

# Design Thinking = Human-centered Design?

Manuel Burghardt<sup>1</sup>, Markus Heckner<sup>1</sup>, Markus Kattenbeck<sup>2</sup>, Tim Schneidermeier<sup>1</sup>, Christian Wolff<sup>1</sup>

Lehrstuhl für Medieninformatik, Universität Regensburg<sup>1</sup>

Lehrstuhl für Informationswissenschaft, Universität Regensburg<sup>2</sup>

## Zusammenfassung

In diesem Beitrag wird die Methode des Design Thinking mit dem Gestaltungsprozess benutzerorientierter Systeme, wie er in der ISO-Norm 13407 („*human-centered design processes for interactive systems*“) vorgeschlagen wird, verglichen. Dabei werden Unterschiede und Gemeinsamkeiten der beiden Vorgehensweisen aufgezeigt und zur Diskussion gestellt.

## 1 Rahmenbedingungen / Geschichte / Kontext

Die Diskussion systematischer Entwurfsmethoden prägt die Gestaltung der Mensch-Maschine-Interaktion im Besonderen, kann aber ganz allgemein im Kontext der „Verwissenschaftlichung“ der Gestaltungswissenschaften im 20. Jahrhundert (Bürdek 2005:273ff) gesehen werden. Die Grundidee der systematischen Lehre von Design als Prozess im Rahmen der Curricula an Universitäten, die im *Design Thinking* durch die sog. *d.schools* prominent vertreten ist, lässt sich bis in die 1960er Jahre zurückverfolgen. Im Frühjahr 1968 hielt Herbert A. Simon am Massachusetts Institute of Technology eine Gastvorlesung, die sich neben der Herausarbeitung des Begriffs „*artificial phenomena*“ (Simon 1969, p ix) auch der Ausarbeitung einer Methodologie für die Erzeugung des *Artifiziellen*, dem Design, aus einer stark formalisierten, prozesshaften Sichtweise heraus, widmet. Design wird dabei als grundlegende menschliche Aktivität innerhalb der Entwicklung von Produkten aufgefasst. Davon ausgehend gilt es Design einen elementaren Bestandteil universitärer Ausbildung werden zu lassen, welche Ende der 1960er Jahre noch stark von naturwissenschaftlichen Herangehensweisen geprägt war (vgl. Simon 1969, S. 55ff). Mit diesem programmatischen Ansatz reflektiert Simon die lange Zeit vorherrschende Auffassung von Design (vgl. den Werdegang von Tim Brown (Brown 2009, S. 4)) und nimmt zugleich die Existenz der *d.schools* gewissermaßen vorweg. Allerdings wird der Design-Begriff im Rahmen von Design Thinking wesentlich akzentuierter aufgefasst: „The evolution from *design* to *design thinking* is the story of

the evolution from the creation of products to the analysis of the relationship between people and products, and from there to the relationship between people and people” (Brown 2009, S. 41-42; Hervorhebungen im Original). Der Ursprung des *terminus technicus* “Design Thinking” lässt sich daher nur ungefähr datieren. Obgleich der Begriff bereits seit 1999 für die Bezeichnung eines Symposiums gebraucht wird (The Design Group 2011) so schreibt doch Tim Brown die Prägung des Begriffs im engeren Sinn David Kelley zu und datiert diese auf den Anfang des 21. Jahrhunderts (vgl. Brown 2009, S. 6).

## 2 Design Thinking vs. Human-Centered Design

Als generische Methode zur praxis- und nutzer-orientierten Generierung von Innovationen und Problemlösungen ist Design Thinking (im Sinne von Tim Brown) grundsätzlich zur Bearbeitung unterschiedlichster Design-Aufgaben geeignet. In der Tradition des Produktdesigns entwickelt, wird es heute auch zur Innovation auf organisatorischer Ebene eingesetzt und weist damit, ähnlich den Zukunftswerkstätten (vgl. e.g. Kuhnt & Müllert 2000), Potenzial zur Lösung sozialer Probleme auf (vgl. Brown 2009, S. 115f. Im Folgenden beschränken wir uns auf den Bezug zum Anwendungsfeld *Entwicklung interaktiver Software / Mensch-Maschine-Interaktion*. Terry Winograd, der an der *d.school* in Stanford *Design Thinking* lehrt, beklagt, dass *Design Thinking* als Methode im Bereich des Software-Design immer noch zu wenig Beachtung findet (Winograd 2008). Diese Einschätzung stützt das Themenspektrum des Design Thinking-Symposiums im Jahr 2010: Hier standen, trotz Fokussierung des Themas auf “Studying Professional Software Design”, weniger Fragen der Gestaltung von MMI bzw. Benutzeroberflächen im Vordergrund; vielmehr wurde der Systementwurf, mithin die Systemarchitektur in den Mittelpunkt der Betrachtungen gerückt (vgl. z.B. Jackson et al. 2010 oder Tang et al. 2010).

Dabei scheint Winograds Aussage nicht unproblematisch, stellen doch *human-centered* bzw. *user experience design* und *usability engineering* seit Jahren einen festen Bestandteil der Softwareentwicklung dar (vgl. Seffah 2005), welcher auch in diversen Normierungsversuchen (z.B. in der ISO-Norm 9142 zur “*Ergonomie der Mensch-System-Interaktion*”) dokumentiert ist. Aus Sicht einer *user-* und *usability-*orientierten Softwareentwicklung stellt sich damit die Frage inwiefern Design Thinking Berührungspunkte mit bestehenden Vorgehensweisen aufweist, und wo die Methode zusätzliche Ideen und Impulse liefern kann. Als Vergleichsgrundlage für die Diskussion dieser Frage wird die ISO-Norm 13407 zur benutzerorientierten Gestaltung interaktiver Systeme (“*human-centered design processes for interactive systems*”) herangezogen. Obwohl diese 1999 veröffentlichte Norm seit März 2010 (vgl. ISO 2011) durch die Norm EN ISO 9241-210 ersetzt wurde, soll sie hier dennoch zum Vergleich herangezogen werden, ist doch Design Thinking noch während der Ära der nunmehr obsoleten ISO-Norm 13407 entstanden.

## 2.1 Grundlagen und Merkmale

Design Thinking ist ein iterativer Prozess zur Ideengenerierung und letztendlich zur Lösung komplexer Problemstellungen (in beliebigen Bereichen), der sich in sechs wesentliche Phasen untergliedern lässt (Plattner, Meinel & Weinberg, 2009): (1) Verstehen des Problemkontexts, (2) Beobachten der Zielgruppe (Empathie), (3) Definieren der Standpunkte, (4) Generieren von Ideen, (5) Prototyping und (6) Testen der Prototypen. Die ISO-Norm 13407 kann hingegen als spezifische "Anleitung für benutzerorientierte Gestaltungsaktivitäten für den gesamten Lebenszyklus rechner-gestützter, interaktiver Systeme" (ISO 13407:1999, S. 3) gesehen werden und beschreibt dabei im wesentlichen vier Phasen (bzw. Aktivitäten) des Gestaltungsprozesses sowie zusätzlich einen Start- (Feststellen der Notwendigkeit einer benutzerzentrierten Gestaltung) und einen Endpunkt (Das System erfüllt die festgelegten Anforderungen an Funktion, Organisation und Benutzerbelangen): (1) Verstehen und Festlegen des Nutzungskontexts, (2) Festlegen von Benutzeranforderungen und organisatorischen Anforderungen, (3) Entwerfen von Gestaltungslösungen, (4) Beurteilen von Gestaltungslösungen gegenüber Anforderungen (ISO 13407:1999). Während sich die ISO-Norm auf die Gestaltung interaktive Systeme beschränkt, kann der Design Thinking Ansatz als generisches Framework zur Entwicklung neuer Produkte verstanden werden. Indessen erscheinen die Design-Phasen beider Ansätze auf den ersten Blick eine enge Verwandtschaft aufzuweisen (vgl. Tabelle 1).

## 2.2 Vergleich der beiden Vorgehensweisen

In diesem Abschnitt werden die wesentlichen Phasen beider Vorgehensweisen gegenüber gestellt und grundlegend diskutiert.

	<b>ISO 13407</b>	<b>Design Thinking</b>
1	Feststellen der Notwendigkeit einer benutzerzentrierten Gestaltung	
2	Verstehen und Festlegen des Nutzungskontexts	Verstehen Beobachten
3	Festlegen von Benutzeranforderungen und organisatorischen Anforderungen	Standpunkte definieren
4	Entwerfen von Gestaltungslösungen	Ideen generieren/ Prototyping
5	Beurteilen von Gestaltungslösungen gegenüber Anforderungen	Testen (Iterieren)
6	Das System erfüllt die festgelegten Anforderungen an Funktion, Organisation und Benutzerbelangen	--

*Tabelle 1: Phasenweiser Vergleich der benutzer-orientierten Systementwicklung nach ISO 13407 mit dem Design Thinking Prozess*

**Phase 1** | Die ISO-Norm 13407 setzt zu Beginn des Prozesses auf eine Analyse des Problemfelds, d.h. es muss zuallererst festgestellt werden ob sich eine spezifische Aufgaben mit

einer benutzer-orientierten Vorgehensweise lösen lässt. Dahingegen wird beim Design Thinking die Angemessenheit der Methodik für spezifische Szenarien eigentlich nie in Frage gestellt, d.h. im Design Thinking gibt es kein Szenario in dem die benutzer-orientierte Methode nicht geeignet ist.

**Phase 2** | Nachdem man sich für die ISO-Norm 13407 entschieden hat wird der Nutzungskontext, d.h. Benutzermerkmale, Arbeitsaufgaben, organisatorische und physische Umgebung untersucht und definiert. Dies geschieht mit Hilfe von geeigneten Quellen, tatsächliche Benutzer oder “durch Personen, die die Benutzerinteressen beim Entwicklungsprozess vertreten” (S.6). Als Ergebnis entsteht eine ausreichend dokumentierte Beschreibung des Nutzungskontexts. Design Thinking verwendet als indirekte Methoden Beobachtung, Interviews oder andere sog. *empathy maps*. Das Ziel ist hier bei beiden Ansätzen ähnlich: Es gilt den Kontakt zur Zielgruppe herzustellen und zu analysieren. Im Design-Thinking wird in dieser Phase zusätzlich das Problem konkretisiert, d.h. in welcher Situation und an welchen Nutzergruppen das Design-Thinking Projekt ansetzen soll.

**Phase 3** | Aus Benutzeranforderungen (funktional, finanziell, gesetzlich) werden in der ISO-Norm 13407 Systemanforderungen (unter Berücksichtigung des zuvor definierten Nutzungskontext) abgeleitet und zusätzlich organisatorische Anforderungen erhoben (Zusammenarbeit, Informationsaustausch, Dokumentation, Änderungsmanagement, ...) (ISO 13407:1999, S. 7). Analog zu den Benutzer und Systemanforderungen wird beim Design Thinking ein Standpunkt definiert, der auf Auswertung und Gewichtung der in den beiden vorhergehenden Phasen gesammelten Erkenntnisse und Grundannahmen aufbaut.

**Phase 4** | Unter Berücksichtigung des “Standes der Technik, der Erfahrungen und Kenntnisse der Teilnehmer sowie der Ergebnisse der Nutzungskontext-Analyse” (ISO 13407:1999, S. 7) werden bei der ISO-Norm 13407 Gestaltungslösungen entworfen. Gestaltungslösungen basieren auf vorhandenem Wissen der Designer, werden anhand von Prototypen konkretisiert, von den Benutzern evaluiert und ggf. gemäß dem Nutzerfeedback iteriert und überarbeitet bis benutzerzentrierte Gestaltungsziele erfüllt sind (ISO 13407:1999, S. 7-8). Design Thinking betont zunächst die Generierung von möglichst vielen Ideen in einem definierten, möglichst kurzen Zeitrahmen, und gibt konkrete Anweisungen wie diese zu finden sind, z.B. unter Zuhilfenahme gängiger Kreativtechniken wie etwa *brainstorming*. Abstrakte Ideen werden dann durch das Erstellen einfacher Prototypen (Pappmodelle, Legosteine, etc.) möglichst früh visualisiert, kommuniziert und damit nachvollziehbar gemacht. Das Generieren von Ideen, und die rasche Umsetzung dieser Ideen in Form von Prototypen sind im Design Thinking nochmals formal in zwei Einzelphasen getrennt und werden damit stärker betont, wohingegen die ISO 13407 einen gemeinsame Phase zum Ideensammeln und -umsetzen definiert.

**Phase 5** | Die Beurteilung von Gestaltungslösungen nimmt in der ISO-Norm 13407 einen prominenten Platz ein und liefert idealerweise in jedem Stadium relevantes Feedback zu bisher umgesetzten Benutzer- und Organisationszielen (ISO 13407:1999, S. 8-10). Diese Beurteilung sollte gut strukturiert, organisiert mit einem eigens dafür erstellten Plan verfolgt werden. Auch beim Design Thinking sollen entworfene Gestaltungslösungen möglichst frühzeitig beurteilt werden. Die späteren Nutzer testen die zunächst als Prototypen realisierten

Ideen der *Design Thinker*, die Annahmen und Konzepte werden früh in Frage gestellt, so dass Rücksprünge zu früheren Phasen möglich sind.

**Phase 6** | Die letzte Phase wird beim Design Thinking überhaupt nicht berücksichtigt (Design Thinking endet vor der finalen Implementierung). In der ISO-Norm 13407 wird hingegen langfristig geprüft ob das letztlich implementierte System die festgelegten Anforderungen an Funktion, Organisation und Benutzerbelange auch im tatsächlichen Anwendungskontext, auf lange Sicht hin, erfüllt. Durch Langzeitbeobachtungen können Lösungen umfassender bewertet werden als im Prototypenstadium des Design Thinking-Ansatzes.

### 3 Diskussion

Design-Thinking behandelt Innovation als Produkt: Für einen gegebenen Kontext soll eine neue Lösung geschaffen werden, die zum Einen Mehrwert für den Nutzer schafft, aber zum Anderen auch das Geschäftsmodell eines Unternehmens ergänzt. Dieser Innovationscharakter beinhaltet auch ein gewisses Maß an Unsicherheit hinsichtlich des Ergebnisses: Ob ein physisches Produkt, eine Dienstleistung oder eine Software am Ende des Prozesses stehen ist unklar. Usability Engineering hingegen zielt von Anfang an auf die Entwicklung einer Software ab. Feedback der Nutzer ist Teil des konstruktiven Prozesses, stellt aber nie in Frage, ob überhaupt eine Software entwickelt werden soll.

Human-centered design und Design-Thinking bedienen sich allerdings aus dem selben Methodenspektrum und haben beide ein gemeinsames Ziel: Die Schaffung eines Ergebnisses, das für alle Projektbeteiligte maximalen Mehrwert generiert. Beide Frameworks akzeptieren Scheitern als Teil ihres Prozesses. Ein erster Entwurf stellt nie die fertige Lösung dar, sondern der Weg zum Ziel ist durch Tests, Feedback und Iteration geprägt.

Augenscheinlich ist Usability Engineering bereits teilweise industrialisiert und in standardisierten Softwareentwicklungsprozessen integriert (vgl. Seffah:2005). Design Thinking steht in dieser Hinsicht noch am Anfang. Aus eigener Erfahrung können die Autoren berichten, dass in deutschen Unternehmen noch einige Überzeugungsarbeit zu Prinzipien und Sinn des Design Thinkings geleistet werden muss.

Neben der entwicklungsgeschichtlichen Einordnung des *Design Thinking*, bei der die (sprach-)philosophische Fundierung, die gerade Terry Winograd für diesen Bereich geleistet hat, zu berücksichtigen ist (Winograd & Flores 1989:267ff), fehlt es bisher auch an einer über wie oben analytische Betrachtung hinausgehende Bewertung der beiden offenkundig ähnlichen Verfahren. Systematische Erfassung der Ergebnisse des Einsatzes von *Human-centered Design* bzw. *Design Thinking* könnte hier ähnlich dem experimentellen Software Engineering den Wissensstand über Einsatz und Wirksamkeit unterschiedlicher Methoden im *usability engineering* verbessern helfen.

**Literaturverzeichnis**

- Brown, T. & Katz, B. (2009). *Change by Design. How Design Thinking Transforms Organizations and Inspires Innovation*. Harper Collins Publishers.
- International Organization of Standardization (2011). *ISO 13407:1999 - Human-centered Design Processes for Interactive Systems*.
- Jackson, M. (2010). Representing structure in a software system design. *Design Studies* 31 (6), 545 - 566.
- Kuhnt, B., & Müllert, N. R. (2000). *Moderationsfibel Zukunftswerkstätten. verstehen - anleiten - einsetzen. Das Praxisbuch zur Sozialen Problemlösungsmethode Zukunftswerkstatt*. Ökotopia-Verlag.
- Plattner, H., Meinel, C. & Weinberg, U. (2009). *Design Thinking*. mi-Wirtschaftsbuch.
- Seffah, A., (ed.). (2005). *Human-centered Software Engineering. Integrating Usability in the Software Development Lifecycle*. Springer.
- Simon, H. A. (1969). *The Sciences of the Artificial*. The MIT Press.
- Tang, A., Aleti, A., Burge, J. & van Vliet, H. (2010). What makes software design effective?. *Design Studies* 31 (6), 614 - 640.
- The Design Group (2011). *Design Thinking Research Symposia*. <<http://design.open.ac.uk/cross/DesignThinkingResearchSymposia.htm>>, zuletzt abgerufen am 28. Juni 2011.
- Winograd, T. & Flores, F (1989). *Erkenntnis Maschinen Verstehen*. Rotbuch.
- Winograd, T. (2008). Interview: Design-Thinking wird sich einschleichen. <<http://www.golem.de/0803/58078.html>>, zuletzt abgerufen am 28. Juni 2011.