



Social Software

Eine Einführung

© Gernot Tscherteu 2003,
alle Rechte vorbehalten.

Schiffamtsgasse 10
A - 1020 Wien

Telefon: +43 1 726 73 71 dw 20
Fax: +43 1 726 73 71 dw 10
gt@realitylab.at
www. realitylab.at



Inhalt

1.	Was ist Social Software	3
	Erster Versuch einer Antwort	3
	Zweiter Versuch einer Antwort:	3
	Man könnte aber noch viel allgemeiner ausholen:	3
	Meine Definition	4
	Nutzen	5
	Gefahren	5
2.	Von „Groupware“ zu „Social Software“	5
	Businessanwendung	7
3.	Social Software – Portraits	7
	Content-Management-Systeme	7
	Wikis	8
	Weblogs	9
4.	Abwicklung eines Social Software - Projekts	10
	Status Quo- und Ziel- Analyse	10
	Teambildung und Projektplanung	11
	Auswahl der Software	11
	Systemdesign	11
	Programmierung	12
	Prototyping und Testing	12
	Iteratives Redesign	12



1. Was ist Social Software

Gegenfrage: Was ist nicht Social Software? Bei genauer Betrachtung wird man bemerken dass fast jede Software auch soziale Schnittstellen aufweist.

Erster Versuch einer Antwort

Social Software ist ein Modewort, mit dem man so unterschiedliche Dinge wie die Zusammenarbeit über Weblogs und Wikis und intelligente Suchalgorithmen a la Amazon und E-Bay sowie neue Trends wie Meetup mit einem neuen gemeinsamen Schlagwort versehen möchte; also ein Schachzug schlauer Berater um das Geschäft anzuheizen. Etwas zynisch, aber mit einem wahren Kern.

Zweiter Versuch einer Antwort:

Programme die menschliche Kommunikation und soziale Interaktion unterstützen sollen. Zunächst ein paar Beispiele für solche Programme:

E-Mail,
Usenet News,
Groupware (a la Outlook und Lotus Notes)
MultiUserDungeons,
WebLogs,
Wikis,
InstantMessaging,
InternetRelayChat,
Collaborative Filtering (Amazon recommendation software, EbayDotCom)

Aber warum nicht auch MS Word: es gibt beispielsweise eine Überarbeitenfunktion, welche die gemeinsame Redaktion von Texten unterstützt.

Und warum nicht auch Handys oder Telefonanlagen.

Telefonanlagen definieren genau wie Anrufe weitergeleitet werden, wer in erster Linie für die Beantwortung von allgemeinen Anrufen verantwortlich ist, wer Anrufe verteilt usw.

Man könnte aber noch viel allgemeiner ausholen:

Komplexere Gesellschaften tendieren dazu soziale Prozesse zu beschreiben bzw. mechanisch abzubilden. Denken wir nur an Gesetze



oder politische Verfassungen. Sie sind in gewisser Weise die ersten Formen sozialer Software: Sie sind soziale Algorithmen, kleine Programme mit „wenn dann“ Schleifen. Sie definieren, was zu geschehen hat, wenn ein bestimmter Fall z.B. ein Diebstahl oder der Rücktritt einer Regierung eingetreten ist. Man könnte auch sagen, dass es das Wesen von sozialer Software ist Verhaltensweisen zu speichern um sie im Bedarfsfall ohne lange Diskussionen abzurufen und umzusetzen. Es handelt sich um so etwas wie „Shortcuts“ zwischenmenschlichen Verhaltens und Kommunikation. Kleine Werkzeuge, welche unsere Kommunikation einerseits erleichtern und sie andererseits in bestimmte Bahnen lenken.

Diese Definition von sozialer Software hat zwar sehr interessante Aspekte ist aber fürs erste, zu weit um damit praktisch arbeiten zu können.

Meine Definition

Meine Definition ist pragmatischer und gar nicht so weit von der ersten Antwort entfernt:

Social Software ist eher als „Brille“ zu verstehen, mit der man auf die sozialen und kollaborativen Aspekt von computergestützter Kommunikation blickt. Social Software ist sicher ein Modewort, das möglicherweise bald wieder verschwinden wird. Es bezeichnet aber recht treffend eine aktuelle Gesamtentwicklung des Internet, die folgende Merkmale aufweist:

- Die allgemeine Computer- und Internet-Literalität hat einen Punkt erreicht, an dem viele Personen von Nutzern auch zu Produzenten von Webinhalten werden. (siehe persönliche Websites, Weblogs, Foren)
- Immer mehr Autoren übernehmen Content von anderen Autoren im Internet wodurch die Anzahl der Verknüpfungen im Internet exponentiell wächst und das Netz immer mächtiger aber auch immer komplexer wird.
- Zur gleichen Zeit entstanden aber auch Technologien, welche das Filtern, Auswählen und Integrieren von Information erleichtern.
- Manche dieser Technologien haben die Form von Programmen (siehe Buchvorschlag bei Amazon), bei anderen handelt es sich einfach um soziales Verhalten (z.B. Empfehlungen eines Autors).

Bei „Social Software“ handelt es sich nun um Programme, in denen sich soziale und maschinelle Prozesse gegenseitig durchdringen und unterstützen und dadurch neuen Nutzen aber auch neue Gefahren produzieren:



Nutzen

- Rascheres Auffinden relevanter Information und damit erhöhte Produktivität
- Etablierung von Wissensnetzwerken zum Austausch von Spezialwissen.
- Demokratisierung der Nachrichtenproduktion
- Kennenlernen von interessanten Menschen
- Organisation von gemeinsamen Interessen
- Unterhaltung
- ...

Gefahren

- soziale Netze werden transparent und öffentlich nachvollziehbar
- wer garantiert dass Nutzerprofile immer anonym bleiben.
- die Profile können auch in falsche Hände geraten z.B. Google Werbung in Eudora.
- ...

Damit die Gefahren in einem vertretbaren Verhältnis zum Nutzen bleiben, muss man „Social Software“ einerseits verantwortungsvoll und aufmerksam nutzen und andererseits auch als ethisch relevantes Problem erkennen. Wir müssen gemeinsam Richtlinien entwickeln, um beurteilen zu können welche Technologien „gut“ bzw. „schlecht“ sind und wie man mögliche Gefahren bannt. Gesetze werden dabei sicher nicht so effizient sein wie die individuelle Verantwortung und das Gewicht der öffentlichen Meinung - einfach deshalb weil Gesetze zu träge sind und nur allgemeine Rahmenbedingungen vorgeben können. Im Einzelfall müssen wir selbst entscheiden.

2. Von „Groupware“ zu „Social Software“

Um das Potential und die Eigenart von „Social Software“ noch besser herauszuarbeiten, möchte ich sie dem etwas älteren „Groupware“ Ansatz gegenüberstellen. Dies geschieht in pointierter Weise und „Groupware“ kommt dabei nicht wohl nicht so gut weg: Tatsache ist aber, dass beides seinen Sinn hat, nur an verschiedenen Orten und für verschiedene Zwecke.

Als jemand der seit 14 Jahren im IT-Bereich tätig ist, habe ich einige Paradigmenwechsel und (fast) alle Entwicklungen im Bereich der kollaborativen Software miterlebt; noch dazu in den unterschiedlichsten Positionen, einmal als Angestellter und Nutzer von Programmen wie Lotus Notes, dann als Unternehmer der selbst unternehmenskritische Aufgaben wie Adressverwaltung und Akquisition optimal unterstützen möchte und schließlich auch als Designer, der selbst kollaborative Software gestaltet.



Das faszinierende und neue an „Social Software“ erscheint mir dabei, dass Sie nicht dem Prinzip bewusster Gestaltung unterliegt sondern eher dem Prinzip der „Selbstorganisation“ d.h.:

Es wird nicht eine „fixe und vorgegebene Idee“ von Organisation und Kommunikation umgesetzt sondern eher umgekehrt, der ohnehin schon natürlich vorhandene Fluss von Kommunikation unterstützt und erweitert. Weblogs etwa haben sich mehr oder weniger von selbst entwickelt und stützen sich auf eine Handvoll von Konventionen, ohne dass ein offizielles Gremium, über deren Einhaltung wacht.

Im Folgenden eine Gegenüberstellung in Schlagworten

Groupware (z.B. Lotus Notes)

Top down
Vorgegebene Organisation
Gruppenorientierte Kommunikation
Kleine bis mittlere Personenanzahl (Betrieb)
„Erzwungene“ Organisation
Klare Grenzen zwischen „Innen und Aussen“
deduktiv
Eher gleichbleibende Strukturen und Workflows
Administratorenkontrolle
Implizite unsichtbare Regeln
Vorgesetztenurteil

Social Software (z.B. Weblogs)

Bottom up
Selbstorganisation
Persönlichkeitszentrierte Kommunikation
sehr viele Personen
Freiwillige Vernetzung
Verschwimmende Grenzen
induktiv
Dynamische Kommunikationsstrukturen
Selbsterfundene Konventionen (co-evolution)
Meist offene Regeln
Social Rating (Kommentare, Punkte)

Matt Jones:

“Usenet and groupware apps were designed to scale from a technical and business point of view, not from a social point of view. That's why they sucked, because they didn't look at how humans work on social scale. That's what's new now I think, is that we're looking much more to the real world being helped by software than software simulating a perfect system that we adapt to.”

Man wird sich also genau überlegen müssen, wo man welche Software einsetzt. Das eine ersetzt das andere nicht. Weblogs sind ideale Werkzeuge für Knowledge-Worker, die ihre Arbeit dokumentieren und sich mit Gleichgesinnten austauschen möchten. Sie sind wohl eher nichts für rein administrative Aufgaben, wo man eher auf Lotus Notes und ähnliches setzen wird. Die Auswahl der Software ist dementsprechend eine wichtige Managementaufgabe, die nicht nur eine technologische Dimension hat sondern auch die Form und Atmosphäre von Zusammenarbeit maßgeblich beeinflusst.



Businessanwendung

Für die Businessanwendung sind unter anderem folgende Fragen zu beantworten:

- Welche Kommunikationsziele werden verfolgt?
- Welche Kommunikationsstrukturen sind bereits vorhanden?
- Welche Strukturen sollen gestärkt welche abgebaut werden?
- Sucht man die Auseinandersetzung auch mit Leuten außerhalb des Unternehmens?
- Ist Innovation überhaupt ein kritischer Erfolgsfaktor?
- Besteht eine grundsätzliche Bereitschaft von Management und Belegschaft sich auf ein neues System einzulassen?
- Wie können konservative und innovative Softwarekomponenten gut miteinander verknüpft werden?

Insgesamt steckt die Businessanwendung von sozialer Software noch in den Kinderschuhen und muss erst auf ihre Praxistauglichkeit untersucht werden. Was sich im Netz im großen Maßstab bewährt hat, muss sich nicht unbedingt im Unternehmen bewähren. Die Integration von sozialer Software und Groupware ist kaum fortgeschritten. So hat es sich beispielsweise bewährt für Projekte spezielle Weblogs zum Austausch zwischen den Projektmitgliedern einzurichten, diese sind dann aber nicht mit der Adress- oder der Terminverwaltung des Unternehmens kompatibel. Das Zusammenwachsen beider Welten wird sicher noch einige Zeit dauern, bietet aber ein faszinierendes Betätigungsfeld.

3. Social Software – Portraits

Content-Management-Systeme

Ich beschränke mich hier auf die kollaborativen Aspekte von CMS: (Fast) jedes CMS das von mehreren Personen benutzt wird, arbeitet mit einem bestimmten Regel- und Rollensystem. Dieses System regelt zum Beispiel

- wer in welchem Bereich der Webseite Beiträge verfassen kann,
- wer Zugriff auf das allgemeine Layout hat,
- wer Beiträge redigiert
- wer Beiträge online bzw. offline stellen, ...

Je nach Größe der Website und der dahinter stehenden Organisation kann dieses System mehr oder weniger komplex sein. Bei manchen Websites, die mit einer Nutzercommunity arbeiten ist ferner geregelt:

- wie neue Benutzer registriert werden
- wer Zutritt zu welchen Bereichen der Website hat
- wer welche Beiträge verfassen bzw. editieren darf.
- Wer wo Kommentare verfassen darf, ...



Dabei haben sich häufig folgende Fehler herausgestellt:

- gerne werden solche Rechte ohne lange Diskussion einfach gesetzt. Oft von Systemadministratoren die meist nur Wartungsfreundlichkeit und Sicherheit im Auge haben und dabei die Benutzbarkeit und sozialen Aspekte übersehen. -> keep it transparent.
- Oft werden Systeme zu komplex und es wird der Umstand übersehen, dass ein Nutzer nicht immer gewillt ist oder genug Zeit hat um sich mit dem Rollen- und Rechteschema eines CMS vertraut zu machen. -> keep it simple.
- Immer wieder glaubt man, dass man ein System einmal einrichtet und dass es dann für immer läuft. Richtig ist vielmehr dass man aus den Erfahrungen laufend lernen muss und das System einer Evaluierung und einem sozialen Redesign unterwerfen muss. -> evaluate and redesign

Ein CMS ist meist die Abbildung eines sozialen Prozesses (Zusammenarbeit, Redaktion) mit elektronischen Mitteln. Ein sozialer Prozess muss sich entwickeln können.

Wikis

Unter Wiki versteht man ein besonderes CMS, das auf besonders flachen Hierarchien und besonders flachen Navigationsstrukturen beruht:

Jeder darf schreiben und auch vorhandene Seiten editieren
Jede Seite ist gleichberechtigt. D.h. es gibt in der Regel keine Menüs sondern nur direkte Verweise auf andere Wikiseiten.

Das ein so einfaches und offenes System funktionieren kann beweist die [Wikipedia](#).

Um Vandalismus auszuschalten verfügen viele über Versionierungswerkzeuge, wodurch der alte Zustand vor der Zerstörung eines Textes wiederhergestellt werden kann. Ein Vergleichstool erlaubt es Unterschiede zwischen alter und neuer Version zu visualisieren.

Um möglichst vielen Menschen die Mitarbeit an Wikis zu ermöglichen sind so genannte What-You-See-Is-What-You-Get (WSIWYG) Editoren im Einsatz. Das Formatieren und Verlinken von Webcontent wird damit so einfach wie die Handhabung üblicher Texteditoren.

Im Businessseinsatz eignen sich Wikis dort, wo z.B. ein Glossar oder ein nach Stichworten organisiertes Handbuch gemeinsam geschrieben werden soll.



Weblogs

Was ist ein Weblog?

Ein *Weblog* ist im Prinzip nichts anderes als eine Art Online-Tagebuch, das von einer einzelnen Person oder von einer Gruppe von Personen geführt wird. Jeder Eintrag – meist in Form eines Absatzes – bekommt einen Datumsstempel und den Namen des Autors angehängt. Der neueste Eintrag wird fast immer ganz oben im *Weblog* angezeigt, ältere hängen sich unten an. Die Inhalte *der Weblogs* sind so verschieden wie die Menschen, die sie führen; Manchmal geht es nur um Tratsch oder den letzten Kinobesuch; Oft wird in Logs aber auch die eigene Arbeit dokumentiert. Ich persönlich nutze mein *Weblog* wie ein erweitertes Gedächtnis um Ideen, Infos und interessante Links, meine Arbeit als Mediendesigner betreffend, abzulegen und zu ordnen. Weltweit gibt es derzeit ungefähr 300.000 *Weblogs*; Man kann also durchaus von einer literarischen Massenbewegung sprechen,

Was auch immer ein Autor oder eine Autorin mit seinem oder ihrem Log vorhat, der oben genannte Aufbau eines *Weblogs* ist prinzipiell immer sehr ähnlich und es gehört zu den kleinen Wundern des Internets, das sich hier ein einfacher aber höchst effizienter Standard^[2] herausgebildet hat, ohne dass es ein Standardisierungsbehörde oder einen mächtigen Softwaremonopolisten gegeben hätte, der diesen Standard durchgesetzt hätte. Die angenehme Folge davon ist, dass man sich in den meisten *Weblogs* recht gut zurechtfindet und andererseits dass man Einträge zwischen den *Weblogs* austauschen kann (mehr dazu weiter unten). Auf diese Weise sind Logs im Gegensatz zu ihren papierernen Vorläufern nicht nur Privatvergnügen, sondern sehr ergiebige, tagesaktuelle Informationsquellen, durch die sich Menschen, die ein gemeinsames Interesse teilen, gegenseitig auf dem Laufenden halten.

RSS und Aggregatoren

Um den Austausch zwischen Autoren zu erleichtern, gibt es einfache aber mächtige, technische Hilfsmittel: *RSS-Feeds* und *Aggregatoren*. Beide verfolgen vereinfacht gesprochen folgendes Ziel: Sie versorgen den *Weblogautor* mit den Neuigkeiten anderer *Weblogs* und ermöglichen es ihm, diese auf einfache Weise (ohne HTML-Kenntnisse) im eigenen *Weblog* zu veröffentlichen. Der Autor muss andere *Weblogs* und Websites gar nicht mehr besuchen. Er sitzt wie die Spinne im Netz und *läßt* alle Neuigkeiten-Fäden bei sich zusammenlaufen.

RSS heißt jener Standard, der eine ganz wesentliche Rolle bei der Übernahme und Verbreitung von Nachrichten spielt. (Auf den meisten *Weblogs* werden sie durch ein „RSS“- Icon angezeigt) Im Prinzip ist ein *RSS-Feed* nichts anderes als eine „*darstellungsunabhängige*“ Textversion eines *Weblogs*. „Darstellungsunabhängig“ soll heißen, dass



man diesen Text in ein beliebiges Medium einlesen und neu formatieren kann (etwa Textgrößen, Schriftarten, Textfluss usw. ändern kann). Auf diese Art und Weise kann man Texte von anderen Autoren in einem speziellen Programm einlesen und dann „auf Mausclick“ (also ohne Kenntnisse von HTML) in seinem eigenen *Weblog* veröffentlichen. Solche Programme heißen *News-Aggregatoren*.

RSS und *Aggregatoren* bewirken einerseits dass Nachrichten rasche und schneeballartige Verbreitung finden, sie bewirken andererseits aber auch, dass sie von den Autoren gefiltert und thematisch geordnet werden. Der *Weblogautor* ist ähnlich einem DJ Arrangeur, Interpret und Komponist zugleich. Jeder Autor wählt natürlich nur jene Neuigkeiten, die ihm wichtig erscheinen und er hat auch seine speziellen Zuordnungskategorien. Wenn man also die *RSS-Feeds* eines bestimmten Autors abonniert, dann vertraut man auf sein Engagement beim Sammeln von News, und auf seine Kompetenz bei der Selektion der „wichtigen“ News. Es entsteht mit anderen Worten ein nachvollziehbares Netz von Wissensflüssen und ein sichtbares Geflecht gegenseitiger Beeinflussung.

4. Abwicklung eines Social Software - Projekts

Grundsätzlich ist jedes Social Software Projekt ein Einzelfall, der eine maßgeschneiderte Lösung erfordert. Nicht immer bedeutet das, dass man Software eigens programmieren muss. Die Auswahl, Anpassung und Einführung im Betrieb erfordert einiges an Erfahrung und einen „dialogischen“ Designprozess.

Status Quo- und Ziel-Analyse

In beiden Fällen ist zu beachten, dass man nicht bloß eine technische Infrastruktur gestalten möchte, sondern eigentlich die soziale Interaktion zwischen den Nutzern. Nicht die Nutzer sollen sich an technische Infrastruktur anpassen sondern umgekehrt. Im Idealfall führt man soziale Interaktionen durch, ohne dass der Gebrauch von Hard- und Software direkte Aufmerksamkeit verlangt. Man arbeitet „through the interface“. Beispiel: Der Gebrauch des Festnetztelefons ist uns mehr oder weniger so geläufig, dass seine Bedienung kaum Aufmerksamkeit abverlangt und wir mit dem Gesprächspartner mehr oder weniger direkt kommunizieren. Im Realfall wird man aber immer Aufmerksamkeit auf die Benutzung des Interface verwenden weshalb man seine Gestaltung und auch die Einschulung im Vorhinein bewusst planen sollte. Hier liegen die entscheidenden Erfolgsfaktoren.



Teambildung und Projektplanung

Im Designteam vertreten sollten folgende Kompetenzen (nicht notwendigerweise in Gestalt eines eigenen Experten):

- Management – jemand, der die Unternehmensseite berücksichtigt und den eigentlichen Auftraggeber repräsentiert.
- Nutzer – jemand der mit der Software arbeiten wird.
- Technologie – jemand der das System implementieren und im laufenden Betrieb warten wird. Möglicherweise braucht man für die Auswahl bzw. die Programmierung des Systems einen externen Experten.
- Interfacedesign – jemand der das Systemoberfläche grafisch ansprechend und benutzbar gestaltet.
- Mediendesign – jemand der bewusst die soziale Interaktion gestaltet.

Je nach Projekt können noch andere Funktionen wie Einschulung und Mediation dazukommen.

Bei der Projektplanung ist wie gesagt auch die Einschulung und die Zeit nach der Einführung des neuen Systems bewusst zu planen. Des Weiteren begleitende Maßnahmen wie Moderation, Werbung und PR, weil es hier in der Regel wechselseitige Abhängigkeiten gibt.

Auswahl der Software

Grundsätzlich ist die Entscheidung zu treffen, ob man Software eigens programmiert oder vorhandene Software anpasst. Wenn man vorhandene Software verwendet oder eigens programmieren lässt, sollte man sich unbedingt folgendes vor Augen halten:

- gibt es den Hersteller noch in ein paar Jahren
- gibt es Ressourcen Vorort für Anpassung und Wartung
- welche sonstigen Abhängigkeiten entstehen

Im Allgemeinen ist Open-Source Lösungen der Vorzug zu geben.

In der weiteren Folge sollte man sich verschiedene Bewertungskriterien überlegen, welche die Software erfüllen sollte und dann verschiedene System auf ihre Brauchbarkeit anhand dieser Kriterien vergleichen.

Systemdesign

Wenn man sich für eine Software entschieden hat (oder völlig neu entwickeln möchte) muss das Aussehen und die Funktion der Anwendung im Detail spezifiziert werden. Einerseits in Form ein grafischen Design und andererseits in Form eines funktionellen Designs, welche die Abfolge von Fenstern in den einzelnen Prozessen, z.B. bei der Registrierung, genau festlegt. Hier ist auch jene Interaktionen zu bedenken, die außerhalb des Systems stattfinden wird. Wenn es zum Beispiel Face-to-Face Meetings



von Nutzern geben soll, so sollte man sich überlegen, wie diese Treffen ablaufen sollen und wie sie im System angekündigt bzw. dokumentiert werden sollen.

Programmierung

Das System wird angepasst oder neu programmiert. In der ersten Version sollte man nicht ein völlig ausdifferenziertes System mit dem vollen Funktionsumfang implementieren sondern eher ein Rumpfsystem das den Test unter realen Bedingungen erlaubt. Ausgefeilte und teure Funktionen sollten zuerst vor dem Hintergrund eines laufenden Systems auf ihren Nutzen hin untersucht werden.

Prototyping und Testing

Jedes Social Software Projekt sollte vor der Einführung als Prototyp parallel zum Status Quo herlaufen und eine Testzeit durchmachen. Wichtig ist nicht nur die Funktionen zu testen, sondern ob das System überhaupt seinen Zweck erfüllt. Dies geht auch über gängige Usability-Tests hinaus. In der Regel ist eine längere Testphase notwendig.

Iteratives Redesign

Auch bei Social Design Projekten sollte man nicht glauben, dass die erste Version auch die letzte sein wird. Auch bei sorgfältiger Planung und genügend Konzeptionszeit sind ist das System in seiner Nutzungsumgebung nicht völlig vorhersehbar. Soziale Systeme sind im Unterschied zu Computerprogrammen nicht völlig determiniert: d.h. Ihr Verhalten ist grundsätzlich unberechenbar. Deshalb sollte man gleich ein zweite Version einplanen die sich noch besser an den gewünschten Zustand annähert. Ansonsten riskiert man, dass sich viele (Interaktions-) Prozesse an die Software angleichen und nicht umgekehrt. Im Unternehmensprozess kann dies zu bedeutenden Effizienzverlusten führen.

Im Allgemeinen sollte man für ein Social Software mindestens vier Monate einplanen, bei komplexeren Projekten, die auf Eigenentwicklungen hinauslaufen, sind auch wesentliche längere Projektlaufzeiten möglich bzw. notwendig. Nicht immer ist eine längere Projektdauer automatisch mit höheren Kosten gleichzusetzen. Oft geht es „nur“ um mehr Zeit für Entscheidungen und gründliches Design. Dies gilt insbesondere für das Redesign; wenn man eine längere Testphase und eine zweite Version von vorneherein einplant so ist der Mehraufwand meist verkraftbar und er wird durch eine Effizienzsteigerung im Unternehmen zumeist wettgemacht.