexität lässt sich durch richtige Zerlegung beseitigen“

- Trotz aller gegenteiliger Erfahrung nicht aussterbender Glaube
- Modularisierung erleichtert den Umgang mit Komplexität – beseitigt sie nicht
- Echte Komplexität lässt sich nicht beseitigen!
- Systeme haben Tendenz zu wachsender Integration / Komplexität

Beispiele

Auto: unabh. Lenkung, Antrieb, Bremse → ESP

Bahn: Lok+Wagen → ICE

Verkaufsportale: Co-Selling, Produktvorschläge

Modularisierungserfahrungen

Nichts ist umsonst - ein (seltener) Erfahrungsbericht

Modularity – Often Desired, but Rarely Achieved

Jens Knodel, Matthias Naab, Balthasar Weitzel

Fraunhofer IESE

Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

{jens.knodel, matthias.naab, balthasar.weitzel}@iese.fraunhofer.de

Abstract—“Everything should be modular” is an exalted goal stated by almost every architect – but is it really possible to achieve this goal? In this experience paper, we share our lessons learned across a number of restructuring projects that went modular. We discuss typical business motivations, restructuring efforts starting with good intentions, and reconstruction reality striking back. In retrospective, we analyze typical pitfalls to be circumvented. Examples illustrate our findings and support a truism too often ignored by architects: everything has its price, and more often than not, the price for modularity is a lot higher than initially estimated.

Modularisierungserfahrungen

Es muss passen!

- Erfahrung: Es gibt eine starke Korrelation zwischen der Organisationsstruktur und der Modularität der Produkte.
- Bekannt als **Gesetz von Conway** (Conway's Law, 1968):
 - *“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.”*
- Schlussfolgerungen
 - Beim Entwurf der Modulstruktur berücksichtigen
 - IT kann schlechte (verworrene) Organisationsstruktur nicht beheben
 - Ggf. mit Verbesserung der Orga-Struktur beginnen
 - Bei Umstrukturierungen den Einfluss auf die IT bedenken

Modularisierungserfahrungen

Bounded Context

Warnung!

- ***Bounded Context*** ist – streng genommen - eine Illusion, weil immer in globalen Kontext eingebunden
- In der Praxis als Konzept in vielen Fällen nützlich
- Gefahr
 - Abhängigkeiten vom globalen Kontext **nicht explizit**
 - Vollständige Beschreibung der globalen Voraussetzungen ist prinzipiell unmöglich
 - Überraschungen nicht auszuschließen
- Beispiele
 - Maßeinheiten (metrisch/Zoll, Celsius/Fahrenheit)
 - Wertebereiche (negative Zinsen)
 - Semantische Abhängigkeiten zwischen entfernten Modulen

Modularisierungserfahrungen

Zusammenfassung

- Zerlegung kann Komplexität nicht beseitigen
- Kritische Faktoren der Zerlegung
 - Granularität
 - Dimensionen
- Modularisierung hat ihren Preis
- Schnittstellendefinition ist **die** Kunst
- Modulstruktur ist nachträglich nicht leicht änderbar
- Modulstruktur muss zur Organisationsstruktur passen
- Implizite Annahmen minimieren

Verteilte Systeme

- Etabliertes Forschungs- und Anwendungsgebiet
- Microservice-Architekturen sind per Definition verteilte Systeme
- Erfahrungen daher übertragbar
- Neuere Entwicklungen wie
 - Container
 - Cloud-Services

vereinfachen die Anwendung, ändern aber nichts an den grundlegenden Herausforderungen und Problemen

- *Achtung:* Ein Teil der Komplexität der Anwendung wird dabei in die Struktur des Systems verlegt →
 - Weniger gut erkennbar
 - Unterschiedliche Ebenen (Beschreibung, Technik, Bearbeiter...)

Verteilte Systeme

Herausforderungen

- Generell schwierige Aufgabe
- Ermöglicht größere Flexibilität und Skalierbarkeit
- Verursacht (erhebliche) Kosten

At the end of the day, managing a single system is easier than a distributed one.

... however, microservices are not free; the trade-off for having flexibility and scalability is having to manage a complicated system.

http://docs.media.bitpipe.com/io_13x/io_134350/item_1433657/Microservices_for_Java_Developers.pdf

Verteilte Systeme

Zusätzliche Fehlerquellen

- Kommunikationsfehler
 - Erkennen
 - Behandeln

One lesson we learned is that large distributed systems are vulnerable to many types of failures, not just the standard network partitions and fail-stop failures assumed in many distributed protocols.

... causes: memory and network corruption, large clock skew, hung machines, extended and asymmetric network partitions, bugs in other systems....,

<http://research.google.com/archive/bigtable.html>

Verteilte Systeme

Weitere Konsequenzen

- Test
 - Schwieriger
 - Aufwendiger
- Sicherheit
 - Sicherheit kostet Zeit und Leistung
 - Viele Angriffspunkte
 - Laufende Ergänzung und Wartung (z. B. Zertifikate)
- Betrieb
 - Verantwortlichkeiten (DevOps?)

Bewertung für Microservice-Architektur

- Kosten-Nutzen-Vergleich schwierig
 - Kosten entstehen in jedem Fall
 - Nutzen stark von „weichen“ Faktoren abhängig (*Time to market,...*)

Trotzdem

- Kosten (Entwicklung + Betrieb) *realistisch* schätzen
 - Wirtschaftlichkeit plausibilisieren (1. Prüfung)
- Potentiellen Nutzen schätzen (optimal + worst-case)
 - Gegen Kosten vergleichen (2. Prüfung)
- Geschäftliche Entscheidung treffen

(Rudimentäre) Fragenliste für Microservices

- Diese Fragen sollten vor dem Einsatz beantwortet werden
- Aus eigenen Erfahrungen entsprechend ergänzen
- Im Prinzip für alle Technologien anwendbar

Grundfrage:

Welche konkreten Anforderungen lassen sich nur mit der Microservice-Architektur und nicht mit einem sauber modularisierten Monolithen erreichen?

Fragen: Organisationsstruktur

- Microservices bringen die Silostruktur zurück:
 - Lassen sich die Geschäftsprozesse in klar abgegrenzte Stränge/Geschäftsvorfälle zerlegen?
- Änderungen der Zerlegung sind kostspielig:
 - Ist die Struktur flexibel genug, um wahrscheinlichen Änderungen des geschäftlichen Umfelds ohne substantielle Umbauten der Service-Struktur entsprechen zu können?
- Schnittstellen definieren die Struktur:
 - Sind die Beziehungen zwischen den Services hinreichend exakt und vollständig definiert, sodass langwierige und konfliktträchtige Abstimmungsrunden vermieden werden können?
- Viele Vorteile erschließen sich erst auf längere Sicht:
 - Haben die einzelnen Stränge hinreichendes Entwicklungspotential, um über einen absehbaren Zeitraum eigenständig neue Features liefern zu können?

Fragen: Entwicklung

- Abhängigkeiten können auch entstehen, wenn Entwicklerteams mehr als einen Service entwickeln oder einzelne Personen in mehrere Teams integriert sind.
 - Sind ausreichende Entwicklungskapazitäten vorhanden, um wirklich parallel – und damit schneller – entwickeln zu können?
 - Der Aufbau und das Zusammenfinden agiler Teams kosten Zeit und Geld. Können die eingespielten Teams dann auch über einen wirtschaftlich sinnvollen Zeitraum hinweg ausgelastet werden?

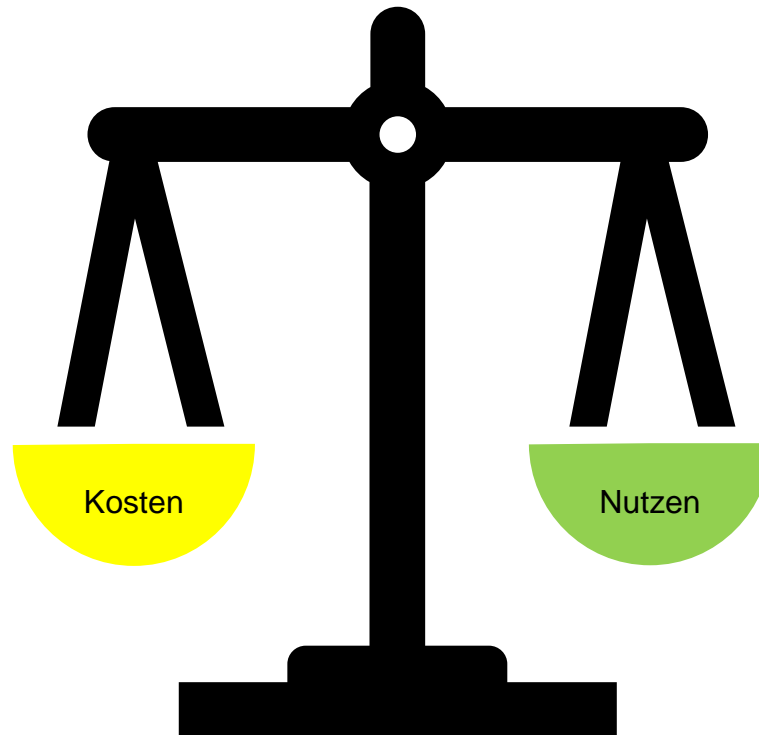
Fragen: Nutzen

- Kann realistischer Weise mit einem so starken Wachstum gerechnet werden (oder ist der Bedarf bereits so groß), dass sich die in die Entwicklung investierten Mehraufwände in überschaubarer Zeit amortisieren?
- Schwanken die Leistungsanforderungen tatsächlich so stark, dass der organisatorische Aufwand für das bedarfsgesteuerte Zu- und Abschalten von Rechnerinstanzen durch eingesparte Nutzungsentgelte gerechtfertigt werden kann?

Fragen: Betrieb

- Eine besonders tückische Kostenfalle lauert in den Verwaltungsfunktionen.
 - Sind beispielsweise die von Cloud-Anbietern standardmäßig bereitgestellten Verwaltungsfunktionen wirklich ausreichend?
 - Rechtfertigt der Umfang der voraussichtlichen Nutzung die gegebenenfalls anfallenden Anpassungsaufwände?

Zum Schluss: Entscheiden



Vielen Dank!



Gerne stehen wir Ihnen bei
weiteren Fragen zur Verfügung.
Sprechen Sie uns an.

Dr. Jürgen Lampe
Principal Consultant

Telefon +49 5251 1581-0
Telefax +49 5251 1581-71
juergen.lampe@agon-solutions.de

S&N Group AG
(zusätzlich Agon/S&N)
Klingenderstraße 5
33100 Paderborn
www.sn-invent.de