

Wie testet man soziale Software und deren soziale Mechanik?

Neue methodische Aspekte und Diskussion eines komplexen Untersuchungsgegenstandes.

Dr. Theo Held
SAP AG, UX
Dietmar-Hopp-Allee 16
69190 Walldorf
theo.held@sap.com

Jürgen D. Mangerich
SAP AG, UX
Dietmar-Hopp-Allee 16
69190 Walldorf
j.mangerich@sap.com

Abstract

Soziale Software und die ihr zugrundeliegenden interaktiven Prinzipien sozialer Mechanik gewinnen in nahezu allen Bereichen der Softwareindustrie zunehmend an Bedeutung. Neben der Frage nach einem attraktiven und gut nutzbaren Design solcher Software gewinnt natürlich auch eine dem Gegenstand angemessene Evaluation immer mehr an Bedeutung. Schon bei oberflächlicher Betrachtung wird klar, dass klassische Untersuchungsverfahren wie formative Usability Tests nur in erweiterter oder modifizierter Form die Spielarten sozialer Mechanik erfassen und zugrundeliegende Fragestellungen beantworten können.

In unserem Beitrag stellen wir einen methodischen Ansatz vor, der weitgehend auf spezielle Aspekte sozialer Software abgestimmt ist. Die Anwendung des Ansatzes illustrieren wir an einem praktischen Beispiel. Wesentlicher Bestandteil des Workshops soll es aber sein, weitere Aspekte zu dieser Thematik zu sammeln und zu diskutieren und darauf aufbauend das einschlägige Methodenrepertoire zu erweitern und somit Praktikern unmittelbar verwendbare Werkzeuge zur Evaluation sozialer Software an die Hand zu geben.

Keywords:

/// Soziale Software
/// Soziale Mechanik
/// Usability Test
/// Paper Prototyping
/// kollaborative Szenarien

1. Einführung und Fragestellung

Mit der enormen Verbreitung und dem großen Erfolg „Sozialer Software“ liegt es nahe, typische Elemente sozialer Software in Business Software zu integrieren. Unter Sozialer Software verstehen wir in diesem Rahmen alle Softwareprodukte oder deren Bestandteile, die die Kommunikation oder Kollaboration von Nutzern fördern oder unterstützen. Der Begriff „soziale Mechanik“ steht in diesem Kontext für die Grundprinzipien sozialer Interaktion, die innerhalb sozialer Software zum Tragen kommen und diese letztendlich zu sozialer Software machen. Beispiele für solche Prinzipien sind „Teilen von Information“, „Kollaboration mit sozialen Kontaktpersonen“ oder auch „Belohnung von Verhalten“. Interessanterweise wurde der Begriff „soziale Mechanik“ von den Soziologen Leon Winiarski und Lester Frank Ward geprägt, die bereits Anfang des zwanzigsten Jahrhunderts versuchten, die Prinzipien der Thermodynamik in Analogie auf

menschliches Sozialverhalten zu übertragen (Ward, 1900; Winarski, 1900).

Zu typischen Elementen Sozialer Software, die von Prinzipien sozialer Mechanik Gebrauch machen, zählen kollaborative Funktionen wie virtuelle Diskussionsräume oder auch dynamische Kommunikationsmöglichkeiten wie Feeds (d. h. automatisch aktualisierte Listen von Informationseinheiten und Kommunikationsbeiträgen). Mit der Verwendung solcher Funktionen geht stets die Frage einher, wie diese Funktionen und deren fundamentale Konzepte in sinnvoller und realitätsnaher (d. h. ökologisch valider) Weise empirisch evaluiert werden können. Insbesondere muss überlegt werden, ob neue Evaluationsmethoden oder Modifikationen von Standardmethoden erforderlich werden.

In klassischen formativen Verfahren wird üblicherweise ausgehend von einer Reihe inhaltlicher Fragestellungen eine Menge von Aufgaben konstruiert, deren Anforderungen und Lösungswege Aufschluss

über die Fragestellungen geben sollen. Typischerweise liegt ein wohl definierter Ausgangszustand vor, der mit Hilfe der zu testenden Software in einen ebenfalls bekannten Endzustand überführt werden soll. Der Ansatz eines klassischen formativen Tests ist somit grundsätzlich determiniert. Für den Weg vom Ausgangszustand zum Ziel existiert meist eine kleine Menge möglicher Alternativen. Es liegt nun nahe, die Sachlage aus dem Blickwinkel der traditionellen Problemlöseforschung zu betrachten. Im Sinne der Problemtypen nach Dörner (1987) liegt bei solchen Tests lediglich eine Synthesbarriere vor, d. h. die Mittel der Zielerreichung sind – im Gegensatz zum Ziel – zunächst unbekannt.

Im Falle sozialer Software hingegen werden Faktoren wirksam, die im klassischen formativen Test nicht berücksichtigt werden (können). In kollaborativen Szenarien oder auch bei den unterschiedlichen Formen dynamischer Kommunikation kommt es häufig zu nicht vorhersehbaren Ereignissen, die eine unmittelbare Reaktion des

Nutzers erforderlich machen und die – in Abhängigkeit von dieser Reaktion – den Ablauf einer Interaktionssequenz erheblich beeinflussen. Der entsprechende Test-Ansatz muss somit dem indeterminierten, probabilistischen Charakter der Situation Rechnung tragen. Typische Ereignisse sind z. B. Anfragen über einen Kommunikationsfeed oder mehr oder weniger beliebige Aktionen von Teilnehmern einer Kollaboration. Unabhängig von der wichtigen Frage, wie solche Ereignisse innerhalb eines Tests simuliert werden können, führen die Ereignisse auch dazu, dass man in solchen Szenarien mit einem völlig anderen Problemtypus wie in klassischen formativen Tests zu tun hat; lediglich der Ausgangspunkt ist üblicherweise – aber nicht zwingend – wohl definiert. Bedingt durch die nicht vorhersehbare Wirkung dynamischer Ereignisse hat man es jedoch mit einer sehr großen Menge möglicher Wege zu einem insgesamt deutlich weniger klar definierbarem Zielzustand zu tun.

Üblicherweise kann als Ziel nur die prinzipielle Erledigung einer bestimmten Aufgabe dargestellt werden, wobei nicht klar vorhergesagt werden kann, in welchem Zustand sich das benutzte System nach Erledigung befindet und auf welchem Weg die Erledigung bewerkstelligt wurde.

In der Terminologie von Dörner (1987) ist von einer Kombination von Syntheschranke und dialektischer Schranke (d. h. die Präzisierung des Ziels erfolgt erst während der Bearbeitung einer Aufgabe) auszugehen. Unsere hauptsächliche Frage ist somit, wie diesem für Usability-Tests speziellen Problemtyp methodisch Rechnung getragen werden kann. Im folgenden Abschnitt schlagen wir einige Ansätze vor.

2. Elemente einer Testprozedur für soziale Software

In dem vorgeschlagenen Ansatz zum Testen sozialer Software wird berücksichtigt, dass

- für den Nutzer unvorhergesehene Ereignisse während des Tests auftreten können,

- der Nutzer in nicht (genau) vorhersehbarer Weise auf diese Ereignisse reagiert,
- das zu untersuchende System während der Testprozedur sehr unterschiedliche und zum Teil unvorhersehbare Zustände annehmen kann,
- der Testmoderator auf alle neuen Situationen in angemessener Weise reagieren muss und somit verschiedene situationsbezogene Varianten von Untersuchungsmaterial verfügbar sein müssen.

Typische unvorhergesehene Ereignisse während des Tests sozialer Software sind das Erscheinen von Nachrichtentexten, die automatische Aktualisierung von Kommunikationsfeeds, das Erscheinen visueller Indikatoren, die z. B. auf neue Informationen hinweisen oder direkte Reaktionen sozialer Kontaktpersonen auf Aktionen des Nutzers. Die Simulation solcher Ereignisse und Interaktionen ist aus Wizard-Of-Oz Experimenten bekannt. In solchen Experimenten werden Ereignisse, deren Verursachung einem Computersystem zugeschrieben wird, von einer unsichtbaren Person in Abhängigkeit von den Aktionen des Probanden an dafür vorgesehenen Stellen erzeugt. In dem von uns beschriebenen Fall ist jedoch für den Probanden üblicherweise klar, dass die Ereignisse durch eine soziale Kontaktperson bedingt sind. Für die von uns vorgeschlagene Vorgehensweise ist es somit auch nicht erforderlich, die Ereignisse „verdeckt“ zu simulieren. Sofern ein funktionaler elektronischer Prototyp in der Testprozedur verwendet wird, ist es aber prinzipiell möglich, dass eine Person außerhalb des Testraums die benötigten Ereignisse und Reaktionen auf die Aktionen des Probanden direkt erzeugt. Diese Vorgehensweise erfordert jedoch einen bereits umfangreich entwickelten Prototyp, da es möglich sein muss, mit ihm alle benötigten Systemzustände (als Reaktion auf die Aktionen des Probanden oder auch des Eventers) zu erzeugen. Befindet sich ein Softwareprodukt oder die ihm zugrundeliegenden Interaktionskonzepte noch in der Entwurfsphase (auf die wir mit unserem Ansatz hauptsächlich abzielen), ist diese Vorgehensweise somit keine Option.

Um möglichst passend auf (Re-)Aktionen des Probanden eingehen zu können und auch einen möglichst hohen Erkenntnisgewinn aus der Testprozedur sicher zu stellen, schlagen wir vor, statt elektronischer Prototypen ausgedruckte Materialien für den Test sozialer Software zu verwenden. Zum einen bietet sich eine Vorgehensweise im Sinne des klassischen Paper-Prototypings an. Ein großer Vorteil der Verwendung von Papier-Prototypen ist die hohe Flexibilität, die dem Versuchsleiter gegeben ist und die es ihm ermöglicht, sehr dynamisch und hoch adaptiv auf alle Aktionen des Versuchsteilnehmers zu reagieren. Eine sehr gelungene und praxisorientierte Einführung in das Thema gibt Snyder (2003). In unserem Ansatz verwenden wir Papierprototypen, deren Erscheinungsbild noch relativ weit vom realen Aussehen eines Bildschirminhalts entfernt ist, sogenannte „low-fidelity Prototypen“. Vorteile dieser eher abstrakten Darstellungsweise sind die niedrigen Entwicklungskosten und die klare Fokussierung auf Interaktion und Informationsarchitektur. Auch versetzt ein „unfertiges“ und skizzenartiges Erscheinungsbild des Reizmaterials erfahrungsgemäß den Probanden in die Lage leichter und freier auf sein kognitives Modell zu assoziieren. Die Verwendung der teureren „high-fidelity Prototypen“ birgt häufig die Gefahr, dass ein weitgehend ausgearbeitetes visuelles Design von den zu einem frühen Zeitpunkt zu untersuchenden kritischen Eigenschaften eines Konzeptes ablenkt. Ein übersichtlicher Vergleich von low vs. high fidelity Prototypen findet sich z. B. bei Rudd, Stern & Isensee (1996).

Wie bei Papierprototypen üblich, werden für unsere Vorgehensweise komplette Bildschirminhalte und deren mögliche Bestandteile vor einem Test hergestellt, um sie dann innerhalb der Testprozedur in Abhängigkeit von den (System-) Ereignissen und den Reaktionen der Probanden darzubieten und zu kombinieren. Die wesentlichen vorgedachten Interaktionssequenzen werden üblicherweise in einer Art „Drehbuch“ festgehalten, dessen Inhalte zumindest der Moderator sehr genau kennen muss. Die Erzeugung spezieller Ereignisse wird vom Moderator oder einer

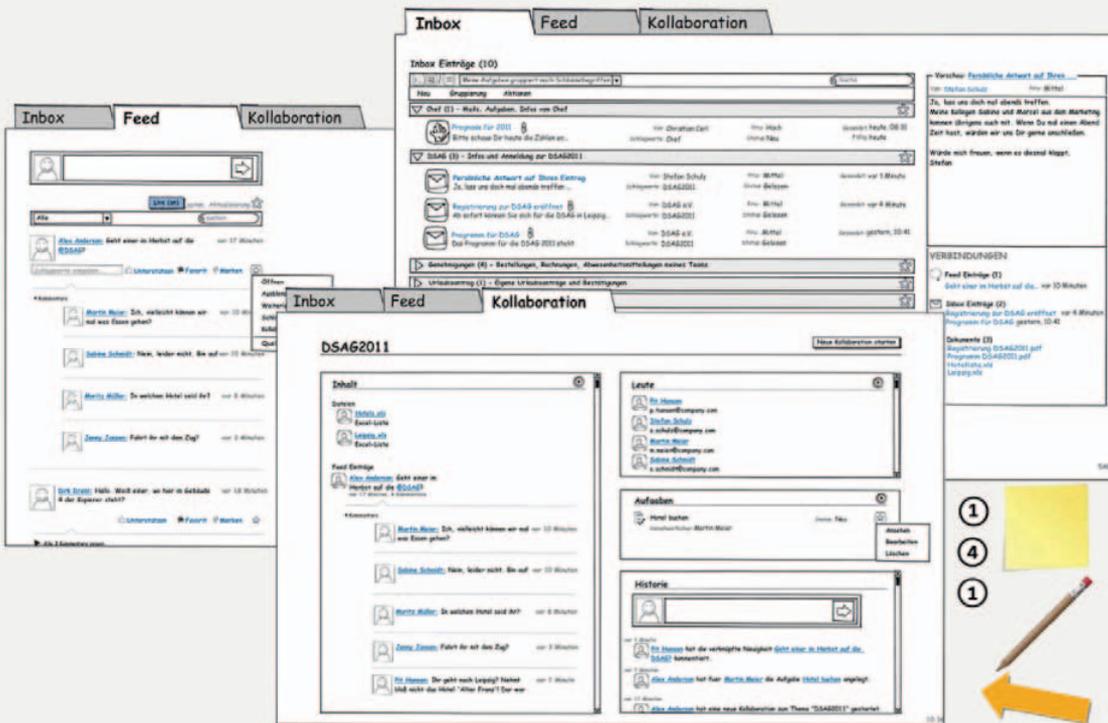


Abb. 1.
Materialien für eine
Untersuchung

direkt in die Testprozedur eingebundenen weiteren Person (dem sogenannten „Event-er“) übernommen. Das bedeutet, dass die Simulation der Ereignisse durch Austausch oder Modifikation von Bildschirmhalten direkt innerhalb der laufenden Prozedur durchgeführt wird. Da der Proband auch diese Aktionen zur Kenntnis nimmt, ist auch von seiner Seite eine gewisse Abstraktionsleistung erforderlich.

Die zur Erledigung einer gegebenen Aufgabe erforderlichen Aktionen und Systemzustände sind teilweise so wenig vorhersagbar, dass auch die Arbeit mit vorgefertigten Elementen einer Nutzungsoberfläche nicht mehr möglich ist. Für diesen Fall schlagen wir vor, die erforderlichen, von Probanden erwarteten Systemzustände unmittelbar in Form handschriftlicher Skizzen herzustellen und mit dem Probanden abzustimmen, ob ein Entwurf seinen Erwartungen entspricht.

Für eine erschöpfende Erfassung eines Versuchsablaufs ist – wie in regulären

Usability-Tests – neben dem Moderator auch ein trainierter Protokollant zwingend erforderlich. Außerdem sollte jede Sitzung zusätzlich per Video aufgezeichnet werden. Die entstehenden Artefakte (Kombinationen vorgefertigter Bildschirmteile sowie speziell angefertigte Skizzen) sollten digitalisiert werden. In vielen Fällen können sie direkt in der Konzeptionsphase der Software weiterverwendet werden.

3. Ein konkretes Beispiel

Der skizzierte Ansatz zum Testen sozialer Software kam in einer Testreihe mit 12 Probanden erstmals zum Einsatz. Testgegenstand waren Konzepte für eine Integration von Nutzerkollaboration, Posteingangsfach und Kommunikationsfeeds. Die Konzepte bzw. Ideen zu deren Zusammenspiel befanden sich zum Zeitpunkt des Tests noch in einer frühen Entwurfsphase. Die Arbeit mit einem elektronischen Prototyp war somit nicht indiziert. Die Testmaterialien lagen als Papierprototypen sowie als

„Blankbildschirme“ zur Erzeugung neuer Systemzustände vor. Abbildung 1 zeigt Beispiele für die verwendeten Materialien. **[Abb. 1]**

Neben den Papierprototypen hatte jeder Proband einen Pfeil aus Pappe zur Verfügung, um die aktuelle Position des Maus-Cursors („Wo würden Sie klicken?“), sowie um den aktuellen visuellen Fokusbereich anzuzeigen. Selbstklebende „Post-Its“ wurden verwendet, um Anregungen und Kommentare der Probanden direkt auf den jeweiligen Seiten zu vermerken. Weitere vorbereitete Elemente wie selbstklebende Ziffern wurden für das Erzeugen von Ereignissen eingesetzt, z. B. um für eine nicht sichtbare Seite anzuzeigen, dass eine neue Nachricht oder ein neuer Feed-Eintrag vorliegt. Abbildung 2 zeigt schematisch den Ablauf für eine Aufgabe, wobei in diesem Beispiel keine während der Prozedur angefertigten Zustände enthalten sind. **[Abb. 2]**

Die Untersuchung wurde von einem Moderator, der auch die Rolle des Eventers

übernahm, und einer Protokollantin durchgeführt. Die verwendeten Events bestanden zum einen im „Einblenden“ neuer Feed-Nachrichten und im Anbringen visueller Indikatoren für neue Nachrichten oder Feed-Einträge. Die Aufgaben waren als allgemeine Handlungsziele formuliert, ohne in irgendeiner Weise auf das geplante System oder die geplanten Konzepte einzugehen. Die Formulierungen der Aufgaben waren wie folgt:

Aufgabe 1: Gerade erscheint eine eMail, dass die Registrierung für die DSAG eröffnet ist.

Vielleicht können Sie die Gelegenheit nutzen, und ein paar alte Kollegen treffen. Finden Sie heraus, wer von Ihren Kollegen auch hinfährt.

Aufgabe 2: Na prima, es scheinen ja doch ein paar mitzukommen.

Um Ihren gemeinsamen Aufenthalt in Leipzig zu planen, führen Sie alle vorhandenen Personen und Informationen zusammen.

Aufgabe 3: Martin ist dafür bekannt, dass er immer die besten und günstigsten Hotels findet. Er bekommt deswegen von Ihnen den Auftrag sich um das Hotel zu kümmern.

Aufgabe 4: Da warnt Pit vor einem bestimmten Hotel. Stellen Sie sicher, dass Martin über Pits Hinweis informiert wird.

Insgesamt wurde die Vorgehensweise von den Probanden sehr gut aufgenommen und das Konzept der intermittierenden Ereignisse wurde gut verstanden. Es war

in jeder der 11 Testsitzungen erforderlich, vom vorgedachten Pfad abzuweichen und ad hoc neue Bildschirm-Varianten herzustellen um den jeweiligen Erwartungen der Nutzer entgegen zu kommen. Die Ergebnisse konnten zum großen Teil direkt in der Weiterentwicklung der Konzepte verwendet werden. Es zeigte sich, dass die Verbindung von Moderation und gleichzeitigem Eventing eine relativ große Herausforderung für den Moderator darstellt. Sofern möglich, ist die Einbeziehung einer Person, die ausschließlich für das Eventing zuständig ist, zu empfehlen. Eine weitere Herausforderung stellt die effiziente Verwendung der vorbereiteten Materialien dar. Da die Verwendung einzelner Seiten oder Seitenelemente schwer vorhersagbar ist, ist eine gute Zugänglichkeit und Organisation der Materialien unbedingt erforderlich.

4. Fazit und nächste Schritte

Der beschriebene methodische Ansatz ist im Wesentlichen eine Kombination bewährter Verfahren, die in ihrer Kombination sehr gut für das Testen sozialer Software geeignet zu sein scheinen. Die Notwendigkeit für die Verwendung dieser speziellen methodischen Kombination ergibt sich insbesondere aus der besonderen Art von Problem mit dem Nutzer bei der Arbeit mit sozialer Software konfrontiert sind. Die meisten Aktionen hängen unmittelbar von den Handlungen, Reaktionen, Signalen oder Entscheidungen anderer Nutzer ab – die Wirkungsprinzipien sozialer Mechanik haben hier einen unmittelbaren Einfluss auf das beobachtbare Verhalten.

Unser empirisches Beispiel verwendet nur einen Teil der denkbaren und praktikablen methodischen Vorgehensweisen. Ein Schritt zur Erweiterung des Ansatzes soll der Workshop auf der gUPA 2012 sein. Wir erwarten uns sowohl neue Sichtweisen in Richtung des Untersuchungsgegenstands und in Richtung der möglichen Fragestellungen beim Testen sozialer Software, als auch Erkenntnisse zur Erweiterung der methodischen Herangehensweise.

Literatur

1. Dörner, D. (1987). Problemlösen als Informationsverarbeitung (3. Auflage). Stuttgart: Kohlhammer.
2. Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *Interactions*, 3(1), S. 76-85.
3. Snyder, C. (2003). Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces (1. Auflage). San Francisco: Morgan Kaufmann Publishers.
4. Ward, L. F. (1900). Social Mechanics, Sociology at the Paris Exposition of 1900 (pp. 1579-1593), Government Printing Office.
5. Winiarski, L. (1900). The Teaching of Pure Political Economy and Social Mechanics in Switzerland, Sociology at the Paris Exposition of 1900 (pp. 1497-1500), Government Printing Office.

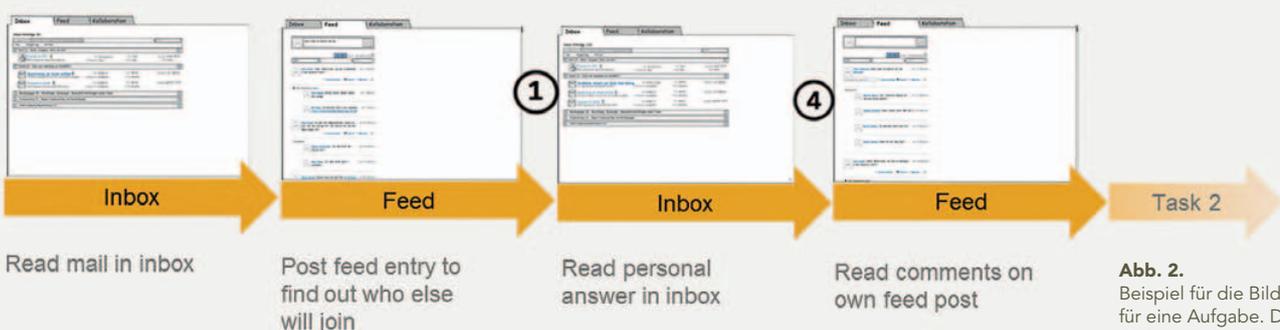


Abb. 2.
Beispiel für die Bildschirmfolge für eine Aufgabe. Die Ziffern in Kreisen stehen für spezifische Events, die vor dem Übergang zum nächsten Bildschirminhalt erzeugt wurden.

