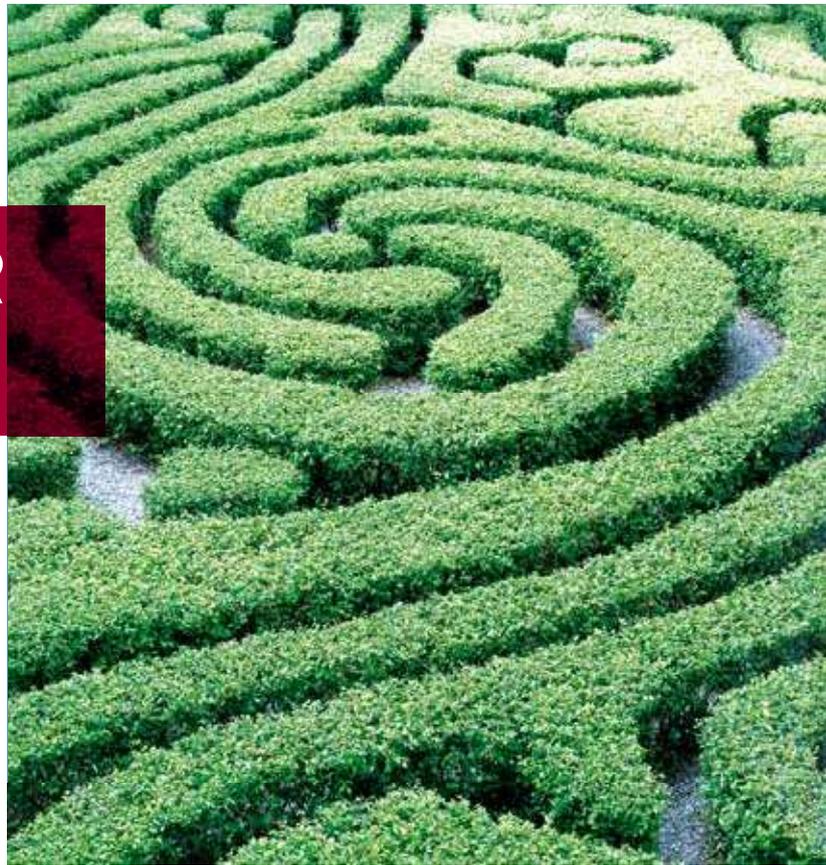


# KOMPLEX ODER KOMPLIZIERT?



## Wie man mithilfe von Wissensmanagement das richtige Projektvorgehen findet

| von UWE KOBLITZ

Softwareentwicklungsprojekte finden oft in komplexen beziehungsweise komplizierten Umgebungen statt. Wobei sich Komplexität oder Kompliziertheit aus der Vielzahl der beteiligten Elemente ergeben, bei denen auch kleine Änderungen große Auswirkungen haben können – beispielsweise aus organisatorischen, technischen oder regulatorischen Gründen. Dieser Artikel beschreibt ein Rahmenwerk zur Entscheidungsfindung und seine Anwendungsmöglichkeiten für ein Großprojekt. Dafür wird eine Brücke vom Wissensmanagement in die Softwareentwicklung geschlagen.

### DAS RAHMENWERK ZUR ENTSCHEIDUNGSFINDUNG

Im Wissensmanagement werden Modelle verwendet, um Probleme, Situationen und Systeme sowie deren Beziehungen untereinander zu beschreiben. Eines dieser Modelle ist das Cynefin-Rahmenwerk von Prof. Dave Snowden<sup>1</sup>, das sich auch bei Softwareentwicklungsprojekten bewährt hat. Es dient der Entscheidungsfindung und der Bestimmung des richtigen Vorgehens.

Das Cynefin-Rahmenwerk hat fünf sogenannte „Problem-Domänen“:

**Einfach (Obvious/Simple):** Die Domäne der bewährten Verfahren. Die Beziehung zwischen Ursache und Wirkung ist für alle offensichtlich. Die Herangehensweise ist Erkennen – Kategorisieren – Reagieren (Sense – Categorise – Respond), es können

bewährte Verfahren (Best Practices) angewendet werden, wie zum Beispiel in der Rechnungsabwicklung oder im Mahnwesen.

**Kompliziert (Complicated):** Die Domäne der Experten. Die Ermittlung der Beziehung zwischen Ursache und Wirkung erfordert die Anwendung von Fachwissen. Die Herangehensweise ist Erkennen – Analysieren – Reagieren (Sense – Analyze – Respond), es können verschiedene gute Verfahren (Good Practices) angewendet werden, deren Auswahl durch Experten getroffen wird. Beispiele sind Brückenbau, Fahrzeugbau oder Softwareentwicklung.

**Komplex (Complex):** Die Domäne der Emergenz. Die Beziehung zwischen Ursache und Wirkung kann nur im Nachhinein und nicht im Voraus wahrgenommen werden. Der Ansatz ist Probieren – Erkennen – Reagieren (Probe – Sense – Respond), es kommen emergente (sich herausbildende) Verfahren (Emergent Practices) zum Einsatz. Beispiele sind Kindergeburtstage oder die Entwicklung einer neuen Benutzeroberfläche.

**Chaotisch (Chaotic):** Die Domäne der schnellen Antwort. Es gibt keine Beziehung zwischen Ursache und Wirkung auf Systemebene. Der Ansatz ist Handeln – Erkennen – Reagieren (Act – Sense – Respond), und es entstehen innovative Verfahren, zum Beispiel Katastropheneinsätze wie bei der Sturmflut 1962 an der deutschen Nordseeküste oder bei den Angriffen auf das World Trade Center 2001.

<sup>1</sup> Vgl. <https://de.wikipedia.org/wiki/Cynefin-Framework> und David J. Snowden & Mary E. Boone: A Leader's Framework for Decision Making. In: Harvard Business Review. November 2007, S. 69–76

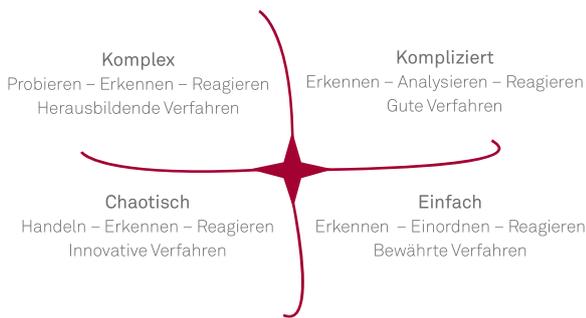


Abbildung 1: Das Cynefin-Rahmenwerk

**Unklar (Disorder):** Die Domäne des Nichtwissens. Verschiedene Perspektiven stehen in Konkurrenz, es gibt Streit um die richtige Ausrichtung. Die Lösung liegt im Herunterbrechen der Situation in einzelne Teile und deren Zuordnung zu den vier anderen Domänen.

Manager oder Projektleiter müssen sich auf die unterschiedlichen Anforderungen bezüglich Vorgehen und Entscheidungsfindung einstellen. Das Vorgehen, das für eine Domäne gute Ergebnisse liefert, wird vermutlich in einer anderen Domäne zu Fehlschlägen führen. Ein Beispiel dafür ist das entschiedene Eingreifen von Helmut Schmidt als Krisenmanager bei der Sturmflut 1962 an der deutschen Nordseeküste, bei der er sich seinen Ruf als „Macher“ erwarb. In einer chaotischen Situation wie damals war schnelles Handeln die beste Reaktion. Dieses Vorgehen wäre ihm in einer anderen Frage, zum Beispiel dem NATO-Doppelbeschluss (einer komplexen Entscheidungssituation), zum Verhängnis geworden.

### AUSWIRKUNG AUF DIE SOFTWAREENTWICKLUNG

Cynefin bietet die Möglichkeit, anstehende Aufgaben, Probleme und Herausforderungen gewissen Domänen zuzuordnen – es ist jedoch kein Vorgehensmodell für die Softwareentwicklung. Nicht alle Methoden der Softwareentwicklung sind für jede Domäne gleich gut geeignet. Im Folgenden wird beschrieben, welche Anforderungen sich in jeder Problem-Domäne stellen und welche Vorgehensweise für diese Domäne geeignet ist.

### KOMPLEXE PROJEKTE

Im Gegensatz zur komplizierten oder zur einfachen Domäne, die beide ein geordnetes System darstellen, können in der komplexen Domäne keine Ursache-Wirkungs-Beziehungen festgestellt werden. Sie können lediglich im Nachhinein erklärt werden (Emergent Behaviour). Anforderungen befinden sich im Fluss und sind nicht vorhersehbar. Es gibt keine richtigen Antworten – im Laufe der Entwicklung bilden sich lediglich bestimmte Muster heraus.

Man befindet sich im Bereich des unbekanntes Unbekanntes (Unknown Unknown) und muss mit vielen konkurrierenden Ideen zurecht kommen. Ein Fortschritt kann nur durch kreative und innovative Ansätze erreicht werden. Führung muss sich deshalb daran ausrichten, kreative und innovative Ansätze zu fördern und neue „Muster“ entstehen zu lassen.

In dieser Situation besteht die Führungsaufgabe darin, eine ideenfördernde Umgebung zu schaffen, in der unterschiedliche Meinungen entstehen und diskutiert werden können. Entscheidend dafür ist ein hohes Maß an Interaktion und Kommunikation zwischen den Beteiligten, die sich idealerweise auch an einem Standort befinden. Ziel ist es, durch Experimente neue Muster zu finden. Eine weitere Führungsaufgabe ist, eine Gruppe von Experten zusammenzubringen, um das zu entwickelnde Gesamtsystem zu analysieren.

Gefahrensignale sind:

- das Zurückfallen in den gewohnten Kommando- und Kontrollmodus,
- die Versuchung, nach Fakten anstatt nach entstehenden Mustern zu suchen,
- der Wunsch nach beschleunigter Problemlösung oder Nutzung von kurzfristigen Vorteilen.

Mögliche Antworten auf diese Gefahrensignale sind:

- Geduld aufbringen und auch Fehlschläge verkraften (die Suche nach Innovation wird immer wieder in Sackgassen führen, was aber besser als ein zu defensives Vorgehen ist),
- Zeit für Reflexion einräumen,
- Stakeholder beim Review von Ergebnissen einbeziehen, um (komplexe) Muster besser zu erkennen und durch regelmäßiges Feedback Verbesserungen schnell einarbeiten zu können.

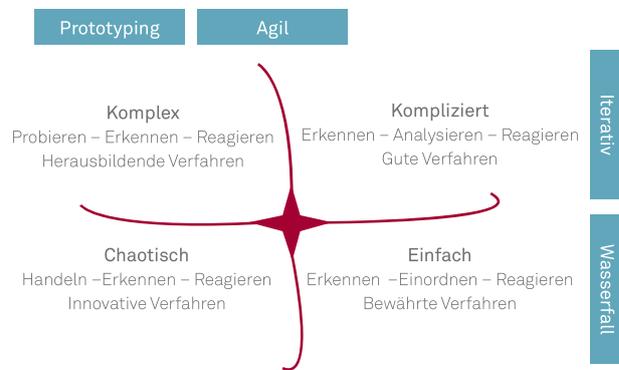


Abbildung 2: Geeignete Vorgehensweisen in den unterschiedlichen Domänen

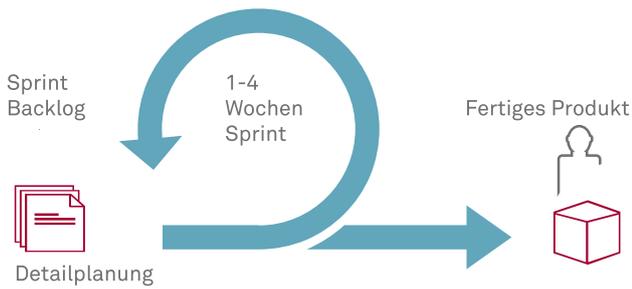


Abbildung 3: Agiles Vorgehen mit Sprints

Sind die Anforderungen noch sehr unbestimmt, dann sind Prototyping oder andere präagile Verfahren geeignet, um mehr Klarheit zu schaffen. Liegen konkrete Anforderungen vor, bieten sich agile Verfahren wie Scrum an, die mit Sprints von ein bis vier Wochen Dauer einsatzfähige Produkte als Grundlage für Rückmeldungen schaffen.

Ein Beispiel für komplexe Projekte ist die Entwicklung neuer Bedienoberflächen. Oft ist nicht vorhersehbar, welche Oberflächen Akzeptanz gewinnen. Hier hilft es, den Benutzern einen Prototyp in die Hand zu geben und zu sehen, wie sie damit zurechtkommen. Die Ergebnisse der Bewertung können dann schnell in eine verbesserte Version des Prototyps einfließen. So kann beispielsweise die Einführung eines neuen Fahrzeugrechners für Verkehrsbusse durch Aufstellen eines Fahrzeugrechnerprototyps im Aufenthaltsraum der Busfahrer beschleunigt werden. Die Fahrer können den Rechner ausprobieren, gewöhnen sich an das neue Gerät und liefern gleichzeitig wichtiges Feedback für eine Verbesserung der Benutzeroberfläche.

## KOMPLIZIERTE PROJEKTE

In der komplizierten Domäne hängt vieles von der Diagnose der Experten ab. Sie analysieren die Ursache-Wirkungs-Beziehungen, die – anders als in der einfachen Domäne – für Außenstehende in der Regel nicht ersichtlich sind. Oft ist mehr als eine richtige Antwort möglich und somit eine Auswahl des für das Projekt am besten geeigneten Vorgehens notwendig. Man bewegt sich im Umfeld des bekannten Unbekannten (Known Unknown), das durch weitere Analysen oder Recherchen reduziert wird. Entscheidungen können aufgrund von Fakten getroffen werden.

Die Führungsaufgabe besteht in dieser Situation darin, eine Gruppe von Experten zusammenzubringen, die das zu entwickelnde Gesamtsystem gründlich analysieren. Dabei gibt es folgende Gefahrensignale:

- Lange Diskussionen über die beste Vorgehensweise; da es bei dieser Domäne mehrere „richtige“ Vorgehensweisen gibt, kann eine Bewertung der unterschiedlichen Alternativen aufwendig werden, insbesondere wenn das Ego eines Experten dahintersteht.
- Ein Experte setzt zu sehr auf die eigene Lösung oder verlässt sich auf die Effizienz alter Lösungen.
- Die Analyse zieht sich in die Länge, da die Experten das Gesamtsystem erst vollständig analysieren wollen, bevor sie Ergebnisse weitergeben.
- Gesichtspunkte von Nichtexperten werden ausgeschlossen, da die Experten zu sehr auf ihr eigenes Wissen setzen.

Mögliche Antworten auf diese Gefahrensignale sind:

- Externe und interne Stakeholder dazu ermutigen, die Experten herauszufordern, die gewohnten Bahnen zu verlassen.
- Die Nutzung von Experimenten und Spielen, um den Beteiligten neue Perspektiven aufzuzeigen.
- Aufspalten des Gesamtsystems in einzelne Teile und Inkremente. Nicht alles muss komplett analysiert sein, solange die Querbeziehungen der Teilsysteme stabil sind und sich in Grenzen halten. Klar umrissene Teilsysteme können dann schon realisiert werden und wertvolles Feedback von Nutzern für die weiteren Teilsysteme liefern. Eine Aufspaltung erfolgt am besten gemäß fachlich abgeschlossener Aufgabengebiete.

Es bieten sich agile beziehungsweise iterative Methoden an, mit denen die Kompliziertheit in kleine und überschaubare Einheiten reduziert werden kann.

Ein Beispiel für ein kompliziertes Projekt ist die Erstellung einer landesweiten Arbeitsvermittlungssoftware für das schwedische Arbeitsamt. Zuerst analysierte ein 20-köpfiges Team die Anforderungen an das Gesamtsystem. Dieser Vorgang dauerte mehr als ein halbes Jahr, ein Ende war nicht abzusehen. Der Zeitplan des Gesamtprojekts kam in Verzug. Das Analyseteam wurde mit der Abarbeitung der zahlreichen, sich teilweise widersprechenden Anforderungen einfach nicht fertig. Das Team war in der Analyse gefangen, ohne verwertbare Ergebnisse zu liefern. Die Fachexperten äußerten wiederholt, erst dann Analyseergebnisse liefern zu können, wenn sie das Gesamtsystem verstanden hätten. Doch ohne diese Ergebnisse konnte die bereits im Projekt etablierte 80-köpfige Realisierungsmannschaft nicht zielgerichtet arbeiten. Erst die Managemententscheidung, das Gesamtprojekt in die Teile Arbeitgeberregister, Register für Arbeitssuchende und offene Stellen zu zerlegen, brachte den Durchbruch. In einem ersten Release konnte das Arbeitgeberregister realisiert und eingesetzt werden, dann folgten die beiden anderen Register in weiteren Releases. Das Vorgehen in Iteratio-

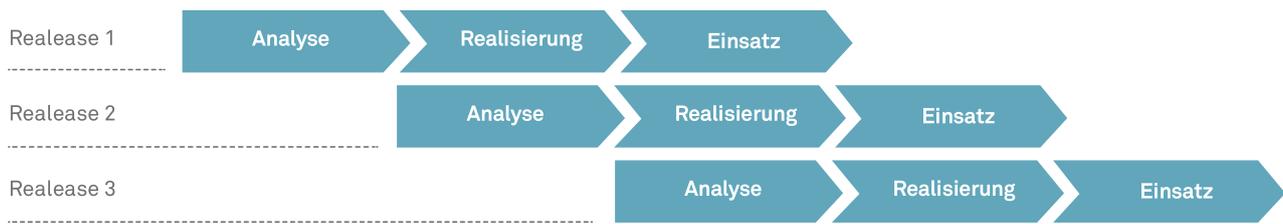


Abbildung 4: Iteratives Vorgehen mit drei Releases

nen sicherte eine gleichmäßige Auslastung des Analyse- und Realisierungsteams und lieferte gleichzeitig wertvolles Feedback aus der Inbetriebnahme des ersten Releases.

### EINFACHE PROJEKTE

In der einfachen Domäne besteht eine offensichtliche Ursache-Wirkungs-Beziehung. Es gibt genau eine richtige Antwort. Hier werden bewährte Vorgehensweisen (Best Practice) eingesetzt, da man sich im Umfeld des bekannten Bekannten (Known Known) bewegt.

Die Führungsaufgabe besteht aus einem faktenbasierten Management und dem Absichern, dass die Entwicklungsprozesse richtig implementiert sind. Eine intensive interaktive Kommunikation, so wie in der komplexen Domäne, ist nicht notwendig. Gefahrensignale in dieser Situation sind:

- Selbstzufriedenheit und Komfort,
- der Wunsch, komplexe Probleme einfach zu machen,
- Denken in eingefahrenen Bahnen,
- fehlenden Hinterfragen von einmal erworbenem Wissen,
- übertriebenes Festhalten an bisher besten Verfahren, auch wenn sich der Kontext verändert.

Mögliche Antworten auf diese Gefahrensignale sind:

- Schaffen von Kommunikationskanälen, um die eingefahrenen Bahnen zu hinterfragen,
- den Wert wie auch die Grenzen von besten Verfahren erkennen,
- nicht davon ausgehen, dass die Dinge einfach sind.



Abbildung 5: Klassischer Wasserfall

Das klassische Vorgehen mit Wasserfallprozessen bietet hier die angemessenste Lösung. Aber Vorsicht: Ist die Einfachheit nur scheinbar gegeben, tritt ein Rutschbahneffekt Richtung Chaos ein – schnelles Handeln ist dann unbedingt erforderlich, um wieder in eine der vorher besprochenen Domänen zu gelangen.

### FAZIT

Das Cynefin-Framework liefert einen guten kognitiven Rahmen, um ein Projekt bezüglich der unterschiedlichen Problem-Domänen – einfach, kompliziert, komplex, chaotisch und unklar – zu bewerten. Jede dieser Domänen erfordert ein unterschiedliches Vorgehen, doch am Projektanfang ist häufig noch unklar, welche Problem-Domäne vorliegt. Hier schafft eine Analyse nach dem Rahmenwerk Klarheit: In der chaotischen Domäne ist schnelles Handeln gefragt, um so schnell wie möglich in eine der anderen Domänen zu gelangen. In der komplexen Domäne haben sich Prototyping und agile Verfahren bewährt. In komplizierten Domäne eignen sich agile und iterative Verfahren. In der einfachen Domäne ist ein Wasserfallvorgehen angemessen. ●

#### ANSPRECHPARTNER – UWE KOBLITZ

Lead Project Manager

Public Sector Solutions Consulting

- +49 172 3731689
- uwe.koblitz@msg-systems.com

