
Systemutvikling i sosiale nettverk

Essay, INFO 214, Høst 2009

1.0 Innledning

Den nye generasjonen av teknologi som legger til rette for interaksjon mellom brukere, går gjerne under begrepet «sosiale nettverk». Disse systemene fungerer ved at innhold, informasjon, meninger, erfaringer og kunnskap blir utvekslet mellom aktørene. *Facebook* og *Twitter* er typiske eksempler på slike nettverk, og er noen av de som blir brukt aktivt av flest mennesker verden over. I følge Qualman (2009), fikk Facebook over 100 millioner brukere på mindre enn ni måneder, de har nå over 300 millioner aktive brukere, mens Twitter har over seks millioner brukere. Disse tallene viser at systemutviklerne i stor grad har lyktes med å lage system som treffer en mengde brukeres behov og ønsker. I denne oppgaven har jeg valgt å ta for meg systemutvikling innenfor sosiale nettverk med hovedfokus på Facebook og Twitter.

1.1 Problemstilling

Jeg vil se på metoder som er blitt brukt i utviklingen av sosiale nettverk, samt strategier og fremgangsmåter de har bygget på. Hvilke faser må man gjennom ved utviklingen av et sosialt nettverk, og hvilken metode bør en forholde seg til? Hva slags effekter har valgene i utviklingsprosessen fått for de endelige brukerne, og hva er spesielt med hvert enkelt nettverks arkitektur og funksjonalitet? Oppnår brukerne det som var utviklernes mål? Dette er spesielt aktuelt innenfor webbaserte system i dag, da vi ser en tendens til at systemer gjerne blir tatt i bruk, før de har gjennomgått en tradisjonell utviklingsprosess. Vi ser dermed at brukerne står i fokus ved å delta i prosessen gjennom å bruke sosiale nettverk.

For å svare på problemstillingen har jeg valgt å analysere bruken av Twitter og Facebook, og vurdere hvilke systemutviklingsmetoder som er best tilpasset utviklingen av sosiale nettverk. Jeg vil ta for meg fasene i noen av de vanligste systemutviklingsprosessene som Weaver (2004) nevner, for så å se på hva som er spesielt med utviklingen av sosiale nettverk.

2.0 Faser i systemutviklingsprosessen

Systemutvikling er prosessen med å utvikle datamaskinbaserte informasjonssystem. For å oppnå en best mulig prosess i utviklingen av et system bør man gjennomgå ulike trinn som er involvert i en systemutviklingsprosess. En systemutviklingsmetode består av teknikker for å utføre de ulike trinnene i prosessen. Den vanligste systemutviklingsmodellen er livssyklusmodellen, som gir et godt rammeverk for håndtering og strukturering av et prosjekt. Eksempler på livssyklusmodeller

kan være Fossefallsmodellen, Spiralmodellen og Inkrementell modell. De nevnte modellene går gjerne under «tradisjonelle metoder» for systemutvikling. Disse fokuserer i stor grad på detaljrik dokumentasjon og planlegging, samt god kontroll gjennom hele prosessen. Den Inkrementelle modellen er mest effektiv på større prosjekter, mens Spiralmodellen passer bedre til mindre systemer der kravene ikke er definert på forhånd. I fossefallsmodellen derimot, må kravene defineres på forhånd, og den egner seg dermed dårlig til utvikling av informasjonssystem der krav endres fortløpende. (Weaver, 2004)

Det finnes også «iterative metoder», som forsøker å begrense risiko ved å utvikle systemet bitvis med hyppigere leveranser. I tillegg finnes agile metoder, som legger vekt på fleksibilitet og hyppige tilbakemelding fra bruker underveis. Flere av de ulike metodene er i utgangspunktet basert på de samme fasene. Stadiene i utviklingsmodellene er altså svært like, uavhengig av hvilke metoder, teknikker og verktøy som brukes. Forskjellen er hovedsakelig rekkefølgen, fremgangsmåten og muligheten for å gå tilbake i en prosess. Extreme Programming (XP) og SCRUM er eksempler på agile metoder, som har en del forskjeller sammenliknet med tradisjonelle metoder. Jeg vil i følgende kort ta for meg noen av de sentrale fasene generelt i en utviklingsprosess. (Weaver, 2004)

2.1 Forskning, planlegging og krav

I en normal utviklingsprosess kreves det mye forarbeid som inkluderer grundig forskningsarbeid. Hva som finnes av liknende system er uvurderlig for utviklingen. Definerings av målgruppene, samt hva som kjennetegner dem og hvilke behov de har, er også et viktig forarbeid som påvirker hele utviklingsprosessen. Brukerens rolle i systemet må nøye beskrives. Det er viktig å tenke på hvordan, og av hvem systemet skal bli brukt av, slik at sluttproduktet ikke ender opp som et overflødig system. (Weaver, 2004)

Å sette opp et team er noe av det første som må gjøres. Hvem er utviklerne? Kan de gjøre alt selv, eller må eksterne ressurser inkluderes? Hva er problemet og hva skal gjøres? Her innkommer arbeid med å forstå problemet som det kommende systemet skal løse. Ressurser, kapasitet og arbeidsinndeling for de ulike oppgavene bør settes opp i en prosjektplan som beskriver alle fasene, hvem som er ansvarlig og hvor mye tid som skal brukes. Det bør også utvikles en PID (Project Initiation Document) som beskriver hovedformålet med prosjektet, målgrupper og risikoer. (Weaver, 2004)

Utviklingen av kravspesifikasjon kommer tradisjonelt i en tidlig fase, mens ved bruk av utradisjonelle modeller, utvikles kravene underveis i prosessen. Kravene bidrar til å definere hva systemet skal kunne utføre og hva det ikke skal gjøre. Et krav er en påstand om et tenkt produkt som spesifiserer hva det skal gjøre og hvordan det skal fungere. For at systemet skal fungere optimalt må vi stille krav til både brukeren og systemet. Det finnes to typer krav: funksjonelle krav og ikke-funksjonelle krav. Funksjonelle krav sier hva systemet skal gjøre, mens ikke-funksjonelle krav beskriver ytelse og begrensninger i systemet. (Weaver, 2004)

2.2 Funksjonalitet/prototyping

I utforming av skisser må systemarkitektur bestemmes, samt nivåer og eventuelle delsystemer. Systemets funksjonalitet utvikles med kravspesifikasjonen som mal. I denne fasen kan det lages en prototype som gjerne viser systemets utseende og funksjoner. Dette kan være en low-fidelity eller high-fidelity prototype. Dersom en prototype kategoriseres som low-fidelity, likner den ikke på det ferdige produktet. Den skal ikke integreres i det ferdige produktet, men fungere som en mal/skisse. En high-fidelity prototype bruker derimot det materialet man forventer seg i det ferdige produktet, og vil svare til forventningene for hvordan det ferdige produktet skal se ut. Prototypen reflekterer de funksjonelle og ikke-funksjonelle kravene som tidligere ble utviklet for systemet. (Sharp, Rogers, Preece, 2007 og Weaver, 2004)

For å vise hvilke data som skal lagres og behandles kan man benytte en databasemodell. Den viser strukturen til den kommende databasen beskrevet med et formelt språk støttet av DMS. Databasemodellen er teorien eller spesifikasjonen som beskriver hvordan en database er strukturert og brukt. (Weaver, 2004)

2.3 Testing, implementasjon og drift

Det finnes flere måter å teste et system på, deriblant *systemtest*. Det går ut på at utviklerne som selv har arbeidet med analysering av systemet, også tester det. Med en systemtest vil utviklerne kunne se om forventet resultat oppstår. Det kan også med fordel foretas en *HCI-analyse*, som går ut på å gå gjennom de ulike delene og elementene som krever et godt og forståelig brukergrensesnitt. Målet er at systemet skal oppføre seg slik det ble fastsatt i kravspesifikasjonen. I tradisjonelle utviklingsmodeller testes systemet i slutten av prosessen, mens i andre modeller foregår det en kontinuerlig test-prosess der brukere ofte er involvert. (Weaver 2004)

Etter det er laget en prototype og databasemodell for systemet, må selve funksjonene i systemet implementeres i en reell database. Dette kan for eksempel gjøres ved bruk av MySQL og PHP. Fasen går altså ut på å realisere systemet, stort sett ved bruk av programmering.

Etter at systemet er ferdig implementert og testet kan det tas i bruk og eventuelt videreutvikles. I tillegg må det ferdige systemet som oftest driftes av en nedsatt gruppe. (Weaver 2004)

3.0 Utviklingen av webbaserte system

Prosessen med å utvikle et webbasert system har mange likhetstrekk med utviklingen av andre system, men har i tillegg flere utfordringer. Et system på web er stort sett tilgjengelig for alle, så lenge det ikke er et intranett eller et annet brukerbegrenset nettverk. Utfordringer blir dermed å lage et system som når en rekke ulike brukergrupper, samt er forståelig og brukervennlig også for de uten høy grad av teknologisk innsikt. I tillegg skal systemet fungere på ulike skjermoppløsninger i ulike nettlesere, verden over. Sist men ikke minst, ulike enheter. Bruken av det mobile nettverket har nemlig opplevd en drastisk økning de siste årene. Utviklerne må dermed ta hensyn til enda flere behov enn ved utviklingen av system rettet mot en enkelt brukergruppe. Dette gjelder spesielt innenfor sosiale nettverk, som nettopp har som hensikt å inkludere flest mulig mennesker for sosialisering.

Både Facebook og Twitter blir i dag brukt til helt andre formål enn hva de først var tenkt. Utviklingsprosessen var tildels ulik, da Facebook ble utviklet som et skoleprosjekt, mens Twitter ble utviklet som et forskningsprosjekt for bruk innenfor et selskap. Likheten var at ingen av dem fra begynnelsen av hadde spesielt store mål med sine prosjekt. Begge disse nettverkene er eksempler på system som er blitt utviklet underveis som nytteverdien har blitt oppdaget. Funksjonaliteten systemene har i dag, var ikke planlagt fra starten, og var dermed ikke en del av en tradisjonell utviklingsmodell. Dersom planen først var klar, og systemene skulle vært utviklet fra grunnen av, kunne de i utgangspunktet brukt en tradisjonell utviklingsmetode. En slik metode ville vært lite aktuelt å ta i bruk etter at første versjon av systemet allerede er lansert, og målet er å videreutvikle det. Videre versjoner vil jo nettopp basere seg på å tilfredsstille brukerne ved å implementere deres forslag og behov. Man kan dermed anta at andre metoder bør benyttes, ettersom det trolig er stor forskjell på å utvikle et system fra scratch eller videreutvikle det. Facebook er et godt eksempel på det, og jeg vil i følgende gå nærmere inn på utviklingen. (Facebook, 2009 og Twitter, 2009)

3.1 Facebook

When we first started Facebook almost five years ago, most of the people using it were college

students in the United States. Today, people of all ages—grandparents, parents and children—use Facebook in more than 35 different languages and 170 countries and territories. - Mark Zuckerberg (2009)

Facebook ble hovedsakelig utviklet som et skoleprosjekt på Harvard University av Mark Zuckerberg i 2004. Først den 26. september 2006, ble Facebook tilgjengelig for enhver med gyldig e-postadresse. I første halvdel av 2007 var det en eksplosiv økning i interessen, hvor en gikk fra anslagsvis 3000 medlemmer ved årsskiftet 2006/2007 til i første halvdel av 2007 å ha godt over 100 000 medlemmer. I dag har Facebook over 900 ansatte. (Facebook, 2009)

Grunnen til at nettsamfunnet tok helt av, til utviklernes store overraskelse, kan ha en sammenheng med situasjonen på web denne perioden. Muligheten Web 2.0 gav for interaktiv kommunikasjon mellom menneskegrupper så nemlig ut til å appellere hos folk flest. For ordens skyld, Web 2.0 er en betegnelse på den nye situasjonen på World Wide Web med interaktive nettjenester. I følge O'Reilly (2005) startet fremveksten rundt 2001, etter det kjente dot.com krasjet.

"The web was more important than ever, with exciting new applications and sites popping up with surprising regularity. What's more, the companies that had survived the collapse seemed to have some things in common." (O'Reilly, 2005)

Web 2.0 ble et vendepunkt for nettet. I O'Reillys tabell kan vi blant annet se at de tradisjonelle "hjemmesidene" gikk over til å bli blogger og profilsider, hvor leserne kan legge igjen sine egne meninger. Web 2.0 dreier seg altså om en ny generasjon nettjenester, som vektlegger samarbeid og deling mellom brukerne. Facebooks visjon er i dag utviklet til å nettopp gi mennesker muligheten til å dele innhold og gjøre verden mer åpen og sammenkoblet.

The full potential of the web is to make the world more open, so everyone has a voice and can share what is important to them. (Facebook, 2009)

Da nettsamfunnet Facebook så ut til å bli godt tatt i mot av flere enn først antatt, ble det satt i gang et utviklingsteam. Nå var det vel og merke ikke snakk om å utvikle systemet av fra grunn – det var jo allerede gjort, men å utvikle det videre. Når det gjelder sosiale nettverk er det på sett og vis rimelig å si at det til en viss grad er brukeren som styrer utviklingen. Deres rolle er sentral for at nettsamfunnet skal fungere, og utviklerne må dermed ta hensyn til deres behov og hva som appellerer. Facebook har valgt å trekke denne grensen veldig langt. I tillegg til at de selv oppdaterer

systemet med ny funksjonalitet, finnes det også muligheter for brukeren å selv utvikle tilleggsapplikasjoner som fungerer sammen med Facebook. Dette kan være alt fra tester og spill til andre «gadgets». Det finnes over en million applikasjonsutviklere fra mer enn 180 land. I tillegg har over 15,000 andre websider, enheter og applikasjoner implementert «Facebook Connect» siden det ble tilgjengelig i desember 2008. Facebook Connect er et set av APIs for utviklere som lar brukere sammenkoble identitet og koblinger over alt. Facebook lar altså systemet integreres med andre systemer brukeren benytter seg av. Et eksempel er tjenesten Flickr, en sosial bildedelingstjeneste, hvor bildene man laster opp på Flickr, også spres på ens Facebookprofil. (Facebook, 2009)

De overnevnte egenskapene øker i stor grad systemenes funksjonalitet. I tillegg har brukeren mulighet for å være aktiv på Facebook uten å være på selve nettsiden, men via utvidelser på andre enheter/applikasjoner.

3.2 Twitter

Twitter ble utviklet av selskapet «Obvious» fra San Francisco med Jack Dorsey i spissen. Det begynte som et forskningsprosjekt, der den første prototypen ble laget på to uker i mars 2006, og tatt i bruk innad i selskapet allerede i august samme året. Tjenesten ble fort veldig populær, og det gav snart mening for Twitter å flytte seg til brukere også utenfor «Obvious». I mai 2007, ble selskapet Twitter Inc. grunnlagt. Ideen bak var enkel;

Jack Dorsey had grown interested in the simple idea of being able to know what his friends were doing. Specifically, Jack wondered if there might be an opportunity to build something compelling around this simple status concept. When he brought the idea up to his colleagues, it was decided that a prototype should be built. (Twitter, 2009)

Twitters mål var i utgangspunktet en microbloggingstjeneste hvor folk kunne poste hva de drev med. Det kan virke som om utviklerne ønsket å forenkle og forkorte kommunikasjonsprosessen, ettersom en melding kun kan inneholde 140 tegn. I følge dem selv, var det da spørsmålet; «Hva gjør du?» som skulle stå i fokus. Dette var vel og merke tenkt skulle gjelde mellom vennegrupper. Ettersom det ikke ble tilfellet, da ens venner gjerne ikke hadde tjenesten i starten, ble det til at man kommuniserte med «fremmede» man hadde i følgerlisten. Det ble da mindre viktig å vite hva den andre spiste til middag, og viktighetsfaktoren i en melding steg til en viss grad. Folk begynte gjerne

å poste nyttige lenker, nyheter og informasjon som kunne være av interesse for andre. Flere grupper i samfunnet, deriblant politikere, bedrifter og journalister oppdaget dermed nytten av Twitters effektive informasjonsspredning. Utviklerne av Twitter har tydeligvis fulgt med på hvordan tjenesten har utviklet seg, og hvordan den ble brukt av mennesker over hele verden. I tillegg til å legge til nye funksjoner, endret de også visjonen i 2009. Nåværende visjon er at mennesker skal kunne dele og oppdage hva som skjer i verden akkurat nå, hvor som helst i verden. Det baseres på at en selv velger hvem man skal følge, og man vil dermed kunne utelukke og filtrere uinteressant informasjon. (Twitter, 2009)

Twitter emphasized their news and information network strategy in November 2009 by changing the question it asks users for status updates from "What are you doing?" to "What's happening?". (Twitter, 2009)

Twitters visjon har altså endret seg fra da det var nyutviklet til nå – etter at brukerne har benyttet det til andre former enn hva som var tenkt. For at det skal fungere med slike endringer underveis, arbeider utviklerne med et webapplikasjonsrammeverk kalt *Ruby on Rails* (Twitter 2009). Dette er et en fri programvare skrevet i programmeringsspråket Ruby. Rammeverket skal gjøre det raskere og enklere å utvikle databasedrevne nettsider ved å automatisere genereringen av kode for å presentere, endre og slette informasjon i en database. At en programvare er fri, vil si at den er utgitt under en fri programvarelisens. Slike lisenser gir brukeren mulighet for å redistribuere og endre programvaren. Free Software Foundation har utarbeidet de tre frie lisensene GPL, LGPL og GFDL. Disse gir brukeren friheter til å studere hvordan programmet virker for å tilpasse det til sine behov, og forbedre programmet for å gi det ut til offentlig eie. Dersom kildekoden er åpen kan de også kjøre programmet uansett hensikt og distribuere kopier for å hjelpe andre. (GNU, 2009 og Rubyonrails, 2009)

Twitter er for øvrig utviklet og designet på Mac-maskiner, ved bruk av *Omnigraffle* og *Photoshop*. Ved å bruke rammeverket Ruby on Rails fikk utviklerne av Twitter mulighet for å arbeide enkelt og effektivt.

«Our team likes to deploy features and changes multiple times per day. Rails provides skeleton code frameworks so we don't have to re-invent the wheel every time we want to add something simple like a sign in form or a picture upload feature.» (Twitter, 2009)

Ruby on Rails er ment til å brukes sammen med en agil utviklingsmetode som blir brukt av

utviklerne som benytter seg av RAD (Rapid application development), en rask systemutviklingsmetode som bruker minst mulig tid på planlegging, og mer på selve utviklingen av prototypen. RAD var for øvrig en respons på Fossefallsmodellen, som har til hensikt å fungere mer effektivt ved å dele prosjektet inn i mange mindre prosjekt som kan levere produkter trinnvis, basert på fleksibilitet. Ruby on Rails kan for eksempel benyttes sammen med Extreme Programming (XP). Kombinasjonen av XP, som er utformet med et sett av enkle metoder, og rammeverket Ruby on Rails, gir utviklerne stor grad av fleksibilitet gjennom hele prosessen. Ved bruk av XP blir systemet testet kontinuerlig, mens utviklerne jevnlig mottar tilbakemeldinger. Vi ser dermed at denne modellen med fordel kan benyttes i utviklingen av sosiale nettverk, der man er avhengig av fleksibilitet. Også SCRUM, som er en kjent agil metode, kan benyttes. Forskjellen på modellene er fokuset, som i XP er basert på rene, stabile koder, mens SCRUM fokuserer på vellykket administrasjon av prosjektet. Det er også mulig å kombinere modellene. (Rubyonrails, 2009 og Martin, 2003)

3.3 Integrering med andre system

Ettersom mange av systemene som finnes i dag er basert på open-source, er det mulig å integrere et eget system med andre. Dette skjer både med Facebook og Twitter. Til Twitter har det for eksempel blitt utviklet en rekke Twitter-klienter, som samler alle kontakter og meldinger i et eget program. Man slipper dermed å forholde seg direkte til nettsiden. Slike klienter er tilpasset og spesielt utviklet både for mobiltelefoner og datamaskiner.

Twitter's core technology is a device agnostic message routing system with rudimentary social networking features. By accepting messages from sms, web, mobile web, instant message, or from third party API projects, Twitter makes it easy for folks to stay connected. (Twitter, 2009)

I følge Qualman (2009) foregår 80 prosent av Twitter-bruken utenfor selve Twitter. Det vil si at mennesker oppdaterer sin status over alt, via for eksempel mobiltelefonen, når som helst. Facebook er også populære via det mobile nettet, og mer enn 65 millioner bruker Facebook via mobiltelefon. Dette øker i stor grad systemets fleksibilitet og kvalitet, da brukerne selv får bestemme hvordan de vil bruke systemet, og ikke er tvunget til å forholde seg til en nettside.

Gjennom Web CMS (Content Management System) på for eksempel blogger, kan brukere la et innlegg postes på Facebook eller Twitter. Dette foregår også andre veien; En melding postet på

Twitter kan også automatisk postes på brukerens blogg. Alle applikasjonene integrerer dermed med hverandre på en måte som samler informasjon, og kobler sammen nettverk. Dette skaper helt nye muligheter for en bruker. Hans/hennes teknologiske innsikter har nå lite å si, for hvor vidt en kan delta på internett. Denne kombinasjonen av teknologi og god brukervennlighet gjør at gjennom sosiale nettverk og CMS får brukeren mulighet til å publisere både tekst, lyd og bilder, via et par enkle tastetrykk.

3.4 Interaksjonsdesign

Facebook's målgruppe er temmelig vid. Alt fra 13-åringer til oldeforeldre benytter seg av det, og innenfor aldersforskjellene finnes det også et hav av ulike brukere; Noen av dem er datanerder, andre vet knapt hva en nettleser er. Dette krever nøye gjennomtenkning fra utviklernes side med tanke på valg av brukergrensesnitt. Utfordringen blir dermed å utforme et grensesnitt som både bestemor kan bruke, og som den erfarne bruker ikke går lei av. Facebook har fått kritikk fra noen for å være for omfattende, det kan rett og slett bli for mye informasjon for noen, spesielt med tanke på alle applikasjonene som er integrert med nettsamfunnet. Dette har utviklerne løst ved å utvikle en såkalt «lite»-versjon av Facebook. Denne versjonen utelukker alt av applikasjoner, og forholder seg til en forenklet forside med statusoppdateringer, som minner om prinsippet med Twitter, samt at man får tilgang til profil, arrangementer og chat. Chat-funksjonen her er tradisjonelt utformet, slik man gjerne kjenner de eldre chat-forumene. Facebook sin enkle navigasjon gir brukerne rask tilgang til hovedsidene og applikasjonene; Profil, venner, nettverk og inboks – disse sentrale sidene plassert øverst og lett tilgjengelig, utgjør en god brukeropplevelse. Ved forståelige symbol, knapper og ikon, kan en i stor grad si at Facebook oppfyller flertallet av de sentrale designprinsippene som gjelder for god nettdesign. (Sharp, Rogers, Preece, 2007)

4.0 Metoder for utvikling av sosiale nettverk

Ved å forholde seg til en bestemt metode, kan utviklerne holde oversikt over budsjett og tidsperspektiv for de ulike fasene, som jeg tok for meg i avsnitt 2. Dette er også et positivt utgangspunkt for sosiale nettverk, men ettersom Facebook ble utviklet som et skoleprosjekt er det lite trolig at det ble brukt en tradisjonell utviklingsmetode. Først etter at Facebook ble tatt i bruk av mange mennesker, og man skjønnte det store potensialet for nettverket, ble det satt i gang en virkelig utviklingsprosess. Antageligvis ble det fra starten av ikke brukt en utviklingsmetode i det hele tatt, og det er også tvilsomt at det ble utviklet krav på forhånd. Facebook er dermed et tilfelle hvor systemet har blitt opprettet før selve systemutviklingsprosessen. For videreutviklingen ble det derimot mest sannsynlig benyttet en metode, og jeg vil i følgende drøfte hvilke metoder som er best egnet for denne type systemutvikling.

Utviklerne av sosiale nettverk har mye å forholde seg til, da nettverket sannsynligvis vil være under utvikling i lengre tid, samtidig som det er i bruk. Dette er tilfellet både for Facebook og Twitter. De kommer stadig med oppgraderinger og nye funksjoner. Dermed kan vi anta utviklerne bruker agile metoder. De tradisjonelle metodene forutsetter at det på forhånd er mulig å klart definere hva som skal gjøres i hver fase. En agil metode derimot, vi si en smidig metode. Dersom forandringer kreves, kan utviklerne gå tilbake i prosessen og endre. (Martin, 2003)

Agile metoder gjør det enkelt å legge til nye funksjoner, ettersom man utvikler programvare som en iterativ prosess. En funksjon blir utviklet for så å bli vist frem til en kunde. På den måten vil man kunne få tilbakemeldinger som man kan benytte i videre utvikling. Agile metoder kan brukes med stort hell i forbindelse med et sosialt nettverk, hvis man antar kundene som brukerne.

Før man kan fastsette krav, må man forstå hvem brukerne er og hva målene deres er ved å bruke systemet. I startfasen av utviklingsprosessen for sosiale nettverk, er det ofte vanskelig å vite direkte hva brukerens mål skal være, og brukerne vet også ganske sjeldent hva de vil ha og hva de trenger. Når de ikke vet hva de kan få, blir det opp til utviklerne å lage nye funksjoner, presentere dem for brukerne, og dermed observere deres bruk, samt motta tilbakemeldinger. Eksempelvis kan dette foregå via spørreundersøkelser, eller man kan foreta målinger av hvor mye en gitt funksjonalitet blir brukt i applikasjonen. (Martin, 2003)

Det finnes for øvrig et eget konsept som heter «Agile web development». Dette er en agil model som er spesielt tilpasset utviklingen av nettbaserte system. Tidsperspektivet denne modellen baserer seg på er kortere og mer effektivt enn andre modeller, og den tillater i tillegg endring av krav sent i prosessen. Fossefallsmodellen, som fastsetter krav i en tidlig fase, ville feilet dersom kravene ble endret senere i prosessen. Bruken av agil webutvikling vil dermed være en stor fordel ved utvikling av sosiale nettverk, med hensyn til at man fra starten ikke alltid kan vite hva brukeren vil ha. Jeg refererer til tidligere eksempler med Twitter, som endret strategi nylig, etter et par års bruk. Ut i fra denne sammenhengen, kan det virke som om Twitter med stor fordel kunne brukt denne metoden. Et viktig konsept med Agile modeller er nettopp at det ikke dreier seg om en komplett systemprosess, men fokuset ligger i stedet på effektiv modellering og dokumentering. Det inkluderer dermed ikke testing etter at systemet er laget, men underveis som systemet modelleres. (Ambler, 2002)

Because AM's focus is on a portion of the overall software process you need to use it with another, full-fledged process such as eXtreme Programming (XP), DSDM, SCRUM, the Agile Unified

Process (AUP), or the Rational Unified Process (RUP). (Ambler, 2002)

I tillegg til rammeverkene Ambler nevner, finnes det også en rekke andre typer av Agile systemutviklingsmetoder en kan ta i bruk for å enklere levere et brukbart funksjonelt system.

4.1 Problemer i og etter utviklingsfase

Det er flere faktorer som kan spille inn når et system på nettet ikke går så bra som utviklerne hadde håpet på. Det kan blant annet oppstå brukere som skaper problemer for andre brukere eller utviklere. Facebook har for eksempel opprettet et eget team som jobber med mobbing.

Open source kan også bli et problem, i den grad av at folk ikke følger prinsipper og regler som gjelder. Andre utviklere kan ta i bruk ens kildekode og dermed utvikle noe bedre.

Et annet problem som kan oppstå er tilretteleggingen. I følge Boyd og Elleson (2007) ble det første nettsamfunnet hvor vi kan finne sosiale tendenser, opprettet allerede i 1997. Dette var nettsiden "SixDegrees.com", som gjorde det mulig for brukere å lage profiler, liste vennene og surfe gjennom sin egen og andres venneliste. AIM og andre chat-applikasjoner hadde naturligvis integrert vennelister tidligere, men ikke der disse vennene var synlig for alle som så ens profil. Selv om dette kunne virke som en spennende løsning, gikk det ikke så bra;

"While SixDegrees attracted millions of users, it failed to become a sustainable business and, in 2000, the service closed. Looking back, its founder believes that SixDegrees was simply ahead of its time".

Hvorfor nettsamfunnet ikke overlevde krasjet kan forklares i at flesteparten av brukerne den gangen ikke hadde utvidede nettverk med venner som ofte var online. Det ble rett og slett for lite å gjøre etter at man hadde "godkjent" vennene. Ut i fra disse observasjonene kan vi anta at det rett og slett var en for tidlig fase for et slikt sosialt samfunn - det var ikke oppdaget av nok mennesker enda. Etter at flere sosiale profilsider oppstår etter 2001, ser folk mulighetene de bringer med seg. Utviklingen av sosiale nettverk øker dermed, og i dag ser vi en enorm vekst av ulike nettsamfunn som lar folk dele innhold og sosialisere seg. Mange av dem har oppstått uten å være fastsatt en utviklingsmetode, men i stedet blitt videreutviklet etter at første versjon er tatt i bruk. (Boyd og Elleson, 2007)

5.0 Videre utvikling

Verken Facebook eller Twitter er ferdige system. Begge jobber hele tiden med nye oppgraderinger og funksjoner, og antall ansatte vokser jo flere brukere nettsamfunnet får.

We continue to follow user behavior and pay close attention to feature requests. We launched our mobile site, m.twitter.com after getting lots of requests for this feature. However, we are also very much guided by our philosophy of keeping things simple and intuitive so we like to restrain ourselves with regard to features. That being said, people seem to be calling for more interesting ways to share feedback with one another and organize themselves and their updates into groups--we're paying attention to these requests. (Twitter, 2009)

Ut ifra hva Twitter skriver om videre planlegging, kan vi anta at det stadig vil bli utvidet, etter hvert som det oppstår behov fra brukerne. Vi ser at alt dreier seg om en typisk agil prosess hvor utviklingen skjer i takt med forespørsler fra brukerne. Dette gjelder også for Facebook.

6.0 Konklusjon

Etter å ha sett på typiske trekk ved utviklingen av sosiale nettverk, med Facebook og Twitter som eksempler, har det vist seg at i flere tilfeller er ikke brukerne og deres behov konkret fastsatt, og det er da ikke mulig å utvikle en fullstendig kravspesifikasjon i starten av utviklingsprosessen. Dermed kan det virke som om de tradisjonelle metodene ikke alltid er det riktige valget i en slik utviklingsprosess. De kan i stedet kombineres med agile metoder, eller metoder som er spesielt tilpasset utviklingen av webbaserte system. Det finnes flere agile metoder som er tilpasset for bruk i utviklingen av nettbaserte system, også rammeverk. Ved at kravene kan utvikles underveis, samt systemet testes mens det er i drift av brukere, oppnår utviklerne den fleksibiliteten de behøver. Det er altså en betydelig forskjell fra system som utvikles fra grunnen, og et system som videreutvikles fra en allerede utviklet førsteversjon.

Om brukerne oppnår det som var utviklernes mål, er ikke tilfellet i de sosiale nettverkene jeg tok for meg som eksempler. Utviklernes mål med både Twitter og Facebook er ulike fra slik systemene fungerer i dag, og vi kan dermed se hvordan både mål og krav har endret seg gjennom prosessen fra første versjon til nåværende versjon. Dette er trolig i stor grad et resultat av brukernes deltakelse i utviklingsprosessen. For at et sosialt nettverk skal appellere hos brukerne, må man gi dem det de vil ha. Disse behovene får utviklerne stor innsikt i, ved å la dem bruke nettverket, samtidig som det er under utvikling. Behovet for tilbakemeldinger fra brukeren blir dermed avgjørende for utviklingen av sosiale nettverk, både direkte tilbakemeldinger og tall over hvilke funksjoner som blir brukt mest.

Systemer som blir tatt i bruk før selve utviklingsprosessen, slik som Facebook og Twitter, kan settes i sammenheng med enkeltpersoners gode idèer og behovet som finnes på nettet i dag. Det handler kanskje like mye om utviklerne, brukerne av systemet og samfunnets situasjon, som de tekniske løsningene.

7.0 Litteratur

Ambler, Scott (2002). *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. John Wiley & Sons, Inc., New York.

Boyd, Danah M. / Ellison, Nicole B. (2007) *Social network sites: Definition, history, and scholarship* [Internett], Indiana.edu, Tilgjengelig fra: <<http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html>> [Nedlastet 21.10.09]

Facebook (2009). [Internett], Facebook.com. Tilgjengelig fra: <http://www.facebook.com>> [Nedlastet 22.11.09]

Gnu (2009). [Internett], Gnu.org. Tilgjengelig fra: <<http://www.gnu.org/philosophy/free-sw.html>> [Nedlastet 01.12.09]

Martin, Robert C. 2003) *Agile Software Development: Principles, Patterns, and Practice*. Prentice Hall, Pearson Education.

Qualman, Erik (2009). *Statistics Show Social Media Is Bigger Than You Think* [Internett], Socialnomics.net, Tilgjengelig fra: <<http://socialnomics.net/2009/08/11/statisticsshow-social-media-is-bigger-than-you-think/>> [Nedlastet 17.11.09]

Reilly, Tim O (30. september 2005). *What is web 2.0?* [Internett], Oreilly.com, Tilgjengelig fra: <<http://oreilly.com/web2/archive/what-is-web-20.html>> [Nedlastet 18.10.09]

Ruby on Rails (2009). [Internett], Rubyonrails.org. Tilgjengelig fra: <<http://rubyonrails.org/screencasts>> [Nedlastet 18.11.09]

Sharp, H. Rogers, Y. & Preece, J. (2007) *Interaction design: beyond human-computer interaction.* 2. utg. Wiley.

Twitter (2009). [Internett], Twitter.com. Tilgjengelig fra: <<http://www.twitter.com>> [Nedlastet 20.11.09]

Weaver, Philip (2004). *Success in your project – a guide to student system development projects*. Prentice Hall, Financial Time.