

TECHNOLOGY DAY / 05.12.2024

Produktlebenszyklus

Was hat er mit Architektur zu tun?



JÖRG MÜLLER



MARTINA FREERS

INNOQ

**Was ist die richtige
Architektur für mein Produkt?**

Es kommt darauf an!

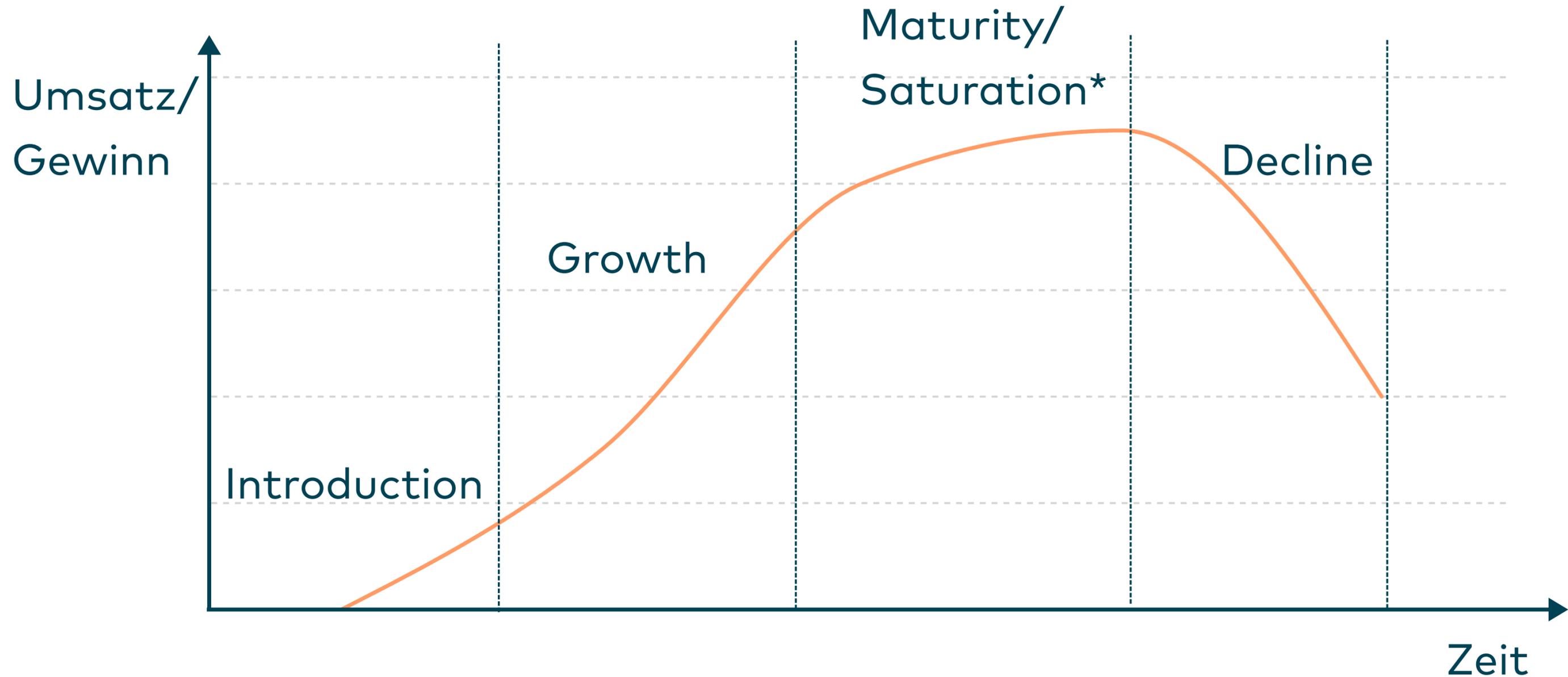
Aber worauf?

**Der Produktlebenszyklus
kann darauf eine Antwort sein**

1966

Raymond Vernon

Der Produktlebenszyklus



* Dies sind eigentlich zwei Phasen, für unsere Zwecke zusammengefasst

Introduction / Einführungsphase

- Umsatz niedrig, Gewinn meistens negativ
- Product / Market Fit noch nicht gut
- **Viele Experimente** / neue Features sind nötig
- **Geschwindigkeit** der Umsetzung zählt



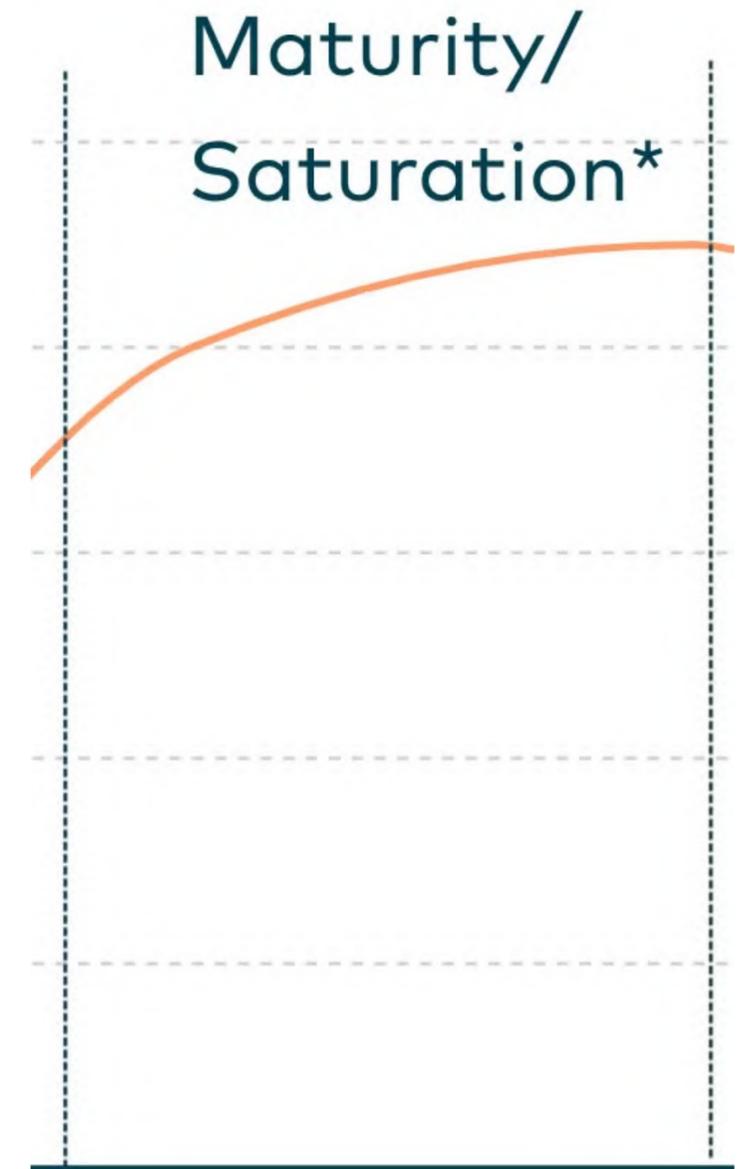
Growth / Wachstumsphase

- Umsatz stark steigend, Gewinn steigend
- Ziel **schnelles Wachstum** des selben Modells
- *Achtung:* Die meisten Produkte schaffen es nicht bis in diese Phase!
- Viele neue Kunden / Regionen unterstützen
- Aber auch weiter viele neue Features nötig



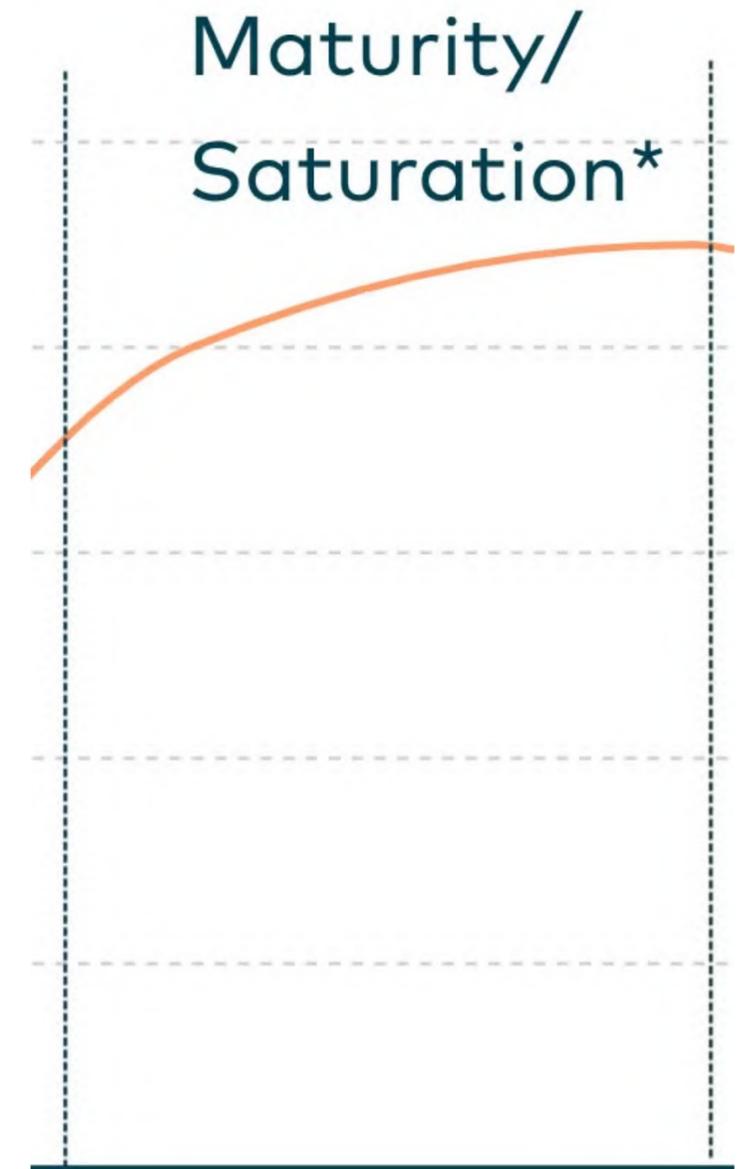
Maturity / Reifephase

- Umsatz schwach steigend, Gewinn relativ konstant
- **Stabilität** wird wichtiger
- Nicht mehr viele neue Features



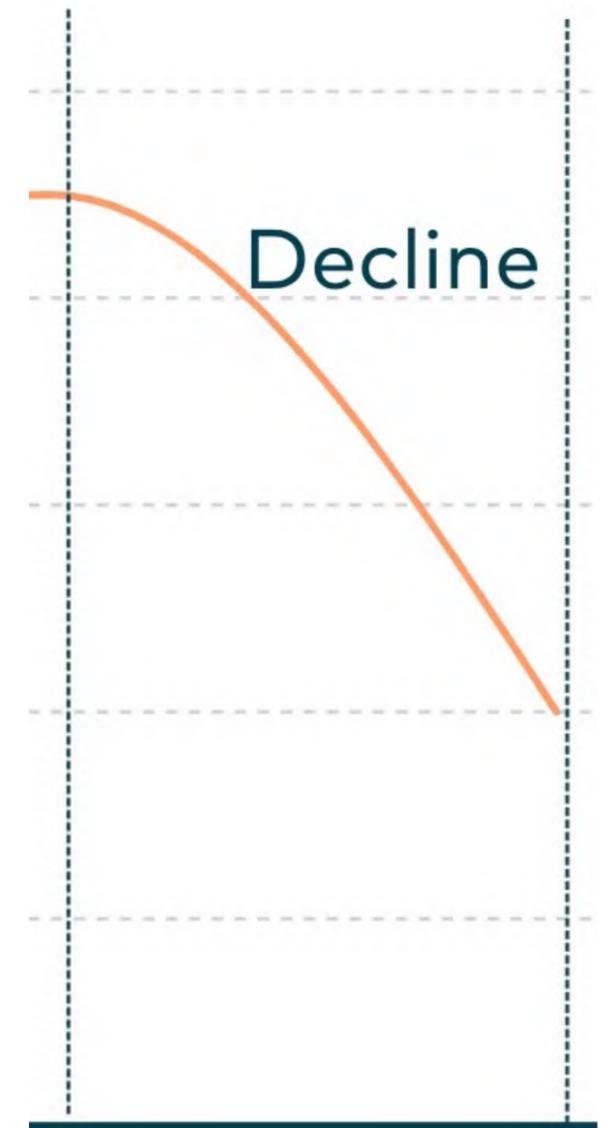
Saturation / Sättigungsphase

- Umsatz relativ konstant, Gewinn rückläufig
- Preisdruck durch Wettbewerb
- Hauptziel: **Kosten reduzieren!**
- Nur noch neue Features, wenn unbedingt notwendig



Decline / Degenerationsphase

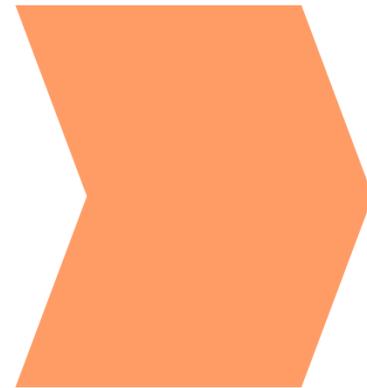
- Umsatz rückläufig, Gewinn stark rückläufig
- Weiter Kosten reduzieren
- Unterstützung des Übergangs in neue Generation
- Langsames **gezieltes Schrumpfen**



Software != Produkt

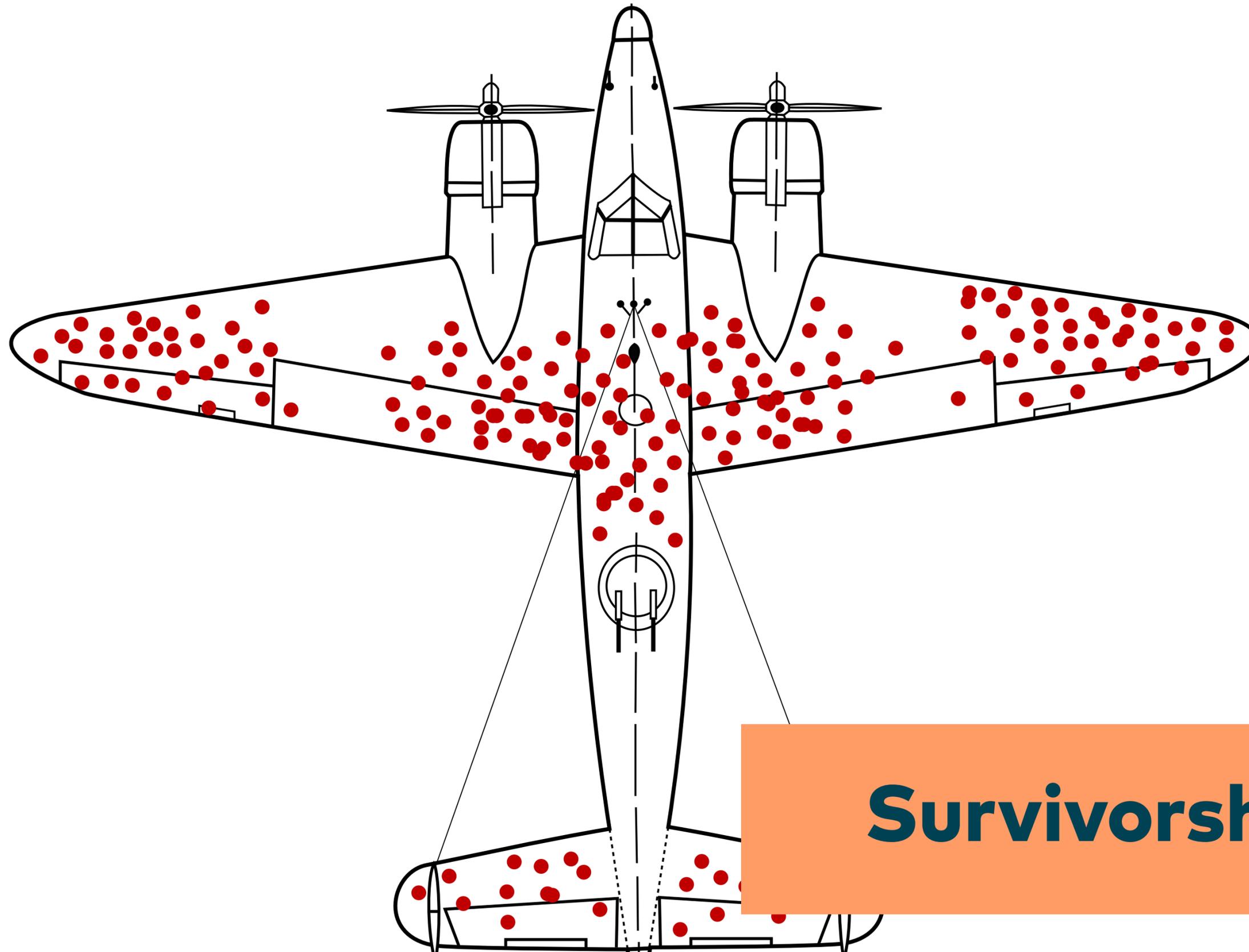
Phase 1 - Introduction

- Viele neue Ideen
- Kosten möglichst niedrig halten
- **Schnelle Experimente!**



- Ein oder sehr wenige Teams!
- "Just enough Architecture"
 - Einfache und vertraute Technologien
 - Monolithen
 - Wenig Infrastruktur

**Wie wichtig ist die
Wartbarkeit der SW in dieser
Phase?**



Survivorship Bias

Diskussion Monolithen

Pro

- Einfacher Start
- Geringe Latenzen innerhalb des Monolithen
- Einzelne Build und Deployment Unit
- Ressourceneffizient in kleinem Maßstab

Con

- Mehr Koordination, wenn Teams wachsen
- Modularität wird nicht so stark erzwungen
- Schlechte Horizontale Skalierung
- Single Point of Failure und Flaschenhals für Performance

Welche der Cons sind in dieser Phase relevant?

Minimal Viable Architecture

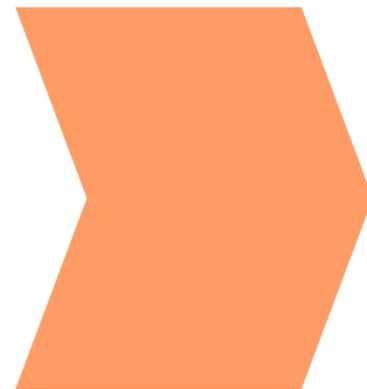
Randy Shoup

- Einige der Ideen in den vorherigen Slides sind direkt von dem Talk übernommen
- Für mehr: <https://www.youtube.com/watch?v=MjPoob7CYnY>



Phase 2 - Growth

- **Schnelle Skalierung** des gefundenen Modells
- Oft auf der berühmten Hockey Stick Kurve
- Viele neue Features nötig



- Horizontale Skalierung wird wichtig
- Es kommen viele Teams dazu
- Das Produkt wird komplexer

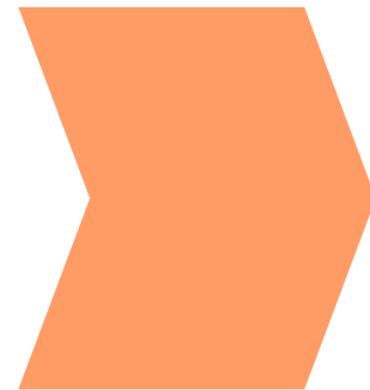
**Wie wichtig ist ein
einheitlicher Tech Stack?**

Autonomie und Wachstum

- Teams wählen die passende Technologie für Ihr Problem
- Wachstum benötigt hohe Autonomie
- Hohe Koordination → Viel Kommunikation → Langsames Wachstum
- Das Einstellen neuer Entwickler:innen wird erleichtert

Phase 3/4 - Maturity / Saturation

- Neue Features werden weniger
- Ausfälle sind extrem teuer, weil viele Nutzer auf dem System
- Ecken und Kanten sind wohlbekannt
- Es existiert Wettbewerb, der oft einen Preiskampf beginnt



- **Stabilität** des Systems hat sehr hohe Priorität
- Teams werden weniger und kleiner
- Standardisierung der Technologien
- Teile der Software durch Standardsoftware ersetzt

Cloud oder On-Premise?



DAVID HEINEMEIER HANSSON

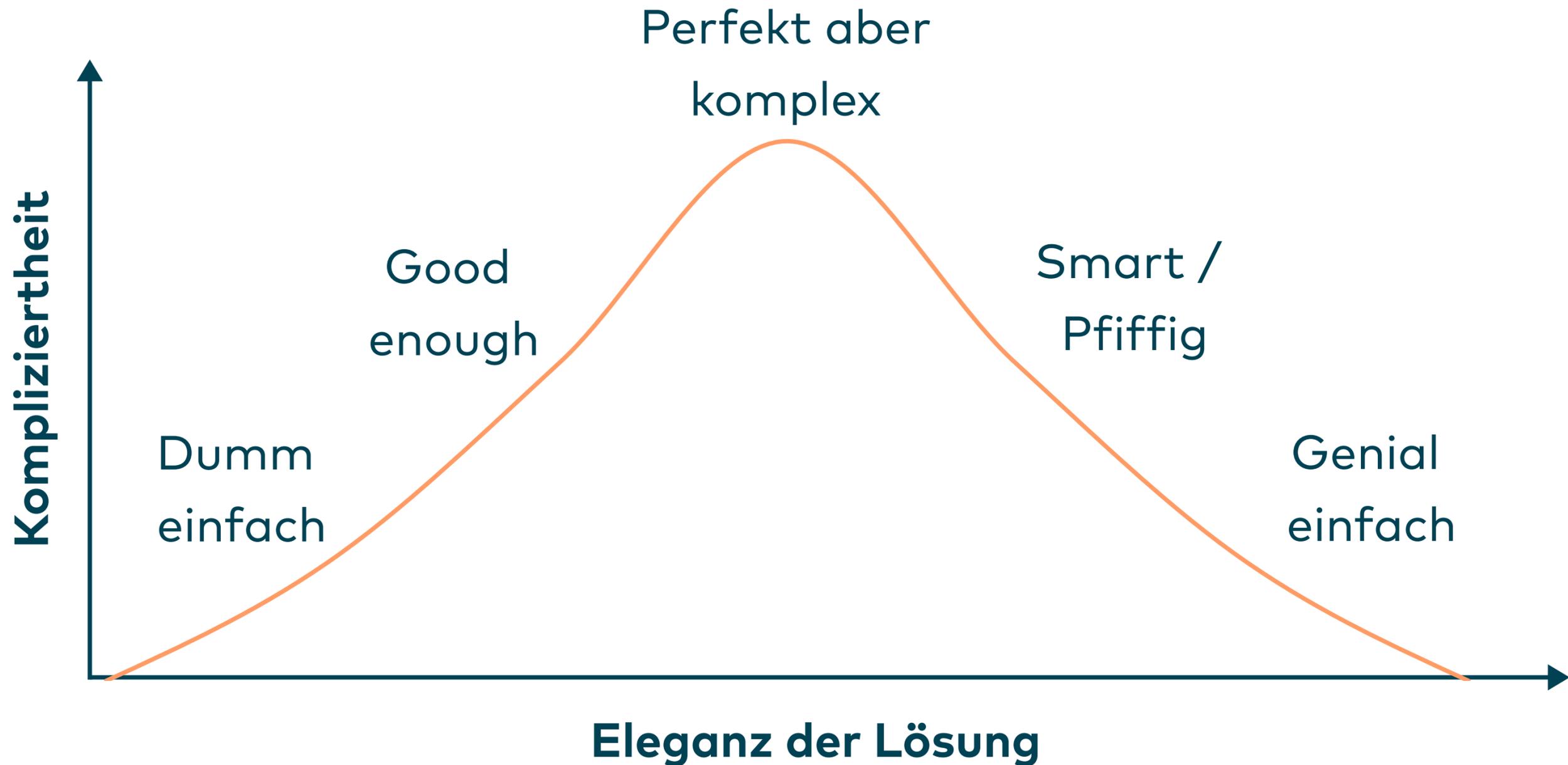
October 19, 2022

Why we're leaving the cloud

[Basecamp](#) has had one foot in the cloud for [well over a decade](#), and [HEY](#) has been running there exclusively since it was launched two years ago. We've run extensively in both Amazon's cloud and Google's cloud. We've run on bare virtual machines, we've run on Kubernetes. We've seen all the cloud has to offer, and tried most of it. It's finally time to conclude: Renting computers is (mostly) a bad deal for medium-sized companies like ours with stable growth. The savings promised in reduced complexity never materialized. So we're making our plans to leave.

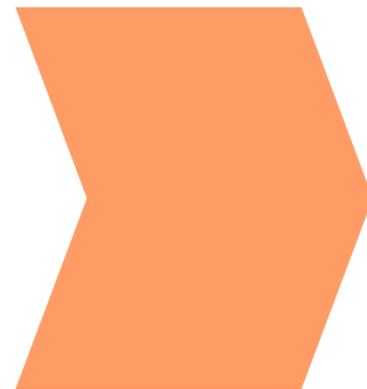
<https://world.hey.com/dhh/why-we-re-leaving-the-cloud-654b47e0>

Die Einfachheitskurve



Phase 4 - Decline

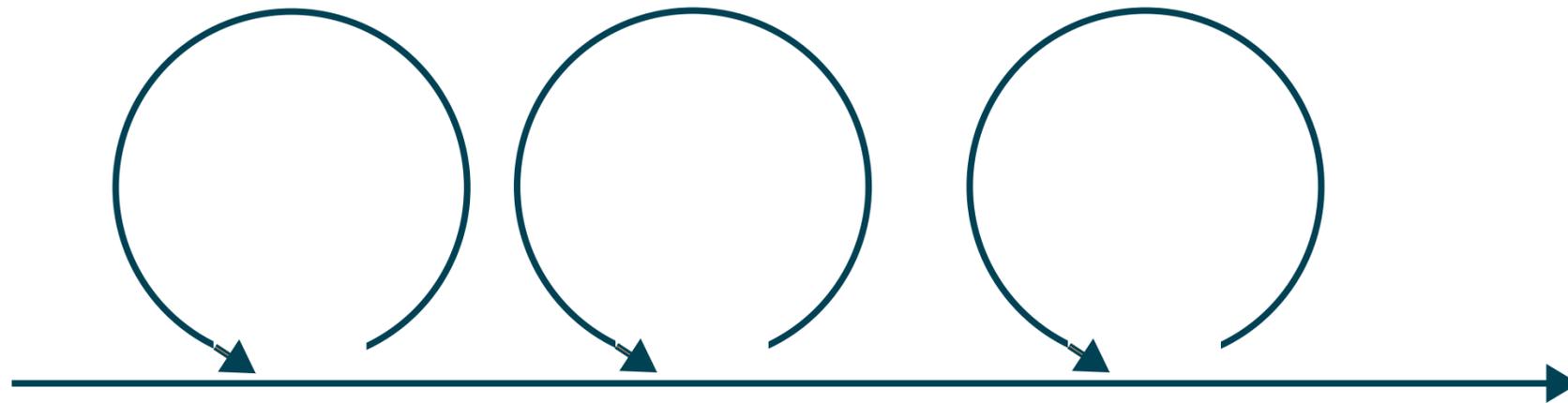
- Keine neuen Features mehr
- Nur noch "am Laufen" halten, Bug beheben, Regulatorien erfüllen
- Unregelmäßige Anforderungen
- Kosten müssen weiter reduziert werden
- Migration auf neues System vorbereiten



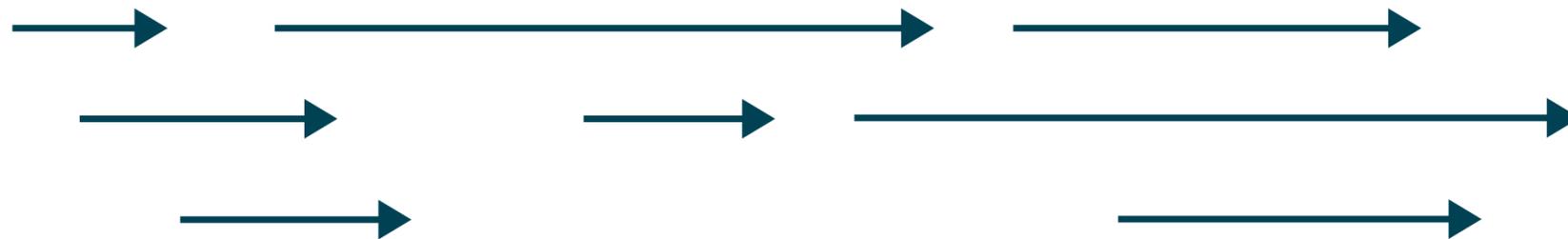
- Nur noch **kleines Team**
- Team eventuell nur Teilzeit
- **Seltene Releases**
- Gute Dokumentation nötig wegen langer Pausen
- Modernere Technologien / Systeme über Anti Corruption Layer anbinden

Welche Entwicklungsmethodik wählen?

Iterations vs. Flow

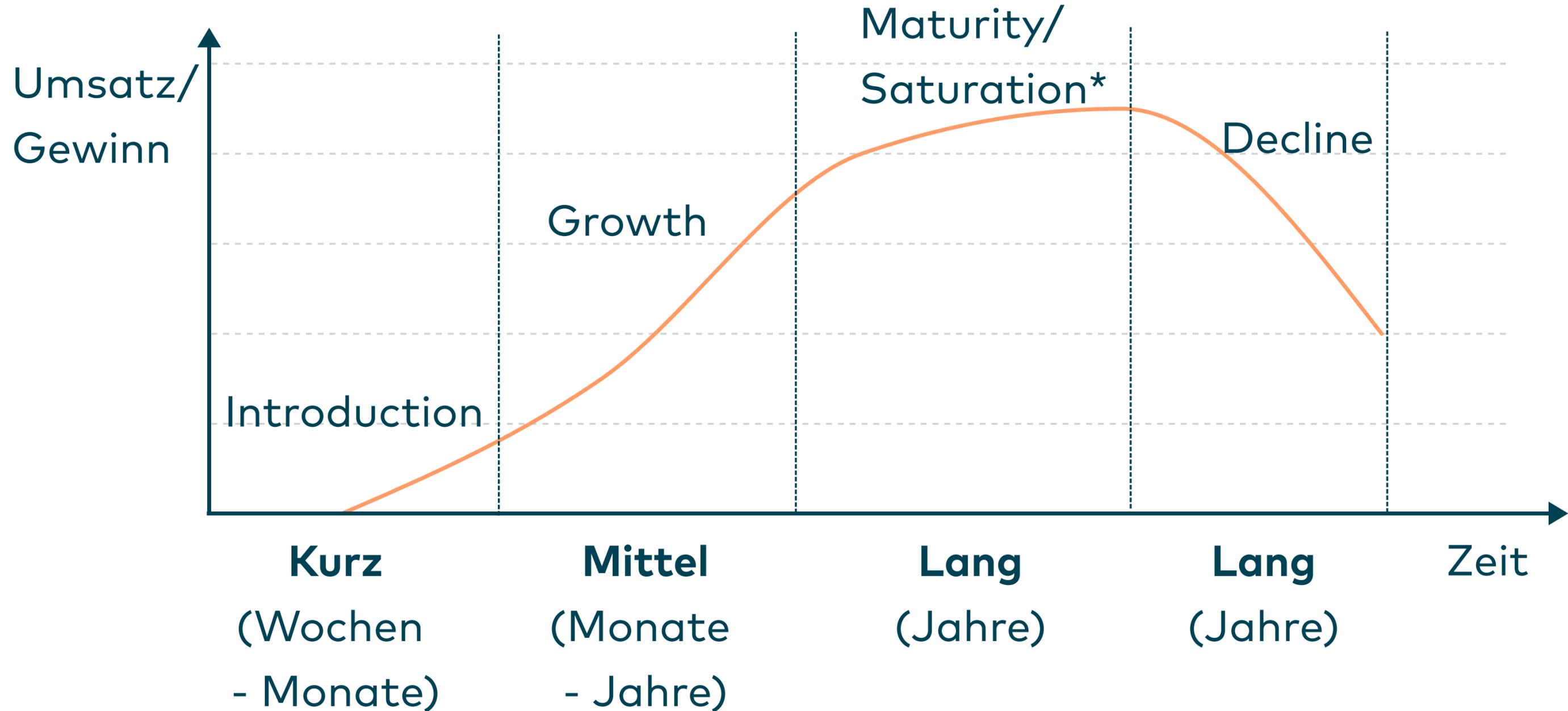


Kontinuierliche Entwicklung
in ähnlich großen
Iterationen



Entwicklung in
unterschiedlich großen
parallelen Paketen mit WIP
Limit

Ideale Dauer der Phasen



Sorry

No *Silver* Bullet

Viele andere Einflussfaktoren

- Internes oder Externes Produkt
- Regulatorik
- Unternehmenskultur
- SaaS oder beim Kunden installiert
- Reine Software oder Software / Hardware Kombination
- ...

Aber ...

**Den Produktlebenszyklus
nicht zu beachten ist leichtsinnig!**

Danke! Fragen?



Jörg Müller

joerg.mueller@innoq.com

+49 151 15676616

mastodon.social/@joergm

[@joergm.bsky.social](https://bsky.social/@joergm)

linkedin.com/in/joerg-m/



Martina Freers

martina.freers@innoq.com

+49 170 4551099

linkedin.com/in/martina-freers/

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 333660

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Wendenstr. 130
20537 Hamburg

Spichernstraße 44
50672 Köln