Komplexität

Christian Silberbauer

Motivation

- Systeme tendieren im Sinne eines evolutionären Prozesses zu Stabilität.
- Für Systeme ist es implizites Ziel erhalten zu bleiben, denn andere sterben eben aus.
- Für Systeme ist es hilfreich Stabilität mit möglichst geringer Komplexität zu erreichen.
- Anforderungen an die Umwelt und die Entwicklung von Werkzeugen spielen dabei eine gewichtige Rolle.

Komplexitätsbegriff

- Kommunikation entsteht durch Kommunikation.
- Kommunikation besteht aus Selektionen.
- Systeme kommunizieren und werden dabei mit Komplexität konfrontiert.
- Komplexität bezeichnet einen Aufwand.

Komplexität und ihre Messbarkeit

- Die Komplexität einer Kommunikation kann als Anzahl damit verbundenen Selektionen bestimmt werden, sofern für diese gilt:
  - Sie sind bestimmbar.
  - Sie sind der Kommunikation zuordenbar.
  - Sie sind ähnlich bzw. ungefähr gleich komplex sein.

- Letzteres ist wahrscheinlicher, wenn nur Kommunikation auf ähnlichem Abstraktionsniveau betrachtet wird.
- Ggf. können bei unterschiedlich komplexen Selektionen Komplexitätsfaktoren berücksichtigt werden.

Komplexität und ihre Messbarkeit

- Die Komplexität einzelner Selektionen wird durch die Selektionsmöglichkeiten bzw. durch *Kontingenz* beeinflusst.
- Die Messung der Komplexität darf die betrachtete Kommunikation nicht beeinflussen oder ihr Einfluss muss ggf. gegengerechnet werden.
- Die Messbarkeit ist einfacher bei technischen Systemen wie Anwendungsprogrammen (z.B. Komplexität des Programms).
- Sie ist schwieriger bei sozialen, psychologischen, biologischen Systemen (z.B. Komplexität des Programmierens).

Geringe Komplexität durch Ähnlichkeit

- Kommunikation ist weniger komplex, wenn sie von Ähnlichkeit geprägt ist.

- Unterstützend ist hierbei die Betrachtung der grundlegenden Differenzierungsaspekte:
  - Gruppierung
  - Linearisierung
  - Beständigkeit
  - Metaisierung

Geringe Komplexität durch Ähnlichkeit

- Gruppierung
  - Kommunikation/Lernen von inhaltlich Ähnlichem
  - Kommunikation/Lernen in ähnlichem Umfeld
  - Ähnlichkeit also systemimmanent und in Bezug auf das Systemumfeld
  - Das Systemumfeld ändert sich für das System generell auf unbekannte Weise. Kontinuierliche Anpassungen sind daher notwendig.

Geringe Komplexität durch Ähnlichkeit

- Beispiele:
  - Ein Studium zu Vertiefen bringt Vorteile.
  - Bei vielen Lebewesen gibt es ein Kümmern innerhalb der Familie.
  - Es ist sinnvoller in der Gegend, in der man sich zumeist aufhält mehr ortskundig zu sein.
  - Ein harmonierendes Projektteam ist produktiver.
  - Einen gemeinsamen Codegenerator für ähnliche Anwendungsfelder zu verwenden ist hilfreich.

Geringe Komplexität durch Ähnlichkeit

- Beispiele:
  - Variablen mit ähnlicher Bedeutung bzw. in ähnlichem Kontext sollten ähnlich heißen.
  - Die Zählvariable einer for-Schleife kann immer „i“ benannt werden; Die der inneren for-Scheilfe als „j“ etc.
  - Klassennamen in PascalCasing, Variablennamen in camelCasing…

Geringe Komplexität durch Ähnlichkeit

- Gruppierung
  - Die Bildung von Abstraktionen als solche trägt zur Vereinfachung bei.
  - Für Menschen ist es vereinfachend, in Schubladen zu denken, obwohl Kommunikation elementar ist.
  - Fremde werden nicht als Individuen, sondern als Fremde eines bestimmten Typus empfunden (Georg Simmel).
  - Sind diese Schubladen zu grobgranular, sollten sie heruntergebrochen werden.
  - Dies gilt natürlich auch für andere, z.B. technische Systeme.

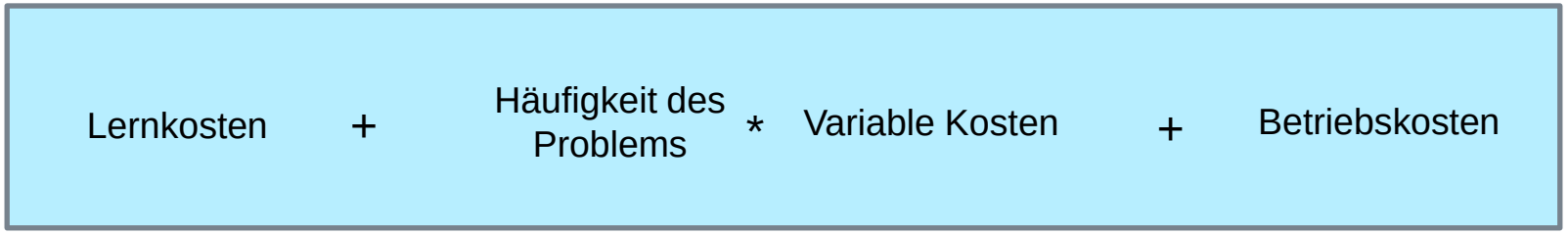
Geringe Komplexität durch Ähnlichkeit

- Linearisierung
  - Kontinuität ist hilfreich für den Lernerfolg.
  - Halbfertiges Wissen bzw. halbfertige Werkzeuge sind oft weniger nützlich.
  - Man sollte daher Lernen mit Schwung angehen.

  - Beispiele:
    - Ein Studium zu Ende bringen ist vorteilhaft.
    - Einen Codegenerator inklusive vollständiger Eingabevalidierung auf Benutzerebene fertigzustellen ist hilfreich.

Geringe Komplexität durch Ähnlichkeit

- Lernen ist eine Investition in die Zukunft bei unklarer Rentabilität.
- Die betriebswirtschaftliche Investitionsrechnung bietet hierfür nützliche Theorien.
- Aufwände werden in diesem Kontext oft als *Kosten* bezeichnet.



- Betriebskosten sind die Infrastruktur zur Problemlösung.
- Die Größen beeinflussen sich gegenseitig.

Geringe Komplexität durch Ähnlichkeit

- Beispiele:
  - Ein Studium zu Ende bringen ist vorteilhaft.
  - Einen Codegenerator inklusive vollständiger Eingabevalidierung auf Benutzerebene fertigzustellen ist hilfreich.

Geringe Komplexität durch Ähnlichkeit

- Beständigkeit
  - Mehr Aufwand in langfristig relevantes Wissen stecken
  - Weniger Aufwand in kurzfristig relevantes Wissen stecken
  - Letztlich ist es für ein System aber unklar, was wie kurz- oder langfristig relevant ist. Es kann nur geschätzt werden.
  - „Vergessen“ von Unwichtigem ist nützlich aufgrund beschränkter Ressourcen.

Geringe Komplexität durch Ähnlichkeit

- Beispiele:
  - Man erinnert den Geburtstag der Mutter, nicht aber den von Einstein (14.3.1879)
  - Die Kenntnis über Design-Patterns in der Programmierung kann das Programmieren nachhaltig verbessern.
  - Das Wissen über SOA hilft längerfristig als das Wissen über die Verwendung konkreter SOA-Plattformen.
  - Vorsicht beim Vergessen: Der Blinddarm schien früher überflüssig zu sein. Mittlerweile ist klar, dass er einen positiven Beitrag zum Immunsystem trägt.
  - In der Softwareentwicklung führt Refactoring zur Bereinigung der Architektur.

Geringe Komplexität durch Ähnlichkeit

- Metaisierung
  - Einheitliche (bewährte) Art und Weise des Lernens/Kommunizierens ist förderlich.
  - Die Wie/was-Unterscheidung bezeichnet Werkzeuge bzw. ihre Anwendung
  - Einheitliche Werkzeuge vereinfachen Anpassungen bzgl. Querschnittsbelangen.

Geringe Komplexität durch Ähnlichkeit

- Beispiele:
  - Andere nutzen ein Hammer, um einen Nagel in die Wand zu schlagen. Das mache ich auch!
  - Informationen über externe Ressourcen durch Codegeneratoren in ein Computerprogramm zu integrieren hat sich bewährt. Ich nutze/entwickle Codegeneratoren auch!
  - In der Automobilindustrie ist C/C++ eine übliche Programmiersprache. Ich nutze das in meiner Abteilung in der Automobilindustrie auch.
  - Ein einheitliches Kommunikationsprotokoll vereinfacht die Berücksichtigung von Querschnittsbelangen (z.B. Rechte auf das eigene Bild im Internet durchsetzen)

Paradigmenwechsel

- Generell verhilft ähnliche Kommunikation zu Fortschritt, birgt allerdings das Risiko, in einem lokalen Optimum stecken zu bleiben.

- Hin und wieder neue Ansätze zu verfolgen – einen Paradigmenwechsel einzuschlagen – kann hierbei Abhilfe schaffen.
- „Sprungfixe Kosten" können ein Game-Changer sein.

- Ein Anzeichen für dessen Notwendigkeit kann eine geringer werdende Verbesserung oder gar eine Verschlechterung sein.

- Häufige Paradigmenwechsel hingegen führen zu Chaos und sind eher destruktiv.

Beispiele

- Ein Patient kommt in eine Psychiatrie.
- Jemand kommt ins Gefängnis.
- Eine Firma führt neue Entwicklungssoftware ein.
- Ein Krieg bricht aus.
- Man erwartet sein erstes Kind.

Beispiele

- In der Psychologie sind zahlreiche Effekte bekannt, wo die Tendenz zu Ähnlichkeit zu irrationalem Verhalten führen und im Nachgang durch Reflektion bzw. einem Paradigmenwechsel korrigiert werden können.

- Z.B.: Present-Bias, Kauslitätsillusion, Priming, Ankereffekt, Framing-Effekt, Halo-Effekt, WYSIATI-Regel, Spielerfehlschluss

- (Solche Effekt lassen sich wiederum den vier Differenzierungsaspekten zuordnen. Siehe https://gwasch.de/slides/psychologische_effekte.pdf)

Beispiele

Present-Bias	Die Unfähigkeit, sich auf langfristige Vorteile zu verpflichten, die unmittelbare Vorteile überwiegen.
Kausalitätsillusion	Kausalitäten werden konstruiert.
Priming	Bezeichnet Phänomene, bei denen zufällig – zumeist unbewusst – eine Information aufgenommen wird, die dann später – ebenso zufällig – unsere Urteilsbildung oder unser Verhalten beeinflusst.
Ankereffekt	Die Neigung, sich bei der Entscheidungsfindung auf eine einzige, in der Regel die zuerst in Erfahrung gebrachte Information zu fokussieren und später hinzugewonnene Erkenntnisse zu ignorieren.
Framing-Effekt	Wahrnehmung je nach Kontext
Halo-Effekt	Wenn wir glauben, eine Eigenschaft einer Person oder einer Sache erkannt zu haben, dann besteht die starke Tendenz, dieser Person oder Sache weitere ähnliche Eigenschaften zuzuordnen, ohne dass es hierfür eine Grundlage gibt.
WYSIATI-Regel	What you see is all there is
Spielereffekt	Jetzt ist schon vier mal hintereinander Kopf gekommen, also ist die Wahrscheinlichkeit groß, dass beim nächsten Mal Zahl kommt. Irrtum!

- Quellen: (Fry & Rutherford, 2023), (Urbaniok, 2020)

Paradigmenwechsel

- Paradigmenwechsel sollten eher konservativ erfolgen, um einen Verlust zu vermeiden und den Status quo zu erhalten.
- …und somit das Streben nach Beständigkeit nicht zu gefährden.
- In der Psychologie kennt man das Phänomen der **Verlust-Aversion**.
- Dass das Gehirn mögliche Verluste gegenüber möglichen Gewinnen stärker gewichtet folgt damit diesem Streben.

Beispiele

- Verlust-Aversion ist ein weiterer Aspekt, der psychologisch zu irrationalem Verhalten führen kann.

- Z.B. **Deklinismus** („früher war alles besser“)

Beispiele

- Der Psychologe Daniel Kahnemann präzisiert in seiner *Neuen Erwartungstheorie* im *viergeteilten Muster* die Tendenz mögliche Verluste höher zu gewichten:

	Gewinne	**Verluste**
Hohe Wahrscheinlichkeit	Risikoscheu	Risikofreudig
Geringe Wahrscheinlichkeit	Risikofreudig	Risikoscheu

- Bei einer hohen Wahrscheinlichkeit auf einen Gewinn überwiegt die Angst, diesen nicht zu bekommen.
- Bei einer geringen Wahrscheinlichkeit auf einen Gewinn überwiegt die Freude, diesen zu bekommen.
- Bei einer hohen Wahrscheinlichkeit auf einen Verlust überwiegt die Hoffnung, diesen zu vermeiden.
- Bei einer geringen Wahrscheinlichkeit auf einen Verlust überwiegt die Angst, diesen in Kauf zu nehmen.

Geringe Komplexität für Systeme

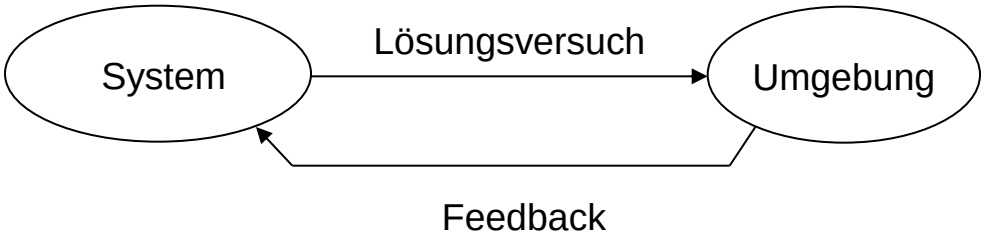
- Von Ähnlichkeit geprägte Kommunikation ist weniger komplex.
- Betrachtet man Kommunikation zwischen bestimmten Systemen, verhilft eine gegenseitige **erwartungsgemäße** Kommunikation zu geringerer Komplexität.
- Der Fehlerfall tritt möglichst nicht ein, weil dies weitere Selektionen nach sich ziehen würde.

Beispiele

- Gegenseitige Zuneigung zeigen bei der Entstehung einer Beziehung zwischen Menschen
- Über einen Witz eines anderen lachen

Geringe Komplexität für Systeme

- Betrachtet man nun den Sachverhalt aus der Perspektive eines Systems, ist folglich die Umgebung des Systems gefordert, eben seine Erwartungen zu erfüllen.



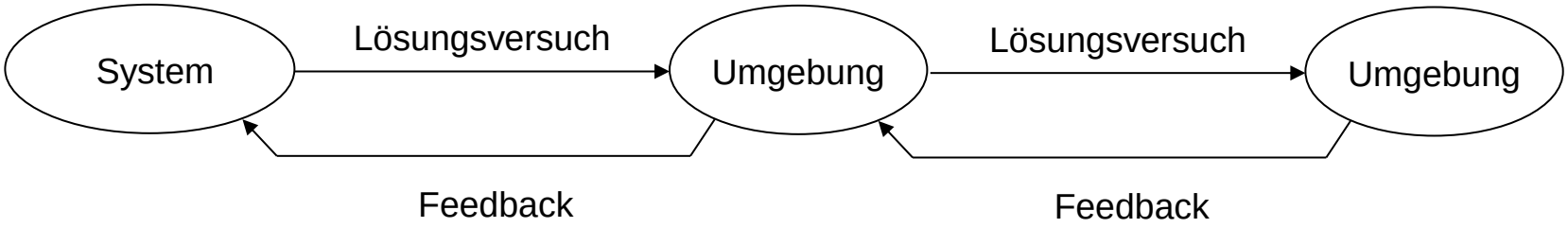
- Die Umgebung *führt* das System zur Lösungsfindung.
- Die Umgebung gibt bei Fehlversuchen hilfreiches Feedback.

Beispiele

- Autovervollständigung einer IDE oder bei mobilen Messengern
- Starke, statische Typsicherheit der Programmiersprache

Geringe Komplexität für Systeme

- Die unterstützende Rolle der Umgebung für sein System lässt sich weiter verbessern, wenn man die Kommunikation der Umgebung mit wiederum seiner (restlichen) Umgebung(en) einbezieht.



Geringe Komplexität für Systeme

- Der Lösungsversuch ist eine **Stimulation** für ein passendes/erwartungsgemäßes/„ähnliches“ Feedback.
- Das System bestimmt mit einer guten Stimulation damit maßgeblich über den Erfolg der Kommunikation.

Beispiele

- Alle wollen einen vergnüglichen Abend. Daher erzählt einer einen Witz.
- Alle wollen tanzen. Ein erstes Paar eröffnet das Parkett.

Geringe Komplexität für Systeme

- Der Grad der Stimulation muss für eine erfolgreiche Kommunikation ausgeprägter sein bei einer unähnlicheren Umgebung.

Beispiele

- In einem Online-Videochat für Erwachsene ist mehr Stimulation des Darstellers erforderlich als bei einem Bordellbesuch.
- In einer Shopping-Mall mit vielen integrierten Restaurants ist mehr Stimulation erforderlich als im Landgasthof.

Geringe Komplexität für Systeme

- Das System darf dabei nicht nur primär aktiv, sondern muss auch reaktiv kommunizierend betrachtet werden.

Beispiele aus der Psychologie

- Das „Selbst" einer Person kann von außen gefördert aber auch entgegen dessen manipuliert werden.
- Letzteres führt zu Schizophrenie, damit zur Depression, also zu Unglück, also technisch zu einer „ungesunden" Instabilität.

- Insbesondere in Kindheit und Pubertät (also insgesamt ca. bis zum 21. Lebensjahr) ist eine Person anfällig Rollen zu adaptieren, die nicht zu ihrer Persönlichkeit kompatibel sind.
- Es führt zur Übernahme fremder Verhaltensweisen aber auch deren Gegenteil (Rebellion).

- Siehe: https://www.youtube.com/watch?v=YMGkqde8BRs,
- https://www.youtube.com/watch?v=nzN9FvdhhRY

Beispiele aus der Psychologie

- Beispiele:
  - Das vierjährige Kind muss Brokkoli essen mögen.
  - Der natürlich Lernwille eines Kindes wird in den ersten Schuljahren ausgebremst.
  - Übernahme dominanter oder devoter Verhaltensweisen durch den Einfluss von Eltern und Geschwistern.
  - Mobbing unter Jugendlichen (durch die Peer-Group)

  - Eine Erzieherin soll Mehrarbeit zur Planung des Sommerfestes leisten, aber auch ihrer Rolle als Mutter gerecht werden und Zeit mit ihren eigenen Kindern verbringen.
  - Eine Softwareentwicklerin erfährt viel Ablenkung durch Kollegen.

Und Philosophie…

„Der Gipfel des Glücks ist erreicht, wenn eine Person bereit ist zu sein, was sie ist."

Erasmus von Rotterdam (1466 – 1536)

- Quelle: (Buckingham et. al., 2011, S. 97)

Beispiel aus der Soziologie

- Antonio Gramsci:
  - Die herrschende Klasse hält ihre Dominanz mithilfe der gesellschaftlichen Institutionen.
  - Mittels subtilen Zwangs setzt sie ihre „kulturelle Hegemonie“ durch.

- Robert Blauner:
  - „Entfremdung existiert dort, wo Arbeiter nicht in der Lage sind, ihren unmittelbaren Arbeitsprozess zu kontrollieren.“
  - Dies basiert auf der Annahme der „Entfremdung“ von Arbeit im Zuge der industriellen Revolution nach Karl Marx.

Geringe Komplexität für Systeme

- Für ein System ist mit geringerer Komplexität konfrontiert, wenn ihre Subsysteme ähnlich sind.
- Das heißt, wenn sie über ähnliches Wissen bzw. ähnliche Werkzeuge verfügen.
- Eben durch Standardisierung

Beispiele

- Schulsystem
- Programmierer arbeiten mit derselben Entwicklungsumgebung.
- Kompatible Charakter in Projektteams
- Weltweite Standardisierung trägt maßgeblich zum Erfolg von Fastfoodketten wie McDonalds bei.

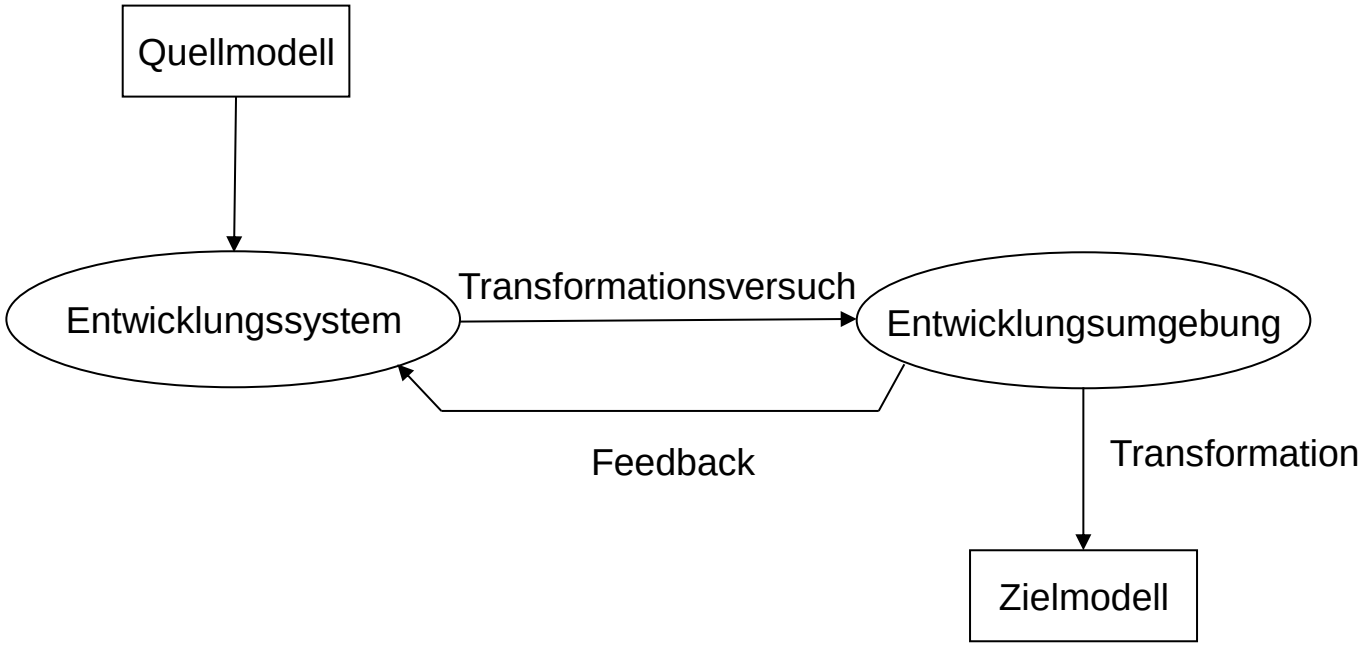
Geringe Komplexität für Systeme

- Ein vorbildliches Handeln ist hilfreich.
- So schafft die Tendenz des Umfelds zu Ähnlichkeit eine Angleichung.

Beispiele

- Soll das Kind gewisse Essgewohnheiten annehmen, ist es hilfreich, dass sein Umfeld diesen auch folgt.
- Lebt im Job ein Chef eine bestimmte Arbeitsweise vor, folgen die Mitarbeiter dieser eher.

Komplexität und Modellierung



Komplexität und Modellierung

- Aus einem Quellmodell wird ein Zielmodell sukzessive abgeleitet.
- Ist das Zielmodell dem Quellmodell ähnlicher, ist auch die Komplexität zur Problemlösung geringer.
- Ähnliche Metamodelle tragen zur Ähnlichkeit jeweils beschriebener Modelle bei.
- Die Ähnlichkeit von Quell- und Zielmetamodell führt zu Ähnlichkeit von Quell- und Zielmodell.

Komplexität und Modellierung

- Auch eine Ähnlichkeit von Entwicklungssystem und Entwicklungsumgebung zu Quellmodell bzw. Zielmodell ist hilfreich.

- Sie sollten mit den jeweiligen Modellen vertraut sein.

Komplexität und Modellierung

- Ein Quellmodell sollte möglichst direkt als Zielmodell abgebildet werden können.
- Voneinander unabhängige Quellmodelle werden so in voneinander unabhängige Zielmodelle abgebildet.
- Abhängigkeiten zwischen Quellmodellen sind analog zwischen Zielmodellen existent.

Komplexität und Modellierung

- „Conway's Law", (Conway, 1968):

> Organizations which design systems […] are constrained to produce designs which are copies of the communication structures of these organizations.

- (Meyer, 1997, pp. 23, 930 ff.) fordert diesbezüglich für den objektorientierten Ansatz „Seamlessness" durchgängig für alle Schritte des Software-Lifecycles, u.a. Analyse, Design, Implementierung und auch Wartung. Die Lücken zwischen aufeinanderfolgenden Aktivitäten sollen minimal sein.

Komplexität und Modellierung

- Aus Sicht der Psychologie (Green & Petre, 1996):

> When seeking information, there must be a cognitive fit between the mental representations and the external representation.

- Basierend auf:
  - (Green, 1977) und (Green, et al., 1987)
  - (Sinha & Vessey, 1992) und (Vessey & Galletta, 1992)

Komplexität und Modellierung

- Die direkte Abbildung von Quell- auf Zielmodell wird als **Domänenzentrierung** bezeichnet.
- Eine semantische Einheit des Quellmodells, also eine Domäne, wird ohne strukturelle Änderung im Zielmodell umgesetzt.
- Domänenzentrierung verhindert **Redundanz** im Zielmodell in der Hinsicht, dass eine Änderung auf Basis des Quellmodells nicht mehrere Änderungen des Zielmodells nach sich zieht.

Komplexität und Modellierung

- (Martin, 2017, p. 14):

> If we'd wanted the behavior of machines to be hard to change, we would have called it *hard*ware.

- Die Verringerung von Redundanz bedeutet gleichermaßen die Erhöhung von *Wiederverwendbarkeit*.

Komplexität und Modellierung

- In der Softwareentwicklung ist Domänenzentrierung eine Erweiterung von Separation-of-Concerns (Dijkstra, 1976).
- Es fordert nicht nur, dass verschiedene Belange/Domänen voneinander getrennt werden sollen, sondern darüber hinaus sollen zusammengehörige Bestandteile einer Domäne auch zusammengehörig programmiert werden.

Komplexität und Modellierung

- Die Domänenzentrierung fokussiert Ähnlichkeit innerhalb der Domäne.
- Dies erlaubt verschiedenen Entwicklungssystemen (z.B. Programmierer) möglichst unabhängig voneinander zu modellieren (z.B. programmieren).
- Der Fokus auf einzelne Domänen trägt damit zur Komplexitätsreduktion bei.
- Beispiele:
  - Ein Softwaresecurityspezialist kümmert sich um Securityaspekte.
  - Für den Hausbau werden Maurer, Elektriker, Architekten etc. beschäftigt.

Beispiele für Aspekte bzw. Domänen

- Concurrency
- Parallel computing
- Event handling
- Network communication
- Time management
- Logging and tracing
- Exception handling
- Software loading
- Variant handling
- Lifecycle management
- Assertions
- Safety
- Security
- Memory management
- Persistency
- UI handling

Komplexität und Modellierung

- Bei komplexer Modellierung ist zudem Hierarchisierung hilfreich.
- Dadurch wird zusätzlich ein heterogenes Entwicklungssystem strukturiert.
- Beispiele:
  - Ein Projektleiter führt das Projektteam.
  - Ein Staat hat eine Regierung.

Komplexität und Modellierung

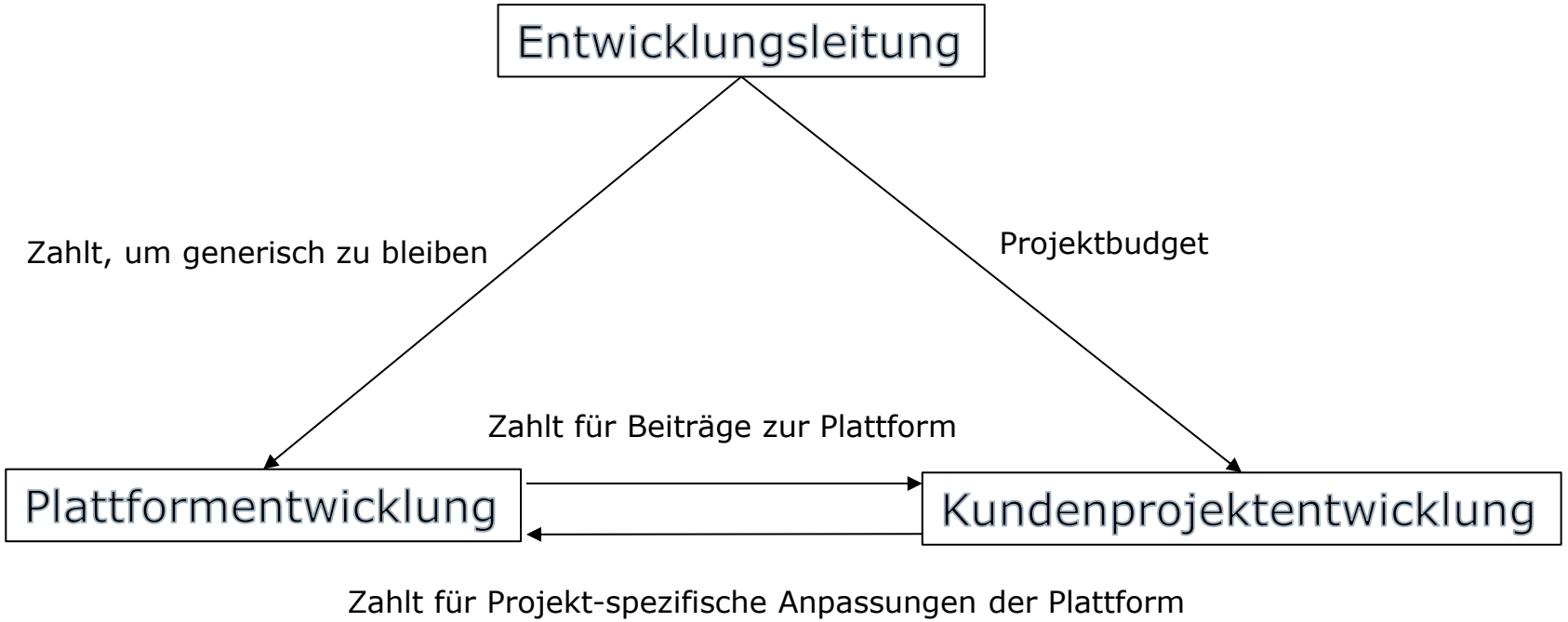
- Sehr komplexe Entwicklungssysteme erfordern:
  - Klare definierte und möglichst voneinander abgegrenzte Domänen und Zuständigkeiten.
  - Sowohl Top/Down- als auch Bottom-up-Kommunikation und damit proaktive (agile) Subsysteme.
  - Redundanz als Sicherungsmaßnahme
  - Mehrdimensionale Hierarchien (je Domäne/Funktion)

- Die Maßnahmen müssen ihre Mehrkosten rechtfertigen.

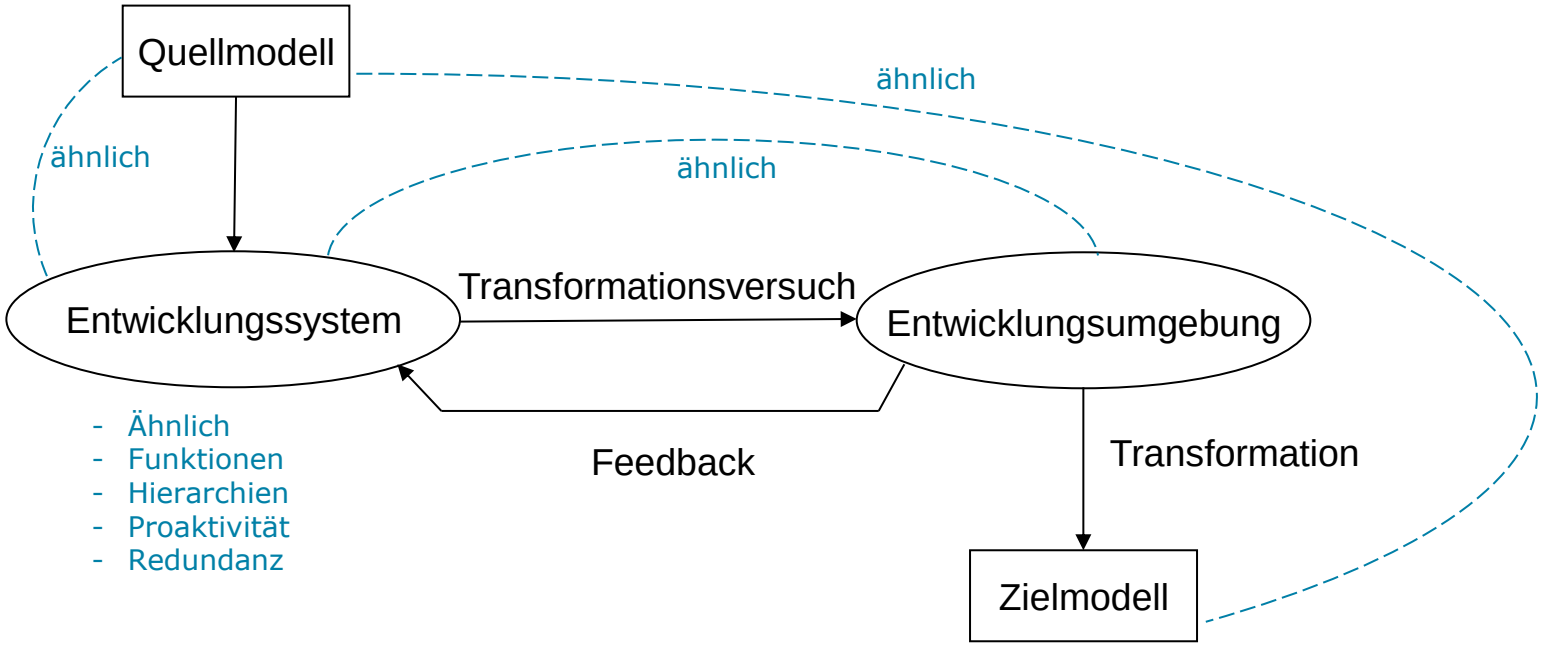
Beispiel Softwareunternehmen

- Sie haben oft eine Hierarchie disziplinarischer Führungsebenen.
- Innerhalb dieser gibt es Projektorganisationen.
- Zudem gibt es Querschnittsdisziplinen, wie z.B. Plattform- bzw. Toolentwicklung.
- Dieser fehlt es oft an Kontinuität, da sie als Zuarbeit für die Projekte betrachtet wird.
- Plattformen werden so regelmäßig entsorgt und neu entwickelt.
- Eine ausgewogene mehrdimensionale Struktur kann hierbei Abhilfe schaffen.
- Ausgewogenheit lässt sich in Unternehmen über den Geldfluss bestimmen.

Komplexität und Modellierung



Komplexität und Modellierung



Komplexität und Modellierung

- Frühe Validierung ermöglicht keine vollständige Stabilität.
- Späte Validierung ist notwendig.

- Das Zielmetamodell steckt den Rahmen des Möglichen für frühe Validierung ab.

- Frühe Validierung zu maximieren, wirkt komplexitätsreduzierend.

Komplexität und Modellierung

- Nr. 1 der Top-Ten-Liste über Software-Metriken aus (Boehm, 1987):

> Finding and fixing a software problem after delivery is 100 times more expensive than finding and fixing it during the requirements and early design phases.

- Ausnahmebehandlungen zur Laufzeit („späte Validierung“) ist eine Domäne für sich und sollte entsprechend gekapselt sein.

Komplexität und Modellierung

- Redundanz ist in Bezug auf Modellierung paradox:
  - Für Entwicklungssysteme erhöht Redundanz Komplexität.
  - Für die daraus hervorgehenden System führt Redundanz zu mehr Absicherung, also mehr Stabilität.
- Damit wird klar, dass Komplexität einerseits eine Notwendigkeit ist, um Stabilität zu entwickeln, sie andererseits aber reduziert werden muss, um dies eben zu vereinfachen.

Komplexität und Modellierung

- Das Zielmetamodell bestimmt das Spektrum möglicher Umsetzungen.
- Es bestimmt Kontingenz also mögliche Selektionen.

- Dabei wirkt für das Zielmetamodell einerseits komplexitätsreduzierend, dass eine möglichst direkte Umsetzung des Quellmodells möglich ist.
- Andererseits sollte es dafür auch nicht zu viele Freiheiten geben.

- Setzen mehrere Programmierer gleiche Anforderungen in ein Programm um, ist es hilfreich, wenn die Programmiersprache zu ähnlichen Lösungen führt.

Literatur

- Boehm, B. W., 1987. Industrial Metrics Top 10 List. *IEEE Software*, Sept., pp. 84-85.
- Buckingham, et. al., 2011. *Das Philosophie-Buch*. DK-Verlag.
- Conway, M. E., 1968. How do committees invent?. *Datamation*, 14(4), pp. 28-31.
- Dijkstra, E. W., 1976. *A Discipline of Programming*. Englewood Cliffs, NJ: Prentice Hall.
- Fry H. & Rutherford A., 2023. *Der ultimative Guide zu absolut Allem*.
- Greeen, T. R. G., 1977. Conditional program statements and their comprehensibility to professional programmers. *Journal of Occupational Psychology*, 50(2), pp. 93-109.

Literatur

- Green, T. R. G. & Petre, M., 1996. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages & Computing*, 7(2), pp. 131-174.
- Kahnemann D., 2012. *Schnelles Denken, Langsames Denken*, 26. Auflage.
- Martin, R. C., 2017. *Clean Architecture: A Craftsman's Guide to Software Structure and Design, 2nd edition*. Upper Saddle River, NJ: Prentice Hall.
- Meyer, B., 1997. *Object-Oriented Software Construction 2nd Edition*. Upper Saddle River, NJ: Prentice Hall.

Literatur

- Sinha, A. P. & Vessey, I., 1992. Cognitive fit in recursion and iteration: an empirical study. *IEEE Transaction on Software Engineering*, 18(5), pp. 368-379.
- Thorpe C. et. al., 2016. *Das Soziologie-Buch*. DK-Verlag.
- Urbaniok, Frank, 2020. *Darwin schlägt Kant*.
- Vessey, I. & Galletta, D., 1992. Cognitive Fit: An Empirical Study of Information Acquisition. *Information Systems Research*, 2(1), pp. 63-84.