

Universität Hamburg
MIN-Fakultät, Department Informatik
Seminar: „*Agile Methoden und Usability-Engineering*“
Leitung: Dr. Axel Schmolitzky
Wintersemester 2006 / 2007

Rapid Contextual Design

14. September 2008

Stefan Westhoff
Methfesselstr. 43
20257 Hamburg
Tel.: (040) 38 68 07 97
E-Mail: stefan.westhoff@gmx.de
Matrikel-Nr.: 5687040

Inhalt

1. *Einleitung* (S. 3)
 - Einordnung in das Studium
 - Aufbau der Arbeit
2. *Rapid Contextual Design grundsätzlich* (S. 4)
 - Usability
 - Aufwandsstufen
 - Einzelne Schritte
 - Team
3. *Ablauf von Rapid Contextual Design* (S. 6)
 - Contextual Interviews
 - Sequence Models
 - Affinity Diagram
 - Personas und User Scenarios
 - Wall Walk und Visioning
 - Storyboarding
 - Paper Mock-up Interviews
4. *Rapid Contextual Design als agile Methode* (S. 15)
 - Das Agile Manifest
 - Iterationen
5. *Bewertung* (S. 18)
6. *Literaturverzeichnis* (S. 19)

1. Einleitung

Die folgende Arbeit ist die schriftliche Ausarbeitung eines von mir im Rahmen des Seminars „Agile Methoden und Usability-Engineering“ (WiSe 2006/07) gehaltenen Referats und befasst sich mit der Entwurfsmethodik des *Rapid Contextual Design* (kurz *Rapid CD*) und dessen Einordnung in agile Verfahren im Bereich des Usability-Engineering. Mit Rücksicht auf eine flüssige Sprache wird darauf verzichtet Personen jeweils doppelt, in weiblicher und männlicher Form also, zu nennen. Wird die eine Form (hier in der Regel die weibliche) genannt, ist damit im Folgenden implizit auch immer die jeweils andere gemeint.

Referenz bei dieser Arbeit ist in erster Linie das gleichnamige Buch von Karen Holzblatt, Jessamyn Burns Wendel und Shelley Wood.¹ *Rapid CD* baut auf der formalen Entwurfstechnik des *Contextual Design*² auf, bleibt anders als diese jedoch näher an der tatsächlichen Entwurfspraxis und Projektorganisation. Während *Contextual Design* theoretisch und analytisch vorgeht, wartet *Rapid CD* mit einer Fülle von Tipps und praktischen Anleitungen für den Projektalltag auf, begründet diese aber oft durch die Projekterfahrung der Autorinnen, also eher „dem gesunden Menschenverstand folgend“ als streng wissenschaftlich.

Zunächst wird ein kurzer Überblick über die Grundzüge des *Rapid CD* gegeben. Danach wird der Ablauf der Methode schrittweise erläutert. Es wird dann dargelegt, was unter einer agilen Methode zu verstehen ist und danach untersucht, inwiefern *Rapid CD* als agil bezeichnet werden kann. Abschließend wird eine vorsichtige Bewertung von *Rapid CD* versucht.

¹ Karen Holzblatt, Jessamyn Burns Wendel, Shelley Wood: „Rapid Contextual Design: A How-to Guide to Key Techniques for User centered Design.“ Morgan Kaufmann Publishers, San Francisco 2005.

² Vgl. Hugh Beyer, Karen Holzblatt: „Contextual Design: Defining Customer-Centered Systems.“ Morgan Kaufmann Publishers, San Francisco 1998.

2. Rapid Contextual Design grundsätzlich

Rapid CD ist eine Usability-Engineering-Technik um Benutzerinnendaten einzuholen und anhand dieser eine *UI*³ zu entwickeln bzw. eine bereits bestehende zu erweitern oder zu verändern. Der Begriff **Usability** beschreibt den Umfang in welchem ein (Software-)Produkt von bestimmten Benutzerinnen in einem bestimmten Kontext benutzt werden kann, um bestimmte Aufgaben effektiv, effizient und zufriedenstellend umzusetzen.⁴ Spezifiziert werden müssen also Kontext, Benutzerinnen und zu erledigende Aufgaben, gewünscht ist eine schnelle und problemlose Zielführung dieser Aufgaben.

Rapid CD lässt sich in Softwareprojekten jeder Größe und Ausrichtung verwenden. Zunächst werden die Benutzerinnen während ihrer Arbeit beobachtet und befragt, und dieser Vorgang dokumentiert. Anhand der so gesammelten Informationen wird dann in einem mehrschrittigen Designprozess das Softwaresystem entworfen, die *GUI*⁵ implementiert und getestet. *Rapid CD* zielt dabei stets darauf ab Ressourcen (also Material, Arbeitszeit und -kraft) zu sparen ohne dabei Qualität beim Endergebnis einzubüßen.

Je nach Umfang des Projekts und verfügbaren Ressourcen lässt sich *Rapid CD* in drei unterschiedlichen **Aufwandsstufen** durchführen: *Lightning Fast* stellt hierbei mit einer Länge von ein bis vier Wochen die ressourcensparende dar, gefolgt von *Lightning Fast +* mit einer Länge von vier bis acht Wochen und schließlich *Focused Rapid CD* mit einer Länge von sechs bis zehn Wochen. *Rapid CD* lässt sich jedoch auch außerhalb dieser Stufen flexibel an die Gegebenheiten anpassen.

Alle drei Stufen beinhalten die Durchführung so genannter *Contextual Interviews*, das Erstellen eines so genannten *Affinity Diagrams* sowie der Methoden

³ „User Interface“, die Benutzerinnenschnittstelle.

⁴ „[E]xtend to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.“ Vgl. International Organisation for Standardization (Hrsg.): „ISO 13407. Human-centered design processes for interactive systems.“ Genf 1999, S. 1.

⁵ „Graphical User Interface“, die graphische Benutzerinnenschnittstelle, Bedienoberfläche.

des *Wall Walk* und *Visioning*. *Lightning Fast +* und *Focused Rapid CD* schließen hieran noch die Durchführung so genannter *Paper Mock-up Interviews* an. Allein bei *Focused Rapid CD* werden zusätzlich nach den *Contextual Interviews* noch *Sequence Models*, nach Anfertigung des *Affinity Diagrams* noch so genannte *Personas* und *User Scenarios* und vor den *Paper Mock-up Interviews* noch *Storyboards* erstellt.

Das durchgängig mit dem Projekt beschäftigte **Kernteam** besteht aus zwei Personen. Um ein möglichst großes Themenspektrum abzudecken ist es jedoch notwendig bei den einzelnen Teilprozessen Mitarbeiterinnen aus verschiedenen Bereichen (Designerinnen, Entwicklerinnen, Technikerinnen, Management, Marketing, auch Auftraggeberinnen usw.) mit einzubeziehen. Hierbei spricht man von einem *Cross-functional Team*.

3. Ablauf von Rapid Contextual Design

Rapid CD eignet sich für die Benutzerinnencharakterisierung bereits bestehender Systeme, für *Usability Fixes*⁶ an diesen, das Erstellen neuer Systemkonzepte, die Evaluation von Webseiten und anderen Bedienoberflächen, den Entwurf eines Systems der nächsten Generation oder auch komplett neuer Systeme. Zunächst muss festgelegt werden, welche Arbeitsvorgänge durch das Softwaresystem unterstützt werden sollen. Größere Projekte können hierbei auch sequentiell oder parallel aufgeteilt werden, womit einander ähnliche Teilprozesse ohne unnötigen doppelten Aufwand unterstützt werden können.

Einen klar abgesteckten Arbeitsvorgang bezeichnet man als *Task*. Ist der zu unterstützende *Task* spezifiziert, wird die zugehörige, diesen ausführende, also die zu unterstützende Arbeitsgruppe ausfindig gemacht und innerhalb dieser eine Auswahl getroffen. In der Regel sollte diese Auswahl aus maximal vier Kernpersonen bestehen, wobei darauf zu achten ist, diese aus möglichst verschiedenen Kontexten innerhalb des *Tasks* zu wählen, so dass sein ganzes Spektrum abgedeckt wird. Diese Personen werden nun während ihrer Arbeit am Arbeitsplatz interviewt.

Jedes **Contextual Interview** besteht in einem ungefähr zweistündigen Einzelgespräch, währenddessen die Interviewe-

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
4-12 Contextual Interviews	6-12 Contextual Interviews	8-12 Contextual Interviews

rin die Ergebnisse notiert oder bei Bedarf mit einem Diktiergerät aufzeichnet. Ziel des Interviews ist es, die tatsächliche Arbeitsweise der Interviewten zu dokumentieren damit diese später im Entwicklungsprozess modelliert und unterstützt werden kann. Spezielles Augenmerk wird dabei auf die von der Interviewten verwendeten Arbeitsmittel, die so genannten *Artefakte* (beispielsweise Formulare oder Dokumente), gelegt.

⁶ Verhältnismäßig einfache Veränderungen an der Bedienoberfläche, oftmals jedoch mit großer Wirkung – beispielsweise Änderung der Farben oder der Schriftformatierung.

Ein *Contextual Interview* lässt sich in vier Phasen unterteilen, genannt *Introduction*, *Transition*, *Field Interview* und *Wrap-up*. Nachdem in der *Introduction*, also der Einleitung, der Interviewten Ablauf und Ziele des Interviews kurz erklärt und die Vertraulichkeit ihrer persönlichen Daten versichert wurden, wird in der *Transition* zum eigentlichen Interview übergeleitet. Dieses dauert circa eineinhalb Stunden und besteht darin, die Interviewte bei ihrer Arbeit zu beobachten und sich diese dabei erläutern zu lassen. Der Begriff „Interview“ ist also missverständlich, da es sich nicht um eine klassische Frage/Antwort-Situation handelt. Dokumentiert wird der Arbeitsvorgang mittels einer circa 15 bis 20 Seiten langen informellen Mitschrift und außerdem durch das Sammeln der bei der Arbeit verwendeten *Artefakte*. Einerseits legt *Rapid CD* Wert darauf, auf den zu unterstützenden Task fokussiert zu bleiben, also nicht „abzuschweifen“, andererseits wird jedoch dazu geraten, sich bietende Möglichkeiten, wie sich vor Ort spontan ergebende anschließende Interviews, zu nutzen. Der *Wrap-up* nach dem eigentlichen Interview besteht in einer 10 bis 15 minütigen Zusammenfassung des Gehörten um noch mögliche offene Fragen zu klären und damit Missverständnisse auszuschließen.

Spätestens 48 Stunden nach dem Interview schließt die so genannte *Interpretation Session* an. Hier werden die gesammelten Informationen in einem *Cross-functional Team* von zwei bis sechs Personen erstmals ausgewertet und die Dokumentation der *Tasks* formalisiert. Resultat sind verschiedene Dokumente, im einzelnen *Userprofil*, *Organisationsprofil*, *Affinity Notes*, *Insights* und bei *Focused Rapid CD* zusätzlich noch ein oder mehrere *Work Models*.

Ein *Userprofil* enthält außer dem *Usercode* zur eindeutigen Identifikation der Benutzerin Angaben zu den so genannten *Job Roles* und zusätzliche, relevante demografische Daten (wie Berufsbezeichnung, Alter, Verantwortlichkeiten, Fertigkeiten, verwendete Software). Unter einer *Job Role* versteht man die Funktion oder Rolle der Benutzerin, die sie während des untersuchten *Tasks* einnimmt. Eine reale Person kann also je nach Kontext durchaus mehrere *Job Roles*

einnehmen, ebenso kann eine *Job Role* von mehreren unterschiedlichen Personen eingenommen werden.

Das *Organisationsprofil* enthält außer dem *Organisationscode* zur eindeutigen Identifikation eine Beschreibung des Geschäftsfeldes mit Angaben zu Kunden, Mitarbeitern, Produkten, Services, Orten und anderen im jeweiligen Kontext relevanten Informationen.

Die *Affinity Notes* bestehen in 50 bis 100 eigenständigen Punkten, die jeweils einen einzelnen Gedanken festhalten sollen. Sie bilden später die Grundlage für das *Affinity Diagram*, das wiederum Basis des eigentlichen Entwurfs ist. Prinzipiell wird bei ihrer Erstellung der Fragestellung nachgegangen, was innerhalb des *Contextual Interviews* im Einzelnen passierte und welche Aussagen man demnach über die einzelnen Arbeitsvorgänge innerhalb des *Tasks* treffen kann. Festgehalten werden Charakteristiken, Einflüsse, Fragen (auf englisch „Questions“, kurz *Qs*), nach wie vor unbekannte Sachverhalte und Zusammenhänge (so genannte blinde Flecken oder Löcher, auf englisch „Holes“) aber auch erste *Design Ideen* (kurz *DIs*), also bereits in diesem Stadium augenfällige Verbesserungsmöglichkeiten der Arbeitsvorgänge.

Insights werden gemeinsam am Flipchart entwickelt und dienen dazu die Detailebene der *Affinity Notes* zu verlassen und sich auf Basis des *Contextual Interviews* umfassendere Gedanken über Aufbau und Ausrichtung des zu entwerfenden Softwaresystems zu machen. Am Ende werden die *Insights* zu den *Affinity Notes* übertragen und mit ihnen vermischt.

Bei *Focused Rapid CD* werden zudem noch *Work Models* erstellt. In der Regel⁷

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
–	–	Sequence Models

handelt es sich hierbei um **Sequence Models**. In einem Flussdiagramm wird der schrittweise Arbeitsablauf der Benutzerin während des entsprechenden

⁷ Je nach Bedarf sind neben dem *Sequence Model* auch andere *Work Models* möglich: So modelliert ein *Artifact Model* Kontext, Herkunft, Wege und Verwendung der im *Task* verwendeten *Artefakte*. Der Schwerpunkt liegt hierbei auf der Darstellung der Arbeitskonzepte, die durch diese *Artefakte* repräsentiert werden. Ein *Physical Model* modelliert den Arbeitsplatz, die dort

Tasks formalisiert festgehalten. Hierbei wird zwischen Anstoßsituation des Arbeitsvorgangs (*Trigger*), Bedeutung eines Arbeitsvorgangs oder Schrittes (*Intend*), einzelnen Arbeitsschritten (*Work Steps, UI Steps*) und Unterbrechung des Arbeitsvorgangs (*Breakdown*) unterschieden.

Mehrere *Sequence Models* können in Zweiergruppen (so genannten *Pairs*) zu einem den *Task* komplett abdeckenden so genannten *Consolidated Sequence Model* zusammengefasst werden. Aus unterschiedlichen Schritten mit gleicher Bedeutung werden so genannte *Abstract Steps* generiert und diese dann wiederum zu *Activities* mit übergeordneten Gesamtintentionen zusammengefasst. Verschiedene Reihenfolgen, Strategien, Loops und mögliche *Breakdowns* innerhalb des *Tasks* werden so festgehalten.

Nach der Interviewphase, bei umfangreichen Projekten möglicherweise zum

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
Affinity Diagram		

Teil auch schon während dieser, wird mittels der *Affinity Notes* und unter Zuhilfenahme der *User-* und *Organisationsprofile* das **Affinity Diagram** erstellt. Dieses Diagramm stellt den ersten Entwurf des Softwaresystems dar. Die *Affinity Notes* und *Insights* werden einzeln ausgedruckt und im *Cross-functional Team* in einem Zeitraum von ein bis drei Tagen an der Wand eines geeignet großen Raumes (beispielsweise eines Konferenzraums oder einer Eingangshalle) im Prozess des so genannten *Affinity Buildings* gruppiert. Gearbeitet wird in einem gemeinsamen, offenen, nicht zentral kontrollierten Prozess. Man verlässt sich darauf, dass sich einzelne Teammitglieder oder Gruppen selbstständig auf bestimmte Themen fokussieren und die so entwickelten Ideen dann von anderen aufgegriffen, weiter verfolgt und letztendlich in größere Kontexte integriert werden.

verwendeten Geräte und Gegenstände, ihre Wege und Kontexte. Bei einem *Flow Model* werden Arbeitsgruppen, die einzelnen *Job Roles* innerhalb dieser und der zwischen ihnen stattfindende Informationsfluss modelliert. Ein *Cultural Model* schließlich modelliert die kulturellen (möglicherweise auch die juristischen) Besonderheiten und deren Einfluss auf die Arbeitsvorgänge.

Das fertige Diagramm hat eine vierstufige hierarchische Struktur: An oberster Stelle stehen grüne Labels, welche die fünf bis acht grundsätzlichen Einzelaufgaben des zu unterstützenden *Tasks* formulieren. Ihnen untergeordnet sind jeweils beliebig viele rosa Labels, beschriftet mit den Schlüsselpositionen der innerhalb dieser Aufgaben verwendeten Daten und *Artefakte*. Diese sind in der ersten Person aus Perspektive der Benutzerin formuliert. Jeder dieser Schlüsselpositionen sind zwei bis sechs blaue Labels untergeordnet, welche – ebenfalls in der ersten Person formuliert – die einzelnen Handlungen der Benutzerinnen bei der Verwendung und Manipulation der Daten und *Artefakte* in so genannten *Stories* zusammenfassen. Jeder dieser *Stories* sind dann ungefähr drei bis sechs einzelne *Affinity Notes* zugeordnet.

Erstellt wird ein *Affinity Diagram* nach der *Bottom-up*⁸-Methode, wobei darauf zu achten ist, die Inhalte einzelner Punkte in der jeweils übergeordneten Hierarchiestufe treffend zusammenzufassen und nicht durch nichts sagende Labels „zu verstecken“ – gruppiert wird nach Sinnzusammenhängen, nicht nach Schlüsselbegriffen oder gar Stichworten, zusammengefasst werden die einzelnen Punkte in Form der Situation in der sie gemeinsam stattfinden.

Bei *Focused Rapid CD* schließt nun die Erstellung von **Personas** und **User Scenarios** an. Diese sind hilfreich um die

Lightning Fast	Lightning Fast +	Focused Rapid CD
–	–	Personas und User Scenarios

bisher entstandenen Modelle innerhalb des nachfolgenden *Wall Walk* gut verständlich zu präsentieren. Eine *Persona* beschreibt in einem ausformulierten Text von etwa einer Seite eine typische Benutzerin des Systems. Zielsetzungen ihrer Arbeit, *Job Roles* und *Tasks* werden dargestellt – ebenso repräsentative demographische Daten. *User Scenarios* beschreiben typische Arbeitsstrategien der Benutzerinnen. Hierzu wird mit einer *Persona* ein *Sequence Model* gedanklich „durchwandert“ und die dabei deutlich werdende Herangehensweise als

⁸ „Von unten nach oben“, zunächst werden die in der Hierarchie am tiefsten stehenden Punkte angeordnet, dann unter dem jeweils nächsthöheren Punkt zusammengefasst, diese dann wiederum angeordnet usw.

ausformulierter Text festgehalten. Jedes *User Scenario* beschreibt jeweils nur eine Strategie. Sowohl *Personas* als auch *User Scenarios* werden trotz ihrer inhaltlichen Verallgemeinerung im Detail authentisch und realistisch formuliert.⁹

Nun werden die bisherigen Resultate im so genannten **Wall Walk** präsentiert.

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
Wall Walk		

Dies ist zum einen eine Möglichkeit um die gesammelten Informationen der Auftraggeberin zu präsentieren und deren Rückmeldung zu erhalten, zum anderen dient der *Wall Walk* dazu aus den bisherigen Materialien (*Affinity Diagram*, *Consolidated Sequence Models*, relevante *Artefakte*, *User Scenarios*, *Personas*) nun den Entwurf zu entwickeln. Der *Wall Walk* besteht im *Affinity Walk*, das heißt der Begutachtung und Auswertung des *Affinity Diagrams*, und bei *Focused Rapid CD* zusätzlich noch im *Sequence Walk*, der Begutachtung und Auswertung der *Sequence Models*. Beim Ersten wird das an der Wand angebrachte *Affinity Diagram* im *Cross-functional Team* gemeinsam still – also ohne Diskussion – betrachtet und dabei gedanklich „durchwandert“. Die einzelnen Punkte werden *top-down*¹⁰ gelesen und dabei von den Teammitgliedern Ideen in Form von *DIs* und *Qs* schriftlich mit Klebezetteln am entsprechenden Label festgehalten. Je höher eine solche Notiz innerhalb der Hierarchie des *Affinity Diagrams* positioniert ist, desto wichtiger ist diese. Die *Consolidated Sequences* werden hingegen in *Pairs* betrachtet und dabei diskutiert. Auch hier werden *DIs* und *Qs* festgehalten, unnötige Schritte innerhalb eines Diagramms eliminiert und bei Bedarf neue generiert. Die Ergebnisse des *Wall Walk* werden anschließend gemeinsam im *Cross-functional Team* diskutiert und dabei zu so genannten *Hot Ideas* und *Hot Key Issues* weiterentwickelt.

⁹ So wird beispielsweise dazu geraten für eine *Persona* einen Namen und ein zugehöriges Porträtfoto zu wählen. Trotz aller Verallgemeinerung soll dem typischen Benutzer dadurch quasi „ein Gesicht“ gegeben werden.

¹⁰ Im Gegensatz zu seiner Erstellung wird die *Affinity Diagram* „von oben nach unten“ gelesen, je besser also die in der Hierarchie höher stehenden Punkte die ihnen untergeordneten repräsentieren, desto einfacher und schneller ist der Inhalt des kompletten Diagramms zu erfassen.

Unter *Hot Ideas* versteht man umfassende, einen großen Teil des Arbeitsvorgangs abdeckende, also in der Hierarchie des *Affinity Diagrams* an möglichst hoher Stelle liegende *DIs*, während *Hot Key Issues* die von der Benutzerin vorrangig benötigten Funktionalitäten zur schnellen und problemlosen Durchführung des Arbeitsvorgangs bezeichnen und somit auch aus den *Qs* entwickelt werden können. Beide helfen den Entwurf weiter zu konkretisieren.

Der zweite Abschnitt dieser Phase von *Rapid CD* besteht im **Visioning** und

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
Visioning		

dauert ein bis zwei Tage. Zunächst finden im großen *Cross-functional Team* drei bis vier einzelne Sitzungen à 30 Minuten statt. In jeder Sitzung wird mit Hilfe der nach dem *Wall Walk* formulierten *Key Issues* und *Hot Ideas* gemeinsam eine so genannte *Vision* skizziert. Diese beschreibt, wie der zukünftige Arbeitsablauf der Benutzerin konkret aussehen könnte. Bei Bedarf werden neue *Job Roles* und Arbeitsvorgänge definiert, die Interaktion mit der *UI* und deren Funktionalität werden beschrieben, konkrete Technologien, Services, Supports und Schulungen festgelegt. Festgehalten wird die *Vision* als diagrammhafter Zeichnung, wobei darauf zu achten ist, nicht einzelne Details sondern vielmehr das gesamte System darzustellen. Je nach Bedarf und Kontext kann die graphische Darstellung einer *Vision* variieren. Am Ende jeder Sitzung werden die so genannten *Pluses and Minuses*, die – in Bezug auf die *Usability* des zu entwerfenden Systems – positiven und negativen Aspekte der zuvor skizzierten *Vision* festgehalten.

Die einzelnen *Visions* werden dann innerhalb eines halben Tages zu einer *Consolidated Vision* kombiniert wobei die *Pluses and Minuses* dazu dienen hierbei die besten Aspekte jeder einzelnen *Vision* auszuwählen. Bei *Lightning Fast* entspricht die *Consolidated Vision* bereits dem endgültigen Systementwurf und die Methode endet an dieser Stelle.

Bei *Focused Rapid CD* schließt nun die Phase des **Storyboarding** an. Mit Hilfe

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
–	–	Storyboarding

der *Consolidated Sequence Models*, der *Personas*, des *Affinity Diagrams* und der *Consolidated Vision* werden in *Pairs*, in 60 bis 90 Minuten andauernden Sitzungen, detaillierte Abläufe einzelner *Tasks* innerhalb des zukünftigen Software-systems – definiert durch die bearbeiteten *Consolidated Sequence Models* – in Form stark schematisierter Bildergeschichten modelliert. Man spricht hierbei von *Storyboards*. Nach Sichten des erforderlichen Materials wird pro *Task* zunächst eine so genannte *Low-level* oder *Mini Vision* (darunter versteht man eine detaillierte, nur einen kleinen, spezifischen Arbeitsvorgang abdeckende *Vision*) inklusive *Pluses and Minuses* erstellt. Mit Hilfe dieser und des zugehörigem *Consolidated Sequence Models* wird dann, mit Fokus auf Strategien, *Intends* und *Breakdowns*, das *Storyboard* erstellt. Dieses stellt die einzelnen neuen *Work Steps* oder *UI Steps* innerhalb des *Tasks* dar und beinhaltet grobe Skizzen der zukünftigen *UI*. Das *Storyboard* muss am Ende den kompletten im *Sequence Model* dargestellten *Task* abdecken, es dürfen also keine relevanten Informationen verloren gehen. Anschließend gibt es eine Präsentation der *Consolidated Vision*, *Storyboards* und *Consolidated Sequence Models* vor dem gesamten Team.

Die letzte Phase des *Rapid CD* besteht in den **Paper Mock-up Interviews**. Innerhalb von zwei bis drei Runden werden

<i>Lightning Fast</i>	<i>Lightning Fast +</i>	<i>Focused Rapid CD</i>
–	4-9 Paper Mock-up Interviews	6-12 Paper Mock-up Interviews

auf Basis der bisherigen Ergebnisse *Prototypen* der *UI* des Systems entwickelt – zunächst aus Papier, dann später in programmierter Form als *GUI*. Die Prototypen werden in Interviews am Arbeitsplatz der Benutzerinnen mit diesen getestet.

Die Interviews laufen prinzipiell ähnlich ab wie die zu Beginn des *Rapid-CD*-Prozesses durchgeführten *Contextual Interviews* und lassen sich in die selben Phasen unterteilen – *Introduction*, *Transition*, *Field Interview* und *Wrap-up*. Man benutzt beim Arbeiten mit dem *Mock-up*¹¹ möglichst echte Daten und versucht

¹¹ „Attrappe, Nachbildung, Modell“ gemeint ist ein in der Regel aus Papier gefertigtes Modell der *GUI*, dessen Reaktionen während des Interviews von der Interviewerin per Hand simuliert werden.

so die Arbeitsvorgänge so realistisch wie möglich zu simulieren. Während die Benutzerin das *Mock-up* bedient, kommentiert sie ihr Vorgehen. *On-The-Fly-Änderungen*¹² des *Mock-ups* während des Interviews sind möglich und erwünscht. Dokumentiert wird ein Interview durch einen zusätzlichen *Note-Taker*, der die einzelnen Schritte als Sequenzen festhält, außerdem werden prägnante Benutzerinnenzitate gesammelt.

Spätestens 24 Stunden nach einem Interview folgt, wiederum analog zu den *Contextual Interviews*, die *Interpretation Session*. Hier wird festgehalten welche Komponenten des *Prototyps* „stabil“ sind, also den gestellten Anforderungen gerecht werden und deshalb nicht mehr verändert werden müssen.

Im Laufe der drei Interviewrunden stabilisiert sich nach und nach der komplette Entwurf, so dass am Ende auf Basis der Benutzerinnendaten ein lauffähiger *Prototyp* der *GUI* entstanden ist, der in einer abschließenden Präsentation zusammen mit ausgewählten Materialien (wie prägnanten *Storyboards* und *User Scenarios*) der Auftraggeberin vorgeführt werden kann.

¹² Gemeint sind kurzfristige, spontane Änderungen während des Interviews mit dem Ziel neue Ideen direkt testen zu können.

4. Rapid Contextual Design als Agile Methode

Das 2001 verfasste **Agile Manifest**¹³ definiert agile Softwareentwicklung anhand von vier Grundsätzen, welche sich auf deutsch wie folgt formulieren lassen: „Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge. Funktionierende Software ist wichtiger als umfangreiche Dokumentation. Kooperation mit Projektbetroffenen ist wichtiger als Vertragsverhandlungen. Reaktion auf Veränderungen ist wichtiger als Festhalten an einem starren Plan.“¹⁴

Viele der einzelnen Arbeitsschritte innerhalb von *Rapid CD* (beispielsweise *Interpretation Session*, *Affinity Building* oder *Visioning*) werden im *Cross-functional Team* durchgeführt. Die Dynamik zwischen den Teammitgliedern, das gemeinsame Entwerfen und Ergründen von Ideen entsprechen dem ersten Grundsatz des *Agilen Manifests*. Die verwendeten Hilfsmittel sind zum größten Teil einfache Büromaterialien wie Stifte, Klebezettel und Flipcharts, bestimmte Software oder andere Werkzeuge können zwar verwendet werden, sind jedoch nicht notwendiger Bestandteil von *Rapid CD*.

Dokumentation ist ein wichtiger Anteil von *Rapid CD*. Informationen werden zunächst gesammelt (*Contextual Interviews*) und dann in einem mehrstufigen Prozess (*Interpretation Sessions*, *Sequence Modelling*) aufgearbeitet bis die Resultate (*Affinity Notes*, *Insights*, *Work Models*) als Grundlage für den eigentlichen Entwurf genutzt werden können. Die Implementierung schließlich bildet nur einen geringen Teil innerhalb der gesamten Methode. Zwar baut der Entwurf letztlich auf der Dokumentation auf (vgl. *Wall Walk*) und der Übergang zwischen beiden kann in gewisser Hinsicht als fließend betrachtet werden (vgl. *DIs*, *Insights*), generell dient *Rapid CD* jedoch eher der Erfassung und Auswertung von Benutzerinnendaten und Modellierung von Arbeitsabläufen und

¹³ Vgl. „Manifesto for Agile Software Development.“ <http://www.agilemanifesto.org>, Stand: 06.09.2008, 13:42 Uhr.

¹⁴ Bernd Oestereich, Christian Weiss: „APM – Agiles Projektmanagement. Erfolgreiches Timeboxing für IT-Projekte.“ dpunkt.verlag, Heidelberg 2008, S. 15.

Geschäftsprozessen als der möglichst schnellen Herstellung lauffähiger Software.

Innerhalb der beiden Interviewphasen ist die direkte Zusammenarbeit mit den Projektbetroffenen unverzichtbar. Basis des *Prototyps* sind die dort gesammelten Informationen. Wichtig ist also was die Benutzerin sagt und tut, nicht, was in Anforderungsspezifikationen zuvor festgelegt wurde. Alle Anforderungen an das System ergeben sich vielmehr erst im Laufe der Methode. Dies entspricht dem dritten Grundsatz des *Agilen Manifests*.

Rapid CD rät dazu, sich bietende Möglichkeiten innerhalb des Arbeitsprozesses (wie zusätzliche Interviews vor Ort oder *On-The-Fly*-Änderungen am *Mock-up*) spontan zu nutzen. Auch Methoden wie *Wall Walk* oder *Visioning* bieten Raum zu Spontaneität und flexibler Arbeitsweise. Ebenso dient das *Cross-functional Team* dem Zweck einen möglichst beweglichen Blickwinkel auf alle Teile des Projekts zu haben und bei Bedarf stets reagieren zu können. Dies entspricht dem letzten Grundsatz des *Agilen Manifests*.

In der Praxis existieren mehrere „klassische“ agile Verfahren, verbreitet sind beispielsweise *Extreme Programming* (XP) oder *Scrum*. Gemeinsam haben diese agilen Verfahren das inkrementelle Arbeiten in einzelnen, überschaubaren **Iterationen**, an deren Ende dann jeweils brauchbare Software steht. Softwareentwicklung lässt sich prinzipiell in vier Aktivitäten unterteilen – Analyse, Entwurf, Implementierung und Test. In der Analysephase werden die Anforderungen an das gewünschte Softwaresystem festgelegt. In der Entwurfsphase werden Lösungen konzipiert, die diesen Anforderungen gerecht werden. In der Implementierungsphase werden diese Lösungen in funktionierenden Programmcode umgesetzt und in der Testphase getestet ob die programmierte Software den in der Analyse gestellten Anforderungen gerecht wird.

Iteratives Vorgehen fußt auf der Beobachtung, dass es oft nicht möglich ist von Anfang an das Endergebnis eines Softwareprojektes klar abzusehen.¹⁵

¹⁵ Vgl. ebenda, S. 3 ff.

Demnach ergibt es oft keinen Sinn zu Beginn direkt alle Anforderungen zu bestimmen und, davon ausgehend, das System komplett zu entwerfen, dann zu implementieren und danach erst zu testen.¹⁶ Iterative Softwareentwicklung bietet eine Möglichkeit sich dem Endergebnis nach und nach anzunähern und nach jeder Iteration ein klareres Bild von diesem zu erhalten, welches einen wiederum Hinweise auf das Vorgehen innerhalb der folgenden Iterationen gibt.

Durch diese Strategie können auch innerhalb weiter fortgeschrittener Projekte noch relativ problemlos Anforderungsänderungen vorgenommen werden, Alternativen können erkannt und Risiken besser abgeschätzt und dadurch reduziert werden.

Rapid CD lässt sich keiner der beiden Vorgehensweisen klar zuordnen. Die Aufteilung größerer Projekte in einzelne hintereinander zu bearbeitende Teile kann als iteratives Vorgehen interpretiert werden und die Strategie innerhalb der *Paper Mock-up Interviews* ist eindeutig iterativ, jedoch ist der eigentliche Aufbau der Methode weitgehend kaskadenartig, Analyse und Entwurf folgen aufeinander, lediglich Implementierung und Test werden im Anschluss daran rundenweise durchgeführt. Im Ganzen betrachtet ist *Rapid CD* also keine inkrementell iterative Methode.

¹⁶ Wegen der kaskadenartigen Struktur spricht man hierbei von einem so genannten *Wasserfallmodell*.

5. Bewertung

Rapid CD geht von der Benutzerin und ihrer Arbeit aus. Auch wenn die Funktionalität eines Softwaresystems in *Wall Walk* und *Visioning* grob festgelegt wird, dient die Methode doch in erster Linie der Erstellung einer *GUI*. Während softwareinterne Funktionalitäten und Schnittstellen immer auf klar definierbaren Werten basieren, die mit Testklassen oder Leistungstests überprüft werden können, lassen sich *GUIs* letztendlich nur durch Anwendung und Beobachtung testen. Ebenfalls können *GUIs* nur eingeschränkt in einzeln kombinierbaren Modulen entworfen und implementiert werden, eine Bedienoberfläche muss immer als Ganzes betrachtet werden. Bestimmte agile Verfahren lassen sich hier also kaum anwenden.

Ein zentraler Punkt bei *Rapid CD* ist die umfangreiche Dokumentensammlung, die beim Ausführen der Methode entsteht. Viele Ressourcen werden bereits benötigt bevor man zu benutzbarer Software gelangt. Daher würde ich die Methode in gewisser Hinsicht als „unagil“ und schwerfällig bezeichnen.

Außerdem möchte ich den Beginn des eigentlichen Entwurfsprozesses als relativ konfus bezeichnen. Nach all dem vorherigen Aufwand verlässt man sich nun allein darauf, dass in der ungeordneten Dynamik des *Affinity Buildings* ein angemessener Entwurf entsteht. Dies halte ich für die größte Schwachstelle von *Rapid CD*. Der Erfolg der Methode steht und fällt mit dem Erfahrungsschatz und den Fähigkeiten des beschäftigten *Cross-functional Teams*. Ob ein ausreichend fähiges Team jedoch eine derart formalisierte Methode letztlich überhaupt nötig hat um ein gutes Ergebnis zu erzielen bleibt fraglich.

6. Literaturverzeichnis

Beyer, Hugh / **Holzblatt**, Karen: „*Contextual Design: Defining Customer-Centered Systems.*“ Morgan Kaufmann Publishers, San Francisco 1998.

Holzblatt, Karen / **Burns Wendel**, Jessamyn / **Wood**, Shelley: „*Rapid Contextual Design: A How-to Guide to Key Techniques for User Centered Design.*“ Morgan Kaufmann Publishers, San Francisco 2005.

International Organisation for Standardization (Hrsg.): „*ISO 13407. Human-centered design processes for iterative systems.*“ Genf 1999.

Oestereich, Bernd / **Weiss**, Christian: „*APM – Agiles Projektmanagement. Erfolgreiches Timeboxing für IT-Projekte.*“ dpunkt.verlag, Heidelberg 2008.

Website: „*Manifesto for Agile Software Development.*“ <http://www.agilemanifesto.org>
(Stand: 06.09.2008, 13:42 Uhr.)