

# Anforderungen klären mit Videoclips

Kurt Schneider

Lehrstuhl Software Engineering  
Leibniz Universität Hannover  
Welfengarten 1, 30167 Hannover  
Kurt.Schneider@inf.uni-hannover.de

**Abstract:** Viele große Softwaresysteme sind heute vernetzt, in Geräte eingebettet und mit Anzeigesystemen verbunden. Sie erscheinen den Nutzern als komplizierte, computergestützte Umwelt, deren Bestandteile kaum zu unterscheiden sind. Die Geräte und ihre Umgebung, die Software und die Bedürfnisse der Benutzer entwickeln sich während des Betriebs ständig weiter. Damit ein Systemteil nützlich und wettbewerbsfähig bleibt, braucht man fortwährend Rückmeldungen und Bewertungen. Die Techniken des klassischen Requirements Engineering reichen dazu aber nicht aus. In diesem Beitrag stellen wir einen Ansatz vor, um mit kurzen und einfachen Videoclips in dieser Situation an Feedback heranzukommen. Bei deren Auswertung werden aktuelle Anforderungen identifiziert und geklärt.

## 1 Einleitung

Viele technische Systeme interagieren heute eng miteinander: Navigationssysteme erhalten Stauinformationen über das Radio, der Lautsprecher wird leiser gestellt, sobald eine Verkehrsfunkanzeige empfangen wird. Im öffentlichen Raum (Bahnhof, Flughafen, Parkhäuser) haben es die Bürger als „Stakeholder“ mit immer enger verflochtenen Systemen von Systemen zu tun. Der Begriff des „IT-Ökosystem“ [NTH09] spielt auf diese Situation an, in der unabhängig entwickelte Systemteile teilweise autonom die Dienste anderer Teilsysteme in Anspruch nehmen. Stakeholder, also betroffene oder nutzende Menschen und Gruppen nehmen das Ganze als mehr oder weniger „intelligente“ Umwelt wahr.

### 1.1 Problemstellung: Anforderungen klären in bereits laufenden IT-Ökosystemen

Auch Geräte und Systeme in solch einer Umgebung müssen Anforderungen erfüllen. Durch die vielfältigen Interaktionen der Systeme und Stakeholder im Betrieb verändern sich diese Anforderungen ständig, ebenso wie die Umweltbedingungen. Softwareunternehmen müssten ständig die Leistungen ihrer Produkte im Gesamtsystem mit den aktuellen Anforderungen der Stakeholder abgleichen, um wettbewerbsfähig zu bleiben [Je08]. Doch dies ist aus mehreren Gründen schwierig:

- Klassisches Requirements Engineering mit Befragung von Stakeholdern in Interviews und Workshops ist äußerst aufwändig.

- Verschiedene Personengruppen und Personen können unterschiedliche Anforderungen an öffentliche Systeme haben. Für Anbieter eines softwaregestützten Systems sind dabei besonders gegenwärtige und potenzielle Kunden relevant. Einzelne aufwändig Interviewte repräsentieren kaum die ganze Zielgruppe. Eine größere Zahl von Personen einzubinden, erzeugt noch höhere Kosten für Marktforschung und Anforderungsinterviews.
- Bei der Befragung oder Validierung im Labor müssen sich Stakeholder an die Interaktion mit dem komplexen System erinnern oder sie nachvollziehen. Die Laborumgebung kann Beobachtungen verfälschen. Beispielsweise lässt sich die Verständlichkeit von Anzeigetafeln auf dem Bahnhof nur schlecht im Labor nachempfinden, weil die Umwelteinflüsse (Lärm, viele Menschen, Eile) fehlen.
- Durch die Interaktion mit neuen, autonomen Teilsystemen ändert sich auch das wahrgenommene Gesamtverhalten ständig. Ein störender Effekt ist kaum reproduzierbar, wenn er aus dem Zusammenwirken mit anderen Teilsystemen entstanden ist. Es ist unmöglich, alle Kombinationen und Interaktionen im Labor zu validieren, da die Umwelt dort nicht in gleicher Weise zur Verfügung steht.

Anforderungserhebung und -validierung in der realen Umgebung durchzuführen (z.B. im Bahnhof), erscheint daher als naheliegende Alternative. Doch bisher mussten teure Experten die Stakeholder beobachten, und die anderen obigen Probleme blieben bestehen.

## 1.2 Ein neuer Ansatz zum Umgang mit Videoclips für Feedback

Software in vernetzten Systemen und IT-Ökosystemen wird nur selten ganz neu erstellt. Meist sind mehrere konkurrierende Komponenten und Produkte im Einsatz, die nach und nach verändert oder ersetzt werden. Welche davon weiter genutzt (und bezahlt) werden, hängt davon ab, wie gut und schnell die Unternehmen den Bedürfnissen ihrer Kunden und Benutzer folgen können. Requirements Engineering entwickelt sich immer mehr von einer einmaligen Entwicklungsaktivität zu einer Tätigkeit, die den ganzen Lebenszyklus begleitet - und stützt. Mit dieser Situation beschäftigen sich auch das Produktmanagement [Je08] und das market-driven requirements engineering [Ka02].

In Zukunft wird die Wettbewerbsfähigkeit von **Unternehmen** noch stärker davon abhängen, ob sie Probleme und veränderte Anforderungen an ihre laufende Software rasch erkennen und umsetzen können [Ka02]. Unternehmen brauchen Rückmeldung und Verbesserungshinweise für ihre Software und ihre Geräte. Mit der Verbreitung von Fotohandys und Flatrates sinkt die Hemmschwelle für die vorgeschlagene Art von Rückmeldungen. Je mehr Stakeholder Feedback schicken, desto größer ist auch die Chancen, ungünstige Konstellationen in komplexen Systemen „zufällig“ zu entdecken. Wenn viele Bürger einfach Rückmeldungen geben können, eröffnet dies Unternehmen einen zusätzlichen Kommunikationskanal zu ihren Kunden, was zu verbesserter Kundenbindung führen kann - wenn die Kritikpunkte dann auch tatsächlich rasch beseitigt werden. Aber auch aus Sicht vieler **Bürger** hat der vorgeschlagene Ansatz Vorteile: Viele Bürger gehen durchaus kritisch mit den Angeboten um, die sie im öffentlichen Raum vorfinden, also in Bahnhöfen, Krankenhäusern und auf Flughäfen. Technisch Interessierte nehmen Fehlver-

halten von Systemen bewusst als solches wahr und sprechen (z.B. am Bahnsteig) über verwirrende Anzeigen und fehlende Funktionen. Viele wären vermutlich gerne bereit, eine kurze Rückmeldung zu geben, wenn sie das Gefühl hätten, damit Systeme verbessern zu helfen, die sie selbst ständig nutzen. Kaum jemand nimmt sich freilich die Zeit, Probleme und Beobachtungen später von zu Hause aus zu melden. Auf Bahnhöfen, Flughäfen und in ähnlichen hochtechnisierten Umgebungen treten dagegen ohnehin oft Wartezeiten auf – sie sind oft Teil des Problems. Diese Wartezeiten bieten eine gute Chance, an Rückmeldungen zu kommen: Manche Stakeholder sind sogar froh, ihren Ärger und ihre Anregung loswerden zu können – statt nur zu warten, sich zu ärgern oder zu langweilen. Kleine Anreize können als extrinsische Motivation hinzukommen, um sinnvolle Einsendungen zu honorieren. Nach Davenport [DP00] müssen Anreize nicht finanzieller Natur oder gar teuer sein. Wichtiger sind Anerkennung und schnelle Verbesserung der gemeldeten Missstände. Frustrationsabbau, aktive Partizipation an der Verbesserung selbst genutzter Systeme sind intrinsische Motivationen vieler technologieinteressierter Bürger. Diese win-win-Situation zwischen Bürgern und Unternehmen sollte man nutzen und fördern.

**In diesem Beitrag wird ein neuer Ansatz beschrieben, um mit einfachen Videoclips Rückmeldungen zu erfassen und Anforderungen zu klären.** Dazu werden ein Prozess und eine Grobarchitektur vorgeschlagen. Vorarbeiten aus verschiedenen Bereichen werden als Bestandteile genutzt, und neuartig kombiniert. Der Ansatz nutzt die drastisch veränderten Möglichkeiten durch Fotohandys, Flatrates und technologieaffine Bürger.

Zur Konkretisierung stellt dieser Beitrag auch ein neues Video-Bearbeitungswerkzeug vor, das in einer Masterarbeit eigens erstellt wurde, um einen wichtigen Teil des neuen Ansatzes noch gezielter zu unterstützen. Die Idee, Videos für die Anforderungsklä rung einzusetzen, ist nicht neu. Kapitel 2 stellt verwandte Arbeiten in verschiedenen Bereichen vor. Kapitel 3 fasst den Ansatz zur interaktiven Verwendung von Videoclips für die Anforderungsklä rung zusammen. In Kapitel 4 wird der Stand der Arbeiten zur Umsetzung und Evaluierung gezeigt und in Kap. 5 wird ein kurzes Fazit gezogen.

## 2 Verwandte Arbeiten

Requirements Engineering vermittelt zwischen Softwareentwicklern und verschiedenen Stakeholdern. Neben den klassischen Befragungstechniken, Interview und Workshop, werden für die Analyse komplexer Probleme und Situationen ethnographische Beobachtungsansätze im realen Umfeld empfohlen [Hu95]. Auch im Produktmanagement [Je08] und im market-driven requirements engineering [Ka02] müssen Anforderungen aus dem Markt erhoben werden. Karlsson et al. [Ka02] weisen auf die Bedeutung von Feedback und neuen Anforderungen im Umgang mit bereits existierenden Produktversionen hin.

**Feedback erheben durch aktive Beteiligung:** Unser Vorschlag setzt auf aktive Beteiligung der Stakeholder: Sie werden nicht beobachtet, sondern beobachten selbst. Vorbild sind Varianten des szenario-basierten Entwurfs, wie das Rollenspiel oder die so genannten „Playthroughs“ [AD02], bei denen man so tut, als hätte man ein gewünschtes Gerät bereits. Die dann mögliche Interaktion wird durchgespielt und auf diese Weise spezifiziert. Wirf-Brocks weist dabei Akteuren die Rollen „Benutzern“ und sogar „System“ zu [Wi95].

**Perspektivenwechsel zur Anforderungsklä rung:** Im Sinne der Anforderungsklä rung muss man versuchen, die relevanten Aspekte eines Rollenspiels oder Videoclips zu identifizieren [St73]. Dazu kann ein Perspektivwechsel nützlich sein. In ihrer Arbeit über Viewpoints haben Nuseibeh et al. diesen Punkt betont [NFK94]. Szenarien und Interaktionssequenzen sind essentielle Bestandteile vieler Spezifikationen in der Praxis [AM04, Co05]. Wie Alexander and Maiden darlegen [AM04], gibt es zahlreiche Typen von Szenarien im RE. Use cases beschreiben beispielsweise die Abfolge der Schritte eines Szenarios und fordern Zusatzinformationen wie Voraussetzung, Auslöser und Stakeholderinteressen ein [Co05].

**Anforderungen ohne klare Adressaten mit Videoclips erheben:** Wenn die Öffentlichkeit bzw. die Kunden Anregungen und Feedback geben sollen, dann muss die Darstellung weitgehend ohne fachspezifische Notation und Einarbeitung auskommen. Das ist bei Videoclips der Fall. Die heutige Situation unterscheidet sich zudem deutlich von 1998, als Haumer et al. [HPW98] den Einsatz von Videos für das Requirements Engineering untersucht haben. Sie schlagen vor, „rich media“ (Videos, Audio, Skizzen, Bilder) gemischt einzusetzen und auf einem Whiteboard Editor zu arrangieren. Dann können die Ergebnisse im Labor diskutiert werden. Bei Haumer et al. [HPW98] werden semi-formale Goal Models in Evaluierungssitzungen mit rich media verknüpft.

**Aufwändige Visionsvideos für Produkte und Einsatzszenarien:** Brügge und Creighton verwenden Visionsvideos. Diese werden in der Werbung, im Marketing und im Requirements Engineering benutzt. Ein bekannter Vertreter ist Apples Vision Video von 1987 (<http://www.youtube.com/watch?v=3WdS4TscWH8>). Bei Brügge und Creighton zeigt ein Visionsvideo beispielsweise, wie sich ein Hersteller die Nutzung eines Computertomographen vorstellt [COB06]. Die Visionsvideos können mit Sequenzdiagrammen angereichert werden, so dass man die relevanten Aspekte der Videos auf die formaleren Modelle abbilden kann. Man will auf diese Weise Abläufe mit „Normalbürgern“ und Stakeholdern validieren und sie dann in UML übertragen, so dass wertvolles Feedback gezielt an die Entwickler weitergegeben wird. Die professionellen Videos und ihre Erweiterung um UML-Modelle sind jedoch äußerst aufwändig und teuer. Ihre Erstellung dauert relativ lange und lohnt sich nur, wenn auch das Produkt entsprechend hohe Gewinne erwarten lässt. In [BSR08] wenden Brügge et al. ähnliche Techniken in der Software-Engineering-Lehre an, wobei wiederum aufwändige und teure Studientechniken, hier das „green screening“, genutzt werden. Ungeachtet des hohen Aufwands belegen die Arbeiten, dass Benutzer anhand von Videos komplexe Handlungsabläufe wirksam diskutieren können.

**Leichtgewichtige Videobearbeitung:** Einige Ansätze beschäftigen sich mit leichtgewichtigen Techniken für „Video-on-the-fly“: Hagedorn et al. [HHK08] stellen einen Ansatz zur Videoannotation vor, was ein Schritt in Richtung leichtgewichtiger Semantikanreicherung und Indizierung ist. Engström et al. beschreiben einen „video jockey (VJ)“, mit dem man interaktiv Videos arrangieren kann, in Analogie zu disc jockeys [EEJ08]. Videoclips eignen sich für einfaches Feedback und für Diskussionen, wenn man sie leicht rekombinieren kann. Maiden et al [Ma06] haben mobile Endgeräte eingesetzt, um zu studieren, wie man diese Geräte zur Anzeige von Checklisten und Fragen über Anforderungen verwenden kann. Wir wollen dagegen nicht Checklisten zeigen, sondern Videos mit solchen Endgeräten erfassen und gleich mit einigen semantischen Annotationen versehen.

### 3 Videoclips für Feedback: Idee, Prozess und Grobarchitektur

Etliche der oben angesprochenen Forscher versuchen, möglichst viel aus Videos und anderen „rich media“ herauszuholen. Sie sind bereit, dafür viel Aufwand in die Erstellung, Annotierung und Diskussion der Videos zu stecken. Alternativ kann man einfach Videos aufzunehmen und ohne jede Verarbeitung versuchen, daraus Anforderungen abzulesen. Dabei wird der Aufwand für die Videoersteller zwar niedriger, die Auswertung ist dagegen umso schwieriger.

Es gab jedoch bisher kein Konzept, wie Videoclips zwar sehr einfach erhoben, andererseits aber aussagekräftig ausgewertet werden können. Anforderungserhebung mit Fotohandys während des laufenden Betriebs erfordert eine durchgängige Idee und einige neue Komponenten. Darauf geht dieser Abschnitt ein.

#### 3.1 Idee: Spezialwerkzeuge erleichtern Feedback und Auswertung

Unser Ansatz liegt in der Mitte zwischen den beiden oben angesprochenen Extremen: Der Aufwand für die Stakeholder soll minimiert werden. Wir investieren dafür in der Forschung relativ viel Aufwand in die Entwicklung von Werkzeugen, die sowohl die Formulierung als auch die Auswertung von Feedback erleichtern. Vollständige Erfassung aller Anregungen und Wünsche ist dagegen hier nicht das Ziel. Man sollte realistischerweise davon ausgehen, dass ein normaler Bürger nur dann ein Video erstellen und einsenden wird, wenn er sich dafür unmittelbare Vorteile verspricht und keine wichtige Tätigkeit unterbrechen muss. Solche Situationen gibt es! Rückmeldungen reagieren in der Regel auf Unterbrechungen und Störungen, wenn die beabsichtigte Tätigkeit ohnehin gestört ist (z.B. Ticket kaufen, Bahn fahren). Es ist für unseren Ansatz wichtig, solche Gelegenheiten zu identifizieren und auszunutzen.

Die Grundprinzipien unseres Ansatzes lauten:

- Die Schwelle, einen Videoclip einzusenden, muss in jeder Hinsicht niedrig sein.
- Videoclips werden nicht professionell bearbeitet und geschnitten. Vielmehr werden sie interaktiv arrangiert, wobei ein Editorwerkzeug hilft.
- Idealerweise enthalten Videoclips auch Metadaten zu Aufnahme, Autor und Absicht, sowie einigen anderen Aspekten, die in einem speziell angefertigten Editorwerkzeug für die Montage und Diskussion genutzt werden können.
- Dabei spielt der Kontext der Rückmeldung eine wichtige Rolle. Er liefert semantische Informationen, um den oder die Adressaten eines Videos zu ermitteln. Eine kurze, simple Interaktion ist dabei akzeptabel, aber nicht mehr.
- Wir nutzen in der Auswertung verschiedene Möglichkeiten, darunter die Gegenüberstellung unterschiedlich formaler Modelle: einerseits Videoclips und rich media, andererseits Use Cases oder ähnliche semi-formale Modelle.

Diese Prinzipien müssen in konkrete Handlungsabläufe (Prozesse) umgesetzt werden.

### 3.2 Prozess und Arbeitsschritte der Anforderungsklärung mit Videoclips

Abb. 1 zeigt einen Überblick über die wesentlichen Schritte beim Einsatz von Videoclips für die Anforderungsklärung. Stakeholder müssen zunächst überhaupt wahrnehmen, dass ein Problem vorliegt oder sie einen Verbesserungsvorschlag haben. Dann wird mit der Videokamera oder dem Handy etwas aufgezeichnet. Bevor dieses Video eingesandt werden kann, muss der geeignete Adressat identifiziert werden – an wen soll das Video also geschickt werden? Der Empfänger ist in der Regel ein Softwareunternehmen, das mit Hilfe der eingehenden Rückmeldungen einen bestimmten Systemteil überarbeitet und verbessert.

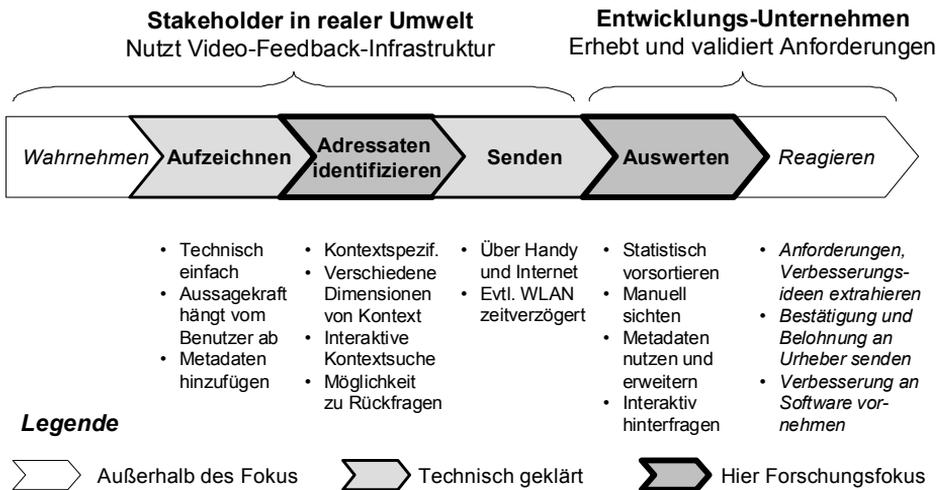


Abbildung 1: Arbeitsschritte der Anforderungsklärung mit Videoclips aus Anwendersicht

Der Forschungsfokus unserer Arbeiten liegt auf dem Gesamtablauf und darin besonders auf den hervorgehobenen beiden Schritten. In *diesem Beitrag* werden die Herausforderungen beider Schritte genannt, aber aus Platzgründen wir nur zur Auswertung auch ein eigens entwickeltes Werkzeug angesprochen.

Die Aktivitäten **Reagieren** und **Wahrnehmen** können hier nicht vertieft diskutiert werden; sie sind aber für das Funktionieren des Ansatzes unverzichtbar. Die ausgewerteten Videoclips werden als Anforderungen interpretiert und zu Change Requests umformuliert. Die ausgewählten Anregungen werden schließlich in die Software eingearbeitet. Am Ende sollte eine Bestätigung bzw. „Belohnung“ der Inputgeber erfolgen. Die eigentliche **Videoaufzeichnung** sowie das **Versenden** von Fotos und Videos stellen heute weder ein technisches noch ein finanzielles Problem dar. Im Idealfall entstehen für den Rückmeldungsgeber keinerlei Kosten. Manche Kunden haben ohnehin Flatrates. Auch kann man Rufnummern einrichten, die für Anrufer kostenlos sind. Um die Bandbreitenunterschiede zu berücksichtigen, kann es sinnvoll sein, ein Video erst dann (verzögert) zu übertragen, wenn sich das Mobilgerät wieder in der Reichweite eines WLAN befindet.

### 3.3 Grobarchitektur für die Unterstützung der wichtigsten Schritte

Abb. 2 zeigt die Verteilung zwischen einer Zentrale, den Repräsentanten für registrierte Teilsysteme eines Anbieters, und den Endgeräten in Benutzerhand. In den registrierten Handys, Kameras und PCs sind Komponenten enthalten, die über das Internet mit der Zentrale und den Repräsentanten für Teilsysteme verbunden sind. Die Anbieter-Komponenten werden von den teilnehmenden Unternehmen betrieben und stehen ihnen für die Auswertung zur Verfügung. Der Verbesserungsprozess unten im Bild stellt die Auswirkung auf die Software in der realen Umgebung dar. Stakeholder erfahren dadurch eine Verbesserung und können auf den neuen Zustand wieder reagieren.

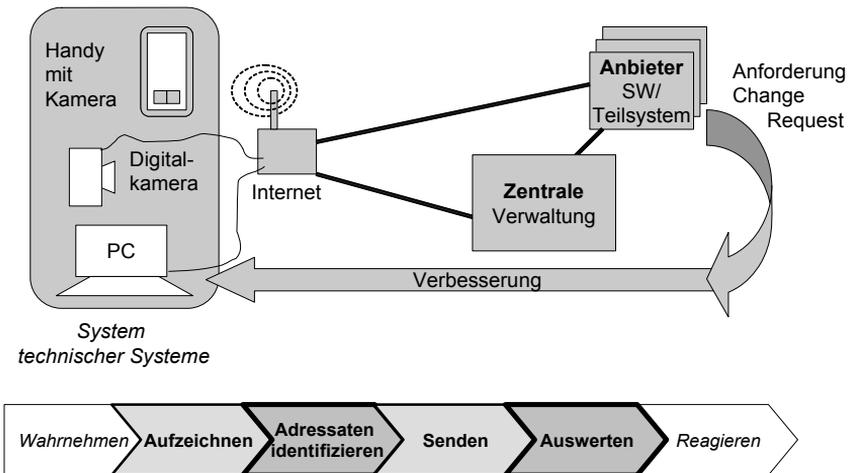


Abbildung 2: Verteilung der Arbeitsschritte auf Komponenten (Grobarchitektur)

Wenn ein Unternehmen seine Softwareprodukte bei der Zentrale registriert, wird dabei auch festgelegt, unter welchen Umständen sie als möglicher Kontext für eine Rückmeldungen betrachtet werden. So lassen sich für die Rückmeldung **Adressaten identifizieren**. Dabei spielen mehrere Dimensionen des Kontextbegriffs zusammen:

- Geographische Nähe, wenn der Stakeholder z.B. vor dem Bankautomaten steht.
- Logische Nähe, wenn im Handy gerade ein Programm geöffnet ist, z.B. bei der Online-Reservierung von Fahrkarten.
- Sender (WLAN, Bluetooth), mit denen der Stakeholder gerade interagieren kann.

Es gibt unterschiedliche Hypothesen, welchen Kontext ein Stakeholder meint, wenn er in einer gewissen Situation Feedback gibt. Stakeholder wissen oft nicht, welche Teilsysteme überhaupt existieren und ob sie für das Gesamtsystemverhalten eine Rolle spielen. Beispielsweise weiß ein Bahnkunde in der Regel nicht, wer im Bahnhof für welche Anzeigetafel zuständig ist. Dagegen kann er durchaus beantworten, ob er gerade eine Rückmeldung zur *geographisch nahen* Anzeigetafel, zur gerade *im Handybrowser*

geöffneten Online-Reservierung oder zum *physischen* Bahnhofsgebäude formulieren möchte. Bei diesem Kontext-Suchvorgang bewegt man sich heuristisch vom Speziellen, Nahen und Naheliegenden zu größeren und generischeren Kontexten, bis der Benutzer durch Tastendruck bestätigt, was er gemeint hat. Wir verwenden diese Heuristik, rechnen aber damit, sie noch verbessern zu müssen.

Das **Auswerten** der eingesandten Videoclips liefert die eigentlichen Anforderungen und Change Requests. Daher liegt diese Tätigkeit im Fokus unseres Ansatzes und dieses Beitrags. Wie oben bereits ausgeführt, sollen dazu Techniken eingesetzt werden, die die Stakeholder und die Anforderungsanalytiker wenig belasten. Einerseits müssen Gelegenheiten zum automatischen Kennzeichnen (Tagging) genutzt werden. Grundlegende Angaben (Absender, Ort, Uhrzeit, evtl. GPS-Standort) kann man technisch automatisiert ermitteln und dem Video als Tags mitgeben. Bereits mit solchen Angaben lassen sich Videos vorsortieren, gruppieren und gezielter nachbearbeiten. Zusätzlich ist es wichtig, Videos mit Audio-Erläuterungen zu versehen: Ein Bahnkunde kann eine defekte Anzeige filmen - und sollte dabei erklären, was daran gerade auffällig und besonders problematisch für ihn ist. Aus diesen Audioannotationen kann man später über die automatisch vergebenen Tags hinaus semantische Angaben extrahieren. Weitergehende Auswertungen sollen durch Spezialwerkzeuge ermöglicht werden; verwandte Beispiele sind [Sc07, Sc08].

#### 4 Stand der Umsetzung

Der wichtigste Beitrag dieses Artikels besteht darin, den Ansatz für einfaches Feedback mit Videoclips vorzustellen. Derzeit sind verschiedene Schritte des Ansatzes in Arbeit. Abb. 3 zeigt noch einmal die Schritte des Ansatzes und gibt an, an welchen Aspekten derzeit an der Leibniz Universität Hannover gearbeitet wird. Dazu gehören auch erste Versuche, den Ansatz zu evaluieren. Eine ausführliche Evaluierung soll sich anschließen. Die annotierten Angaben werden nun anhand von Abb. 3 von links nach rechts erläutert. Zuletzt wird der Spezialeditor etwas ausführlicher dargestellt.

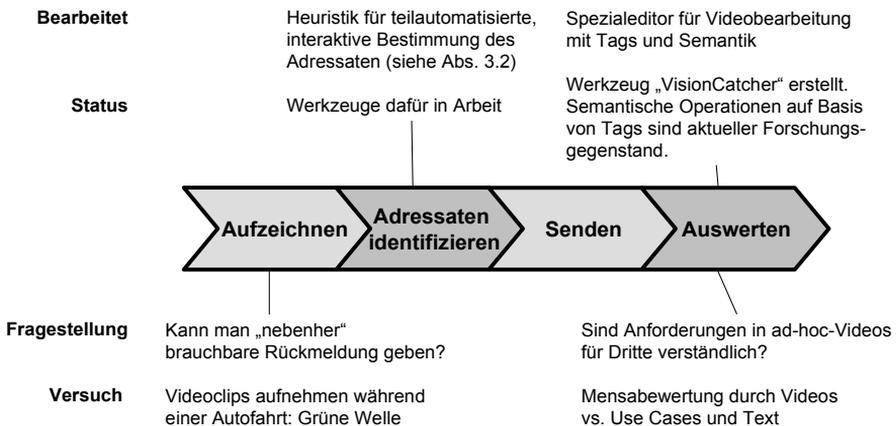


Abbildung 3: Übersicht über laufende Arbeiten, Status und Evaluierungsaktivitäten

**Kann man „nebenher“ brauchbare Rückmeldung geben?** Hierfür wurden kurze Videoclips während einer Autofahrt aufgenommen. Aus der Position des Beifahrers wurden Fahrsituationen, der Fahrer selbst und ein fiktives, erwünschtes Gerät gefilmt. Teile davon sind in Abb. 4 zu sehen. Besonders beim Warten an Ampeln hat der Fahrer Kritik und Wünsche gesprochen, während gefilmt wurde. Während der ca. 20-minütigen Fahrt entstanden 18 kurze Videoclips. Das Video wurde in weniger als einer Stunde mit Mockups eines „Grüne-Welle-Assistenten“ kombiniert und dann bei verschiedenen Gelegenheiten gezeigt. Die Zuschauer konnten die Intention erkennen, wiedergeben und sie waren in der Lage, Alternativen oder weitere Anforderungen für den Assistenten zu formulieren.

**Heuristik für halbautomatisierte Identifikation von Adressaten:** Die Videoclips werden mit handelsüblichen videofähigen Handys (oder Digitalkameras) aufgenommen. Derzeit laufen studentische Softwareprojekte, in denen die teils automatisierte, teils interaktive Identifikation des besten Adressaten für eine Rückmeldung implementiert wird. Dafür stehen Android-Handys mit integrierter Videofunktion zur Verfügung. In diesem Schritt muss die Architektur aus Abb. 2 konkretisiert und eingesetzt werden. Der identifizierte Adressat, also das angesprochene Teilsystem aus Abb. 2, soll bis zu drei kurze Rückfragen stellen können, wenn es Feedback erhält. Sie sollen möglichst durch einfache Auswahl oder ja/nein zu beantworten sein, um den Aufwand für den Feedback-Geber gering zu halten. Durch solche vorstrukturierten Zusatzangaben ist es möglich, große Mengen von Feedback zu kanalisieren, sie statistisch vorauszuwerten und auch die Videoclips automatisch mit Anmerkungen bzw. Tags zu versehen.

**Sind Anforderungen in einem ad hoc-Video für Dritte verständlich?** Eine Person erklärte einer Gruppe von vier Doktoranden die Idee eines „Mensa-Bewertungssystems“. Dann sollten zwei der Doktoranden das System mit Use Cases und Text beschreiben, die beiden anderen sollten mit Hilfe einer Videokamera ein kurzes Video drehen. Dafür standen jeweils 30 Minuten zur Verfügung, anschließend noch 15 Minuten für den Schnitt auf dem Storyboard bzw. Ausformulieren der Texte. Zwei Softwareentwickler (als Empfänger der Anforderungen) und zwei Studierende, die oft in die Mensa gehen (als Stakeholder) wurden unabhängig gebeten, auf der Basis des Videos bzw. der Use Cases wesentliche Eigenschaften des beschriebenen Systems zu identifizieren (speak-aloud). Dann wurden noch Fragen gestellt. Die wichtigsten 13 Anforderungen waren zuvor schriftlich festgehalten worden, um als Referenz zu dienen. Die Zahl der erkannten Anforderungen lag in beiden Fällen bei knapp über der Hälfte, was für die kurze Erstellungszeit akzeptabel ist. Das bestätigt, dass auch sehr kurzfristig und schnell erstellte Videos wirksam Anforderungen transportieren können. Wie zu erwarten, konnten Entwickler mehr Anforderungen identifizieren als Stakeholder; Entwickler haben das gelernt. Im subjektiven Urteil schnitten die Videos bei den Entwicklern erstaunlicherweise sogar besser ab als bei den Stakeholdern.

**Spezialeditor für Videobearbeitung mit Tags und Semantik.** Eine Herausforderung bei der Auswertung besteht darin, einerseits den Aufwand für alle Beteiligten gering zu halten, und andererseits doch hilfreiche Schlüsse aus den Videoclips zu ziehen.

Zu diesem Zweck sollten ad-hoc-Videos mit Annotationen versehen und während eines Interviews validiert und nachbearbeitet werden. Dazu wurde eine Masterarbeit definiert,

in der die Kernfunktionalität für Videoclip-Kombination mit Funktionen zum Annotieren („Tagging“) integriert werden sollte. Kitzmanns rich media-Mischpult mit dem Namen „VisionCatcher“ greift die Idee des Whiteboard Editors auf [HPW98]. Verschiedene Medien (Videoclips, Fotos, Handskizzen, Audio usw.) können einfach eingelesen und neu angeordnet werden. Längere Videos werden zunächst in kurze Clips zerlegt. So lassen sich leichter Varianten kombinieren und vergleichen. In Interview und Auswertung arbeiten Softwarefachleute des Unternehmens und können bei Bedarf einzelne Kunden gezielt zu ihrer Interpretation und ihren Präferenzen befragen. Dazu braucht man viel weniger Zeit als bei einer „Schrotschuss-Befragung“ ohne vorherigen Videoinput.

Kitzmanns Mischpult soll auf einem TabletPC während der Interviews eingesetzt werden. Es ähnelt auf den ersten Blick kommerziellen Storyboards (Abb. 4), Bedienbarkeit der Hauptfunktionen war hier aber wichtiger als ein großer Funktionsumfang. Links ist die Palette von Videoclips, Bildern usw. zu sehen. Durch drag-and-drop kann man diese Elemente auf den Filmstreifen (im unteren Bildteil) ziehen. Den ganzen Filmstreifen kann man sofort abspielen: Mit der Play-Taste oben startet man den Film, der sich aus den Clips, Skizzen und anderen Elementen zusammensetzt. Standbilder sind eine vordefinierte Zeit lang zu sehen. Der Film wird rechts im Bild angezeigt. In Abb. 4 läuft gerade der mittlere Videoclip. Er ist im Filmstreifen „aufgeklappt“ und kann bearbeitet werden:



Abbildung 4: Der VisionCatcher von Kitzmann [Ki09] für Videoclipbearbeitung mit Tags

Bei der Auswertung entwickeln sich häufig interessante Diskussionen. Diese kann man über ein Mikrofon aufzeichnen (siehe runder Aufnahmeknopf oben). Außerdem kann man am TabletPC jederzeit mit dem Stift (rechts) Annotationen auf allen gezeigten Elementen

vornehmen, also „dazu schreiben“. Sie werden mit den Diskussionsaufnahmen und dem Film zusammen gespeichert. Dies fördert Interviews und anschließende Auswertung. Schon damit ist VisionCatcher ein Spezialwerkzeug zur Unterstützung des hier vorgestellten Ansatzes. Kitzmanns Arbeit [Ki09] illustriert darüber hinaus, wie semantische Annotationen (Tags) in diesem Prozess eingesetzt werden können. In Abb. 4 ist gerade die Filterfunktion aktiviert, das zugehörige Teilfenster überdeckt teilweise die Palette. Die Palette zeigt nur solche Elemente, die zu den gewählten Tags passen. Tags stehen unter den Palettenelementen.

Gerade ist ein administrativer Tag „KPS-2.3.09“ (Absender-ID und Datum) ausgewählt. Nur Elemente mit diesem Tag werden in der Palette angezeigt. Die Palette kann durch Auswahl weiterer Tags weiter eingeschränkt werden. Besonders wertvoll sind in diesem Arbeitsschritt auch explizite Audiokommentare. Der Fahrer hat im obigen Beispiel beim Warten an der roten Ampel mündlich erklärt, wieso er hier eine Grüne Welle erwartet hätte. Relevante Kommentare kann man manuell in Tags umsetzen.

Das dritte Element im Filmstreifen („Grüne Welle“) ist ein Mockup, also eine Skizze, mit dem der Interviewer eine Lösung für die Grüne Welle vorschlägt: Der Assistent im Auto empfiehlt die Geschwindigkeit, bei der auch die nächste Ampel bei Grün erreicht wird. Im Interview können Stakeholder zu diesem Vorschlag Stellung beziehen. Dieses Beispiel stammt aus dem Forschungsprojekt „IT-Ökosysteme“ [NTH09], in dem der hier beschriebene Ansatz zur Kontrolle eng vermaschter IT-Systeme verwendet werden soll.

## 5 Schlussfolgerungen

Videos wurden schon früher im Requirements Engineering eingesetzt. Der hier vorgestellte Ansatz geht aber von einer veränderten Situation in der Gesellschaft aus und nutzt sie. Handys mit Videokamera und Flatrates haben viele technikinteressierte Bürger zu ernstzunehmenden Stakeholdern bei der Verbesserung von Softwaresystemen gemacht.

Ein leichtgewichtiger und durch Werkzeuge gut unterstützter Ablauf wird vorgeschlagen. In diesem Beitrag werden die Schritte identifiziert und diskutiert, die zur Klärung von Anforderungen durch Videoclips führen. Werkzeuge sind dazu erforderlich. Daher wird eine Grobarchitektur für die Kommunikation vorgestellt und ein speziell entwickeltes Werkzeug zur Auswertung gezeigt. Kitzmanns VisionCatcher ist in einer Masterarbeit an der Leibniz Universität Hannover entwickelt worden.

In diesem Beitrag sollte die Vision und der Ansatz von einfachen Rückmeldungen und dennoch reichen Auswertungsmöglichkeiten gezeigt und anhand eines speziell dafür entwickelten Werkzeugs exemplarisch konkretisiert werden. Dadurch wird aber auch deutlich, dass noch erhebliches Forschungspotenzial besteht, um den Ansatz in allen Schritten umzusetzen und auszubauen. Die Herausforderungen reichen von verbesserten Heuristiken für die Identifikation von Kontext und Adressaten über intelligenteres Tagging bis hin zu semantisch reicheren Auswertungsmechanismen. Auch statistische Auswertung von Videoclips mit Tags bieten sich an. An diesen Aspekten werden wir weiterarbeiten.

## Literaturverzeichnis

- [AM04] Alexander, I.F. and N. Maiden, eds. Scenarios, Stories, Use Cases. Through the Systems Development Life-Cycle. 2004, John Wiley&Sons: Chichester, England.
- [AD02] Austin, R., Devin, L.: Beyond requirements: software making as art. IEEE Software, 2002. 19.
- [BSR08] Brügge, B., Stangl, H., Reiss, M.: An Experiment in Teaching Innovation in Software Engineering. in OOPSLA' 08. 2008. Nashville, Tennessee, USA.
- [Co05] Cockburn, A.: Writing Effective Use Cases. 14th Printing ed. 2005: Addison-Wesley.
- [COB06] Creighton, O., Ott, M., Brügge, B.: Software Cinema: Video-based Requirements Engineering. in 14th IEEE Internat. Requirements Engineering Conference. 2006. Minneapolis/St. Paul, Minnesota, USA.
- [DP00] Davenport, T., Probst, G.: Knowledge Management Case Book - Best Practises. 2000, München, Germany: Publicis MCD, John Wiley & Sons. 260.
- [EEJ08] Engström, A., Esbjörnsson, M., Juhlin, O.: Mobile Collaborative Live Video Mixing. in MobileHCI 2008. 2008. Amsterdam, the Netherlands.
- [HHK08] Hagedorn, J., Hailpern, J., Karahalios, K.: VCode and VData: Illustrating a new Framework for Supporting the Video Annotation Work. in AVI 08. 2008. Naples, Italy.
- [HPW98] Haumer, P., Pohl, K., Weidenhaupt, K.: Requirements Elicitation and Validation with Real World Scenes. IEEE Transactions on Software Engineering, 1998. 24(12): p. 1036-1054.
- [Hu95] Hughes, J., O'Brian, J., Rodden, T., Rouncefield, M., Sommerville, I.: Presenting ethnography in the requirements process. in Second IEEE International Symposium on Requirements Engineering, March 27 - 29, 1995. 1995. York, England: IEEE Computer Society.
- [Je08] Jesse, S.:Tutorium: Erhebung von Produktanforderungen durch den Requirements Engineer. in Industrialisierung des Software-Managements. 2008.
- [Ka02] Karlsson, L., Dahlstedt, A., Natt och Dag, J., Regnell, B., Persson, A.: Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study. in Eighth International Workshop on Requirements Engineering: Foundation for Software Quality. 2002. Sept 9-10th, 2002 Essen Germany.
- [Ki09] Kitzmann, I.: Konzept und Implementierung eines Werkzeugs für multimediale Anforderungserhebung und -validierung, in Masterarbeit, FG Software Engineering. 2009, Leibniz Universität Hannover.
- [Ma06] Maiden, N., Seyff, N., Grünbacher, P., Otojare, O., Mitteregger, K.: Making Mobile Requirements Engineering Tools Usable and Useful. in Requirements Engineering Conf., 2006. RE 2006. 14th IEEE International. 2006. Minneapolis.
- [NTH09] NTH: IT Ökosysteme: Autonomie und Beherrschbarkeit Software-intensiver Systeme - Projekt der Niedersächsischen Technischen Hochschule (NTH). 2009.
- [NKF94] Nuseibeh, B., J. Kramer, and A. Finkelstein: A framework for expressing the relationships between multiple views in requirements specification. IEEE Transactions on Software Engineering, 1994. 20(10): p. 760-773.
- [Sc07] Schneider, K.: Generating Fast Feedback in Requirements Elicitation. in Requirements Engineering: Foundation for Software Quality (REFSQ 2007). 2007. Trondheim, Norway.
- [Sc08] Schneider, K.: Improving Feedback on Requirements through Heuristics. in 4th World Congress for Software Quality (4WCSQ). 2008. Bethesda, Maryland, USA.
- [St73] Stachowiak, H.: Allgemeine Modelltheorie. 1973, Wien, New York: Springer Verlag.
- [Wi95] Wirfs-Brock, R.: Designing Scenarios: making the case for a use case framework. The Smalltalk Report, 1995. 3 (3): p. 7-10.