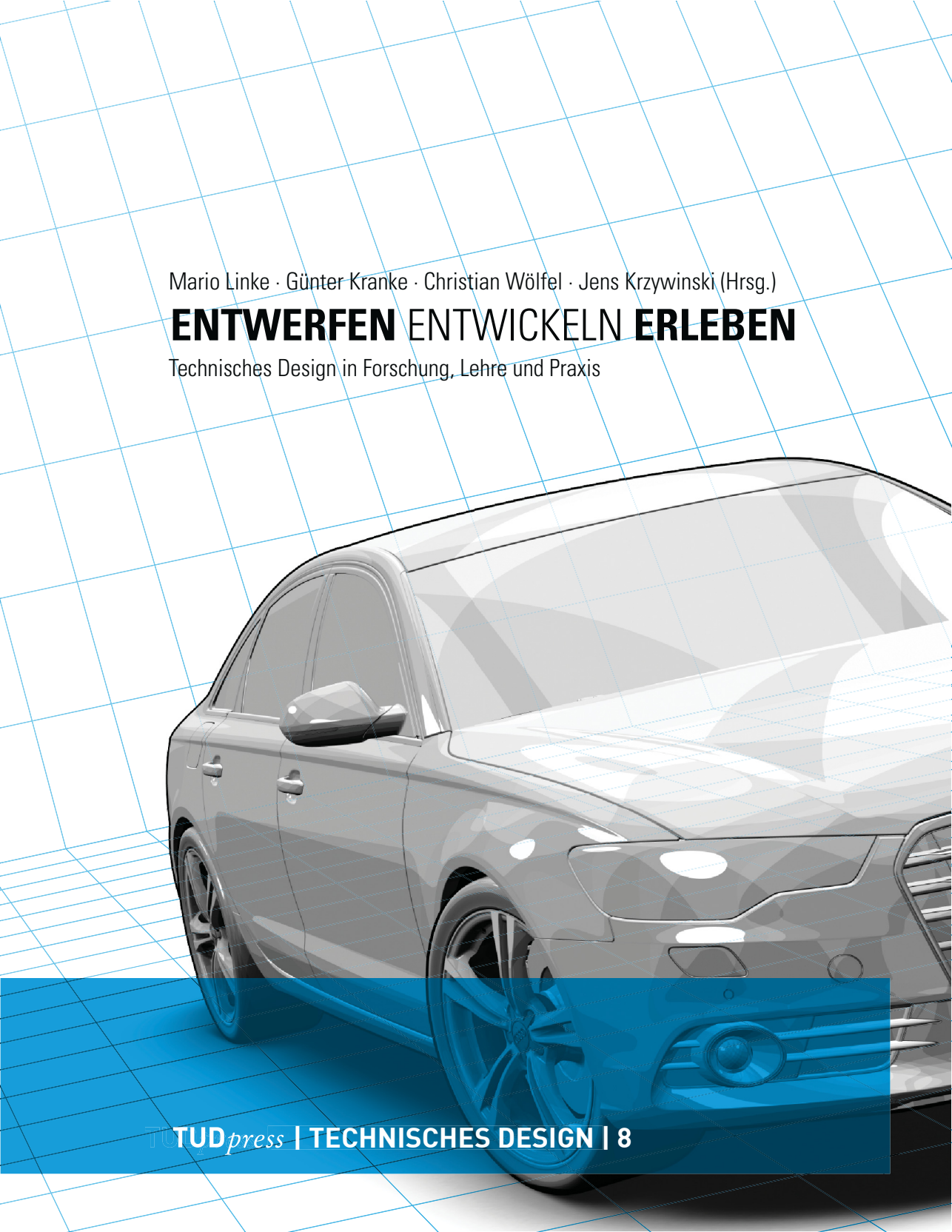
Mario Linke · Günter Kranke · Christian Wölfel · Jens Krzywinski (Hrsg.)

ENTWERFEN ENTWICKELN **ERLEBEN**

Technisches Design in Forschung, Lehre und Praxis



TUD*press* | TECHNISCHES DESIGN | 8

Mario Linke · Günter Kranke · Christian Wölfel · Jens Krzywinski (Hrsg.)

ENTWERFEN ENTWICKELN **ERLEBEN**

Technisches Design in Forschung, Lehre und Praxis

TUD*press* **| TECHNISCHES DESIGN | 8**

Mario Linke, Günter Kranke, Christian Wölfel & Jens Krzywinski (Hrsg.)

TUD*press* **| TECHNISCHES DESIGN**

In der Reihe Technisches Design sind bisher erschienen:

- Johannes Uhlmann:
  Die Vorgehensplanung Designprozess (Nr. 1)
- Norbert Hentsch et al. (Hrsg.):
  Industriedesign und Ingenieurwissenschaften (Nr. 2)
- Norbert Hentsch et al. (Hrsg.):
  Innovation durch Design (Nr. 3)
- Mario Linke et al. (Hrsg.):
  Design – Kosten und Nutzen (Nr. 4)
- Jens Krzywinski:
  Das Designkonzept im Transportation Design (Nr. 5)
- Jan-Henning Raff: *Lernende als Designer (Nr. 6)*
- Christian Wölfel: *Designwissen (Nr. 7)*
- Mario Linke et al. (Hrsg.):
  Entwerfen – Entwickeln – Erleben (Nr. 8)

Weitere Informationen finden Sie unter
reihe.technischesdesign.org und *tudpress.de.*

Mario Linke · Günter Kranke · Christian Wölfel · Jens Krzywinski (Hrsg.)

ENTWERFEN ENTWICKELN **ERLEBEN**

Technisches Design in Forschung, Lehre und Praxis

Entwickeln – Entwerfen – Erleben.
Technisches Design in Forschung, Lehre und Praxis
Herausgeber: Mario Linke, Günter Kranke, Christian Wölfel, Jens Krzywinski

Reihe Technisches Design Nr. 8
reihe.technischesdesign.org

Wir bedanken uns für die Unterstützung bei
ma design, Tedata, Continental, xPLM, B.I.M. Consulting und Reiss Büromöbel

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

ISBN 978-3-942710-75-6

Verlag der Wissenschaften GmbH
Bergstr. 70 | D-01069 Dresden
Tel.: 0351/47 96 97 20 | Fax: 0351/47 96 08 19
http://www.tudpress.de

Layout und Satz: Sandra Olbrich/Technische Universität Dresden.
Umschlaggestaltung: TU Dresden, Illustration Audi A6 Limousine © 2012 Audi AG
Printed in Germany.

Sven Richter

Agiles Entwerfen – Lektionen aus einem Experiment

Erfolg macht attraktiv – das gilt auch für die Prinzipien und Methoden der agilen Softwareentwicklung. Diese finden momentan eine weite Verbreitung, denn sie geben offenbar gute oder zumindest bessere Antworten auf die Probleme, mit denen die herkömmlichen Projektmethoden nicht zurande kommen (Royce 1970). Agile, manchmal auch »leichtgewichtig« genannte Methoden vermeiden übermäßige Planung und Spezifikation, sie ersetzen sie durch eine schnelle Abfolge von Zyklen aus Aktion-Reflexion und Neuausrichtung. Sie bevorzugen die Kollaboration unter gleichberechtigten Experten, die Kommunikation statt Weisung. Und sie beziehen den Kunden und späteren Nutzer bereits in die Entwicklungsarbeit mit ein, denn der Kunde ist die wichtigste Informationsquelle, er bestimmt, inwiefern das Produkt für ihn nützlich ist. Durch solche Prinzipien ist es möglich, Zeit- und Budgetüberschreitungen besser zu vermeiden, einen produktiven Umgang mit Ungewissheit und wechselnden Situationen zu entwickeln und schonender mit der menschlichen Arbeitskraft umzugehen (Abrahamsson et al. 2003). So hat sich das Konzept der »Agilität« auch auf andere Bereiche übertragen, z.B. auf die Gründung von Unternehmen (»Lean Start Up«, Ries 2011) oder die Gestaltung von Organisationsstrukturen (»Agile Organisation«, Richardson 2005).

Der Architekt hat beim Entwerfen von Gebäuden genau diese Probleme, die die agilen Prinzipien adressieren: Der Entwerfer muss mit diffusen Zielstellungen beginnen, er arbeitet aus einem Feld komplexer Informationen heraus, die Ausgangssituation und Rah-

menbedingungen ändern sich häufig im Laufe der Arbeit. Zeitdruck und drohende Budgetüberschreitung sind normal. Der Kontakt zum Kunden - in diesem Fall dem Bauherrn -ist vor allem in den frühen Phasen des Entwurfs intensiv. Zugleich stehen dem Architekten unzählige Entwurfsstrategien zur Verfügung ohne dass es einen Königsweg zum gelungenen Entwurf gäbe.

Daher stellt sich die Frage: Lassen sich die agilen Prinzipien der Softwareentwicklung nicht ins architektonische Entwerfen übertragen um deren Vorteile zu nutzen? Diese Frage hat sich das Laboratory for Architecting Innovation (LAI) an der Wissensarchitektur der TU Dresden in einigen experimentellen Modellversuchen gestellt. Die Aufgabe bestand darin, Bürogebäude zu entwerfen, in denen Wissensarbeit, also Forschungs- und entwicklungsarbeit stattfinden soll. Der Genauigkeit halber muss gesagt werden, dass es nicht *die* agilen Prinzipien gibt, es handelt sich eher um ein Bündel von Wertvorstellungen, Philosophien, Rezepten, ausgebauten Vorgehensmodellen, Tools und Methoden. In den Versuchen des LAI war es eher wichtig, agile Ansätze in drei Bereichen zu verfolgen: wie die zu erledigende Arbeit definiert wird, auf welchem Verständnis die Zusammenarbeit im Entwurfsprojekt basiert und wie der Entwurfsprozess gestaltet wird. Außerdem schreibt das Konzept der Agilität keine bestimmte Entwurfsstrategie vor, es sagt nicht inhaltlich, welche Arbeitsschritte in welcher Abfolge zu machen sind. Vielmehr ist es ein formales Rahmenwerk, das Prämissen und gewissermaßen Spielregeln, aber nicht die konkreten Spielzüge festlegt.

Die Frage muss vorerst mit Nein beantwortet werden: Agile Prinzipien lassen sich nicht ohne weiteres auf das architektonische Entwerfen übertragen. So zumindest das Ergebnis der Versuche am LAI. Die Gründe für das Nein sind folgende:

- Die Arbeit beim Entwerfen lässt sich nicht inkrementell begreifen und erledigen.
- Eine Aufgabenteilung zwischen jemanden, der Anforderungen erstellt, und einem, der die Anforderung realisiert, ist im Entwerfen nicht produktiv.
- Ein »schlankes« Entwerfen analog zum »Lean Development« in der Softwareentwicklung ist nicht sinnvoll.

1 Inkrementelles Entwerfen

In der agilen Softwareentwicklung spielt die Frage, wie die anstehende Arbeit gegliedert und dann schrittweise erledigt werden soll, eine wichtige Rolle (Larman & Basili 2003). Anstatt am Ende eines Projekts das Produkts einmal als Ganzes abzuliefern, teilt man die Arbeit in kleine Produktteile – Inkremente – auf, das jedes für sich möglichst schon vom Kunden zu nutzen ist. Bildlich gesprochen bäckt man einen Kuchen nicht Schicht für Schicht und liefert ihn dann im Ganzen ab, sondern liefert jeweils schon ein essbares Kuchenstück – wenn es auch nur ein Sechzehntel des gesamten Kuchens ist. Das bringt zum einen den Vorteil, dass man durch die Auslieferung des Inkrements an den Kunden schon Einnahmen erzielen kann, zum andern vermeidet man, dass man bei Zeitmangel oder fehlendem Budget am Ende einer langen Projektlaufzeit Gefahr läuft, noch gar nichts geliefert zu haben.

Für das architektonische Entwerfen wäre ein solches inkrementelles Vorgehen interessant. Kleine Ergebnisse permanent auszuliefern, brächte dem Entwerfer eine gewisse Sicherheit, er würde nicht alles auf die eine Karte des endgültigen Entwurfs setzen. So könnte auch die so häufige Panik und übermäßige Arbeit am Ende der Entwurfsphase vermieden werden. Die Deadline nötigt zwar einen Schub der Kreativität ab, aber die so wichtige Arbeitsressource Kreativität wird so sehr verbraucht, dass sie sich nach Abgabe erst wieder sehr langsam erholt.

In seinen Experimenten verwendete das LAI sogenannte User Stories, um die Inkremente zu definieren. Man denkt dabei vom bereits fertiggestellten Produkt her und beschreibt, wie ein Nutzer mit dem Produkt umgehen möchte, z. B. in dieser Form: »Als Teammanager möchte ich in Sichtnähe meines Entwicklungsteams sitzen, um bei Gesprächsbedarf sofort erreichbar zu sein.« Die User Story hat immer den Aufbau, dass eine Person einen bestimmten Gebrauch machen möchte und dafür Gründe oder Ziele angegeben werden. In der Architektur gibt es häufig sehr stark spezifizierte Raumprogramme, die benötigte Flächen, ihre Funktionen, Raumtiefe, Lichtbedarf, usw. festlegen. Die User Story geht einen Schritt weiter und erzählt sogar die spätere Benutzung durch Personen. Es gab einen Backlog

von User Stories, d.h. eine nach Wichtigkeit priorisierte Liste der User Stories. Diese Liste legt die Reihenfolge der Abarbeitung fest. Das Ergebnis der Versuche am LAI war, dass die inkrementelle Arbeitsaufteilung im architektonischen Entwerfen nicht funktioniert.

1.1 Keine inkrementelle Arbeitsgliederung

Die Arbeit beim Entwerfen lässt sich nicht so aufteilen, dass sie inkrementell abzuarbeiten und abzuliefern wäre. Die einzelne User Story kann man formulieren, aber sie ist nicht als für sich stehende funktionale Einheit im Entwurf umsetzen. Die Beschreibung durch eine User Story lässt sich nicht isoliert in eine bauliche Gegebenheit umsetzen. Die Anforderung z.B. die Verkehrswege für Mitarbeiter in einer Gebäudeebene zu minimieren, um den Widerstand gegen den Weg zum persönlichen Gespräch zu verringern, kann nicht unabhängig von anderen baulichen Gegebenheiten wie Erschließungen oder Raumtiefen entworfen werden. Die User Stories stehen in so einem starken Abhängigkeitsgefüge, dass der Entwerfer hier nicht additiv vorgehen kann, sondern immer das Ganze Beziehungsgeflecht der Anforderungen berücksichtigen muss. Bildlich gesprochen geht es nicht darum eine Fotografie Pixel an Pixel aufzubauen, sondern als Ganzes allmählich deutlicher werden zu lassen. Die User Stories können im Nachhinein als Kriterien dienen, ob alle Anforderungen erfüllt worden sind. Sie dienen aber nicht dazu, die Arbeit in kleine für sich bearbeitbare Teile zu segmentieren.

1.2 Kein inkrementelles Abarbeiten

Ein Entwurf lässt sich ebenso wenig inkrementell abarbeiten. Dem widerspricht auch hier widerspricht der Umstand, dass die einzelnen Anforderungen in einem starken Abhängigkeitsverhältnis zueinander stehen. Das macht es schwer möglich, Teile des Entwurfs in schon fertiger, d.h. auslieferbarer, Form vorzulegen. Natürlich können erste Ideen in Form von Zeichnungen dem Bauherrn vorgelegt werden, um zu prüfen, ob der Entwurf die Intention trifft. Aber das sind noch vorläufige Ergebnisse. Andererseits können sicherlich auch Bauausschnitte als Entwurfsdetail vorgelegt werden, aber diese Details sind nur vom Ganzen des Entwurfskonzepts her zu begreifen. An eine eindeutige Priorisierung und folgegerechte Abarbeitung ist im Entwerfen nicht zu denken. Zu viele Rekursionen, Abänderungen,

Umdeutungen und Korrekturen des bisher entstandenen sind nötig. Zuletzt ist die Abschätzung des Zeitaufwands für die Bearbeitung einer Anforderung sehr schwer abschätzbar, noch weniger einzugrenzen, als es auch in der Softwareentwicklung möglich ist.

1.3 Kein Gewinn für die Kreativität

Eine inkrementelle Abarbeitung bringt wenig Gewinn in Bezug auf einen nachhaltigen Umgang mit der Kapazität für kreative Lösungen. Auch wenn es nicht immer so dramatisch geschieht, kommt doch die entscheidende Entwurfsidee sprungweise nach einer starken Anstrengung. Ist noch genügend Zeit, erkundet der Entwerfer alle Möglichkeiten. Je weniger Zeit er hat, desto mehr ist er bereit, einen bestimmten Satz an Rahmenbedingungen anzuerkennen - er ist bereit, sich einschränken zu lassen und bestimmte Rahmenbedingungen sogar zu ignorieren. Diese Bereitschaft zum Weglassen und Konzentrieren entsteht bemerkenswerterweise nur unter großem Zeitdruck oder aber bei großer Disziplin. Der Entwerfer nimmt aber diese Haltung nicht dadurch ein, dass die Arbeit inkrementell konzipiert ist.

2 Verteiltes Entwerfen

In der agilen Produktentwicklung findet man häufig eine Rollenverteilung, die die Verantwortung für das Produkt zwischen Personen aufteilt: Eine Person erarbeitet und verwaltet die Anforderungen und eine oder mehrere andere realisieren sie, indem sie programmieren, testen, usw. Es gibt also eine Person – Product Owner genannt -, diese kümmert sich darum, was zu entwickeln ist, und auf der anderen Seite gibt es Entwickler, deren Expertise darin liegt, dass sie am besten wissen, wie die Anforderung umzusetzen ist (Schwaber & Beedle, 2002). Damit ergibt sich eine klare Trennung: die Kunden- und Anforderungsseite und die technische Seite. Daraus folgt auch ein Kooperationsmodell: die gleichberechtigte Zusammenarbeit zwischen Experten. Es entsteht ein produktives Tauziehen zwischen Kundenwunsch und technischer Machbarkeit. Dieses Modell führt zuweilen soweit, dass es den weisungsberechtigten Manager gar nicht mehr gibt: die Verantwortung für das Gesamtprodukt liegt in den Händen beider Seiten, die sich für ihren eigenen Aufgabenbereich eigenständig organisieren.

Für das architektonische Entwerfen würden in dieser Rollendefinition wiederum einige Vorteile liegen: Zunächst kann sich jeder auf seinen Zuständigkeitsbereich konzentrieren und dadurch die Arbeit intensivieren. Der permanente Klärungsprozess zwischen dem, was der Nutzer bräuchte, und seiner technischen Machbarkeit führt zu einem Qualitätsgewinn. Außerdem vereinbaren der Product Owner und die Entwickler üblicherweise Akzeptanzkriterien, in welchem Fall die Anforderung als erfüllt anzusehen ist. Das geschieht vor der Bearbeitung und gibt damit dem Entwickler zusätzliche Informationen, was er zu erarbeiten hat.

Bei den Versuchen des LAI wurde also ein Product Owner installiert. Dessen Aufgabe war es, die User Stories zu erstellen und sie zusammen mit dem Raumprogramm zu verwalten. Er priorisierte die User Stories nach der Wichtigkeit für den Kunden und formulierte Akzeptanzkriterien, ab wann die Anforderung als realisiert gilt. Auf der anderen Seite gab es mehrere Entwerfer, die sich untereinander bei selbständig koordinierten. Sie stimmten jeweils für einen Entwurfszyklus mit dem Product Owner ab, in welchem Zeitrahmen sie welche User Stories erarbeiten würden. Dieses Rollenmodell erwies sich für den Entwurfsprozess als nicht so produktiv wie erhofft.

2.1 Keine eindeutige Rollentrennung

Zunächst ist im Entwerfen gar nicht eindeutig und leicht zwischen Was und Wie der Anforderung zu unterscheiden. Sobald die User Story konkreter werden sollte, ging diese Arbeit schon in einen Entwurf über. Das wie es zu machen wäre, mischte sich bald ein in die Diskussion. Es gab permanente Überschreitungen, die an sich nicht schlimm sind, die aber das Rollenmodell untergraben. Die Trennung zwischen Anforderung und Entwurf funktioniert nur auf einem sehr abstrakten Level. Offenbar sind im Entwerfen die Klärung des Was und des Wie zu stark verwoben.

2.2 Keine objektiven Akzeptanzkriterien

Akzeptanzkriterien lassen sich im Fall des Entwurfs nicht so objektiv wie gewünscht formulieren – zumindest nicht im Vorhinein. Die Akzeptanzkriterien sind oft vielfältig. Sie betreffen ästhetische Aspekte,

soziale, technische und noch manche andere. Man kann sie nicht auf eine einfache, gut bestimmbare Input-Output-Logik wie im Falle eines Stücks Programmiercodes bringen. Die Kriterien sind im Entwerfen dagegen eher Anlass für Diskussionen, sie machen Unklarheiten deutlich, können aber nicht scharf definiert werden. Bemerkenswerterweise erzeugt der Entwerfer erst indem er zeichnet oder modelliert die Akzeptanzkriterien mit. Denn erst so wird anschaulich, was tatsächlich einer Akzeptanzprüfung unterworfen werden könnte – ob es hier mehr um ästhetische Gesichtspunkte geht, die Gestaltung eines Details, oder doch eher um technische Fragen des Materials.

2.3 Keine Figur eines Product Owners

Ein Vorgang wie der Entwurf eignet sich nicht dazu, von einem Product Owner zu sprechen, der die Verantwortung für das Gesamtprodukt trägt. Der Entwurf gehört viel mehr dem Entwerfer, die Identifikation mit dem, was entsteht, ist beim Architekten groß. Er liefert nicht bloß eine abgearbeitete Anforderung ab, sondern ein persönliches Verständnis, seine Sicht der Dinge. Er kann das nicht leichterdings dem Product Owner übergeben und damit die Verantwortung abgeben. Für die Erfüllung technischer Anforderungen mag das möglich sein. Wenn es um die grundlegende Idee eines Entwurfs geht oder um ästhetische oder soziale Auffassungen, die im Entwurf stecken, ist dies nicht mehr so leicht möglich.

3 Schlankes Entwerfen

Die agilen Entwicklungsprinzipien haben ein großes Vorbild im »Lean Manufacturing«, das vom sogenannten Toyota-Production-System herstammt (Womack & Jones, 2003). Schlank oder eben lean bedeutet in diesem Verständnis, dass man alles im Produktionsprozess vermeidet oder los wird, was nicht Wert für den Kunden schafft. Den Wert bestimmt allerdings der Kunde. Das wendet sich gegen den bedauerlichen und noch weit verbreiteten Irrtum, der den Wert einer Sache damit gleichsetzt, wie viel Mühe und Aufwand die Herstellung gekostet hat. Dieses Prinzip gibt einen Maßstab für Verbesserungen vor, vor allem stetige Verbesserungsversuche, bis unnötige Wartezeit, Arbeitsaufwand und Materialeinsatz getilgt sind.

Natürlich ist es ein weiter Weg von der Fertigung eines Automobils über die Entwicklung von Softwaresystemen bis zum architektonischen Entwurfsprozess. Aber auch hier gibt es den Bedarf an Effizienz, sowohl im Aufwand, den der Kunde bezahlen muss, als auch im Aufwand, den der Architekt selbst betreibt. Und sicherlich ist auch eine Verbesserungskultur gewünscht, ein Erlernen und Ausarbeiten effizienter Entwurfsstrategien. Dafür wäre ein Zielpunkt – der Wert, den der Kunde definiert – nützlich.

Aber auch der Versuch, dieses Prinzip im Rahmen des Entwerfens zu etablieren, hat bislang nicht funktioniert. In den Versuchen des LAI hat das Entwurfsteam nach bestimmten Arbeitszyklen im Entwurfsprozess sogenannte Retrospektiven durchgeführt. Dabei diskutierte das Team die Schwierigkeiten, die beim Entwerfen aufgetaucht sind. Das Team beriet über Änderungen, wie den Schwierigkeiten abgeholfen werden kann. Das Team bewertete auch den Erfolg bisheriger Verbesserungsversuche.

3.1 Wertfindung ist die Aufgabe des Entwerfens

Im Entwurfsprozess sucht man erst noch den Wert für den Kunden. In allgemeiner Form kann er wohl benannt werden. Aber Entwerfen ist vor allem dieser Klärungsprozess, was dem Kunden wichtig ist oder zumindest sein könnte. Ein großer Teil der anfänglichen Entwurfsphasen beschäftigt sich damit, das zu trennen, was der Bauherr will, und das, was er braucht. Das lässt sich nur als Hypothesen im Form von Entwürfen dem Bauherrn vorlegen, damit auch er selbst in eine Klärung seines Bedarfs gehen kann. In diesem Sinne ist jede Zeichnung oder jedes Modell eine gut begründete Vermutung über den Wert, den das spätere Gebäude haben soll. Also liegt die Wertschöpfung im Entwerfen eher in diesen sekundären Werten des Entwurfs: Ob der Kunde die Entwurfsidee gut verstehen kann, ob sie ihm gut kommuniziert wird per Modell, ob die Diskussion über das Gebäude und das, was in ihm stattfinden soll, schnell den richtigen Punkt treffen kann.

3.2 Strategievielfalt im Entwurfsprozess

Es gibt im Entwerfen sehr viele verschiedene Wege, zum Ziel zu kommen. Je nach besonderer Problemstellung der Bauaufgabe

müssen die Methoden und Strategien gewählt, neu angeordnet oder manchmal auch neu entwickelt werden. Dadurch ist es nicht einfach möglich, in kontinuierlichen Verbesserungsdurchläufen die einmal gewählte Strategie zu verbessern. Was in einem Fall eher störender Mehraufwand ist – im Sinne des lean thinking – Verschwendung, kann im anderen Fall zielführend sein. Auch hier geht die versuchsweise Anwendung von Methoden oder Routinen vor einer Tilgung von Mehraufwand. Diese Versuche müssen gemacht werden, sie berechtigen sich selbst erst im Falle des Erfolgs.

3.3 Variabilität ist wichtig

Im Entwerfen kommt es gerade darauf an, anfangs eine große Variabilität der möglichen Lösungen zu entwickeln. Schlankes Entwerfen würde eine Fokussierung verlangen. Diese ist erst in späteren Phasen angebracht, wenn einmal die grundlegende Entwurfsidee gefunden ist. Für das Entwerfen gilt eher das Prinzip des leistbaren Verlusts, es geht darum rechtzeitig zu bemerken, dass der Weg nicht weiterführt und darum aufgegeben werden muss. Jeder Weg erzeugt aber ein Wissen darüber und eine schärfere Bestimmung dessen, was zu tun ist. So wird sich auch ein Irrweg im Nachhinein als nützlich erweisen, weil er dem Entwerfer Beschränkungen klar macht, die vorher nicht offenbar waren. Aber ob ein Weg wirklich diese wenn auch negative Qualität hat, ist bei der Arbeit nicht im Vorhinein festzustellen. Letztendlich findet im Entwerfen so etwas wie ein intelligenter evolutionärer Mechanismus statt: Der nächste Lösungsversuch geht nicht völlig blind los, sondern schon sehr gewählt und gerichtet. Ob er erfolgreich ist – gewissermaßen überlebt –, hängt nicht allein von objektiven Umweltbedingungen ab, sondern auch davon, ob der Entwerfer bestimmte Umweltbedingungen überhaupt anerkennt.

4 Diskussion

Es ist auch nicht in allen Bereichen der Softwareentwicklung möglich, agile Prinzipien anzuwenden. Vor allem dann, wenn es um Arbeiten geht, die selbst Entwurfs- bzw. Designcharakter haben, so z.B. wenn es die Gestaltung des Interface zum Nutzer geht, bei der gute Benutzbarkeit entscheidend ist. Auch hier schafft der Benutzer

Bezüge zwischen Funktionen, die mit einer einfachen Input-Output Funktionalität nicht zu begreifen sind. Auch hier muss im Ganzen entworfen werden.

Ohnehin sind Einwände berechtigt, die sagen, agile Methoden sind ausschließlich für die Produkt*entwicklung* anzuwenden, sie sind gedacht für die Implementierung eines Konzepts, aber nicht für dessen Entwurf. Es sind zwei verschiedene Dinge, eine Idee für ein Produkt zu erarbeiten, und diese Idee dann auszudifferenzieren und zu realisieren. Agile Methoden setzen die Idee voraus. Das ist sicherlich richtig – sollte aber nicht daran hindern, mit den nötigen Abänderungen einen Transfer auch für die Entwurfsphase zu versuchen.

Die hier beschriebenen Versuche haben probierenden, experimentellen Charakter. Sie beweisen nicht, dass es prinzipiell unmöglich ist, die agilen Prinzipien auf das Entwerfen zu übertragen. Sie haben sogar eine gewisse Naivität. Es ging um das Ausprobieren, entgegen möglichen Einwänden, auf die man bereits kommt, wenn man die Übertragbarkeit durchdenkt. Erst der tatsächliche Versuch kann solche Einwände auch bestätigen.

5 Zusammenfassung

Zusammenfassend kann man sagen: Der Versuch, agile Prinzipien auf den Bereich des architektonischen Entwerfens zu übertragen, hat keinen Erfolg gebracht. Die gewünschte Sicherheit in der Zeitplanung und der Qualität der Ergebnisse hat sich nicht eingestellt. Die Ressource Kreativität lässt sich auch agil nicht besser steuern.

Architektonisches Entwerfen ist in einem gewissen Sinn bereits »agil« – es setzt eine hohe Reaktionsfähigkeit auf sich ändernde Situationen voraus. Kein vorgefertigter, von jedem Entwerfer übernehmbarer Prozess führt zum Ziel. Agile Prinzipien hätten hier die Aufgabe, mehr Ordnung zu schaffen - ganz im Gegensatz zu dem, was sie im Verhältnis zu den herkömmlichen Methoden in der Softwareentwicklung leisten, denn dort sorgen sie für mehr Beweglichkeit, für weniger Gewicht. Der Prozess des architektonischen Entwerfens hätte hier den Bedarf nach mehr Struktur, denn

so kontrolliert und diszipliniert er auch vollzogen werden mag, es bleibt ein großer Anteil an Unvorhersehbarkeit und Individualität. Die Frage bleibt, ob ein prinzipieller Unterschied zwischen Entwurf und Entwicklung den Transfer verhindert - oder ob es einfach bislang nicht gelungen ist, den richtigen Ansatz zu finden.

Literaturverzeichnis

Abrahamsson, P., Warsta, J., Siponen, M.T., & Ronkainen, J. 2003:
New Directions on Agile Methods: A Comparative Analysis. In: ICSE 2003.

Larman, C. Basili, V. 2003: Iterative and Incremental Development:
A Brief History. In: Computer, vol. 36 (6), 47–56

Richardson, K. (Hrsg.) (2005). Managing Organizational Complexity:
Philosophy, Theory, and Application. Greenwich, CT: Information Edge Press.

Ries, E. 2011: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to
Create Radically Successful Businesses. New York: Crown Business.

Royce, W. 1970: Managing the Development of Large Software Systems.
In: Proceedings of IEEE WESCON, 26, 1–9

Schwaber, K., Beedle, M. (2002): Agile software development with Scrum.
Upper Saddle River: Prentice Hall.

Womack, J., Jones, D.T. 2003: Lean Thinking, London: Simon & Schuster.

Kontakt

Sven Richter, M. A.
Technische Universität Dresden
Wissensarchitektur
LAI – Laboratory for Architecting Innovation
Zellerscher Weg 17
01062 Dresden
www.architectinginnovation.net