



mitp

Frederick P.
Brooks

Erfolgreiches Design

**Essays über universelle Designprozesse mit
Beispielen aus IT und Software-Entwicklung**

Teil 1

Modelle für den Designprozess



(Corbis)

Spiralförmiges Treppenhaus

| KAPITEL 1 |

Die Designfrage

Wenn jemand auf Rätsel stößt und darüber nachdenkt, entstehen neue Ideen aus den Zusammenhängen und dem Transfer der Beobachtungen eines Handwerks auf ein anderes.

SIR FRANCIS BACON [1605], THE TWO BOOKS OF THE PROFICIENCY AND ADVANCEMENT OF LEARNING, ZWEITES BUCH, 10
(DEUTSCH: ÜBER DIE WÜRDE UND DIE FÖRDERUNG DER WISSENSCHAFTEN)

Nur wenige Ingenieure, Dichter und Komponisten können miteinander so über die Inhalte ihrer jeweiligen professionellen Arbeiten reden, dass alle Seiten davon profitieren. Ich schlage vor, dass sie derartige Gespräche über Design führen und einander dabei ihre Erfahrungen mit kreativen, professionellen Designprozessen mitteilen sollten.

HERBERT SIMON [1969], THE SCIENCES OF THE ARTIFICIAL, 82

© des Titels »Erfolgreiches Design« (ISBN 978-3-8266-9080-8) 2011

by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.

Nähere Informationen unter: <http://www.mitp.de/9080>

Hat Bacon recht?

Wir wollen Sir Francis Bacons Hypothese hinterfragen. Gibt es unveränderliche Eigenschaften des Designprozesses selbst, die für eine breite Palette zu gestaltender Dinge gelten? Falls dem so ist, scheint es wahrscheinlich, dass Designer eines Bereichs besonders schwierige Probleme zu lösen haben, über die sie einige dieser Prinzipien besser als Designer aus anderen Bereichen verstehen. Weiterhin können einige Bereiche, wie das Gebäudedesign, bei Design und Metadesign (dem »Design des Designs«) auf eine längere Geschichte zurückblicken. Wenn dies zutrifft und Bacons Schlussfolgerung richtig ist, müssten Designer aus Erfahrungen und Einsichten von Designern aus anderen Bereichen für das eigene Handwerk lernen können.

Was ist Design?

Die Tätigkeit des Designens lässt sich als »gedankliche Entwicklung eines Plans oder Entwurfs für die spätere Ausführung oder Umsetzung« definieren.

Die wesentlichen Elemente dieser Definition sind *Plan*, *gedanklich* und die *spätere Realisierung*. Bei einem Design handelt es sich daher um eine Schöpfung, die dem gestalteten Gegenstand zwar vorgeschaltet und zu diesem in Beziehung steht, sich von ihm aber unterscheidet. Dorothy Sayers, die englische Schriftstellerin und Dramatikerin, unterteilt den kreativen Prozess in ihrem großartigen und inspirierenden Buch *The Mind of the Maker* weiter in die drei getrennten Aspekte Idee, Energie (oder Implementierung) und Interaktion,¹ und damit in:

1. Die Formulierung der konzeptionellen Konstrukte
2. Implementierung mit realen Mitteln
3. Interaktion mit Benutzern beim praktischen Einsatz

Bücher, Computer oder Programme existieren dieser Auffassung gemäß zunächst als ideelle Konstrukte, die außerhalb von Zeit und Raum entwickelt werden, aber im Kopf der Urheber im Wesentlichen vollständig sind. Implementiert werden sie in Zeit und Raum, mit Stift, Tinte und Papier oder Silizium und Metall. Der Schöpfungsprozess ist

abgeschlossen, wenn jemand das Buch liest, den Computer benutzt oder das Programm startet und auf diese Weise mit deren Schöpfern interagiert.

In einer früheren Abhandlung habe ich die Aufgaben der Softwareentwicklung in *essenzielle* und *beiläufige* unterteilt.² (Dadurch sollen Letztere nicht herabgewürdigt werden.) Bei den *essenziellen* Komponenten der Softwareentwicklung handelt es sich um die mentale Ausgestaltung des konzeptionellen Konstrukts, bei den *beiläufigen* um dessen Implementierungsprozess. Zur *Interaktion* (Sayers dritter Schritt) kommt es, wenn die Software genutzt wird.

Beim Design handelt es sich daher um die mentale Formulierung, die Sayers »Idee« nennt und die bereits vor Beginn der Realisierung abgeschlossen sein kann. Mozarts Antwort auf eine Rückfrage seines Vaters zu einer Oper, die der Herzog in drei Wochen erhalten sollte, ist zwar ein wenig erstaunlich, verdeutlicht aber das Konzept:

Alles ist bereits komponiert, nur noch nicht niedergeschrieben.

BRIEF AN LEOPOLD MOZART [1780]

Wenn wir Dinge herstellen, wird uns zumeist erst bei der Implementierung klar, wo unsere Ideen noch Lücken aufweisen oder inkonsistent sind. Für Theoretiker sind daher Schreiben, Experimentieren und »Ausarbeiten« wesentliche Lehrfächer.

Die Phasen Idee, Implementierung und Interaktion funktionieren rekursiv. Bei der Implementierung entstehen Leerräume, die in weiteren Designzyklen gefüllt werden müssen. Entsprechend hat Mozart seine gedankliche Vorstellung (die Idee) von seiner Oper mit der Feder auf Papier implementiert. Der Dirigent interagiert mit Mozarts Werk, interpretiert es seinen eigenen Vorstellungen entsprechend und implementiert sie mit Orchester und Sängern. Über die Interaktion mit dem Publikum wird der Prozess dann abgeschlossen.

Ein *Design* ist ein erschaffenes Objekt, mit dem ein *Designprozess* (oder *Gestaltungsprozess*) einhergeht. Das entsprechende Verb lautet dann *designen* oder *gestalten*. Da beim Designprozess oft auch nur vom Design die Rede ist und die Begriffe eng zusammenhängen, kann es hier gelegentlich zu Missverständnissen kommen, die der Kontext aber eigentlich ausräumen sollte.

Was ist wahr? Das Designkonzept

Gewöhnlich vertreten wir den Standpunkt, dass wir bei allen Vielheiten, denen wir denselben Namen geben, eine ideelle Einheit annehmen. Verstehst du das?

Ja, ich verstehe.

Also gehen wir auch jetzt von beliebigen Vielheiten aus, wie zum Beispiel Stühlen und Tischen.

In Ordnung.

Ideelle Einheiten gibt es von diesen Gerätschaften allerdings nur zwei: eine vom Stuhl, eine vom Tisch.

Klar.

Wir können üblicherweise sagen, dass sich die Hersteller bei der Produktion der beiden Gerätschaften an ideellen Einheiten orientieren, wobei die einen Stühle und die anderen Tische für den praktischen Gebrauch fertigen. Die abstrakte, ideelle Einheit davon fabriziert aber kein menschlicher Hersteller, denn das ist unmöglich.

PLATON, DER STAAT [360 V. CHR.], ZEHNTES BUCH

Beim *Design Thinking Research Symposium 7* stellten die Redner 2008 jeweils Analysen derselben vier Treffen des Designteams vor.³ Videos und Kopien waren bereits rechtzeitig im Vorfeld verteilt worden.

Rachael Luck von der *University of Reading* identifizierte in Gesprächen über Architektur ein Element, über das bis dahin zwar noch niemand gesprochen hatte, das uns aber allen bekannt war, nämlich das *Designkonzept*.⁴

Und wirklich bezogen sich der Architekt und sein Kunde von Zeit zu Zeit auf dieses unsichtbare Element. Wenn sie darüber sprachen, deuteten die Redner üblicherweise vage auf die Zeichnungen, dabei war aber klar, dass nicht die Zeichnungen oder deren Bestandteile gemeint waren. Immer ging es dabei um die konzeptionelle Integrität des zu entwickelnden Designs.

Durch Lucks Erkenntnis wurde das *Designkonzept* zu einem eigenständigen Element. Dies entspricht im Wesentlichen auch meinen Erfahrungen. Bei der Entwicklung der Gesamtarchitektur der System/360-»Mainframe«-Rechnerfamilie (1961–1963) war ein derartiges Element im Architekturteam ständig präsent, auch wenn es namentlich nie erwähnt wurde. Mit Gerry Blaauw zusammen hatten wir auf der Grundlage seiner brillanten Einsichten die Designaktivitäten beim System/360 konkret in *Architektur*, *Implementierung* und *Realisierung* untergliedert.⁵ Das Basiskonzept bestand in einer Computerfamilie, mit einem für die Programmierer einheitlichen Gesicht (die Architektur) und verschiedenen Implementierungen, die auf unterschiedlichen Punkten der Preis-Leistungs-Kurve angesiedelt waren (Kapitel 24).

Die verschiedenen Parallelimplementierungen mit unterschiedlichen Vorkämpfern aus Technik und Management lenkten die gemeinsame Architektur in Richtung Allgemeingültigkeit und Reinheit und bewahrten uns vor den kleinen Kompromissen, die die Kosten verringern sollen. Diese Zielsetzung diente den Architekten, die jeweils einen tadellos funktionierenden Rechner entwickeln wollten, aber eher als Schutzschild für ihre Instinkte und Wünsche.⁶

Mit dem Fortschreiten des Designs der Architektur konnte ich etwas zunächst recht Seltsames beobachten. Für das Architekturteam war das *wirkliche* System/360 das Designkonzept, ein Computer als *ideelle Einheit* gemäß Platon. Bei den physischen und elektrischen Geräten Model 50, Model 60, Model 70 und Model 90, die sich bei den Ingenieuren noch im Bau befanden, handelte es sich nur um Platons *Vielheiten* vom wirklichen System/360. Die vollständigste und reinste Verkörperung des Systems war nicht aus Silizium, Kupfer und Stahl, sondern befand sich in den Beschreibungen und Diagrammen des Maschinensprache-Programmierhandbuchs (*IBM System/360 Principles of Operation*).⁷

Ähnliche Erfahrungen machte ich beim Strandhaus View/360 (Kapitel 21). Sein Designkonzept wurde bereits lange vor Baubeginn real. Und es bestand über viele Versionen von Zeichnungen und Pappmodellen hinweg weiter.

Interessanterweise habe ich bei der OS/360-Softwarefamilie kein derartiges Designkonzept-Element bemerkt. Vielleicht war das bei seinen Architekten anders, vielleicht war ich nur nicht intim genug mit

dem konzeptionellen Gerüst vertraut. Vielleicht offenbarte sich das Designkonzept mir auch nur deshalb nicht, weil es sich bei OS/360 eigentlich um eine Fusion vier irgendwie separater Teile handelte, einer Datenverwaltung, einem Steuerprogramm, einem E/A-Steuersystem und einem großen Paket, das aus Compilern und Dienstprogrammen bestand (Kapitel 25).

Und der Nutzen?

Hat es einen positiven Nutzen, wenn man ein unsichtbares Designkonzept als reales Element in Designgesprächen identifiziert? Ich denke schon.

Erstens besitzen hervorragende Designs konzeptionelle Integrität, bilden eine Einheit, sind wirtschaftlich und klar. Sie funktionieren nicht nur, sondern können uns *entzücken*, wie es Vitruvius erstmals formuliert hat.⁸ Wenn wir über Brücken, Sonaten, Schaltkreise, Fahrräder, Computer oder Smartphones sprechen, verwenden wir Begriffe wie *elegant*, *klar* und *rein*. Wenn wir das Designkonzept als eigenständiges Element verstehen, können wir bei eigenen Designarbeiten und auch im Team besser nach dessen Integrität streben und dies auch unseren Jungdesignern vermitteln.

Wenn man zweitens häufig über das Designkonzept an sich spricht, unterstützt dies stark die Kommunikation innerhalb eines Designteam. Das einheitliche Konzept ist das Ziel, das sich nur über umfangreiche Konversation erreichen lässt.

Wenn das Designkonzept an sich und nicht abgeleitete Darstellungen oder Detailsaspekte im Mittelpunkt stehen, lässt sich viel direkter kommunizieren.

So nutzen Filmemacher Drehbücher, um sich in ihren Designgesprächen auf das Designkonzept und nicht auf Details der Umsetzung konzentrieren zu können.

Durch das Eingehen auf Details werden natürlich unterschiedliche Auffassungen des Konzepts deutlich, die beseitigt werden müssen. Bei der System/360-Architektur wurde beispielsweise ein dezimaler Datentyp benötigt, der den Tausenden vorhandenen Benutzern von IBM-Dezimalrechnern als Überbrückungshilfe dienen sollte. In unserer sich

noch in der Entwicklung befindlichen Architektur gab es bereits etliche Datentypen, zu denen ein 32-Bit-Integer mit Festpunkt in Zweierkomplement-Darstellung und eine Zeichenkette variabler Länge zählten.

Der dezimale Datentyp konnte an beide angepasst werden. Welche Wahl entsprach aber mehr dem System/360-Designkonzept? Für beide Darstellungen gab es starke Argumente, deren Stringenz von den jeweiligen Auffassungen des Designkonzepts abhängig war. Das implizite Designkonzept einiger der Architekten basierte auf früheren wissenschaftlichen Rechnern, während sich andere eher an früheren Unternehmensrechnern orientierten. Und System/360 sollte explizit beide Anwendungen gleichermaßen gut unterstützen.

Wir entschieden uns bei der Darstellung des dezimalen Datentyps für die Zeichenkette, mit der die größte spezifische Anwendergemeinde des dezimalen Datentyps (IBM-1401-Anwender) vertrauter war. Ich würde mich wieder so entscheiden.

Über den Designprozess nachdenken

Das Nachdenken über Designs kann auf eine lange Geschichte zurückblicken, die bis zu Vitruvius zurückreicht, der etwa 15 vor Christus starb. Sein *De Architectura* ist das wichtigste Buch zum Design aus der Klassischen Periode. Bedeutende Meilensteine sind Leonardo da Vincis *Notizbücher* (1452–1529) und *Die vier Bücher zur Architektur* von Andrea Palladio (1508–1580).

Über den Designprozess selbst wurde erst sehr viel später nachgedacht. Pahl und Beitz folgen, inspiriert von der zunehmenden Mechanisierung, dem Gedankengang von Redtenbacher [1852].⁹ Für mich stellen *Notes on the Synthesis of Form* (Christopher Alexander, 1962), *The Sciences of the Artificial* (Herbert Simon, 1969), *Konstruktionslehre* (Pahl und Beitz, 1977), die Gründung der *Design Research Society* und der Start der Zeitschrift *Design Studies* [1979] bedeutende Meilensteine dar.

Bei Margolin und Buchanan [1995] handelt es sich um eine bearbeitete Sammlung von etwa 23 Essays aus der Zeitschrift *Design Issues*, in denen es vorwiegend um Designkritik und Designtheorie und »gelegentliche Abstecher in philosophische Gefilde, die sich auf das Designverständnis beziehen«, geht (S. xi).

Mein *The Mythical Man-Month* [1975, 1995] befasst sich mit dem OS/360-Designprozess, aus dem später MVS und weitere Betriebssysteme hervorgingen. Es konzentriert sich bei diesem Design- und Entwicklungsprojekt auf menschliche, team- und verwaltungsspezifische Aspekte. Von besonderer Relevanz für das vorliegende Werk sind hier die Kapitel 4 bis 6, die sich damit befassen, wie sich konzeptionelle Integrität beim Teamdesign erreichen lässt.

Computer Architecture: Concepts and Evolution von Blaauw und Brooks [1997] enthält eine ausführliche Erörterung des Designs von IBM System/360 (und System/370–390–z), dessen Architektur, der Beziehungen zwischen Designaspekten und der Gründe für Dutzende Designentscheidungen. Auf den Designprozess oder die menschlichen Aspekte des Gestaltens wird darin überhaupt nicht eingegangen. Abschnitt 1.4, in dem Gütekriterien für das Design von Computerarchitekturen behandelt werden, ist aber tatsächlich von besonderer Relevanz für das vorliegende Werk.

Arten von Design

Systemdesign versus künstlerisches Design

Im vorliegenden Buch geht es um das Design komplexer Systeme aus der Sicht des Ingenieurs, der sich nicht nur auf Nützlichkeit und Leistungsfähigkeit, sondern auch auf Effizienz und Eleganz konzentriert.

Dies steht im Gegensatz zu vielen Designarbeiten von Künstlern und Autoren, die uns entzücken und Bedeutungen übermitteln wollen. Architekten und Industriedesigner können natürlich in beide Lager fallen.

Routinemäßiges, adaptives und originales Design

Das Brückendesign stellen wir uns oft als hohe Kunst der Architektur vor, bei der konzeptionelle oder technische Durchbrüche zu dramatischen und deutlich erkennbaren Kosten, Funktionen und ästhetischen Konsequenzen führen.

Die allermeisten Straßenbrücken sind aber kurz, weshalb es sich beim Design einer 20 Meter langen Betonbrücke um einen routinemäßigen und automatisierbaren Prozess handelt. Der Entscheidungsbaum für

das Design, die Beschränkungen und die Wunschvorstellungen sind Bauingenieuren längst bekannt und wurden bereits vor langer Zeit in Handbüchern verewigt. Dasselbe gilt überwiegend auch beim Design von Compilern für etablierte Programmiersprachen auf neuen Plattformen. Es gibt viele Bereiche mit routinemäßigen, automatisierten Designaufgaben.

Der Schwerpunkt dieses Buches liegt beim Originaldesign, das im Gegensatz zum routinemäßigen Redesign von Objekten mit veränderten Parametern oder auch zum adaptiven Design steht, bei dem im Wesentlichen bereits existierende Designs oder Objekte für neue Zwecke modifiziert werden.

Anmerkungen und Quellen

- ¹ Sayers [1941], *The Mind of the Maker*. Deutsch: *Homo Creator. Eine trinitarische Exegese des künstlerischen Schaffens*.
- ² Brooks [1986], »No silver bullet«.
- ³ McDonnell [2008], *About Designing*. Bei diesem Buch handelt es sich um die bearbeiteten Unterlagen des *Design Thinking Research Symposiums 7* (DTRS7).
- ⁴ Luck [2009], »Does this compromise your design?« abgedruckt in McDonnell [2008], *About Designing*.
- ⁵ Blaauw und Brooks [1964], »Outline of the logical structure of System/360«. Blaauw untergliedert Sayers »Energie« weiter in Implementierung und Realisierung, wobei ich diese Unterscheidung für höchst nützlich halte.
- ⁶ Janlert [1997] stellt in »The character of things« fest, dass Designobjekte einen Charakter besitzen und erörtert, wie sich dieser gestalten lässt.
- ⁷ IBM Corp. [1964], *IBM System/360 Principles of Operation*.
- ⁸ Vitruvius [22 v. Chr.], *De Architectura*.
- ⁹ Pahl und Beitz [1984], *Engineering Design*.