

# **VIRALE ENTERPRISE PRODUKTE**

*Das Ende der  
Rollout-Projekte*

<b>TYPISCHE ROLLOUT PROJEKTE</b> .....	4
<b>HERAUSFORDERUNGEN UND POTENTIALE BEI ROLLOUT PROJEKTEN</b> .....	5
Budgetgeber ≠ Benutzer .....	7
Fokus auf Benutzer & deren Feedback .....	7
Design ≠ Realität .....	10
Minimum Viable Product und inkrementelle Verbesserung .....	10
Ablehnung zentraler Systeme .....	11
Kollaborative Zusammenarbeit.....	12
<b>ZUSAMMENFASSUNG</b> .....	15

# EINLEITUNG

International agierende Unternehmen stehen immer wieder vor der Herausforderung, interne Software regional, für unterschiedliche Marken oder Märkte oder gar global auszurollen. Dieser Prozess, der als Rollout bezeichnet wird, gelingt selbst in Unternehmen, die damit Erfahrung haben, nicht immer optimal.

Eine effektive Alternative zum klassischen Rollout von IT-Systemen und einer Push-Strategie für die Skalierung eines neuen Systems sind so genannte virale, digitale Enterprise-Produkte. Dabei impliziert schon die Änderung der Bezeichnung von IT-System hin zum digitalen Produkt ein anderes Verständnis und Vorgehen. Virale, digitale Produkte sind vergleichbar mit viralen Videos in sozialen Netzwerken. Sie erzeugen in einem Unternehmen einen Nachfragesog oder "Pull-Effekt", der das typische Rollout-Projekt obsolet macht. In diesem Artikel beschreiben wir Ansätze, um virale Enterprise Produkte zu kreieren.

## **TYPISCHE ROLLOUT-PROJEKTE:**

Wir alle kennen digitale Produkte, wie Apps oder allgemeine Services auf Webseiten, die uns das Leben einfacher machen. Sie stellen das Kerngeschäft vieler Unternehmen dar. Neben diesen digitalen B2C-Produkten gibt es einen weiteren Produkttyp: Digitale Enterprise Produkte. Diese werden nur innerhalb eines Unternehmens von Angestellten genutzt und helfen den Mitarbeitern bei ihren alltäglichen Arbeitsprozessen.

Für solche Produkte findet man in großen Unternehmen oft die besagten Rollout -Projekte. Ihr Ziel: Die zentral entwickelten Enterprise-Produkte bei neuen Benutzergruppen in verschiedenen Abteilungen oder Regionen reibungslos einführen. Letztendlich sollen damit die erwarteten Vorteile, die durch die Nutzung der digitalen Produkte entstehen (z. B. durch Standardisierung von Geschäftsprozessen oder Skaleneffekte), realisiert werden.

Für ein erfolgreiches Rollout-Projekt müssen folgende Punkte adressiert werden:

- Die Software muss für die neuen Benutzergruppen technisch verfügbar und nutzbar sein – dies geht manchmal mit neuer IT-Infrastruktur einher

- Die Software muss ggf. an lokale Anforderungen und Gegebenheiten angepasst werden – das können Währungen oder gesetzliche Vorgaben sein
- Die Unternehmen und Mitarbeiter müssen ggf. neue Geschäftsprozesse und Abläufe einführen und diesen folgen
- Die Unternehmen und Mitarbeiter müssen die Software nutzen und dazu ggf. den Umgang damit erlernen, soweit dieser nicht intuitiv ist

## **HERAUSFORDERUNGEN UND POTENZIALE BEI ROLLOUT -PROJEKTEN**

Bei digitalen B2C-Produkten gibt es eine Reihe von Möglichkeiten, diese schnell, ohne viel Aufwand und mit Hilfe von direktem Kundenfeedback auf den Markt zu bringen bzw. erst einmal zu erfahren, was die Kunden wirklich wollen. Die dazu etablierten Methoden und Werkzeuge werden unter der Disziplin Produktmanagement zusammengefasst.

Bei internen Enterprise-Produkten hat sich dieses Verständnis noch nicht konsequent etabliert. Oft bezeichnen und betrachten

Unternehmen ihre intern genutzte Software als System und nicht als Produkt. Diese Sicht hat oft folgende Konsequenzen:

- Budgetgeber und Benutzer sind unterschiedliche Gruppen
- Anwendung der Software ist in der Realität anders als designed
- Ablehnung der Benutzer gegenüber zentral entwickelter Software

Die Disziplin des Produktmanagements und der agilen Softwareentwicklung zeigen Ansätze, um diesen Herausforderungen zu begegnen und den Erfolg des Unternehmens durch digitale Enterprise-Produkte zu steigern. Im Folgenden beschreiben wir jeweils zunächst die Herausforderung und im Anschluss daran Ansätze zur Verbesserung und Entwicklung eigener viraler Enterprise-Produkte. Dafür haben wir jeder genannten Herausforderung einen Lösungsansatz zugeordnet:

Herausforderung	Verbesserungspotenzial
Budgetgeber ≠ Benutzer	Fokus auf Benutzer und deren Feedback
Design ≠ Realität	MVP und Inkremente
Ablehnung zentraler Systeme	Kollaborative Zusammenarbeit

*Tabelle 1: Übersicht Herausforderungen und Verbesserungspotenzial*

### **Budgetgeber ≠ Benutzer**

Die Organisationsstruktur von großen Unternehmen bedingt, dass es oft separate Budgetgeber für Rollout-Projekte gibt. Dies sind mitunter nicht die Personen, die später das digitale Produkt nutzen. Außerdem haben diese häufig kein konkretes Bild von den Arbeitsabläufen und Anforderungen z. B. durch örtliche Distanz.

Die Kultur großer Unternehmen führt ebenfalls dazu, dass die Budgetgeber viele Entscheidungen in Bezug auf die Features digitaler Produkte treffen. Sie tun dies – bestimmt im besten Interesse – auf Basis von Theorien und beeinflussen die Durchführung der Projekte damit stark. Da Theorie und Praxis oft auseinanderlaufen, birgt dieses Vorgehen ein Risiko: Aufgrund der fehlenden Nähe zu den konkreten Abläufen, trifft der Budgetgeber falsche oder mangelhafte Entscheidungen. Im angelsächsischen wird dies als HIPPO = “Highest Paid Person’s Opinion” bezeichnet.

Die Herangehensweise, ohne konkretes User Feedback Entscheidungen zu treffen und eine Software zu entwickeln, ist ein Indiz dafür, dass die Produkt-Denke für unternehmensinterne Vorgänge noch fremd ist.

### **Fokus auf Benutzer und deren Feedback**

Modernes Produktmanagement sieht vor, die Entscheidungen ausschließlich auf Basis von echtem Feedback der Endbenutzer zu treffen. Was sich logisch anhört, ist in der Realität zunächst

ungewohnt: Es Bedarf besonderer Schritte, um an Feedback zu kommen und dies für Entscheidungen zu nutzen.

Wir erzeugen Feedback immer durch direkte Interaktion mit den späteren Benutzern. Dies geschieht in unterschiedlichen Formen und in Abhängigkeit von der Reife des digitalen Produktes – hier einige Beispiele.

*Anfangsphase – keine oder kaum realisierte Funktionen verfügbar:*

- Beobachtung und Befragungen von End-Nutzern machen den aktuellen Prozess verständlich und ermöglichen es, Verbesserungspotenziale zu entdecken und zu diskutieren. Schließlich gilt es nicht nur, den Status Quo eines neuen Produkts abzubilden, sondern alle technischen Möglichkeiten zur Unterstützung und Verbesserung auszunutzen (z. B. durch Machine Learning oder intelligente Nutzung von Geschäftsdaten).
- **Low-Fi Prototypen** ermöglichen die Diskussion von Skizzen der Screens und dem Flow der Anwendungen mit zukünftigen Benutzern bevor diese implementiert werden. Damit bekommt man ohne Programmierarbeit bereits ein erstes Feedback zur Gestaltung, zur Nutzbarkeit und zur Abbildung der Geschäftsprozesse.



*Reifephase – einige oder viele Funktionen verfügbar:*

- Alle Methoden aus der Anfangsphase sind noch immer valide und nützlich.
- Das Vorgehen der [Hypothesen getriebenen Entwicklung](#) beschreibt für jede neue Funktion ein Experiment, das Ursache und Wirkung in Korrelation setzt und nach einem Beweis für den Nutzen der neuen Funktion verlangt.
- A/B Testing ist eine typische Vorgehensweise, um die Wirkung einer neuen Funktion zu belegen.
- Monitoring im Allgemeinen zeigt, wie oft Funktionen genutzt werden, wo ggf. der Fluss in der Anwendung stockt. Außerdem hilft es, den tatsächlichen Geschäftswert von Funktionen zu messen, zu artikulieren und Trends abzuleiten.

Durch Nutzer-Feedback erhält man eine verlässliche Basis für korrekte Entscheidungen. Wenn die Budgetgeber nun den Teams die Entscheidungen überlassen, die für die jeweiligen Produkte verantwortlich sind, dann ist das Risiko schlechter Entscheidungen durch HIPPOS in Bezug auf Enterprise-Produkte eliminiert.

## **Design ≠ Realität**

Man hört immer wieder von Softwareprojekten, bei denen die implementierten Funktionen am tatsächlichen Bedarf vorbei gehen. Die Ursachen dafür sind vielfältig:

- Falsch getroffene Entscheidungen (siehe oben)
- Lange Zeiträume zwischen Spezifikation und Nutzung – die Welt dreht sich weiter und Features werden obsolet (Up-Front Design)
- Komplizierte und detailverliebte Spezifikation – viele goldene Wasserhähne und wenig echter Nutzen
- Spezifikation ohne tiefes Technologie-Knowhow – Features nutzen nicht das volle Potenzial der neuesten Technologie
- Schlechte Priorisierung – Features mit großem Wert werden erst spät realisiert

Alle genannten Punkte bergen das Risiko, dass ein suboptimales Produkt gebaut wird, welches in vielerlei Hinsicht mehr Geld kostet, mit unnötigem Aufwand verbunden ist und die Arbeitszeit des Entwicklungsteams bindet.

## **Minimum Viable Product und inkrementelle Verbesserung**

Um den Herausforderungen von Design ≠ Realität geschickt zu begegnen, gilt es zunächst in möglichst kurzer Zeit ein erstes

Produkt zu erstellen und es in Produktion zu nehmen. Mit dem Ziel, den Nutzen des Produktes im Einsatz zu beweisen und die avisierte Anwendung zu bestätigen.

Dieses erste Produkt wird Minimum Viable Product (MVP) genannt. Als Daumenregel gilt, dass ein MVP in einem Enterprise-Umfeld nach drei bis sechs Monaten verfügbar sein sollte. Ansonsten dauert es zu lange und die Wahrscheinlichkeit wird größer, dass eines der genannten Risiken eintritt.

In vielen weiteren Schritten und Iterationen werden neue Funktionen hinzugefügt, die bestehenden erweitert oder neue Benutzergruppen (z. B. Regionen oder Abteilungen) freigeschaltet. Dabei ist der Spielraum jeder Iteration so klein und die Releases der Inkremente sind so hochfrequent, dass es kaum Abweichungen zwischen Design und Realität geben kann.

Bei jeder Iteration spürt man die Konflikte zwischen verfügbarer Entwicklungskapazität oder dem Budget sowie der nächsten wichtigen Funktion, die realisiert werden soll. Häufig gibt es zu wenig Budget und zu viele wichtige Funktionen. Wie man dennoch geschickt priorisiert, Werte fokussiert und gute Budgetentscheidungen trifft, wird in dem Buch [Lean Enterprises](#) beschrieben.

### **Ablehnung zentraler Systeme**

Oft werden in großen Unternehmen neue Systeme angekündigt und vieles versprochen. Am Ende gehen jedoch Planung und

Realität, wie gerade beschrieben, auseinander. Statt bei der Arbeit zu helfen und die Produktivität zu steigern, fehlen wichtige Funktionen, die Bedienung ist nicht intuitiv und die neue Software wird als umständlich empfunden.

Umständlich aufgebaute Software bedarf einer längeren Einführungszeit, die Prozessharmonisierung dauert länger, die Nutzer sind unzufrieden und Motivation und Produktivität sinken. All dies wirkt sich auf den Erfolg der Unternehmung aus. Denn mit einem Produkt, das nicht funktioniert wie geplant, können auch festgelegte Geschäftsziele nicht erreicht werden. Und am Ende rechnet sich der Business Case für das neue System gar nicht.

In manchen Unternehmen ist dieser Ablauf zur Regel geworden und hat über die Zeit zu Vorsicht und Skepsis bei den Nutzern von Enterprise-Produkten geführt. Misstrauen und Ablehnung sind die Standardreaktion bei der Ankündigung neuer Systeme.

### **Kollaborative Zusammenarbeit**

Wir haben gesehen, dass Benutzergruppen in Unternehmen zunächst einige positive Erfahrungen machen müssen, um wieder Vertrauen in "die IT" zu fassen und Spaß an der gemeinsamen Entwicklung von digitalen Enterprise-Produkten zu finden.

Dazu etablieren wir stets ein Produktteam, das die Aufgabe hat, das digitale Enterprise-Produkt zu realisieren und

erfolgreich zu machen. Das [Produktteam](#) besteht sowohl aus technischen Experten als auch aus Vertretern der Benutzer, der Fachbereiche und anderen Stakeholdern.

Positive Erfahrungen, die schnell innerhalb eines Produktteams umsetzbar sind, sind z. B.:

- Transparenz und offene Kommunikation
- Kollaboration und direkt Umsetzung von Feedback in der Software
- Berücksichtigung von Wünschen und Prioritäten bei der Realisierung

Um Missverständnissen vorzubeugen, kommunizieren alle Mitglieder im Team regelmäßig über den Stand der Entwicklung sowie mögliche Änderungen und Herausforderungen. In wöchentlichen Showcases zwischen den Technologen, den Benutzern und anderen Stakeholdern zeigt und diskutiert das Team die Fortschritte anhand der laufenden Software.

Bei den Showcases sind zwei Erfolgsfaktoren besonders wichtig:

1. Die Darstellung des aktuellen Status muss anhand der laufenden Software erfolgen. Nur so ist der Eindruck konkret und relevant.
2. Der Rhythmus der Showcases muss hochfrequent sein (wöchentlich). Nur so können die Benutzer sofort Feedback geben, spüren die Transparenz und sind in die tägliche Arbeit eingebunden.

Mit Showcases wird das Risiko einer nachträglichen Enttäuschung eliminiert, denn alle Parteien besprechen Änderungen, Herausforderung und mögliche Lösungen gemeinsam.

Der enge Kontakt in den Produktteams, die auch interdisziplinäre oder cross-funktionale Teams genannt werden, ermöglicht notwendige Änderungen an bestehenden Funktionen oder nützliche, neue Funktionen schnell und direkt zu artikulieren und umzusetzen. Dieser direkte Einfluss auf das digitale Produkt ist für die meisten Benutzer neu. Die Erfahrung, dass das eigene Feedback innerhalb von Tagen oder Wochen umgesetzt wird, ist sehr mächtig und schafft auf Seiten der Anwender viel Vertrauen in das Produkt und das Team.

Die Benutzer haben Einfluss bei der wöchentlichen Sprintplanung genauso wie bei der Gestaltung der Roadmap und bei der Abstimmung der weiteren Prioritäten z. B. für das nächste Quartal. Die Planung wird jeweils an der Priorität und dem Wert der Funktionen für die Endnutzer und das Business ausgerichtet. So vermeidet das Team gemeinsam, dass ein distanzierter Business Case ggf. nicht erfüllt wird und stellt sicher, dass jede Funktion relevant ist und Anwendung findet.

## ZUSAMMENFASSUNG

Die IT eines Unternehmens ist wie ein eigener kleiner Markt. Die eingesetzten Systeme sind digitale Enterprise-Produkte, deren Funktionen von den Nutzern bestimmt werden sollten. Erfolgreiche Enterprise-Produkte werden intensiv genutzt, verbreiten sich im Unternehmen viral und unterstützen so mit kleinem Investment den Geschäftsnutzen.

So eliminiert man Risiken aus typischen, internen IT-Projekten, die sich durch schlechte Business Cases, lange Vorlaufzeiten und Entscheidungen sowie Spezifikationen weitab von der Realität ergeben.

Die Disziplin, um digitale Enterprise-Produkte zu entwickeln, erfolgreich zu machen und zu skalieren, ist das Produktmanagement. Und sie wird immer mehr in der Unternehmens-IT angewendet. Das macht sowohl für digitale Produkte als auch für [Plattformen](#) und [APIs](#) Sinn.

# ThoughtWorks®

ThoughtWorks ist eine globale Softwareberatung,  
die führenden Unternehmen hilft, neue  
Absatzmärkte durch Technologie zu nutzen.

[\*info-de@thoughtworks.com\*](mailto:info-de@thoughtworks.com)

[\*thoughtworks.com\*](https://www.thoughtworks.com)