



LUNDS
UNIVERSITET

Project Management och mjukvaruutveckling inom Open Source

Författare: Alexander Crespo Nyqvist

Utgivare: Institutionen för informatik
Handledare: Markus Lahtinen
Publiceringsår: 2012
Uppsattstyp: Kandidatuppsats 15hp
Språk: Svenska
Nyckelord: Open Source, project management, planering, mjukvaruutveckling, innehållsanalys, OSS

ABSTRAKT

Syftet med studien är att pröva hypotesen att *project management* inträffar vid mjukvaruutveckling inom Open Source. Studien avser endast att undersöka planeringsfasen av *project management* vid utveckling av mjukvara inom Open Source och inte att beskriva hur planering sker i praktiken. Genom att analysera den dokumenterade kommunikationen inom Open Source-*communities* undersöks hur aspekter av *project management*, med fokusering på planering, hanterades, diskuterades och applicerades. Riktad innehållsanalys ligger till grund för analysen och en utarbetad teoretisk referensram ligger till grund för studien. Den teoretiska sammanställningen är utarbetad efter publicerad litteratur inom ämnet och tidigare publicerad forskning. Resultatet av studien pekar på en frånvaro av traditionella *project management*-utföranden samtidigt som bevis för tillämpning av *project management* hittas.

INNEHÅLLSFÖRTECKNING

Sid.

ABSTRAKT	2
INLEDNING	6
1.1 Bakgrund	6
1.2 Problemformulering	8
1.3 Frågeställning	9
1.4 Syfte	9
1.5 Komponenter inom Open Source	9
1.5.1 The Open Source Initiative	9
1.5.2 Open Source Software	9
1.5.3 Utveckling inom Open Source-communities	10
1.5.4 Free Software	10
1.6 Avgränsningar	10
1.7 Begreppsdefinitioner	11
1.7.1 Open Source-projekt	11
1.7.2 Open Source-community	11
1.7.3 Project management	11
1.7.4 Objectives och Goals	11
1.7.5 Open Source-fenomen	11
2. FORSKNINGENS SYN PÅ PROJECT MANAGEMENT OCH RELATIONEN TILL OPEN SOURCE	12
2.1 Organisationer och projekt i relation till Open Source	12
2.2 Project management	12
2.3 Syfte med planering	13
2.4 Projektplanering	13
2.4.1 Mål	14
2.4.2 Budget	14
2.4.3 Schema	15
2.5 Utgångspunkter för studien	15
2.6 Projektplanering och Open Source	16
2.7 Undersökningsmodell	17
3. METOD	19
3.1 Litteratursökning och kritisk granskning	19
3.2 Studieobjekt	19
3.3 Empiri	21
3.4 Riktad innehållsanalys	23
3.5 Validitet och reabilitet	23
4. ANALYS AV STUDIEOBJEKTEN	25
4.1 Empiriska fynd	25
4.2 Sammanställning kopplad till forskningsfrågan	30
5. PROJECT MANAGEMENT INOM OPEN SOURCE	32
5.1 Empiri i relation till forskningens syn	32
5.1.1 Planering	32
5.1.2 Mål	32
5.1.3 Budget	33
5.1.4 Schema	33
5.1.5 Tidigare studier	33
5.1.6 Analys	34
6. SLUTSATS	36

6.1 Egna reflektioner.....	36
REFERENSER	
BILAGOR	

TABELLFÖRTECKNING

Tabell 2.1 Project management-kategorier enligt Koontz & Wehrich, Kerzner och PMBOK.....	12
Tabell 2.2 Planeringens komponenter enligt Antvik & Sjöholm, Kerzner och PMBOK.....	14
Tabell 2.3 Sammanställning av komponenter och faktorer som enligt samtliga författare i litteraturgenomgången utgör projektplanering.....	15
Tabell 3.1 Sammanställning av studieobjekten.....	20
Tabell 4.1 En sammanställning av analysenheten i relation till studieobjekten.....	27

FIGURFÖRTECKNING

Figur 2.1 Undersökningsmodell.....	18
Figur 3.1 Utdrag ur Androids mailinglista.....	21
Figur 3.2 Utdrag ur WebKits arkiv.....	22
Figur 3.3 Utdrag ur OADs forum.....	22

With my community gone, to continue as before was impossible
(Richard Stallman, 1999)

1. INLEDNING

1.1 Bakgrund

Mjukvaruindustrin är en industri som omsätter miljarder dollar årligen (229 miljarder dollar 2010 enligt Rönkkö, Peltonen & Pärnänen, 2011) och sysselsätter tusentals människor världen över. Mjukvaruindustrin har kommit att bli en betydande industri i samhället, och i takt med att Internet växer, en betydande faktor för dess fortsatta utveckling. Healy och Schussman (2003) menar att samhället är i ständig utveckling där teknologin, däribland mjukvara (källkod), är en avgörande faktor för företag, institutioner och stater. Open Source (OS) och Open Source Software (OSS) har blivit en betydande del av mjukvaruindustrin.

Från att ha existerat i lokala miljöer har mjukvaruutveckling generellt utvecklats till en global arbetsmetod där människor i olika tidszoner och kulturer samarbetar med att utveckla mjukvara för en eller flera organisationer. Företag har strävat efter att uppnå en global utvecklingsmiljö med syfte att effektivisera produktionen och uppnå strategiska fördelar gentemot konkurrenter och marknader. Med hjälp av IT-infrastruktur har olika verktyg, som t ex kommunikationskanaler, skapats. Verktygen möjliggör koordinering av arbete som är spritt över planeten. 1962 skapades Internet vid MIT-universitetet. Internet har legat till grund för den globalt digitala kommunikation som vi etablerat idag och möjliggör globala utvecklingsmiljöer. "Before the Internet came about, the power of a "collective mind," so to speak, had never been realized" (Tiwari, 2010). Sedan dess har företag och institutioner utforskat och beskrivit informationsteknologi för att hitta lösningar och metoder som bidrar till fortsatt utveckling inom det högteknologiska samhället (Raymond, 2003). IT har sedan starten varit ett forskningsområde som kan undersökas och vidareutvecklas av alla, från företag och institutioner, till privatpersoner.

Företags försök att kontrollera mjukvarumarknaden genom licenser och användaravtal har resulterat i ett fenomen som kom att förändra industrin. Ett exempel på ett icke-kommersiellt projekt är mjukvaruprojektet GNU (GNU's Not Unix) som Richard M. Stallman etablerade 1984 med syfte att skapa ett fritt operativsystem. "I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free." (gnu, 2012). När Richard Stallman anställdes 1971 vid MIT Artificial Intelligence Lab (AIL) blev *software-sharing* en självklar del av hans arbete. Genom att vara en del av AIL's mjukvaruutveckling och medlem i "The AI lab hacker community" bestod arbetsuppgifterna även av att hantera andra företag och universitet som var intresserade av att få tillgång till system och mjukvaror (Stallman, 1999). Samarbeten som etablerades utgick från en ömsesidig grund och informationen dem emellan delades öppet.

We did not call our software "free software", because that term did not yet exist; but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalize parts of it to make a new program. (Stallman, 1999, s. 1)

Systemen som utvecklades på 1960-talet var få och hade inte förmågan att tillgodose företagets behov på 1980-talet. TCP/IP-protokollet utvecklades under 1980-talet av The Defense Advanced Research Projects Agency [DARPA] (Raymond, 2003) vilket

resulterade i att tidigare system var obsoleta. AIL valde att ersätta det tidigare systemet (the Incompatible Timesharing System [ITS]) med ett "non-free timesharing system" (Stallman, 1999; s. 1) med tillhörande operativsystem. Detta resulterade i att man var begränsad i användandet av en dator. Programmerare arbetade fritt med att utveckla mjukvara (operativsystem, program m m) för datorn och var inte begränsade mot ett operativsystem. I och med att manipulering av mjukvara förbjöds blev källkod hemligstämplad. "[...] you had to sign a nondisclosure agreement even to get an executable copy" (Stallman, 1999, s. 1). Från att ha utvecklat i en öppen miljö där källkod och mjukvaror kunde spridas fritt hade Stallman svårt att acceptera den verklighet han tvingats in i.

This meant that the first step in using a computer was to promise not to help your neighbor. A cooperating community was forbidden. The rule made by the owners of proprietary software was, "If you share with your neighbor, you are a pirate. If you want any changes, beg us to make them." (Stallman, 1999, s. 2)

Samtidigt blev många aktiva utvecklare i The AI lab hacker community anställda av andra företag där de utförde samma arbetsuppgifter som tidigare men utvecklade mjukvara under licenser med syfte att skapa produkter. Effekten av detta var att *communitiet* mer eller mindre försvann. Stallman ansåg att mjukvaruindustrin inte längre var tilltalande under dessa premisser och vidtog åtgärder. Resultatet blev införandet av ett nytt fritt operativsystem och utformandet av en licens. 1985 startade Stallman The Free Software Foundation och vad viktigare var, han designade licensen The General Public License (GPL) (Raymond, 2003). Syftet med GPL var att hindra andra att göra anspråk på rättigheter till källkod. Rörelsen som Stallman satt i gungning fick ett ordentligt uppsving när Linus Torvald 1991 valde att släppa källkoden till sin Unix-kernel (Linux) under GPL-licensen (Bar & Fogel, 2003). I kombination med Unix-verktygen som GNU-projektet erbjöd kunde GNU och Linux erbjuda ett komplett fritt operativsystem.

Genom att etablera konceptet "fri mjukvara" i den kommersiella världen med exempelvis fonder och organisationer kunde OS-fenomenet utarbetas till en delmängd av den existerande mjukvaruutvecklingen. Det var nu möjligt att, i enlighet med lagar och förordningar, rubricera mjukvara och definiera "fri mjukvara". Termen Open Source började användas först 1998 (Raymond, 2003) av människor som var aktiva inom utvecklingen av fri mjukvara för att beskriva vad de jobbade med (Stallman, 1999). "Fri mjukvara" har i dagens läge fått en övergripande benämning - Open Source - och fenomenet har inte bara etablerats inom mjukvaruindustrin utan också inom flertalet andra industrier. När konceptet Open Source blivit accepterat av det globala samfundet var det möjligt att utnyttja de kommunikationskanaler och tillvägagångssätt som erbjöds. OS och OSS är intressanta att titta på eftersom de utgör en betydande del av mjukvaruindustrin. "OSS is becoming famous in the field of software development and million of people are getting the benefit from this type of development" (Tawari, 2010, s. 124).

Med en mjukvaruutveckling baserad på en disciplin som förespråkar ideellt arbete, men som inte fokuserar på vinstdrivandet, är utveckling av mjukvara inom OS ett annorlunda fenomen sett till traditionell utveckling. Vidare kan man ställa sig frågan hur OS-fenomenet förhåller sig till teorier som beskriver liknande verkligheter. Kan man, sett till det faktiska utvecklandet av mjukvara (planering, utförande, deadlines o s v) hitta gemensamma nämnare mellan OS-utveckling och traditionell utveckling eller är dessa skilda åt?

Ett OS-projekt startas av en eller flera personer som av intellektuella, personliga eller affärsmässiga skäl anser projektet intressant.

Every good work of software starts by scratching a developer's personal itch... too often software developers spend their days grinding away for pay at programs they neither need nor love. But not in the (open source) world... (Hippel, 2003, s. 211).

Initiativtagarna till projektet blir oftast projektledare och/eller projektansvariga och utvecklar en första version av mjukvaran som beskriver det önskade slutresultatet. Källkoden görs sedan tillgänglig för allmänheten. Initiativtagarna förser projektet med mailinglistor och forum för att underlätta kommunikation mellan deltagarna. I dessa mailinglistor och forum kan deltagarna söka hjälp, sprida koder och information om projekten. Implementering av OS-projekt börjar gradvis bli enklare och effektivare eftersom projektdesignen är bättre förstådd och färdiga lösningar för projektdrivandet erbjuds via webben. Hur ett OS-projekt bedrivs, sett ur ett organisatoriskt perspektiv, varierar från projekt till projekt.

Project management avser att leda projekt i symbios med organisationens strategi (Larson & Grey, 2011). Det finns flera teorier inom *project management* men det existerar ingen universal teori som kan appliceras på alla organisationstyper. Med det menas inte att *project management*-teorier är alltigenom olika. Efter en mer djupgående granskning av teorier kan det konstateras att gemensamma attribut och aspekter existerar med syfte att genomföra uppgifter koordinerade med flera deltagare. Genom att sammanfoga organisationen med projektet utformas ramarna för projektets utförande och en aktivitet skapas där organisationen bistår med resurser.

OS-*communities* kan liknas vid en social enhet av människor och beskrivas som en organisation som bedriver projektdriven verksamhet. Det är därför inte fel att anta att OS-fenomenet bedriver någon form av *project management*.

1.2 Problemformulering

De organisationer som utövar *project management* har struktur, hierarki och ekonomiska resurser för att genomföra samordnade aktiviteter. Detta resulterar i att organisationer kan förvänta sig prestation och engagemang av organisationens medlemmar som i slutändan bidrar till att planer realiseras och mål uppnås (Cleland, 1999). Med detta i åtanke har den traditionella organisationen resurser som inte är tillgängliga inom OS där arbetet är frivilligt (Hippel, 2003). Avsaknaden av ekonomiska resurser förmodas bidra till att en bestämd, given organisatorisk struktur saknas. Den eftertraktade kompetensen finns inom OS-*communities* men möjlighet att säkra kompetens och resurser saknas. Filosofin inom OS är att engagera efter eget intresse, vilket resulterar i att ekonomisk vinning inte är en drivande faktor. Prestation och engagemang är inte aspekter som kan kalkyleras då dessa är oberäkneliga.

Planering är en av komponenterna inom *project management*. Planering är det första steget vid genomförande av projekt och grundläggande för projektets utfall. Vilka aspekter av *project management* är aktuella vid planering av mjukvaruutveckling inom OS när grundläggande element saknas eller är annorlunda till sin natur? Sett till utvecklingen av mjukvara inom OS finner man fördelaktiga aspekter som inte är närvarande inom den

traditionella mjukvaruutvecklingen. Vid sammankoppling av två olika organisationstyper är försök att definiera bryggan mellan traditionella organisationer och OS aktuella. Hur mjukvaruutveckling inom OS förhåller sig till *project management* kan ge svar på skillnader och likheter mellan traditionella organisationer och OS-*communities*. Förståelse för detta kan klargöra styrkor och svagheter i mjukvaruutvecklingen inom OS.

1.3 Frågeställning

Uppsatsens syfte kommer att uppnås genom att följande frågeställning besvaras:

Vilka aspekter av *project managements* planeringsfas kan identifieras vid utveckling av mjukvara inom Open Source?

1.4 Syfte

OS är ett globalt fenomen som involverar många. Att utveckla mjukvara i grupper på distans är en vardaglig företeelse som ställer krav på organisering. Genom att identifiera planeringsfasens aspekter uppnås studiens syfte som var att pröva hypotesen att *project management* inträffar vid mjukvaruutveckling inom Open Source.

1.5 Komponenter inom Open Source

1.5.1 The Open Source Initiative

Open Source Initiative (OSI) är ett icke-vinstdrivande internationellt nätverk vars syfte är att utbilda och förespråka kring fördelarna med OS. OSI anses vara en brygga mellan olika OS-*communities* med fokus på sammankoppling mellan dessa. OS har blivit en utvecklingsmetod för mjukvara som utnyttjar kraften av distribuerad förhandsgranskning och processtransparens. En av de viktigaste aktiviteterna inom OS är arbetet kring OS-definition, underhåll och utveckling av licenser. OSI's Approved License Trademark and Program skapar en gemenskap och tillit mellan utvecklare, användare, företag och regeringar (opensource, 2012).

1.5.2 Open Source Software

Vid nedladdning eller köp av OS-mjukvara (Open Source Software [OSS]) erhålls i princip tre delar: maskinkod, källkod och licens. Den stora skillnaden mellan OS-mjukvara och kommersiell mjukvara är att också en källkod för programmet erhålls vid införskaffning av ett program. Förutom källkoden ger licensen också köparen rättigheter att ändra och distribuera den nya källkoden. Den mest kända licenstypen inom OS är General Public License (GPL). Gemensamt för alla OS-licenser är att källkoden, som lanseras under en licens, i någon mån måste vara öppen för allmänheten.

Diskussionen kring huruvida OS är ett lönsamt koncept ur ett framtidsperspektiv är relevant och involverar många intressenter. Utveckling av OS-mjukvara är en realitet eftersom många projekt inom OS är aktivt utvecklade på global basis. Deltagarna är samlade i OS "*developer community*" där de, med hjälp av tekniska verktyg, kan uppnå samarbete kring mjukvaruutveckling. Dessa "*developer communitie*." är spridda runt planeten och det finns inte en organiserad instans som underhåller, strukturerar eller möjliggör mjukvaruutvecklingen (opensource, 2012).

1.5.3 Utveckling inom Open Source-communities

Utvecklare kan delta i en utveckling, ”projects” (Sourceforge, 2012; code.google, 2012) genom att bli medlemmar i en OS-*community*. Stora och små organisationer och företag har etablerat ramverk och plattformar för att bidra till utvecklingen inom OS. Några exempel på sådana organisationer och företag är Sourceforge, GitHub och Google Code. I dessa *communities* är det möjligt för vem som helst att starta ett projekt och tillsammans med medlemmarna inom *communities* utveckla ett program. Ett OS-projekt är i grund och botten realiserat av människor med intresse för mjukvaruutveckling. Deltagarna i ett OS-projekt har ingen ekonomisk motivation eftersom de generellt inte tjänar pengar på att delta eller bidra till ett projekt. ”Thousands of software programmers are spending tremendous amounts of time and effort writing and debugging software, most often with no direct monetary compensation” (Madey et al., 2002, s. 1806). Ett OS-projekt har inga begränsningar vad gäller antalet medlemmar som kan involveras i en utveckling.

Kommunikationen inom OS sker främst i form av skrift. OS-*communities* bistår med forum och mailinglistor för att bidra till kommunikationen. Enligt Madey et al. (2002) visar fallstudier av utvecklingen inom OS på att det finns potentiella lärdomar att ta del av för traditionell mjukvaruutveckling. Det sägs att OSS producerar felfri kod snabbare än den stängda utvecklingen. Detta är dock ett påstående som måste undersökas ytterligare (Madey et al., 2002). OS-teamet är utspritt över planeten och deltagarna har oftast inte träffats i verkliga livet. OS-projekt är självorganiserande och genererar ofantliga mängder kod och frekventa uppdateringar.

1.5.4 Free Software

Free Software syftar inte till att mjukvaran är gratis utan till friheten att använda den. Free Software delar egenskaper med OSS men är två skilda begrepp. ”Open source is a development methodology; free software is a social movement” (gnu, 2012). OS är ett begrepp som började användas 1998 för att beskriva vad utvecklingen av Free Software var. Idag skiljer sig Free Software och OSS åt sett till värderingar och mål.

1.6 Avgränsningar

Studien avser inte att analysera utveckling av mjukvara inom OS utifrån någon annan aspekt än allmänna projektplaneringsteorier. Beskrivningar av hur utvecklingen planeras eller vilka verktyg och metoder som används kommer inte att framgå. Studien avser endast att undersöka relationen mellan traditionell projektplanering och planering inom utveckling av OS-mjukvara; inte att definiera utveckling inom OS. Benämningen av utförandet som sker inom OS-utveckling förblir odefinierad genom studien. För att undersöka hur OS-*communities* bedriver sin verksamhet jämförs aktiviteten (mjukvaruutvecklingen) mot traditionell *project management*. Studieobjekten analyseras utifrån de aspekter som sammanfaller med *project management*.

1.7 Begreppsdefinitioner

1.7.1 Open Source-projekt

Utvecklare kan delta i en utveckling av mjukvara, ”projects” (Sourceforge, 2012; code.google, 2012), genom att bli medlemmar i en OS-*community*. För att underlätta läsningen av studien kommer begreppet “OS-projekt” att användas och fortsättningsvis syfta till dessa projekt.

1.7.2 Open Source-community

För att enkelt kunna uttrycka den virtuella mötesplats, där människor som engagerar sig inom OS träffas, kommer begreppet “OS-*community*” att användas.

1.7.3 Project management

Den teoretiska genomgången har utarbetats utifrån engelska källor. För att tydligt beskriva grunden för den litterära studien, och för att utesluta eventuella feltolkningar, har begreppet “*project management*” använts. Termen *projektledning* används inte eftersom jag anser att *project management* omfattar fler organisatoriska mekanismer än projektledning. Begreppet *project management* ligger till grund för den litterära sökningen.

1.7.4 Objectives and Goals

I litteraturgenomgången har begreppen “objective” och “goal” använts. Oenigheten kring vad dessa benämningar speglar är utbredd. Vissa författare menar att “objective” belyser organisationens mål och “goal” projektets mål, medan andra författare menar att “objective” avser projektets mål. I denna studie används både “objective” och “goal” för att belysa projektets mål.

1.7.5 Open Source-fenomen

Genom studiens gång avser termen ”OS-fenomen” att belysa alla kringliggande aspekter av OS som utveckling, organisation och mjukvara och inte enbart syfta till öppen källkod.

2. FORSKNINGENS SYN PÅ PROJECT MANAGEMENT OCH RELATIONEN TILL OPEN SOURCE

För att besvara forskningsfrågan utvärderas relationen mellan *project management* och projektplanering samt vilka aspekter som är viktiga vid projektplanering enligt planeringsfasen av *project management*. Därefter presenteras hur planeringsfasen av *project management* relateras till OS.

2.1 Organisationer och projekt i relation till Open Source

En organisation definieras som en social enhet av människor som är strukturerad och förvaltd för att uppnå ett syfte eller mål. Organisationens struktur bestämmer relationer mellan funktioner och positioner samt etablerar roller, ansvar och befogenheter för att genomföra definierade uppgifter (businessdictionary, 2012). Enligt Koontz & Wehrich (1988) är medlemmarna inom en organisation medvetna om funktion och syfte. Genom att tillsammans sträva efter att realisera organisationens strategi uppnås organisationens mål (Koontz & Wehrich, 1988). Att bedriva utveckling inom OS är ett vedertaget fenomen på global nivå. Genom nätverk och *communities* samverkar deltagare med ambitionen att utveckla mjukvara. Dessa nätverk och *communities* kan likställas med en organisation.

Verksamheten inom OS-*communities* kan rubriceras som projekt driven utveckling. "A project is a temporary endeavor undertaken to create a unique product, service, or result" (Project Management Body of Knowledge [PMBOK], 2008, s. 5). Utvecklingen inom OS har gemensamma aspekter med projekt eftersom det skapas produkter, tjänster och resultat.

2.2 Project management

"The management of projects entails the crossing of functional and organizational boundaries" (Cleland, 1999, s. 43). *Project management* syftar till applicering av kunskap, färdigheter, verktyg och tekniker till aktiviteter för att tillgodose ett projekts krav (PMBOK, 2008). *Project management* innebär design och skapande av en miljö som underhåller en grupp individer att under samarbete uppnå ett gemensamt mål med hänsyn till tid, pengar och resurser (Koontz & Wehrich, 1988). *Project management* är, enligt flera författare, indelat i områden som utgör grunderna för vad man väljer att fokusera på under projektets gång. Tabell 2.1 redogör för Koontz & Wehrichs, Kerzners och PMBOK:s syn på vilka kategorier som *project management* kan indelas i.

Tabell 2.1 – *Project management-kategorier enligt Koontz & Wehrich, Kerzner och PMBOK*

Koontz & Wehrich (1988)	Kerzner (1992)	PMBOK (2008)
Planering	Planering	Initiering
Organisering	Organisering	Planering
Bemanning	Bemanning	Utförande
Ledning	Ledning	Övervakning och kontroll
Kontroll	Kontroll	Slutförande

I tabellen går det att utläsa att samtliga författare anser planering vara en av de grundläggande kategorierna för *project management*.

2.3 Syfte med planering

Planering kan generellt beskrivas som en funktion där policys, procedurer och nödvändiga program utformas för att uppnå organisationens mål (Kerzner, 1992). En plan, så även en projektplan, är en framtida kedja av händelser utformade för att uppnå ett mål (Kerzner, 1992). Planering är en övning i problemlösning och beslutsfattande som är nödvändiga för att utnyttja mänskliga och icke-mänskliga resurser (Koontz & Weihrich, 1988; Kerzner, 1992). Nästan alla projekt är beroende av en formell och noggrann planering. På grund av deras relativt korta livslängd och prioriterade resursanvändningar är det viktigt att samordna de olika aktiviteterna inom projektet (Kerzner, 1992). Syftet med projektplanen är, enligt PMBOK (2008), att inge förståelse genom ett dokument som godkänns och ligger till grund för projektets utförande, övervakning, kontroll och avslutning. En projektplan syftar till att lyfta fram milstolpar, aktiviteter och resurser för att på så sätt fastställa riktlinjer som klargör hur ett projektmål ska hanteras, och när det ska hanteras (Kerzner, 1992). Riktlinjerna fastställs genom att belysa milstolpar, aktiviteter och resurser. Projektplanen och projektdokumentationen utgör en fullständig bild av projektets omfång, tid, kostnad, kvalitet, kommunikation, risk och anskaffning (PMBOK, 2008). Kerzner menar även att ett av planeringens syften är att definiera allt nödvändigt arbete så att det är identifierbart för varje projektmedlem.

2.4 Projektplanering

Att genomföra en projektplanering kan likställas med uträttandet av förutbestämda handlingar inom en förutsägbar miljö för att uppnå ett mål (Kerzner, 1992). Cleland (1999) beskriver projektplanering som en fas där man, för att säkra resurser, analyserar framtida relationer mellan projekt och organisation. Projektplanen är både operativ (short-term) och strategisk (long-term) vilket innebär att både operativt och strategiskt tänkande är nödvändiga vid skapandet av planen (Cleland, 1999). Teoretiskt är projektplanen indelad i underkategorier där man mer detaljerat specificerar hur man analyserar de nödvändiga aspekterna (komponenter) för att uppnå en lyckad planering (Kerzner, 1992).

I tabell 2.2 framgår vilka komponenter Antvik & Sjöholm, Kerzner och PMBOK anser bör ingå i projektplaneringen.

Tabell 2.2 – Planeringens komponenter enligt Antvik & Sjöholm, Kerzner och PMBOK

Antvik & Sjöholm (2007)	Kerzner (1992)	PMBOK (2008)
Mål	Mål	Syfte
Resurser	Program	Mål
Schema	Schema	Beskrivning
Budget	Budget	Risk
Omfattning	Beräkningar	Schema
Anskaffning	Organisation	Budget
Kvalitet	Policy	
	Procedur	
	Standard	

I tabellen konstateras att mål (*objective*), budget (*budget*) och schema (*schedule*) är de komponenter som källorna i tabellen är överens om borde ingå vid projektplanering. I följande avsnitt utvecklas dessa och förklaras vad de i praktiken innebär och hur de, enligt författarna, bör utformas.

2.4.1 Mål

Enligt Koontz & Wehrich (1988) är målet inte bara slutpunkten för planering utan också slutmålet för organisering, anställning, ledning och kontroll. Målen kan vara short-term/long-term, breda eller specifika. Koontz & Wehrich (1988) menar också att "mål" har en hierarkisk struktur där delmål utgör grunden för projektets mål. Projektmålen kan beskrivas utifrån termer som kostnad, schema och teknisk prestation (Cleland, 1999). Kerzner (1992) bekräftar detta genom att, när projektets mål väl är identifierat, belysa delmålen, vilket resulterar i att kostnad och prestation kan övervakas. Kerzner menar att det är viktigt att utforma delmål som är anpassade till organisationen eftersom projektansvariga delegerar arbete till projektmedlemmarna utifrån dessa. Organisation och delmål måste kunna kopplas samman.

Vid planering av ett projekt måste arbetet struktureras i mindre aktiviteter (Kerzner, 1992). PMBOK (2008), Kerzner (1992), Koontz & Wehrich (1988), Cleland (1999) och Antvik & Sjöholm (2007) menar att projektets mål indelas i mindre aktiviteter genom att definiera "The Work Breakdown Structure" (WBS). WBS är ett verktyg för att i detalj kunna redogöra för vilka aktiviteter projektets mål utgörs av och för att dela in projektet i grupper och sedan dela in grupperna i aktiviteter. Med en utarbetad WBS ges möjligheten att tillgodose aktiviteter med resurser (Cleland, 1999). Det är genom definiering av WBS som projektet ges möjlighet att strukturera resurser, fördela ansvar, schemalägga och skapa budgetar (Kerzner, 1992).

2.4.2 Budget

Budgeten är en beskrivning av förväntade resultat i numeriska termer och anses vara det fundamentala planeringsinstrumentet (Koontz & Wehrich, 1988). Budgeten tvingar projektet att utföra en framtida analys av *cash-flow*, utgifter och använda mantimmar respektive maskintimmar (Koontz & Wehrich, 1988). Kerzner (1992) menar att budget och kvalitet går hand i hand. Det är därför nödvändigt att beakta den önskade

kvalitetsnivån vid budgetering. Budgeten är nödvändig för kontroll av projektet, men kan inte utgöra en grund om den inte speglar projektplanen (Koontz & Wehrich, 1988). Budgeten avser att presentera projektet utifrån dess utgifter i relation till olika tidsintervaller och visa vecko-, månads- och årlig nivå samt klargöra utgifter för mantimmar, underhåll, tillverkning och konstruktion (Cleland, 1999). Budgeteringen kan variera inom olika organisationer. I vissa projekt är budgeten utarbetad separat utifrån olika organisationsdelar som är aktiva inom projektet (Koontz & Wehrich, 1988). Budgeten fokuserar oftast på två dimensioner; dels speglar den de aktiviteter som är strukturerade i ett WBS, dels tidsaspekten i relation till WBS (Antvik & Sjöholm, 2007).

2.4.3 Schema

Att utarbeta ett schema innebär analysering av sekvenserna aktiviteter, varaktighet och resurskrav (PMBOK, 2008). I stora projekt är ett schema oftast inte tillräckligt (Antvik & Sjöholm, 2007; Kerzner, 1992). I dessa fall utformas delschema vilka belyser enstaka aktiviteter (Antvik & Sjöholm, 2007). Cleland (1999, s. 57) anser att "the project master schedule", tillsammans med "the supporting schedules", är den främsta outputen för projektplanering. Syftet med att schemalägga aktiviteter är att generera en plan över när aktiviteter ska startas och avslutas. Utveckling av scheman är oftast en iterativ process. Schemat behöver ständigt uppdateras allteftersom projektet utvecklas och nya faktorer påverkar (PMBOK, 2008). Vid utformning av projektets schema måste, enligt Cleland (1999), hänsyn tas till definition av projektets mål och delmål, utveckling av aktiviteter, estimering av tid och kostnad och bekräftelse att schemat överensstämmer med organisationens resurser.

2.5 Utgångspunkter för studien

Utifrån beskrivningen i kapitel 2 av vad projektplanering innebär sammanställs här de aspekter som är viktiga att analysera. I tabell 2.3 presenteras de komponenter (planering, mål, budget, schema) och dess faktorer som enligt författarna i litteraturgenomgången bör hanteras för att genomföra en lyckad projektplanering.

Tabell 2.3 – Sammanställning av komponenter och faktorer som enligt samtliga författare i litteraturgenomgången utgör projektplanering

Källor	Planering (<i>planning</i>)	Mål (<i>goal/ objective</i>)	Budget (<i>budget</i>)	Schema (<i>schedule</i>)
Koontz & Wehrich (1988) Kerzner (1992)	Resurser (<i>resources</i>)	Resurser (<i>resources</i>)	Kostnad (<i>cost</i>)	Tid (<i>time</i>)
Antvik & Sjöholm (2007) PMBOK (2008)	Aktiviteter (<i>activity</i>)	Aktiviteter (<i>activity</i>)	Aktiviteter (<i>activity</i>)	Datum (<i>date</i>)
Cleland (1999)	Dokumentation (<i>documentation</i>) WBS (<i>wbs</i>)	WBS (<i>wbs</i>)	WBS (<i>wbs</i>)	Resurser (<i>resources</i>) Aktiviteter (<i>activity</i>) WBS (<i>wbs</i>)

Som framgår av tabellen är vissa faktorer överlappande och därför avgörande inom flera områden. Detta kommer inte att påverka det fortsatta utförandet av studien då dessa kategorier avses mätas likvärdigt oavsett upprepningsgrad. Aspekterna i tabell 2.3 utgör grunden för studien och det är dessa som avses undersökas för att besvara forskningsfrågan.

Den del av *project management*-teori som berör organisationen och projektet kommer att syfta till OS-*communities* och utvecklingen av mjukvara. OS-*communities* kommer att benämnas som organisationer och utvecklingarna av mjukvara kommer att benämnas som projekt.

2.6 Projektplanering och Open Source

Aspekterna i kapitel 2.5 ligger till grund för vilka studier som valts att analyseras för att presentera en bild av tidigare forskning relaterad till forskningsfrågan. Studierna angriper projektplanering och/eller *project management* inom OS i någon form.

Aberdour (2007) avser att undersöka kvalitetsaspekten av OS-mjukvara. I hans studie behandlas *project management* i relation till kvalitet. Aberdour anser att OS-utveckling bevarar vissa grundläggande aktiviteter applicerade på traditionell mjukvaruutveckling som *project management* men menar att dessa hanteras annorlunda.

Yutaka Yamauchi and his colleagues argued that this results in a culture of spontaneous work that is coordinated afterward—a major shift from traditional development where coordination and planning come first. They summarized the OSS process as action, experimentation, and innovation followed by coordination, peer review, improvement, and stability. [...] However, other studies have found coordination occurring before action. In the Apache case study, when the core team found alternative solutions to a problem, they invited feedback from the developers' mailing list, and the core team then decided on a solution before it was developed. (Aberdour, 2007, s. 61)

Aberdour hävdar att konceptet ”deadline” inte innebär samma behov av kontroll som inom traditionell utveckling och därför kan vara en anledning till varför organiseringen inom OS-*communities* angriper *project management* olika.

Jaccheri & Østerlie (2007) diskuterar OS-utveckling som ett tema för att bidra till kunskap för studenter. Vid urval av projekt fanns det kriterier att uppfylla. Ett av kriterierna var ”Formal techniques (project planning, etc.) are used in the OSS development” (Jaccheri & Østerlie, 2007, s. 3). I studien hittades projekt som uppfyllde detta krav.

Nakagawa et al. (2008) har, genom att i olika fallstudier observerat mjukvaruarkitektur inom OS och hur det påverkar mjukvarukvalitet, insett att planering inte är lika fokuserad inom OS-utveckling som inom traditionell projektledning. “[...] planning, documentation and modeling is not the main focus of these projects” (Nakagawa et al., 2008, s. 1238).

Genom att undersöka hur OS förhåller sig till kunskap och ”knowledge management”, konstaterar Ulhøi (2004):

In open source innovation, innovators freely choose the assignments they want to work on, typically based on personal preferences such as for example needs, skills and intellectual challenge, which means their motivation is at a maximum. In the closed source innovation model, management is normally involved in detailed planning and monitoring, and has sole power regarding matters of HRM and financial allocations. (Ulhøi, 2004, s. 1110)

Ulhøi (2004) menar att det existerar en avsevärd skillnad mellan mjukvaruutveckling inom OS och traditionell utveckling sett till hur projekt bemannas och till att grunden i en OS-utveckling bygger på erfarenhet och intresse hos de involverade.

Hippel (2003) avser att studera OS ur ett organisatoriskt perspektiv. Genom att analysera hur OS-utveckling förhåller sig till organisatoriska begrepp belystes hur OS-utveckling förhåller sig till *project management*. Hippel menar att utveckling är mål och behov tillgodosedda genom kommunikation inom OS-*communities* och att organisering sker utifrån vad som behöver utvecklas.

“Users make their behavior logically plausible to the community, base decisions on technical performance criteria, and have a bias for action/programming rather than project planning” (Hippel, 2003, s. 220).

Madey et al. (2002) avser att undersöka hur OS-*communities* fungerar och är strukturerade. Madey et al. (2002) menar att organiseringen inom OS-*communities* är skilda från traditionella utföranden och strukturer inom organisationer.

The Open Source Software movement is a prototypical example of a decentralized self-organizing process. There is no central control or central planning. It challenges conventional economic assumptions, it turns conventional software engineering and project management principles inside out [...] (Madey et al., 2002, s. 1807)

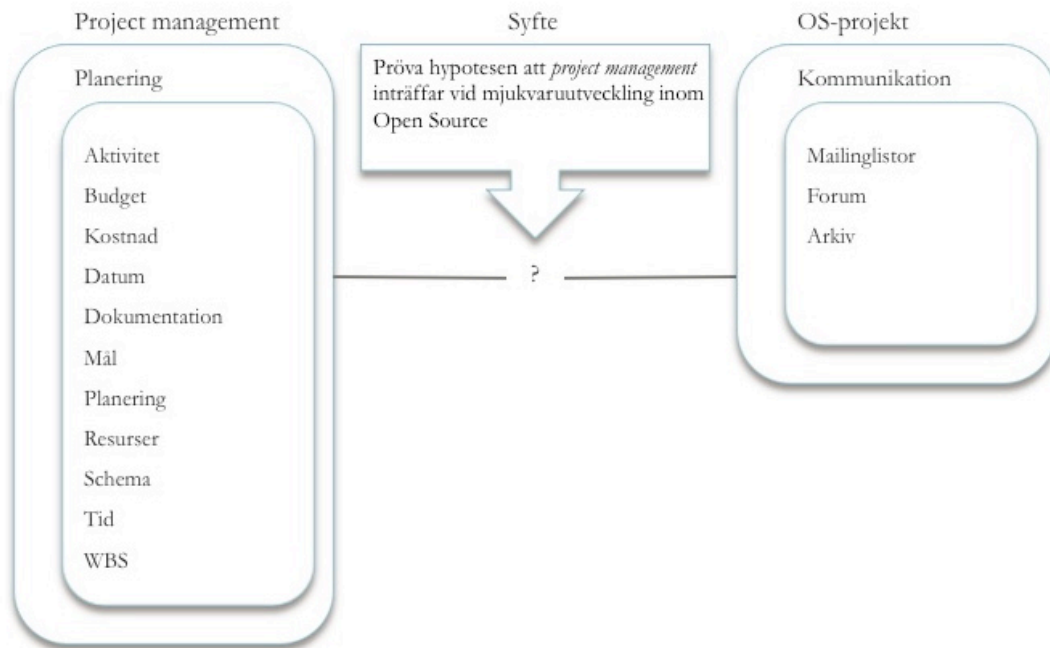
“The programs, some of which are extremely large and complex, are written without the benefit of traditional project management [...]” (Madey et al., 2002, s. 1206).

Sammanfattningsvis är antagandet att *project management*, utifrån forskningen syn, inte förhåller sig som ett grundläggande verktyg för att uppnå mål, strukturera aktiviteter och samordna människors engagemang, eller ens är nödvändigt inom OS. Det framgår också att det inom forskningen generellt inte finns en bestämd uppfattning om huruvida detta är sant eller är en sanning som är generell för alla OS-utvecklingar.

2.7 Undersökningsmodell

Undersökningsmodellen i figur 2.1 ligger till grund för studien och har utarbetats för att besvara forskningsfrågan. Modellen lyfter fram studiens syfte som är att pröva hypotesen att *project management* inträffar inom OS-utveckling. Undersökningsmodellen presenterar studieobjektet och de aspekter som ska analyseras; i detta fall dokumenterad kommunikationen i form av mailinglistor, arkiv och forum samt de aspekter inom *project management* som ska undersökas (aktivitet, budget, kostnad, datum, dokumentation, mål, planering, resurser, schema, tid, wbs) och som är förankrade i tidigare teorigenomgång.

Figur 2.1 – Undersökningsmodell



3. METOD

Syftet med metodavsnittet är enligt Backman (1998) att underlätta replikation och evaluering av studien. Evalueringen syftar till att ge utomstående möjlighet att värdera och bedöma det empiriska förfarandet (Backman, 1998). I metodavsnittet presenterar jag tillvägagångssätt, urval av studieobjekt och urval av empiri.

3.1 Litteratursökning och kritisk granskning

Enligt Backman (1998) ska artiklar, böcker och rapporter grunda den teoretiska sammanställningen. I detta avsnitt presenterar jag hur litteratursökningen och granskningen vid utförandet av den teoretiska sammanställningen gick till. Litteraturundersökningen är utförd enligt rekommendationer som Wallén (1996) förespråkar. Den systematiska litteraturundersökningen utgick från frågeställningen. Under studiens gång har frågeställningen utvecklats vilket har resulterat i att litteratursökningen förändrats. Här är det viktigt att påpeka att det är urvalet som ändrats, inte metoden. Litteraturstudien utarbetades utifrån följande frågor:

Vad är *project management*?

Hur genomförs planering?

Hur relateras *project management* till OS?

Eftersom jag utvecklade frågeställningen ytterligare utökades urvalet av användbara sökord. Detta ledde till att sökresultaten förfinades. Litteratursökningen har främst skett i databaser och bibliotek. Exempel på sökordskombinationer presenteras i bilaga 1, tabell 1. Abstrakt har legat till grund för avgöranden huruvida en artikel, studie eller bok varit relevant för studien. För att granska litteraturens trovärdighet har författarnas yrke och sysselsättning spelat roll. Den teoretiska genomgången har utgjorts av aktiva inom den vetenskapliga akademien eller av aktiva som tillämpat teorin i arbetslivet. För att stärka tillförlitligheten har teoretiska utsagor, då detta varit möjligt, bekräftats från flera källor. För att få tillgång till en större litteraturbas valde jag engelsk litteratur. Jag valde att utgå från en teoretisk bakgrund som definierar aspekter och begrepp av vad som skulle undersökas som Backman (1998) förespråkar. Detta angreppssätt syftar till att utforma studien med relevant information och ge stöd för val av metod.

Den litteratur som jag använde för att undersöka forskningens syn på frågeställningen arbetades fram först efter att den teoretiska sammanställningen av *project management* var etablerad. Detta för att styrka relevansen av forskningens syn. Litteraturen av forskningens syn granskades ur samma kritiska synvinkel som övrig litteratur. I bilaga 1, tabell 1, presenteras söktermerna som användes vid sökningen.

Efter att ha utarbetat en teoretisk referensram var det möjligt att undersöka forskningens syn på forskningsfrågan. Genom att förankra sökord ur den teoretiska sammanställningen (kap. 2.6) hittade jag nedan presenterade utsagor.

3.2 Studieobjekt

Jag avsåg att analysera dokumenterad kommunikation inom tre OS-projekt för att undersöka betydelsen av de aktuella aspekterna (kap. 2.6). Projekten hittades i de tre OS-

communities Sourcefourge, Github och Google Code (kap 1.5.3). Tanken var att studera objekt som var aktiva, d v s att medlemmar var engagerade och att detta skulle speglas i aktiviteten inom OS-*communitiet*.

Vid urval av studieobjekt fanns kriterier att gå efter såsom omfattning (målgrupp), etablering (popularitet) och komplexitet (svårighetsgrad). Målgruppen avsåg att avgöra om studieobjekten var avsedda för barn, vuxna, män eller kvinnor. Populariteten avgjorde storleken på studieobjektens användarbas. Svårighetsgraden avsåg att avgöra studieobjektens komplexitet sett till mjukvaruutvecklingen och antalet aktiva utvecklare. För att få en spridning av resultatet ville jag analysera objekt som förhöll sig olika till dessa kriterier, d v s väletablerade projekt i kontrast till mindre väletablerade projekt. I dokumentationen på studieobjektens officiella hemsida har jag utvärderat om objekten är av relevans för studien. Efter sökning och utvärdering hittade jag följande objekt:

Android är en plattform/operativsystem som främst används inom mobiltelefoni. Android utvecklas inom OS-*communities* med Google som främsta sponsor. Plattformen är vid studiens genomförande aktuell för en bred marknad av mobiltelefoner (HTC, Samsung, Nokia m fl). Med tanke på produktens användningsområde anses Android vara ett väletablerat studieobjekt. Android hittades i OS-*communitiet* Google Code och är det största projektet i studien sett till antalet aktiva utvecklare.

WebKit är en “web browser engine” vilket innebär att WebKit används för att strukturera och presentera information i en webbläsare. WebKit är ett OS-projekt som i grunden sponsras av Oracle. Alla stora webbläsare, som t ex Explorer, Chrome, Safari och Firefox, stödjer WebKit vid studiens genomförande. Med avsikt att standardisera och optimera en webbläsare (som utgör en stor del av datoranvändandet) anses WebKit vara ett omfattande studieobjekt. WebKit hittades i OS-*communitiet* Sourceforge.

0AD är ett strategispel och utvecklat inom OS-*communities*. 0AD är utvecklat för tre olika plattformar; Windows, iOS och Linux. 0AD har en öppen utveckling med hänsyn till funktionalitet och innehåll, men en stängd utveckling ur design-perspektiv. 0AD är fullt utvecklat inom OS-*communities* och ingen extern aktör påverkar utvecklingen. 0AD är i Alfa-Beta stadiet, vilket innebär att en officiell publicering inte inträffat. 0AD hittades i OS-*communitiet* Github och är det minsta projektet i studien sett till antalet aktiva utvecklare.

I tabell 3.1 ges en övergripande sammanställning av kriterierna för studieobjekten.

Tabell 3.1 – Sammanställning av studieobjekten

Android	WebKit	0AD
<ul style="list-style-type: none"> - Innehar flest antal aktiva utvecklare - Används flitigt som operativsystem i mobiltelefoner - Är allmänt känt - Google är involverad i utvecklingen 	<ul style="list-style-type: none"> - Innehar aktiva utvecklare - Används flitigt i webbläsare - Är inte allmänt känt - Oracle är involverad i utvecklingen 	<ul style="list-style-type: none"> - Innehar minst aktiva utvecklare - Spelas av ett mindre antal användare - Är inte allmänt känt - Ingen extern aktör är involverad i utvecklingen

3.3 Empiri

Dokumentationen som legat till grund för analysen har hittats i olika OS-communities (Sourcefourge, Github, Google Code) på studieobjektens officiella hemsidor eller i mailinglistor, arkiv och forum. Då dokumenteringen av kommunikationen inom OS sker i form av forum och mailinglistor (kap. 1.5.3) har jag fritt sökt efter forum och mailinglistor på Internet i relation till studieobjekten. Efter sökning och utvärdering hittade jag följande källor för de tre studieobjekten.

Android, Google har utarbetat ett system för mailinglistor (fig. 3.1), Google Groups. Android Contributors-mailinglista är den officiella kommunikationskanalen för allmänheten vid utveckling inom Android. Androids mailinglista låg till grund för analysen.

Figur 3.1 – Utdrag ur Androids mailinglista

Topic	Posts	Views	Author	Date
Adding a new HOST_ARCH (3)	3	21	Steve Jones +1	Oct 30
bugreport analyzer tool (2)	2	18	Pal Szasz +1	Oct 19
Extending AbsListView (1)	1	9	Chris Sarbora	Oct 17
Spanish LatinIME	4	20	Juan Rodriguez +1	Oct 12
Gerrit trick: commenting on your own changes (2)	2	64	Jean-Baptiste Queru +1	Oct 11
More questions on CCLA, Item 7 (2)	2	10	Ryan V. Bissell +1	Oct 11
repo sync failing with "Idevice/samsung/toroplus: not a Gerrit project" (3)	3	9	Duane Sand +1	Oct 10
Status of incremental dexing (10)	10	187	Charlie Groves +5	Oct 3
how create a new project in AOSP (7)	7	107	jim.zhang +2	Sep 26
Should we spend more time on reviewing the fixes before shipped a release (2)	2	29	Roger Chen +1	Sep 26
Adding Group MMS functionality to AOSP (18)	18	726	Robert Nye +6	Sep 13
Contribution for gingerbread CTS in libcore/luni repository (4)	4	26	Fabien Duvoux +1	Sep 13
Question on CCLA, Item 7 (4)	4	24	Ryan V. Bissell +1	Sep 11

WebKit har utarbetat ett arkivsystem där bl a utvecklingsdiskussioner arkiveras (fig. 3.2). Jag analyserade nio månaders dokumenterad kommunikation ur detta arkiv.

Figur 3.2 – Utdrag ur WebKits arkiv

```
From oszi at inf.u-szeged.hu Mon Jan 2 02:34:57 2012
From: oszi at inf.u-szeged.hu (Osztrogonac Csaba)
Date: Mon, 02 Jan 2012 11:34:57 +0100
Subject: [webkit-dev] Tools/Scripts/build-webkit --gtk
--debug --makeargs="-j1" taking up to 80% of memory
In-Reply-To: <5CC6D293-DA7A-4C20-BED8-8CDF10B2D54@webkit.org>
References: <CAKh9THsXgMBuTfATGw4EuKcy+B-ef34uBACMuE0B-5_YX6D6-A@mail.gmail.com>
<5CC6D293-DA7A-4C20-BED8-8CDF10B2D54@webkit.org>
Message-ID: <4F018851.9050308@inf.u-szeged.hu>

Hi All,

32 bit debug build is impossible long time ago, because linker runs
out of 4Gb address space. Maybe cross compiling on a 64 bit machine
can solve this problem. Is there anyone interested in 32 bit build?

br,
Ossy

Zeno Albisser ?rta:
> Hi Sachin,
>
> We had the same/a similar problem with QtWebKit.
> Ranlib ran out of memory on a 32bit system when building debug, because the library got bigger than 4GB.
> I dont think there is a proper solution for this problem. The only thing that would help would be to reduce the size of WebCore. For now we just disabl
debug builds on 32bit systems. I heard some people managed to build debug by disabling SVG. - never tried it myself though.
>
> Best Regards
>
> -- Zeno
>
> On Dec 27, 2011, at 6:45 AM, sachin nikam <sknikam at gmail.com> wrote:
>
>> I synced up the latest webkit code base and am trying to build webkit
>> gtk on ubuntu 11.10 with 4gb of RAM and 4 CPUs
>>
>> I tried with --makeargs="-j2" but still got "ld process terminated
>> signal[9]" error which indicates that it ran out of memory.
>> I suspect i will still the get the same error with --makeargs="-j1".
>> Is there any other flag where we can restrict the memory usage?
>> Regards
>> Sachin

From philn at igalia.com Mon Jan 2 02:40:49 2012
From: philn at igalia.com (Philippe Normand)
```

OAD har ett centralt forum för diskussioner rörande OAD där bl a utvecklingen diskuteras (fig. 3.3). OAD's forum låg till grund för analysen av studieobjektet.





Figur 3.3 – Utdrag ur OADs forum

Wildfire Games Community Forums → O.A.D. → Game Development & Technical Discussion

Game Development & Technical Discussion






[Are you a programmer, artist, or animator? Are you interested in helping with O.A.D. development?](#)

Subforums

 Applications and Contributions Post here to put in your application to join Wildfire Games and help finish O.A.D. If you're just want to donate works as more of a freelance OS contributor, you can start a thread here too!	53 topics 919 replies	 Animator application - Marc... By Lok1950 Today, 05:26 PM
 Bug reports Post bug reports here. Errors, crashes, and other bugs.	343 topics 1,875 replies	 Crash By Josh Today, 12:39 AM

Page 1 of 21 [NEXT](#) [»](#) Mark this forum as read

Recently Updated [Start Date](#) [Most Replies](#) [Most Viewed](#) [Custom](#)

PINNED Technologies Started by Mythos_Ruler, 14 Oct 2012 1 2	HOT 35 replies 1,165 views	 Mythos_Ruler Yesterday, 04:37 PM
PINNED Progress reports on funded work Started by Ykkrosh, 23 Oct 2011 1 2 3 13 →	HOT 250 replies 24,812 views	 Lion.Kanzen 28 Oct 2012
PINNED Audio Design 5 - Voice List Started by Acumen, 09 Dec 2003 1 2 3	HOT 48 replies 940 views	 Lion.Kanzen 20 Oct 2012
PINNED Alternative Combat System Started by Mythos_Ruler, 08 Mar 2011 1 2 3 4	HOT 63 replies 6,383 views	 Lion.Kanzen 14 Oct 2012
PINNED Stances Started by Mythos_Ruler, 28 Oct 2010 1 2 3 4	HOT 65 replies 5,830 views	 Pedro Falcão 25 Sep 2012

Med dessa källor till empiri hade studien en bred bas för analys (se bilaga 1, tabell 2).

3.4 Riktad innehållsanalys

En av de kvalitativa undersökningsmodellerna är riktad innehållsanalys (Statens beredning för medicinsk utvärdering [SBU], 2012). En innehållsanalys syftar till att analysera dokument och texter (Bryman & Bell, 2005) samt skriftlig, verbal eller visuell kommunikation (SBU). Riktad innehållsanalys avser att koda utifrån teori, i mitt fall *project management*-teori. Eftersom all kommunikation inom OS-utveckling sker i skrift och teorin förespråkar en detaljerad dokumentation ansåg jag att en textorienterad analysmetod var användbar. Jag antog att aspekterna som jag valde att undersöka (tabell 2.3) framgick i text och var väl dokumenterade. Därför valde jag riktad innehållsanalys som möjlig, och tillräcklig metod, för att genomföra studien. *Project management*-teori har använts som ett orienterande filter i studien och analyserats objektivt i relation till studieobjekten.

Genom att använda en systematisk och replikerbar metod vill man kategorisera innehåll utifrån tidigare definierade kategorier (Bryman & Bell, 2005). Vidare menar Bryman & Bell (2005) att det vid innehållsanalytisk forskning är viktigt att besluta om analysenheten ska utgöras av ord, fraser eller satser. Jag avsåg att analysera utifrån bärande ord (tabell 2.3) vilka utgjorde analysenheten, samt utföra innehållsanalysen manuellt. Inget datoriserat system granskade texterna eller utarbetade ett resultat. Innehållet kategoriserades genom att jag sökte efter ord som ansågs vara av teoretisk relevans. Kategorierna som utgjorde sorteringen av relevant material baserades på aspekterna planering, mål, budget och schema, och presenterades i den teoretiska sammanställningen.

För att resultatet skulle kunna ge en generell bild av de tre projekten som analyserades sattes ramar för empirin upp. Kravet för empirin var att analysenheten skulle vara relevant inom två studieobjekt och summan av antalet träffar minst fem. Vid inskaffning av empiri inleddes en systematisk sökning av analysenheterna där relevansen av varje analysenhet noterades (se bilaga 1, tabell 3). Resultatet av denna filtrering var att analysenheten förfinades genom att budget och WBS försvann som analysenheter. Sedan analyserades varje studieobjekt i relation till analysenheten. För att sammanställa varje studieobjekts syn på analysenheten utarbetades en tabell (tabell 4.1) där studieobjekten presenterades i relation till analysenheten. Den generella bilden av analysenheterna baseras på tabellen där analysenhetens tolkning överensstämmer i minst två studieobjekt.

3.5 Validitet och reabilitet

Validitet och reabilitet är två viktiga begrepp vid evaluering av kvalitativ forskning (Bryman & Bell, 2005). Reabilitet speglar forskningsresultatets tillförlitlighet vilket är viktigt eftersom samma resultat ska uppnås vid upprepning av studien. Genom att i detalj beskriva den litterära sökningen, motiveringen för metodvalet, genomförandet av analysen, motiveringen och innehållet av urvalet för studieobjekten samt kommentera empirin avser jag att styrka reabilitet och validitet för studien. För att uppnå en hög tillförlitlighet var min avsikt att ställa frågor med teoretisk utgångspunkt och systematiskt undersöka publik information samt i detalj beskriva utförandet av den kvalitativa undersökningen. Empirin är offentlig och hittad på Internet och kan därför kontrolleras. Därmed är utgångspunkter och tillvägagångssätt för studien pålitliga (upprepningsbara) och eventuella feltolkningar kan avgöras.

Validitet ämnar belysa huruvida resultatet är förankrat i materialet från den empiriska undersökningen (Bryman och Bell, 2005) d v s om studien undersöker det den avser undersöka utifrån frågeställningen. Frågeställningen och den teoretiska sammanställningen har legat till grund för beslut vid utformningen av studien med avseende på tidigare nämnda aspekter för att styrka validitet. Med en kvalitativt inriktad innehållsanalys existerar möjligheten att vid upprepning av studien konstatera ett annat resultat. För att stärka validiteten av studien har jag valt att tolka empirin ur ett objektiva perspektiv förankrat i litteraturgenomgången. För att stärka reabilitet och validitet är studien utarbetad utifrån den teoretiska sammanställningen.

4. ANALYS AV STUDIEOBJEKTEN

I detta kapitel redovisas analysenheten ur det empiriska materialet baserat på de tre studieobjekten Android, WebKit och OAD. I tabellerna 1-10 i bilaga 2 presenteras den individuella tolkningen av studieobjekten i relation till analysenheten. Nedan följer en sammanfattning av varje analysenhet. Reflektioner av studieobjekten som inte är direkt kopplade till forskningsfrågan diskuteras också. I denna studie används både “objective” och “goal” för att belysa projektens mål.

4.1 Empiriska fynd

Inom Android och OAD diskuteras aktivitet (*activity*) när man syftar till en mjukvaras aktiviteter eller aktivitet i forum. Mjukvaror kan ha olika funktioner och utträta olika aktiviteter. Aktivitet i forum syftar till utvecklarnas engagemang i skriftliga diskussioner. Aktivitet kan också syfta till aktiviteter som är relevanta för utvecklarna i projektet, d v s uppgifter som kan kopplas till utförandet av projektet. Inom OAD relaterar aktivitet även till CPU-aktivitet och aktiviteter i spelet. WebKit hade ingen träff på *activity*.

Budget (*budget*) har inte analyserats då tolkning av analysenheten inte kunde härledas till studien.

Inom studieobjekten sammanfaller kostnad (*cost*) med “performance cost” vid analys. “Cost” och “expensive” (expensive operation) i OS-utveckling kan, förutom att referera till en kostnad som relaterar till finansiella faktorer, också relatera till hur mycket mjukvaran belastar datoriserade system. Det är oftast i samband med Google’s resurser (anställda) som kostnaden inom Android avser att belysa en ekonomisk faktor eller för att t ex belysa belastningen på batterier. Inom WebKit är kostnaden en ekonomisk faktor i samband med underhåll av servrar eller utomstående plattformar och system samt inläring av funktioner (inlärningskurva). Inom OAD har kostnad en betydelse i spelet (virtuell kostnad) för objekt som kan köpas.

Inom studieobjekten relateras ett datum (*date*) oftast till “release-date” vilket syftar till datum då en uppdatering ska offentliggöras. Datum förekommer ofta i samband med “up-to-date”, “update” eller ”out of date” för att definiera om något är relevant ur ett tidsperspektiv (updated/up-to-date). Inom Android och WebKit används datum i samband med “update” (updatera mjukvara) och diskuteras när en uppdatering är nödvändig, bristfällig eller under konstruktion. Datum används också när medlemmar i utvecklingen vill uppdateras inom ett specifikt område. Inom OAD och Android är datum en aspekt i mjukvaran.

Inom studieobjekten är dokumentation (*documentation*) utspridd i och över olika *communities* samt över nätet och inte indexerade utifrån standard. Utan bestämmelser för dokumentation kan det uppstå svårigheter att hitta relevant information. Dokumentation utifrån engagerade medlemmars initiativ och utvecklarna är positiva till bidrag. Det finns inget krav på att dokumentera efter stadgar vilket bidrar till att dokumentationen är bristfällig eller frånvarande. Inom Android faller all dokumentation under OS-licensen. Inom OAD är intresset för autogenererad dokumentation (av kod) större än manuell dokumentation. Dock finns det inte resurser för att stödja den mängd dokumentation som är nödvändig för projektets helhet. Inom WebKit uppstår problem vid hänvisning till dokumentation som inte är uppdaterad. Inom OAD och WebKit diskuteras hur och när dokumentation bör hanteras.

Inom studieobjekten diskuteras målen (*goal/objective*) inom gruppen för att klargöra om de är relevanta att utföra. Mål kan både syfta till projektets och mjukvarans mål samt funktionen som mjukvaran ska uppfylla. Målen är individuella och definierade av varje medlem. Mål är oftast personliga, vilket innebär att de inte involverar alla i projektet; de påverkar inte övrigas arbete eller egna uppsatta mål. Inom WebKit framgår att mål i viss utsträckning rubriceras som long-term/short-term. Inom OAD har mål betydelse vid diskussion om ett objekts förflyttning inom spelet; mål avser den position som spelaren har definierat för ett objekt att förflytta sig till. Inom Android benämns dokumentering som ett mål. Inom Android och OAD finns mål för olika forum och mailinglistor. Diskussioner i forum och mailinglistor bör fokusera på att lösa eller uppnå en fördefinierad uppgift.

Inom Android kan termen "objective" översättas till mål och aktiviteter som utträttas eller diskuteras. Medlemmarna har önskemål om hur målen ska uppnås i samarbete med Google. Diskussionerna handlar oftast om stöd för olika funktioner och om hur Android är konstruerat för att möjliggöra en interaktion mellan flera system. Mål och aktiviteter avser inte att utveckla Androids kärna, utan är kompletterande system som antingen utnyttjar, eller tillför, funktion till Android. "Objective" är på individuell basis. Inom OAD förhåller sig termen "objective" i de flesta fall till "objective" inom spelet - det finns mål för spelaren att uppnå. "Objective" är även en term som används för att beskriva mål för utvecklingen av OAD. Inom OAD sammanfaller "objective" med "objective observation". Inom WebKit och OAD sammanfaller "objective" med namnen på programmeringsspråk "Objective C" och "Objective C++".

Inom studieobjekten sker planering (*planning*) utifrån en utvecklarens perspektiv; en plan innefattar en eller flera utvecklare som individuellt bestämmer sin planering. Planering av framtida aktiviteter, liksom planering av datum, utarbetas utifrån diskussioner i forum och mailinglistor. Alla medlemmar har rätt att uttala sig om projektets olika planeringar. Planering avser att beskriva och informera om utveckling. Inom Android existerar planering i viss mån och är utarbetad efter behov. Planering involverar människor som vill engagera sig i en aktivitet. Ibland relateras planering till planer som long-term/short-term. Planer diskuteras för att införskaffa support till andra system och användare. Inom WebKit existerar inte en central plats för planering vilket medför att en central plan saknas. Inom OAD är planering en aktivitet inom spelet.

Inom studieobjekten relateras inte resurser (*resources*) till studieobjektens resurser i OS-utvecklingssammanhang. Resurser speglar inte projektens resurser när temat diskuteras i grupp utan beskriver mjukvarans resurser. Mjukvaran har resurser i form av andra mjukvaror. Diskussionerna i forum och mailinglistor handlar oftast om hur resurser ska användas eller hittas av andra mjukvaror. Inom Android kan resurser existera i form av språkstöd för applikationer. Det finns två aspekter; dels resurser i form av mjukvarutyp, dels resurser i form av människor. Oftast är resurser i form av människor en anspelning på Googles resurser. Inom OAD används resurser främst för att benämna resurser i spelet (food, wood, stone etc) och diskussioner om hur dessa ska hanteras i spelet.

Inom studieobjekten framgår att schema (*schedule*) avser att bestämma datum framför allt för "release-date". Schema har också en teknisk aspekt. Oftast avser schema att belysa en teknisk funktion, d v s ett schema behövs för att utföra något frekvent i mjukvaran. Schema används vid enstaka fall för att koordinera arbetet. Inom Android och OAD avser schema att schemalägga aktiviteter i en mjukvara (schemalagda aktiviteter). Inom samtliga studieobjekt avser scheman också att belysa tidsaspekten av

uppgifter men är i detta sammanhang kopplat till “release-schedule” (schema för när en uppdatering ska släppas). Inom Android framgår i enstaka fall att en aktivitet har utförts med ett bakomliggande schema. Inom OAD är schema individuellt. Schemat kartlägger och strukturerar enstaka uppgifter som utvecklarna avgör. Schemaläggning sker på personlig nivå.

Inom studieobjekten används tid (*time*) för att beskriva hur lång tid det tar att utveckla funktioner inom ett projekt. Tid är en avgörande faktor vid strukturering av arbetet mellan medlemmarna i utvecklingen. Tid diskuteras ofta och handlar t ex om tidsåtgång, om en uppgift är tidskrävande eller om en lösning är lönsam ur ett tidsperspektiv. Inom Android kan tid också kopplas till diskussioner gällande servers tillstånd och hur länge dessa har varit offline. Tid används inom Android också för att beskriva hur lång tid det tar att exekvera specifika funktioner. Eftersom Android är ett komplext system och många aktiviteter är igång samtidigt (parallella aktiviteter) avser tid även att belysa aktiviteter inom systemet. Inom OAD är tid i vissa sammanhang kopplat till “runtime”; tiden då mjukvaran körs. Tid är även kopplad till vad som uträttas. Tiden avgör hur många medlemmar som kan engagera sig i utvecklingen. Tid är en aspekt i spelet. Inom WebKit hittar man ”long-time users”.

WBS (*wbs*) har inte analyserats då tolkning av analysenheten inte kunde härledas till studien.

Tabell 4.1 – En sammanställning av analysenheten i relation till studieobjekten

Analysenheter	Android	WebKit	OAD
Activity	<ul style="list-style-type: none"> - Aktivitet inom mjukvara - Uppgifter som utformas inom projektet - Aktivitet i mailinglistan 	<i>Inga träffar</i>	<ul style="list-style-type: none"> - Aktivitet inom mjukvara - Uppgifter som utformas inom projektet - CPU-aktivitet - Aktivitet i forumet - Aspekt i spelet
Budget	<i>Inga träffar</i>	<i>Inga träffar</i>	<i>Inga träffar</i>
Cost	<ul style="list-style-type: none"> - Prestandaperspektiv - Google’s resurser - Kostnad av licenser - Kostnad vid utveckling 	<ul style="list-style-type: none"> - Prestandaperspektiv - Underhåll av servrar - Kostnad av utomstående system och plattformar - Vid inläring 	<ul style="list-style-type: none"> - Prestandaperspektiv - Aspekt i spelet
Date	<ul style="list-style-type: none"> - Release-date - Diskussion kring uppdatering (update) 	<ul style="list-style-type: none"> - Release-date - Diskussion kring uppdatering (update) - Relevans (up-to-date) 	<ul style="list-style-type: none"> - Release-date - Relevans (up-to-date)
Documentation	<ul style="list-style-type: none"> - Eget initiative - Inga krav - Utspridd - Bristfällig/frånvarande - OS-licens 	<ul style="list-style-type: none"> - Eget initiativ - Inga krav - Utspridd - Bristfällig/frånvarande 	<ul style="list-style-type: none"> - Eget initiativ - Inga krav - Utspridd - Bristfällig/frånvarande - Automatisk dokumentering

Goal	<ul style="list-style-type: none"> - Belysa mål och uppgifter - Mjukvarans mål - Personliga 	<ul style="list-style-type: none"> - Belysa mål och uppgifter - Mjukvarans mål - Personliga - Long-term/short-term 	<ul style="list-style-type: none"> - Belysa mål och uppgifter - Mjukvarans mål - Personliga - Position av objekt
Objective	<ul style="list-style-type: none"> - Mål och aktiviteter - Önskemål om samarbete med Google för att uppnå mål - Avsedda för kompletterande system - Personliga 	<i>Inga träffar</i>	<ul style="list-style-type: none"> - Mål och aktiviteter - Aspekt inom spelet - Personliga - <i>Objective C++</i> - Objektiv observation
Planning	<ul style="list-style-type: none"> - Frivilligt/eget intresse - Personliga - Long-term/short-term - Diskuteras i forum och mailinglistor - Medlemmar har rätt att uttala sig om planer - Utarbetad efter behov - Planera support 	<ul style="list-style-type: none"> - Beskriva och informera om aktiviteter - Frivilligt/eget intresse - Personliga - Diskuteras i forum och mailinglistor - Medlemmar har rätt att uttala sig om planer - Ingen central plan 	<ul style="list-style-type: none"> - Beskriva och informera om aktiviteter - Frivilligt/eget intresse - Personliga - Diskuteras i forum och mailinglistor - Medlemmar har rätt att uttala sig om planer - Aspekt i spelet
Resources	<ul style="list-style-type: none"> - I form av andra mjukvaror - Mänskliga resurser (Google's) - Språkstöd - Fokus på att använda och hitta resurser 	<ul style="list-style-type: none"> - I form av andra mjukvaror - Fokus på att använda och hitta resurser 	<ul style="list-style-type: none"> - I form av andra mjukvaror - Aspekt i spelet - Fokus på hantering av resurser i spelet
Schedule	<ul style="list-style-type: none"> - Schemalägga aktiviteter i en mjukvara - Release-schedule - Aktivitet kan relateras till scheman - Inte genomgående 	<ul style="list-style-type: none"> - Schemalägga aktiviteter i en mjukvara - Release-schedule - Aktivitet kan relateras till scheman 	<ul style="list-style-type: none"> - Release-schedule - Aktivitet kan relateras till scheman - Personligt
Time	<ul style="list-style-type: none"> - Lönsam lösning ur tidsperspektiv - Servers tillstånd - Belysa aktiviteter inom systemet som körs parallellt - Tid sett till prestanda 	<ul style="list-style-type: none"> - Lönsam lösning ur tidsperspektiv - Medlemmars engagemang ur ett tidsperspektiv 	<ul style="list-style-type: none"> - Lönsam lösning ur tidsperspektiv - Runtime - Medlemmars engagemang ur ett tidsperspektiv - Aspekt i spelet
WBS	<i>Inga träffar</i>	<i>Inga träffar</i>	<i>Inga träffar</i>

Efter analys av empirin av Android kan aspekter som inte är direkt anknutna till forskningsfrågan, men som är relevanta för studien, lyftas fram. Det framgår i diskussionerna vem som är ansvarig för utvecklingen vilket visar på att hierarki existerar inom organisationerna. Det finns nackdelar med att ett OS-projekt är sponsrat av kommersiella organisationer. När en utomstående organisation är involverad är delar av utvecklingen beroende av traditionella utvecklingsprocesser. Eventuella problem måste hanteras enligt traditionella *project management*-metoder vilket resulterar i att resurser och tid blir krävande och avgörande. Det tar Google betydligt längre tid att lösa ett avgörande problem (groups.google, 2012). Detta påverkar OS-utvecklarna negativt eftersom de inte

kan fortsätta arbeta. Om den utomstående organisationen väljer att inte stödja en innovation som efterfrågas i *communitiet* är utvecklingsmöjligheterna begränsade och utvecklingen inte lika fri som inom självständiga OS-utvecklingar. Google försöker inte applicera egna "straight-forward" utvecklingsmetoder (krav på dokumentation o s v) inom *communitiet* för att strukturera utveckling enligt egna principer utan vill driva Android-utveckling enligt OS-normer och metodik. Begrepp för att förklara hur en utveckling har utförts, eller ska utföras, diskuteras inte. Diskussionerna kretsar kring förklaringar av mjukvaror och termer som beskriver funktioner i system. Empirin speglar sammanfattat hanteringen av aktiviteter, resurser, problemlösningar och målsättningar.

Diskussionerna i WebKit är främst fokuserade på "bug-fixes" där lösningar på problem har hög prioritet (bilaga 2, tabell 7). Även om WebKit är sponsrat av en kommersiell organisation är närvaron av denna inte genomgående. Utvecklare använder sig av mailinglistor och forum som ett verktyg för support. OS-projektet WebKit är ett självgående projekt i den mån att utvecklare har möjlighet att utnyttja resurser och implementera system fritt utan att tillhöra eller involveras i aktiviteterna inom kärngruppen eller *communitiet* överlag.

OAD är det OS-projekt som mest kan liknas vid ett fullblodat OS-projekt. All kodutveckling är öppen men har en sluten grafisk utveckling vid sidan om. Med hänsyn till en liten utvecklingsgrupp blev "bug-fixing" en självklar dominerande aktivitet. Med detta menas inte att "bug-fixing" är högsta prioritet. Det existerar en kärngrupp som gemensamt tilldelar varandra ansvarsområden som närmast kan liknas vid traditionell mjukvaruutveckling (wildfiregames, 2012). När utvecklingen sträcker sig utöver organisationens möjlighet att bidra med resurser tenderar struktur, ansvar och målsättningar att falla bort.

De tre studieobjekten utvecklas under former som hanterar parallella utvecklingar (groups.google, 2012; wildfiregames, 2012; lists.webkit, 2012). Bevis finns för att det existerar ansvarsområden och auktoritet inom utvecklingarna d v s att utvalda utvecklare har möjlighet att besluta i vissa frågor som påverkar den övriga utvecklingen. Fall då utvecklare blivit tillrättavisaade p g a dåliga arbetsinsatser, inte följt avtal, uppnått mål eller i någon mån kontrollerats sett till arbetstuppgifterna har inte framkommit. Samtliga studieobjekt har koppling till den traditionella marknaden i form av stöd för utvecklingen eller som aktiva deltagare av utvecklingarna. Samarbetet påverkar båda segmenten, mestadels i form av begränsningar för OS, men OS är den som bestämmer kriterierna medan de traditionella organisationerna måste leverera, ändra och anpassa sig för att tillgodose dem. Sett till utvecklingarna ur ett definitionsperspektiv diskuteras eller utvärderas inte huruvida applicerade utvecklingsmetoder är korrekta, nödvändiga eller tillräckliga.

Inom OS-*communities* är hierarkin utarbetad efter erfarenhet och prestation. Inom varje enskilt projekt finns en hierarki samtidigt som det inom OS-fenomenet existerar en hierarki på global nivå. Hierarkin inom ett OS-projekt består av en "core-group" som ansvarar för projektet och fattar avgörande beslut (groups.google, 2012; wildfiregames, 2012). Övriga medlemmar bidrar till projektet i form av källkod, lösningsförslag, utvecklingsidéer och kunskap. Programmerarna kan vara aktiva i flera projekt samtidigt; det är insatsen som räknas. Programmerare som bidrar med lösningar och kunskap genom medverkan i ett projekt blir erkända som skickliga programmerare inom hela OS-fenomenet. Istället för ekonomisk kompensation tilldelas programmerare en viss status inom organisationen. Kunskap är därmed en avgörande faktor för att lyckas. Att tillhöra

eliten av programmerare inom OS motiverar. På grund av OS-fenomenets globala spännvidd sammanförs skickliga programmerare som kan mäta sina kunskaper med varandra och samtidigt dela med sig av dem. OS-fenomenet har inte bara strukturerat en självorganiserande, global mjukvaruutvecklingsmiljö utan också etablerat en mötesplats där programmerare kan visa upp sig, vilket lett till att spetskompetenser inom organisationen lyfts fram. Även om företag säkrat kompetens i form av skickliga programmerare med avtal och ekonomiska kompensationer är det möjligt för programmerarna att vara delaktiga i OS-projekt vilket resulterar i att OS-*communities* har en unik tillgång till resurser.

Det finns utvecklingsmöjligheter för programmerare inom OS-*communities* (bilaga 2, tabell 3). Där allt arbete är frivilligt och arbetsbelastningen avgörs av programmeraren kan uppgifter som inte ligger inom ramen för programmerarens spetskompetens vara tilltalande i utvecklingssyfte. Projektmedlemmarna har tillgång till hela projektet och kan därför välja att studera och diskutera okända aspekter i projektet. Även om kunskapspridning är en central del av OS-utveckling är det inte alltid att en programmerare får tips och råd av andra programmerare. Om programmerarna inte intresserar sig för en idé eller utveckling engagerar de sig inte i diskussionerna. Det existerar fler individuella delmål inom ett projekt än vad som är dokumenterade eller allmänt kända. Vid utveckling är kontroll av resultat det väsentliga. Om en programmerare inte presterar som förväntat påverkas utvecklingen negativt och programmeraren riskerar få dåligt rykte inom OS-*communitiet*.

Kodpraxis är en grundläggande princip inom OS (groups.google, 2012). Att skriva kod som är smart och effektiv är inte till någon nytta om den är obegriplig. Samtidigt som det organiseras arbete inom OS-*communities* har OS-fenomenet utarbetat regler för hur kod ska se ut. Dynamiska lösningar är oftast effektiva sett till prestanda men kan bara hanteras och förstås av programmeraren som skrivit dem. Mjukvaruutveckling inom OS är en långsam process eftersom kod måste skrivas ”dum” och primitiv. Funktioner och metoder är utvecklade i en serie mindre delar, en modulbaserad utveckling, där varje enskild modul utför en specifik aktivitet. Varje modul fungerar självständigt och möjliggör att andra programmerare fritt kan återanvända kod och implementera moduler. KISS – Keep It Simple, Stupid – är ett uttryck som enkelt beskriver hur OS-fenomenet förhåller sig till kodutformande och myntades då Unix utvecklades.

4.2 Sammanställning kopplad till forskningsfrågan

Aktivitet syftar till aktiviteter i mjukvaror men kan också relatera till uppgifter som är relevanta för utvecklarna i projektet. Vid analys av uttrycket kostnad sammanföll detta ord med “performance cost”. “Cost” och “expensive” (expensive operation) i OS-utveckling kan referera till hur mycket mjukvaran belastar datoriserade system och inte endast till en kostnad som relaterar till finansiella faktorer. Ett datum är oftast relaterat till “release-date” vilket syftar till datum då en uppdatering ska offentliggöras. “Date” förekommer ofta i samband med “up-to-date” eller “update” vid diskussion av uppdateringar av mjukvara och används också när medlemmar i utvecklingen vill uppdateras inom ett specifikt område. Generellt är dokumentationen utspridd över olika *communities*. Dokumentation är inget krav och sker på eget initiativ. Ett projekt har flera mål och aktiviteter. Målen diskuteras inom gruppen för att klargöra om de är relevanta att utföra. Mål kan både syfta till projektets mål och mjukvarans mål. Generellt är målen individuella och definierade av varje medlem. Planering sker utifrån en utvecklarens perspektiv. En plan innefattar en eller flera utvecklare som individuellt bestämmer sin

planering och den är frivillig att utföra. Planering av framtida aktiviteter, liksom planering av datum, utarbetas utifrån diskussioner i forum och mailinglistor. Medlemmarna har rätt att uttala sig om projektets olika planeringar. Planering avser att beskriva och informera om utveckling. Resurser speglar inte projektets resurser när temat diskuteras i grupp. Resurser beskriver mjukvarans resurser; mjukvaran har resurser i form av andra mjukvaror. Det framgår att schema avser att bestämma datum framför allt för “release-schedule”. Schema har också en teknisk aspekt. Oftast avser schema att belysa en teknisk funktion. Ett schema behövs för att utföra något frekvent i mjukvaran. Schemat används vid enstaka fall för att koordinera arbetet. Konceptet tid används för att beskriva hur lång tid det tar att utveckla funktioner inom ett projekt. Tid är en avgörande faktor vid strukturering av arbetet mellan medlemmarna i utvecklingen.

5. PROJECT MANAGEMENT INOM OPEN SOURCE

Detta kapitel avser att redogöra för vad som har konstaterats utifrån den empiriska undersökningen i relation till teori och tidigare forskning. Genom att återge teoretiska grunder och innehåll i analysen som kan återkopplas till frågeställningen genomförs en djupare analys för att besvara forskningsfrågan. Resultatet besvaras utifrån aspekterna planering, mål, budget och schema som presenteras i tabell 2.3.

I bilaga 3, tabell 1-7, presenteras underlaget för resultatet. Tabellerna har tre kolumner; en kolumn Teori där teorin som förespråkas i kapitel 2 sammanfattas, en kolumn Analys där empirin från kapitel 4, med relevans för frågeställningen, presenteras, och en kolumn Teori vs Analys som avser jämföra teorin med empirin. För att genomföra en relevant analys utan upprepningar och med tydligare budskap har vissa av analysenheterna sammanförts, se tabellerna 2, 4 och 7.

5.1 Empiri i relation till forskningens syn

5.1.1 Planering

Den största skillnaden mellan traditionella projekt och OS-projekt är bristen på projektdokumentation. Enligt teorin är grunden för *project management* att samtliga projektaspekter ska dokumenteras vid projektets start. Överlag görs inte detta inom OS-utveckling. Att dokumentera framtida aktiviteter och schemalägga deras utföranden har knappast någon betydelse för projektets välbefinnande inom OS-utveckling. Teorin förespråkar en detaljerad aktivitetsuppdelning och vikten av att samordna dessa för ett lyckat projektutförande. Inom ett OS-projekt är den aktivitet som främst schemaläggs lanseringen av en ny uppdatering. Övriga aktiviteter och önskade utvecklingar förblir oplanerade ur ett teoretiskt perspektiv. De förblir oorganiserade och utarbetas och uppmärksammas endast av några få inom OS-*communitiet*. Teorin menar att planering är nödvändig för att projektet ska bistås med resurser från organisationen. Inom ett OS-projekt är resurserna främst i form av andra mjukvaror och det saknas en bakomliggande planering för att hantera dessa.

5.1.2 Mål

Där teorin belyser vikten av att strukturera och utforma aktiviteter väljer OS-projekten att utforma aktiviteter i efterhand och efter eget intresse. Resultaten uppnås genom olika förfaranden och prioriteringar. Inom OS pågår en ständig utveckling och definiering av mål. Nya förslag till aktiviteter och mål uppmuntras inom OS-utvecklingen och ses som något positivt. Inom den traditionella utvecklingen ses nya förslag till aktiviteter och mål som ett problem eftersom nya planeringar måste genomföras. Inom OS-projekten är resurserna främst i form av andra mjukvaror som är tillgängliga för alla inom projektet för att uppnå ett mål. Enligt teorin kan mål beskrivas utifrån termer som kostnad, schema och teknisk prestation. Inom ett OS-projekt är fokuseringen främst på den tekniska aspekten som oftast poängterar mjukvarans uppgifter. Målen som sätts upp inom OS-utveckling är av stor differens och kan rubriceras som delmål. I slutändan uppnår båda projektutförandena samma sak där aktiviteter blir strukturerade och beskrivna och delmål identifierade.

5.1.3 Budget

Den ekonomiska faktorn och dess betydelse för ett OS-projekt framstår som mindre relevant i den empiriska analysen. Teorin förespråkar en detaljerad bild av projektets rörliga kostnader och avser att beskriva projektet i numeriska termer. Aktiviteter som berör utförandet av projektet inom OS är inte kostnadsbaserade då arbetskraften är frivillig. Om kostnader existerar för ett OS-projekt diskuteras inte dessa inom projektets forum och mailinglistor. Kostnader avgör i vilken riktning en mjukvara ska utvecklas och vilka stöd för andra plattformar som ska implementeras i mjukvaran, men de är inte avgörande, eller sätter ramar, för hur projektet ska bedrivas. Där teorin pekar på "avgörande" och "viktig" för projektets utförande och organisationens medverkande, förefaller budgeten vara mindre relevant och avgörande inom OS-utvecklingen. Generellt är resurser inte en aspekt som måste kalkyleras eftersom de ofta är gratis.

5.1.4 Schema

Teorin lyfter fram schema som en avgörande faktor vid resurshantering, budgetering och strukturering av utarbetade aktiviteter. Inom OS-utveckling är den enda viktiga aspekten att schemalägga och följa lanseringen av nya uppdateringar, d v s bestämma datum för "release-date". Aspekter, som resurser, deadline och aktiviteter är inte schemalagda i detalj såsom förespråkas i teorin. I motsats till teorin utformar OS-utvecklare personliga scheman för att koordinera det gemensamma arbetet men detta är inget som schemalägger projektet i sin helhet. Att följa scheman och utföra aktiviteter i enlighet med dem är inte avgörande för ett OS-projekts utveckling. Den aspekt av tid som är relevant inom ett OS-projekt är medlemmarnas möjlighet att bidra till utvecklingen.

5.1.5 Tidigare studier

Forskningen menar att *project management*, och då även projektplanering, inom OS-utvecklingen inte alltid förhåller sig som sig bör enligt teorin. Min studie överensstämmer med tidigare forskning. Som Aberdour konstaterar; "a major shift from traditional development where coordination and planning come first" (Aberdour, 2007, s. 61), förhåller sig OS-utveckling distanserad till traditionella *project management*-teorier.

Nakagawa et al. menar att "[...] planning, documentation and modelling is not the main focus of these projects" (Nakagawa et al, 2008, s. 1238). Mina undersökningar i denna studie tyder på att Nakagawa et al.'s påstående är korrekt.

Ulhøi påstår att "In open source innovation, innovators freely choose the assignments they want to work on" (Ulhøi, 2004, s. 1110), vilket också överensstämmer med resultatet i min studie.

Hippel's (2003) och Ulhøi's (2004) påståenden att aktiviteter och mål inom OS-*communities* är utarbetade utifrån tekniska kriterier, vars lösning baseras på tekniska kunskaper snarare än på en utarbetad projektplanering, visar sig vara sant.

Jaccheri och Østerlie (2007) menar att de hittat OS-projekt som utformar traditionell projektplanering och menar att detta är ett fenomen inom OS-utveckling. Studieobjekten i denna studie står sig annorlunda till detta eftersom de inte använder traditionell projektplanering.

Madey et al. (2002) menar att organiseringen inom OS-*communities* är skilda från traditionella utföranden och strukturer inom organisationer. Utifrån teorin som granskats i denna studie stämmer resultatet i denna studie överens med detta påstående.

Min tolkning är att avsaknaden av projektplanering inom OS-projekt som presenteras av tidigare forskning i kapitel 2.6 varken kan motbevisas eller påstås ha felaktiga uppfattningar. Det är viktigt att poängtera att ingen av de studier som nämns fokuserar på att undersöka *project management* inom OS. Fynden som presenteras har hittats under studiernas gång. Med hänsyn till bristen på tidigare empiriska studier som undersöker *project management* inom OS är det svårt att djupare analysera tidigare studiers påståenden.

5.1.6 Analys

Min första tolkning av den djupare analysen var att *project management* är frånvarande inom utveckling av OS-mjukvara. Om man ska jämföra terminologi utifrån dess bakomliggande betydelse och syfte är det enkelt att påvisa skillnader mellan OS-utveckling och *project management*. Inom OS är det inte nödvändigt att strukturera ett projekt såsom *project management* förespråkar. Det framgår av studien att planering inte är en aktivitet som utarbetas innan projektet initieras. Det visar sig dessutom inte vara nödvändigt, eller avgörande, att genomföra en detaljerad planering såsom *project management* rekommenderar för ett lyckat projektutförande. Metoder som förespråkas av *project management* är inte applicerade inom OS; syftet med metoderna uppnås på andra sätt. Man kan enkelt hitta bevis som styrker tidigare forskningspåståenden (kap. 2.6).

Jag konstaterar att planering och dess underkategorier inte hanteras utifrån *project managements* regler och bestämmelser, och att de i vissa fall är helt frånvarande (ex. budget och WBS). Istället utarbetas de "on-the-fly" under projektets gång vilket kan lyftas fram som den största avvikelser från traditionell *project management*. Även om teorin inte i detalj beskriver strukturen i ett projekt för att utöva *project management* är det tydligt utifrån empirin att OS-projekt har en friare och bredare struktur. Med detta påstår jag inte att OS-projekt inte har en ledning som fattar avgörande beslut, utan att ansvar fördelas och åberopas friare i utvecklingsstadiet. *Project management* anser att projektdokumentation är grundläggande, men är inte organiserad eller prioriterad på samma sätt inom OS. Efter den empiriska undersökningen kan man konstatera att det inom OS sällan hänvisas till extern dokumentation utanför mailinglistorna och att dokumentation överlag är bristfällig.

Angriper man empirin utifrån ett statistiskt perspektiv inser jag att studien inte är ensidig i budskapet i den mening att tidigare påståenden är de enda existerande. Genom att räkna ihop antalet träffar av analysenheten ser jag att begrepp valda att analyseras upprepas med undantag för budget och WBS (bilaga 1, tabell 3). Denna förenkling av empirin visar på statistik som tydligt tar avstånd från den tidigare tolkningen och pekar på att situationen förhåller sig tvärtom. Begreppen som ligger till grund för analysenheten har valts ut då de utifrån den teoretiska genomgången använts för att definiera eller utgöra *project management*. Sett till statistiken är dessa begrepp utspridda inom OS-utveckling och i många fall väl använda.

I den teoretiska genomgången framgår det t ex att resurser definieras som något som måste mätas, kontrolleras, hanteras och distribueras. Även om det framgår att resurser kan vara maskiner och människor, framgår det inte vad en resurs inte är. Därför går det inte att utesluta resurser i form av andra mjukvaror inom OS som korrekta resurser

utifrån teori. Resurser är resurser och dessa måste hanteras, bearbetas och koordineras oavsett typ. Om man väljer att ställa samtliga begrepp (analysenheten) mot teorin med syfte att utesluta dem från *project management* inser man att begreppen inte är tillräckligt definierade i *project management*-teori, d v s teorin klargör inte vad dessa begrepp inte är. Vid tillfällen där begrepp används flitigt i två separata verkligheter, men som utifrån övergripande definitioner har mycket gemensamt och i grunden ser likadana ut, bör begrepp (kap. 2.5) antingen kunna uteslutas grundat i teoretiska påståenden eller accepteras som befintliga, likvärdiga element tills motsatsen kan bevisas.

Vid utförande av en projektplan är kommunikationen ett viktigt verktyg inom organisationen. Projektmedlemmar ska informeras, resurser fördelas och syfte och mål definieras. Kommunikationen genomsyrar alla nivåer inom organisationens hierarki. När projektutvecklingen påbörjas blir avdelningarna inom organisationen självständiga. Avdelningarna har en plan att följa och fördefinierade aktiviteter att utföra. Mjukvaruutveckling inom OS är i ständig förändring och kommunikation en vital mekanism genom utvecklingens gång.

De aspekter av *project management* som den traditionella organisationen applicerar innan projektnitiering tillämpas konstant genom hierarkin för projektet i en OS-utveckling. Förutsättningarna för projektet (organisatoriska förändringar, resurstillgångar och utarbetandet av nya mål) ändras under projektutförandet. *Project management* manifesterar sig i kommunikationen inom OS-fenomenet. Ser jag till *project managements* uppgift att leda projekt i symbios med organisationens strategi är detta kärnan inom OS. Jag vill påstå att det inom OS är uppenbart att organisationen bistår projekten. Genom att effektivisera kommunikationen är det OS-*communitiets* uppgift att möjliggöra resursfördelning till OS-projekten. OS-*communitiet* har utvecklats till att enbart fokusera på detta. Verktyg som skapas och används för upprätthållandet av mjukvaruutveckling inom OS har som syfte att sammanfoga organisationen med projekten.

Med hänsyn till att statistiken talar för en närvaro av *project management* och att det i praktiken upprättas en symbios mellan organisation och projekt anser jag att OS-fenomenet uppfyller *project managements* syfte och mål.

6. SLUTSATS

Statistik talar för en närvaro av *project management* vilket överensstämmer med min tolkning av empirin att OS-*communities* huvudsakligen fokuserar på att sammanföra organisation med projekt. Min slutsats är därför att *project management* inträffar vid mjukvaruutveckling inom OS. En form av *project management* som inte följer de traditionella reglerna och rekommendationerna, men som fortfarande har gemensamma aspekter med organisatoriska mekanismer, har med tiden utvecklats till att bli en unik *project management*-metod. Ett koncept som utgick från att definiera öppen källkod har utvecklats till ett koncept som specialiserat sig på att sammankoppla människor, organisera, förse med kommunikationsmöjligheter och realisera mjukvaruutveckling i en virtuell värld på global nivå.

Där forskningen pekar på bristande *project management* inom OS, och sett till definitioner är jag beredd att hålla med, menar jag trots detta att det i slutändan är en missvisande bild av verkligheten. Med verktyg för att sammankoppla världen på global nivå ställs nya krav på organisering och hantering av aktiviteter. Vid strukturering av mjukvaruutveckling har OS-*communitiet* större friheter, vilket innebär att t ex mål, budget eller scheman nödvändigtvis inte måste utarbetas i detalj, eller i förväg, eller som vid budgetering och WBS, inte alls. Oberoende av detta är OS-fenomenet i behov av organisatoriska mekanismer som möjliggör koordinering av aktiviteter inom OS-*communities* med syfte att uppnå resultat. Även om det inte är självklart att *project management* appliceras inom OS visar analysen att *project management*-mekanismer praktiseras spontant vid mjukvaruutveckling av OSS. Mjukvaruutvecklingen inom OS utnyttjar möjligheter i en global självorganiserande organisation som introducerar en ny form av *project management*.

6.1 Egna reflektioner

Från början var min tanke att genomföra intervjuer för att få svar på frågeställningen. Jag kontaktade aktiva inom OS-utveckling via mail. Endast ett fåtal besvarade min förfrågan om att delta i en intervjustudie. Efter att ha presenterat mitt syfte avtog intresset. Jag blev upprepade gånger ombedd att ytterligare utveckla vad jag ämnade undersöka. Mina kontakter konstaterade att projekten var små och att planering inte var relevant. Även aktiva inom stora projekt var motvilliga att medverka i intervjuer. Det visade sig att nyckelpersoner inom OS-*communitiet* tidigare tillfrågats om de ville medverka men att de då skriftligen tagit avstånd från att låta sig intervjuas. Det var därför inte möjligt att realisera intervjuerna med den tidsram som gavs.

Vid ytterligare granskning av frågeställningen konstaterade jag att en riktad innehållsanalys föll inom ramarna för den kvalitativa ansatsen. Efter att ha undersökt frågeställningen inom OS-*communities* förvånades jag över hur lite projektplanering intresserade de aktiva. I kontakter med aktiva inom OS-utveckling har det under studiens gång poängterats att OS-projekten är frivilliga arbeten och att man därför inte kan begära engagemang utöver kodbidragande. Min tolkning är att man inom ett OS-projekt generellt inte är intresserad av hur ett projekt styrs eller planeras, utan endast av utvecklingen, egna prestationer och vad de resulterar i. Det är uppenbarligen bevisat att projekt går att bedriva, utveckling går att genomföra och resultat är möjligt att leverera utan en traditionell planering. Utveckling fungerar oavsett om allt är dokumenterat eller i förväg genomtänkt såsom *project management* förespråkar.

Det har varit en intressant och givande process att genomföra studien. Samtidigt som arbetet har inneburit en övning i att skriva, strukturera och genomföra en vetenskaplig rapport har det också resulterat i en djupare förståelse för samhällsvetenskaplig forskning. Att ta del av, förstå och anpassa sig till vetenskapliga konventioner har varit lärorikt och introducerat ett nytt tankesätt. Min syn på vetenskap och forskningsrapporter har utvecklats och jag har tagit till mig kunskap som i framtiden kommer att hjälpa mig utvärdera vetenskapliga studier.

REFERENSER

Aberdour, M. (2007). Achieving Quality in Open Source Software. *The IEEE Computer Society*, 58–64.

Antvik, S., & Sjöholm, H. (2007). *Project management and methods*. Västerås: Projektkonsult Håkan Sjöholm AB.

Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.

Bar, M., & Fogel, K. (2003). *Open Source Development with CVS* (3rd ed.). Arizona: Paraglyph Press.

Bryman, A., & Bell, E. (2003). *Företagsekonomiska forskningsmetoder*. Lund: Liber.

Cleland, D. I. (1999). *Project Management – Strategic Design and Implementation*. Singapore: McGraw-Hill.

Healy, K., & Schussman, A. (2003). *The Ecology of Open-Source Software Development*. University of Arizona

Hippel, E. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science, *Organization Science*, Vol. 14 No. 2, 209–223.

<http://www.businessdictionary.com>, (2012-06-20)

<http://www.code.google.com>, (2012-04-20)

<http://www.gnu.org>, (2012-08-22)

<https://groups.google.com/forum/?fromgroups#!forum/android-contrib>, (2012-04-20)

<https://lists.webkit.org/pipermail/webkit-dev/2012-February.txt>, (2012-04-25)

<http://www.opensource.org>, (2012-04-20)

<http://www.sourceforge.com>, (2012-04-20)

<http://www.wildfiregames.com/forum/index.php?s=ad6b7033feca36ec63aa8026d44f3c91&showforum=312>, (2012-04-25)

Jaccheri, L., & Østerlie, T. (2007). Open Source Software: A Source of Possibilities for Software Engineering Education and Empirical Software Engineering. *First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007)*, 1–5.

Kerzner, H. (1992). *Project management – A Systems Approach to Planning, Scheduling, and Controlling*. New York: Van Nostrand Reinhold.

Koontz, H., & Weihrich, H. (1988). *Management* (9th ed.). Singapore: McGraw-Hill.

Larson, E. W., & Grey, C. F. (2011). *Project Management – The Managerial Process* (5th ed.). New York: McGraw-Hill.

Madey, G., Freeh, V., & Tynan, R. (2002). The Open Source Software Development Phenomenon: An Analysis Based On Social Network Theory. *Americas Conference on Information Systems (AMCIS2002)*, 1806–1813.

Nakagawa, E. Y., Machado de Sousa, E. P., Gabriel de Faria Andery, K. M., Morelli, L. B., & Maldonado, C. (2008). Software Architecture Relevance in Open Source Software Evolution: A Case Study, *COMPSAC '08 Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, 1234–1239.

Project Management Institute. (2008). *A Guide to the Project Management Body of Knowledge* (4th ed.). Newtown Square: Project Management Institute.

Raymond, E. S. (2003). *The Art of Unix Programming*.

SBU. (2012). *Utvärdering av metoder i hälso- och sjukvården: en handbok*. Stockholm: Statens beredning för medicinsk utvärdering (SBU).

Stallman, R. M. (1999). *OPENSOURCES Voices from the Open Source Revolution*. USA: O'Reilly

Tiwari, V. (2010). Some Observations on Open Source Software Development on Software Engineering Perspectives. *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 2, No 6, 113-125.

Ulhøi, J. P. (2004). Open source development: a hybrid in innovation and management theory, *Management Decision*, Vol. 42 Iss: 9, 1095–1114.

Wallén, G. (1996). *Vetenskapsteori och forskningsmetodik* (2:a rev. uppl.). Lund: Studentlitteratur.

BILAGA 1

Tabell 1 – Söktermer och antal träffar vid sökning av litteratur

Databas	Sökord	Träffar
Libris	“project management”	533
	“project management” and “Open Source”	3
	“project management” and planning	103
	“project management” and theory	32
	“project management” and theory and planning	9
LibHub	“Open source”	97 387
	“Open source” and planning	18 242
	“Open source” and Planning and objective	6 934
	“Open source” and Planning and objective and budget	1 984
	“Open source” and Planning and objective and budget and schedule	822
	“Open source” and Planning and objective and budget and schedule and time	813
	“Open source” and Planning and objective and budget and schedule and time and resource	776
	“Open source” and Planning and objective and budget and schedule and time and resource and cost	742
	“Open source” and Planning and objective and budget and schedule and time and resource and cost and date	13
	“Open source” and Planning and objective and budget and schedule and time and resource and cost and date and goal	13
	“Open source” and Planning and objective and budget and schedule and time and resource and cost and date and goal and project	1

Tabell 2 – Empiriskt material

WebKit	https://lists.webkit.org/pipermail/webkit-dev/2012-May.txt
	https://lists.webkit.org/pipermail/webkit-dev/2012-April.txt
	https://lists.webkit.org/pipermail/webkit-dev/2012-March.txt
	https://lists.webkit.org/pipermail/webkit-dev/2012-February.txt
	https://lists.webkit.org/pipermail/webkit-dev/2012-January.txt
	https://lists.webkit.org/pipermail/webkit-dev/2011-December.txt
	https://lists.webkit.org/pipermail/webkit-dev/2011-November.txt
	https://lists.webkit.org/pipermail/webkit-dev/2011-October.txt
Android	https://groups.google.com/forum/?fromgroups#!forum/android-contrib
0AD	http://www.wildfiregames.com/forum/index.php?s=ad6b7033feea36ec63aa8026d44f3c91&showforum=312

Tabell 3 - Antalet träffar av analysenheten som påträffas inom de olika studieobjekten

	Android	WebKit	0AD
Objective	2	2	92
Planning	105	56	42
Budget	1	0	2
Schedule	5	5	6
Goal	11	36	47
Resource	52	20	260
Activity	13	1	19
Cost	5	43	91
Time	319	250	420
Date	7	8	126
Documentation	22	13	54
WBS	0	0	0

BILAGA 2

Tabellerna 1-10 presenterar studieobjekten var för sig i relation till analysenheten. Källor se bilaga 1 tabell 2.

Tabell 1 - Analysenhet "Activity"

Android	WebKit	OAD
<p>Termen "activity" belyser en aktivitet inom mjukvara och mailinglista. Mjukvaror kan ha olika funktioner och då utträta olika aktiviteter, men avser också att benämna aktivitet i form av en uppgift som utformas av medlemmarna inom projektet.</p> <p>"Android APIs also contain methods for sending keypresses and touch screen events in the Java code, but these work only inside the activity that is calling these methods, so other applications can't be affected."</p> <p>"It looks like the easiest approach would be to rewrite the Search Activity, the Suggestions Provider, and the Preferences to plug into an instance of an interface rather than specifically the Google classes."</p> <p>"Mention the provider in respective manifest configure the required url(bing,Yahoo etc) in provider and return cursor so that from your activity you can launch browser appending the query string."</p> <p>"Hi, Currently this activity is in the testing/debugging phase. As soon as this is completed we plan to submit for review."</p> <p>"Note however that I'll have to temporarily step away from my Android tasks for a good chunk of time this month, and AOSP activity will probably be slower during that period"</p>	<p>Inga träffar.</p>	<p>Termen "activity" belyser en aktivitet inom mjukvara, spel och forum. Mjukvaror kan ha olika funktioner och då utträta olika aktiviteter, men avser också att benämna aktivitet i form av en uppgift som utformas av medlemmarna inom projektet. "Activity" kan referera till CPU-aktivitet.</p> <p>The C code could play background music depending on the current game condition, and also keep track of the objects in the screen area and play sounds for what ever activity they were currently taking part in.</p> <p>Also I looked on Trac and didn't see a ticket and there was little activity on the forum or IRC, so I just started working on it. I wasn't trying to duplicate effort or step on toes.</p> <p>A search of the forums says Jeru and Erik are set to update the documentation, but I haven't seen much evidence of activity in that respect.</p> <p>This causes a lot of JS activity (formations and stuff) in addition to all the AI startup stuff thats going on at the same time (UnitAI and Computer AI startup). This isn't a high priority, but it does make the game really sluggish at the start so it's something we should think about.</p>

Tabell 2 - Analysenhet "Cost"

Android	WebKit	OAD
<p>Termen "cost" avser att belysa ekonomiska faktorer och kostnaden för en mjukvara att genomföra aktiviteter ur ett prestandaperspektiv. I de fall där kostnaden avser att belysa en ekonomisk faktor är det oftast i samband med Googles resurser eller utomstående plattformar och system (licenser).</p> <p>Very quickly, we reach a point where the cost of merging such an external contribution in Google's internal master tree is higher than the cost of having a Google engineer make that change directly there.</p> <p>Yet, as pointed out earlier, that comes at a cost and the developer needs to be able to control these costs.</p> <p>You sign a 2 > year contract to obtain GNex at a lower cost for two years and support > as a development device is taking a way two months into contract.</p>	<p>Termen "cost" avser att belysa ekonomiska faktorer och kostnaden för en mjukvara att genomföra aktiviteter ur ett prestandaperspektiv. I de fall där kostnaden avser att belysa en ekonomisk faktor är det oftast i samband med underhåll av servrar, eller utomstående plattformar och system. "Cost" används för att avgöra om ett system har en effektiv inlärningskurva.</p> <p>While Chromium takes an approach to hard-code the rage of acceptable values, such an approach has a high maintenance cost and prone to problems such as having to increase the range periodically as the score slowly degrades over time.</p> <p>While we could have added another dedicated apache server with all nice features Graph Server's native backend provides, the maintenance cost of maintaining such a server seemed too high.</p> <p>I wrote our own backend using Google App Engine <http://code.google.com/appengine/> for its low maintenance cost and ease of use; note App Engine is already used by commit-queue and flakiness dashboard.</p> <p>The only downside would seem to be the extra overhead associated with making changes to GoogleURL. Has that sort of cost been an issue with ANGLE? Do you anticipate wanting to make a lot of significant code changes, or would you primarily just be concerned about the ease of bug fixing?</p> <p>I don't really get this point. With WTF linked statically, no matter how "largish" WTF, it will not cost much to use. You say you don't want to convert <code>std::string->WTF::String</code> and <code>WTF::String</code> at the browser level, but aren't you doing that a lot more with the current code?</p>	<p>Termen "cost" avser att belysa ekonomiska faktorer och kostnaden för en mjukvara att genomföra aktiviteter ur ett prestandaperspektiv. Termen "cost" har en betydelse i spelet (virtuell kostnad) för objekt som kan "köpas".</p> <p>The general approach is not great though since it should be taking the cost of items into account when using the priorities since this is what the QueueManager does.</p> <p>The point of doing that work is that multiple units with the same goal can use the same flowfield - moving a group of 100 units has the same flowfield-generation cost as moving 1 unit, and the units can still move independently of each other (so they move as a swarm, not as a formation).</p> <p>As for fishing boats, should they be counted as food gatherers too? Third, are citizensoldiers basically workers but their food cost is converted to metal (or something like that?)? And finally, for the healer, I assume they're not allowed to carry resources, correct?</p>

Tabell 3 - Analysenhet "Date"

Android	WebKit	OAD
<p>Termen "date" används i samband med "update" (uppdatera) och diskuteras när en uppdatering är nödvändig, bristfällig eller under konstruktion. "Date" används också när medlemmar i utvecklingen vill uppdateras inom ett specifikt område.</p> <p>Github uses a different path configuration for the repositories, so it would require quite a manual work to update the manifest file, if all the repositories used by android are mirrored at github, that is.</p> <p>The best solution to date is to change the XML resource to something like android:default="@bool/mirror", and in the code change it to getBool("mirror", getResources().getBool(R.bool.mirror).</p> <p>Any update on this. I request to please add this project so that we >>>> can begin proposing the changes for very configurable font rendering >>>> engine system in Android.</p> <p>The code has been checked in and also updated to the latest release.</p>	<p>Termen "date" används i samband med "update" (uppdatera) och diskuteras när en uppdatering är nödvändig, bristfällig eller under konstruktion. "Date" avser också att definiera om något är relevant ur ett tidsperspektiv (up-to-date).</p> <p>I will update this page whenever EFL >> port coding style guide is changed.</p> <p>Update other build systems as is applicable.</p> <p>Personally I would have told you not to update our files and just let them be broken if we didn't update them. The fact that so much work is needed from developers besides the port developers to maintain a more slowly developing port indicates a problem with WebKit.</p> <p>This is exponential growth! How is a small port possibly suppose to stay up-to-date under those conditions? Of course this just means that WebKit is a vibrant project and you cannot be faulted for that, but I think with this onslaught of changes it may be time to reconsider past methods of keeping ports up-to-date.</p> <p>As far as keeping ports up-to-date goes, it seems like (in my limited experience) there's roughly four kinds of changes you have to work about: changes to tests that require new baselines, changes to the build system (adding, renaming, deleting files), changes to platform-specific core functionality (usually o/s-specific hooks), and changes to platform functionality (e.g., webaudio).</p>	<p>Termen "date" avser att definiera om något är relevant ur ett tidsperspektiv (up-to-date).</p> <p>I run the game on a up to date version of windows vista longhorn home premium.</p> <p>This file will most likely be kept up-to-date manually.</p> <p>Just as a reminder, please keep the "EngineProfiling" wiki page up to date when integrated.</p>

Tabell 4 - Analysenhet "Documentation"

Android	WebKit	OAD
<p>Termen "documentation" sker utifrån engagerade medlemmars egna initiativ. Det finns inget krav på att dokumentera efter stadgar men ibland efterfrågas dokumentation. Dokumentationen är utspridd i <i>communitiet</i> eller över nätet generellt och dokument faller under OS- licensen. Generellt är dokumentation bristfällig/frånvarande.</p> <p>The documentation is mostly spread around the framework source tree, as comments in the source code that get extracted at build time. The "root" of the documentation is probably slightly spread between the build/ directory, the development/ one and frameworks/base. If I remember correctly that should be enough to generate all of d.android.com and the offline documentation that's distributed in the SDK. We accept contributions to the documentation like we accept contributions to the code.</p> <p>Similar to adding copyright notices in code, how do you prefer this being handled when contributing documentation as no copyright notices are present in the document source code.</p> <p>The problem the Android SDK has had with Developers has been the lack of Documentation.</p> <p>We open-source the documentation, web site and samples as part of the Android Open-Source Project, and we routinely accept contributions there, following the regular contribution process.</p> <p>Write a specification document for this project, detailing the boundaries of what you want to support, proposed architectural choices, and submit it there for discussion.</p>	<p>Dokumentation sker inte på regelbunden basis och är inte indexerad eller centralt lagrad. Det finns inget krav på att dokumentera efter stadgar men ibland efterfrågas dokumentation. Dokumentationen är utspridd i <i>communitiet</i> eller över nätet generellt. Dokumentation uppdateras efter behov.</p> <p>It should've been written in the instruction. Thanks for bringing this to my attention. This is a fairly recent requirement. I have updated the documentation</p> <p>I didn't find any documentation about porting squirrelfish extreme on Arm. I tried to google it, and found some posts where said that there is squirrelfish extreme on Arm, but I can't find source code, or any documentation about it(Can anybody help me. Any information about squirrelfish extreme is important for me:)</p> <p>I'm not sure there's much documentation. If you have a specific question, I can try to answer it.</p>	<p>Dokumentation sker inte på regelbunden basis och är inte indexerad eller centralt lagrad. Det finns inget krav på att dokumentera efter stadgar men ibland efterfrågas dokumentation. Uppfattningen kring projektet varierar då det inte går att säkerställa den egentliga statusen eller kvalitén på den totala dokumentationen av projektet. Intresset för auto-genererad dokumentation går före manuell dokumentation. Regler för vem som ska utföra dokumentering är inte tillräckligt framlyfta.</p> <p>I want to add the wall_builder.js rmgem lib documentation to the wiki.</p> <p>The wiki currently has extremely limited documentation. A search of the forums says Jeru and Erik are set to update the documentation, but I haven't seen much evidence of activity in that respect. I'm trying to work on a few simple tickets, but without solid documentation I'm having a tough time of locating everything that needs to be located. So, is a structure already in place for documenting the game? If so, what is it or where can I find it? I've begun in a format based on the object or action being referenced here, if anyone cares to throw in what they know. Along with documenting past changes, I think that for every addition/change in functionality that could warrant a change to the documentation, the change should be required before the ticket is closed. If the person programming the patch doesn't want to do it, perhaps tagging the ticket with "document"?</p> <p>I think that's the system you are looking for, as Doxygen is able to scan the source code and rebuild documentation based upon whatever recent changes have occurred. Making coders do it manually</p>

		is unlikely to work, because we're lazy like that... Edit: Also it must be documented (or we risk historic bruno's wrath).
--	--	---

Tabell 5 - Analysenhet "Goal"

Android	WebKit	OAD
<p>Termen "goal" avser att belysa mål och uppgifter; inte bara projektets mål utan även mjukvarans mål med olika funktioner. "Goal" är oftast personliga, vilket innebär att de inte involverar alla i projektet och påverkar inte övrigas arbete och egna uppsatta mål.</p> <p>Agree with most of your comments on the high level goal. Any comments on the plan for iteration#2? Otherwise we start off with an upload and we can continue the discussion on code level.</p> <p>Recovery is a stripped down system whose primary goal is to modify the system partition (which is easy, since the full-size system isn't running at that point).</p> <p>The goal here is to prepare an image of a cache partition that gets recognized by recovery and installed over the equivalent files in the system partition, without touching any of the proprietary hardware-related files.</p> <p>Our goal is to be able to read/write on the two HID channels (interrupt/control) from a Java application.</p> <p>Our main goal is to make Android toolchain (current gcc version = 4.4.3; android.git.kernel.org) work for x86, and at least build system has to be changed accordingly.</p>	<p>Termen "goal" avser att belysa mål och uppgifter; inte bara projektets mål utan även mjukvarans mål med olika funktioner. "Goal" är oftast personliga, vilket innebär att de inte involverar alla i projektet och påverkar inte övrigas arbete och egna uppsatta mål. Det framgår att "goal" i viss utsträckning rubriceras som long-term/short-term.</p> <p>The main goals are to put a compile / runtime switch in for mouse lock, get the IDL in place, and pass the new movementX/Y members around.</p> <p>I don't know if splitting part of RenderLayer is totally against the goal of having that part composited.</p> <p>The overall goal is to simplify image download progress reporting by supporting roughly the same progress events as XHR and the File API for image elements.</p> <p>I think > that's what Adam tried ~ 15 months ago. By the way, this my long-term goal. My plan has been to align browser behavior before bringing this topic back to the IETF. That will make it much clearer exactly what we need to document.</p> <p>The URL you pasted shows Jon removing a feature that isn't used by the web (not analogous to this situation), after no objection (not analogous to this situation), and proposing removing it for the whole project on the same grounds (consistent with the goals I've stated).</p>	<p>Termen "goal" har betydelse vid diskussion om ett objekts förflyttning inom spelet, d v s "goal" avser den position som spelaren har definierat för ett objekt att förflytta sig till. "Goal" avser även att belysa mål i form av uppgifter som är på personlig basis. Generellt avser "goal" att belysa projektets mål.</p> <p>Given that passability grid, a unit can pathfind to a single goal tile, using a flowfield algorithm. That algorithm computes a vector for every tile, such that a unit can follow the vectors and will end up at the goal. See another screenshot which makes it clearer (the goal is the bottom-centre of the screen).</p> <p>Tonight I'm going to get some writing done. If I hit my goal of 10 pages (I'm at 3 pages now), then I'll spend all of tomorrow doing O A.D. planning and stuff. I'll prioritize thinking about and listing changes and features for new dude and dudette meshes.</p> <p>I think the important goal is that building the game should be easy and automatic, not that it should be fast.</p> <p>Obviously this would include more gameplay specific stuff like some of the things mentioned in this thread, so keep that in mind there is a long term planning goal here for getting what everyone needs to get done.</p> <p>Ease of modding has always been a design goal: most data is specified in XML files, and much of the gameplay logic is in JavaScript (at least in the current ongoing redesign of the gameplay system), and we've tried to avoid putting game-specific stuff in the C++ engine code.</p>

Tabell 6 - Analysenbet “Objective”

Android	WebKit	OAD
<p>Termen “objective” kan översättas till mål och aktiviteter som uträttas eller som diskuteras inom Android. Medlemmarna har önskemål om hur målen ska uppnås i samarbete med Google. Diskussionerna handlar oftast om stöd för olika funktioner och om hur Android är konstruerat för att möjliggöra en interaktion mellan flera system. Mål och aktiviteter avser inte att utveckla Androids kärna, utan är kompletterande system som antingen utnyttjar, eller tillför, funktion till Android. “Objective” är på individuell basis.</p> <p>My objective is to have a way for GDB to identify whether it is looking at a regular arm-linux-gnueabi binary or arm-linux-androideabi.</p> <p>As far as I know, the existing security solutions are limited and cannot do much more as blocking applications or scan the user space for malwares. It is not enough according to our vision. Our objective is to enhance the Android's security model, and in order to do this, we need the access to network layer. In few words, the idea is to redirect the packets through our filtering environment and return or drop them after they have been analyzed. Is there no way to consider such a possibility ?</p>	<p>Inga träffar.</p>	<p>Termen “objective” förhåller sig i de flesta fall till “objective” inom spelet, d v s det finns mål för spelaren att uppnå. “Objective” är även en term som används för att beskriva mål för utvecklingen av OAD, referera till programmeringsspråket Objective C och i vissa fall till ”Objective Observation”.</p> <p>The player is informed when an objective has been completed, or the objectives list is updated.</p> <p>Pauses the game when the objectives are viewed.</p> <p>Displays hints, instructions, history and goals or victory conditions. The button flashes when objectives are completed/updated.</p> <p>Well, the automatic battalions (red) benefit from any bonuses that make them meaningful formations as opposed to simple AoE-style groups (a key objective of OAD if I understand it correctly): Think of graphical changes, like units moving and fighting in an actual phalanx for example, or of statistical advantages (higher HP etc.), special attack actions... you name it.</p>

Tabell 7 - Analysenhet "Planning"

Android	WebKit	OAD
<p>Termen "planning" existerar i viss mån och är utarbetad efter behov. "Planning" involverar människor som vill engagera sig i en aktivitet. Planer är utarbetade på individnivå efter eget intresse. Ibland refereras till olika typer av planer som long-term/short-term och kan syfta till bearbetning av support för system och användare. Planerna diskuteras och utvärderas i forum och mailinglistor. Alla medlemmar har rätt att uttala sig om projektets olika planeringar.</p> <p>If we can come up with a good plan for this before starting that would be good:</p> <ul style="list-style-type: none"> - externalization method: There were several to choose and we chose to use org.eclipse.osgi.util.NLS. - one or several properties file(s)? - naming scheme for the strings: Should they show context? should they indicate placeholder for String.format? <p>I plan to make improvements to the algorithm soon.</p> <p>I am very interested in the changes you plan to make and is willing to help you out testing and reviewing them. I think the main problem here is as you write the statistical problems that result from overfitting the curve, at least from my observations.</p> <p>What about Hebrew translation, is that in the short-term plans?</p> <p>I'm planning to tag the CTS releases as soon as I can release the matching source code.</p>	<p>Termen "planning" avser att beskriva och informera om aktiviteter. Planer är utarbetade på individnivå efter eget intresse. En central plats för planering existerar inte vilket medför att det inte existerar en central plan. Planerna diskuteras och utvärderas i forum och mailinglistor. Alla medlemmar har rätt att uttala sig om projektets olika planeringar.</p> <p>I wanted to let you know that we are planning to implement the Shadow DOM specification (http://dvcs.w3.org/hg/webcomponents/raw-file/tip/spec/shadow/index.html) in WebKit.</p> <p>Also, Robert Helmer (rhelmer), a Mozilla contributor who is actively working on the Graph Server, told me that Mozilla is planning to replace the backend with a key-value database.</p> <p>But are you planning to run test builds against live websites?</p> <p>It's not really a practical option at this time. Perhaps in due course, CFURL could be changed to use WTFURL under the covers, or offer a mode to do so. But planning that sort of thing would be outside the scope of this mailing list.</p> <p>I am planning to document IDLs being used in WebKit: https://trac.webkit.org/wiki/WebKitIDL Before starting documenting, I would like to clean up the IDLs in WebKit. Here is a cleanup plan. There are 117 IDLs, and I am planning to remove 44 IDLs and rename 47 IDLs: https://docs.google.com/spreadsheet/ccc?key=0AlobCOyvTnPKdG1iaFJhNVZGMWI3eWZid0ZRvFdxU3c This is a meta bug: https://bugs.webkit.org/show_bug.cgi?id=77393 If you have any concern, please let me know. Comments are appreciated!</p>	<p>Termen "planning" avser att beskriva och informera om aktiviteter. Planer är utarbetade på individnivå efter eget intresse. En central plats för planering existerar inte vilket medför att det inte existerar en central plan. Planering är en aspekt i spelet. Planerna diskuteras och utvärderas i forum och mailinglistor. Alla medlemmar har rätt att uttala sig om projektets olika planeringar.</p> <p>I don't know whether there are plans to animate them soon.</p> <p>Mythos and I are planning to write a detailed guide that includes a step-by-step blender export section.</p> <p>This is a completely theoretical plan which I haven't done any code for at all. It is below a few other priorities so will be a while before I could do any work on it.</p> <p>This would be helpful for attack planning and defence i.e. build fortresses blocking every approach.</p> <p>That is a rough outline of what my AI does and what I plan to do to develop it further. Hopefully this will help people testing to see what is a bug and what is unimplemented and let the other AI people know what I am doing.</p>

Tabell 8 - Analysenbet "Resources"

Android	WebKit	OAD
<p>Termen "resources" används för att påvisa två aspekter; dels resurser i form av mjukvarutyp (system, funktioner och språkpaket); dels resurser i form av människor. Oftast är resurser i form av människor en anspelning på Googles resurser. Detta gäller också för mjukvara. Generellt är dock att termen "resources" används för att diskutera en mjukvaras resurser i form av andra mjukvaror och hur resurser ska användas och hittas.</p> <p>It would be much better, if there was a project manager for AOSP > contributions, whose main role was to aid communication between > (wannabe) contributors, and the actual Google developers, fight for > review resources with the internal subsystem leads/PMs ... etc. You > would be great for this position, but you seem to have too many roles > to count already in AOSP. :)</p> <p>So if you're interested in improving the accessibility of the platform, please consider submitting patches to add these accessibility hooks in the places where the default behavior isn't correct (or file bugs if you notice something broken and don't have the resources to fix it yourself).</p>	<p>Generellt är att termen "resources" används för att diskutera en mjukvaras resurser i form av andra mjukvaror och hur resurser ska användas och hittas.</p> <p>Everything is processed in a stateful manner. The behavior of a JavaScript file depends on what JavaScript files were included earlier. Decoding of most resources depends on what the referring page encoding is.</p> <p>I believe someone at one time implemented a Firefox extension that rewrote pages to capture resources and use local references.</p>	<p>Termen "resources" används främst för att benämna resurser i spelet (food, wood, gold etc). Generellt är att termen "resources" används för att diskutera en mjukvaras resurser i form av andra mjukvaror och hur resurser ska användas och hittas.</p> <p>There is a gray area between player given and unit AI given orders however like: Gather/return resources and some future commands 'auto explore' and 'attack move'.</p> <p>If the aim of stances is to make citizens stay alive a 'return resources if attacked' or 'auto garrison into a building with an attack if attacked' behavior would do better IMO.</p>

Tabell 9 - Analysenbet "Schedule"

Android	WebKit	OAD
<p>Termen "schedule" avser att schemalägga aktiviteter i en mjukvara, d v s schemalagda aktiviteter. Scheman avser också att belysa tidsaspekten av uppgifter men är i detta sammanhang kopplat till "release-schedule" (schema för när en uppdatering ska släppas). I enstaka fall framgår det att en aktivitet har utförts med ett bakomliggande schema men det är inte standard.</p> <p>Allow applications to force the Android platform to schedule GPS refreshes, instead of solely relying on the native code to properly report its GPS scheduling capabilities.</p> <p>The reasoning is that the SDK not a single component but several ones, each of them released and updated on different schedule.</p> <p>For all practical purposes, you shouldn't assume that those will ever get reviewed or merged. Based on the schedule, those changes were based on 2.3.5 or older versions, and I assume that most of them would be irrelevant anyway in the context of Google's master branch (the split between those happened almost 2 years ago).</p> <p>The development phase completed on schedule a while ago, but there's a very annoying bug that makes Gerrit impossible to deploy in public at the moment and that we can't manage to pinpoint.</p> <p>Of course, that option won't help us in the short term, but our project's deadline is still up for debate, and it might be possible to submit such a patch to Android in time for a release to line up with our schedule.</p>	<p>Termen "schedule" avser att belysa tidsaspekten av uppgifter men är i detta sammanhang kopplat till "release-schedule" (schema för när en uppdatering ska släppas).</p> <p>Once AppleWin is ready we'll schedule a date for the transition and announce it one this thread.</p>	<p>"Schedule" är individuellt, d v s kartlägger och strukturerar enstaka uppgifter som utvecklarna avgör. Schemaläggning sker på personlig nivå. "Schedule" avser att belysa tidsaspekten av uppgifter men är i detta sammanhang kopplat till "release-schedule" (schema för när en uppdatering ska släppas). "Schedule" avser att schemalägga aktiviteter i en mjukvara, d v s schemalagda aktiviteter.</p> <p>I have no schedule besides going to school, so it would be difficult to give an exact time-estimate. Though, I should be able to put-in however much time is required -- without issue.</p> <p>You will mostly create your own schedule. Working as a team means that everyone has a say in what goes into the project, and everyone has their piece of the puzzle.</p> <p>Yeah, I think it'll be necessary to make pathfinding asynchronous - a unit requests a path on one simulation turn, and doesn't get the result until the next turn (or even later), so the engine can schedule the computations to spread the load.</p>

Tabell 10 - Analysenbet "Time"

Android	WebKit	OAD
<p>Termen "time" diskuteras ofta och handlar t ex om tids-åtgång, om en uppgift är tidskrävande eller om en lösning är lönsam ur ett tidsperspektiv. "Time" kan också kopplas till diskussioner gällande servers tillstånd och hur länge dessa varit offline. Tid avser också att belysa aktiviteter inom systemet då Android är ett complex system och många aktiviteter är igång samtidigt (parallella aktiviteter vid samma tidpunkt). Man hänvisar till tid när man vill avgöra om en funktion i systemet är prestandakrävande, dvs om det tar lång tid att utföra funktionen.</p> <p>At this time, I think the only thing we would look into is downloading the SDK components on first start from our own repository.</p> <p>We have a reproducible crash that is happening when the concurrent gc is running at the same time as System.arraycopy, and are looking for feedback on how to address this, with the intention of contributing a fix.</p> <p>I apologize for the continued downtime of the AOSP Gerrit instance.</p> <p>I know the claims that with software "adding people to a late project only makes it later", but if there are only a few issues and the purpose is not writing new code, I think it has the potential to save time rather than wasting it.</p> <p>Is there any better time estimate after almost 4 months down ?</p>	<p>Termen "time" diskuteras ofta och handlar t ex om tids-åtgång, om en uppgift är tidskrävande eller om en lösning är lönsam ur ett tidsperspektiv. "Time" avser också generella diskussioner kring medlemmars engagemang ur ett tidsperspektiv.</p> <p>I remember last time this came up, there was some controversy, both within the WebKit community and among browser implementors more broadly. Kudos for writing a much more comprehensive spec and taking more of the feedback into account.</p> <p>If removing deprecatedFrameEncoding isn't possible at this time, we should revert http://trac.webkit.org/changeset/104723 because it causes crashes. Rather than get into a revert war, however, I believe the project would be better served by talking out this issue.</p> <p>I don't think that enough has changed to make spending time on this again very useful. The whole topic is quite isolated and inconsequential, so the intensity of argument (particularly in bug 67882 and its patch) does not appear adequate.</p> <p>All that being said, if you don't wish to comply with this standard at this time, that's your choice. I'm just asking for an ifdef so I can turn this non-standards compliant code off in the Chromium port.</p> <p>Anyway, thanks for taking the time to engage in this discussion. I'm going to write up a summary of the various points of view and propose a path forward.</p>	<p>Termen "time" diskuteras ofta och handlar t ex om tids-åtgång, om en uppgift är tidskrävande eller om en lösning är lönsam ur ett tidsperspektiv. "Time" är i vissa sammanhang kopplat till "runtime"; tiden då mjukvaran körs. "Time" är även kopplat till vad som utträttas, dvs att tiden avgör hur mycket människor kan engagera sig i utvecklingen. Tid är också en aspekt i spelet.</p> <p>OK, I see. It's a good developing way for saving much effort and time.</p> <p>We ought to support reinitialisation on Windows and Linux too, so we can change monitor resolution and antialiasing etc at runtime without restarting the whole engine, but they don't need reinitialisation when toggling fullscreen so it's not a problem there yet</p> <p>There are many other situations the units refuse to actually perform the players commands especially but not exclusively when time is of the essence like in battles.</p> <p>So if I have some OAD time, what should I do with it? I could go back and work on JuBot, but I guess it's probably been massively superseded by now (correct me if I'm wrong). Or I could do scenario design perhaps, or figure out how to write random map scripts, or something else entirely (as long as it stays out of the bits that need C-family coding).</p>

BILAGA 3

Här presenteras underlaget för resultatet. Tabellerna har tre kolumner; en kolumn Teori där teorin som förespråkas i kapitel 2 sammanfattas, en kolumn Analys där empirin från kapitel 4, med relevans för frågeställningen, presenteras, och en kolumn Teori vs Analys som avser jämföra teorin med empirin. För att genomföra en relevant analys utan upprepningar och med tydligare budskap har vissa av analysenheterna sammanförts, se tabellerna 2, 4 och 7.

Tabell 1 - Aktiviteter (*activity*)

Teori	Analys	Teori vs Analys
Projekt består av aktiviteter som utgörs av delmålen som kan övervakas utifrån kostnad och prestation. Organisationen bistår med resurser för att utarbeta aktiviteterna. Aktiviteter är för organisationen något som kan hanteras och planeras.	Aktiviteter relaterar till uppgifter som är relevanta för utvecklingarna i projektet.	Det existerar fall då aktiviteter avser att belysa vad som måste uträttas i utvecklingen. Att utarbeta genomförandet i aktiviteter är inte en genomgående process eller nödvändigt för utvecklingen. Inom OS-utveckling är det inte kritiskt att utifrån kostnad och prestation övervaka en aktivitet. Aktiviteter är inte heller utarbetade i detalj (dokumenterade eller totalt sätt utarbetade) som möjliggör vidare kontroll. Organisationen har ingen skyldighet att bistå med resurser för att genomföra en aktivitet. Den enda resurs som organisationen bistår med är kommunikations- och kodhanteringsverktyg samt resurser i form av andra mjukvaror.

Tabell 2 - Budget (*budget/cost*)

Teori	Analys	Teori vs Analys
<p>Budgeten avser att presentera projektet i numeriska termer. Den huvudsakliga uppgiften för budgeten är att analysera pengaflödet inom projektet, d v s belysa utgifter, använda mantimmar respektive maskintimmar. Budgeten måste även relatera utgifter och tid för att ge en ekonomisk syn av projektet i alla olika stadier. Budgeten ska spegla projektplanen. Budgeten ska utarbetas utifrån faktiska kostnader som mäts i valuta och som gör det möjligt att definiera projektet utifrån dessa utgångspunkter. Ett projekt har utgifter i form av mantimmar och maskintimmar.</p>	<p>Ibland diskuteras "cost" i samband med utvärdering av införskaffning av hårdvara och kostnaden för användning av andra plattformar.</p>	<p>Budget som något formellt verktyg för att utarbeta en projektplan är inte nödvändigt inom OS och används inte som standard. Mängden ekonomiska faktorer inom OS är liten. Om det existerar utgifter av olika slag är dessa inte diskuterade eller relevanta för utvecklingen, d v s de belyses inte i form av budgetering. Behovet att kartlägga mantimmar är inte heller relevant då det inom OS i flesta fall inte existerar ekonomisk kompensation.</p>

Tabell 3 - Dokumentation (*documentation*)

Teori	Analys	Teori vs Analys
<p>Dokument ligger till grund för projektets utförande, övervakning, kontroll och avslutning. Projektdokumentationen utgör en fullständig bild av projektets omfång, tid, kostnad, kvalitet, kommunikation, risk och anskaffning. Dokumentationen utgör kärnan i projektet då alla aspekter som hanteras vid planering utförligt dokumenteras.</p>	<p>Generellt är dokumentation (documentation) utspridd över olika <i>communities</i> och inte indexerade utifrån standard. Utan bestämmelser för dokumentation kan det uppstå svårigheter att hitta relevant information. Dokumentation sker på eget initiativ men efterfrågas ibland.</p>	<p>Vikten av att sammanställa alla detaljer av projektet i form av dokumentation är inte prioriterat inom OS. Dokumentation är av intresse och de flesta medlemmar är medvetna om dokumentationens betydelse, men det existerar inga bestämmelser som klargör hur dokumentering av projektet ska utföras. Dokumentering sker på eget initiativ. Detaljerad dokumentation som schemalägger, eller beskriver, mål saknas.</p>

Tabell 4 - Mål (*objective/goal*)

Teori	Analys	Teori vs Analys
<p>Projektets mål är den viktigaste aspekten att ”mäta” vid utförande av planering. Projektmålen kan vara breda eller specifika. Mål är oftast utarbetade i mindre delmål. Mål beskrivs utifrån termer som kostnad, schema och teknik. Mål är grunden för projektets olika framtida aktiviteter.</p>	<p>Ett projekt har flera mål (goal/objective). Målen diskuteras inom gruppen för att klargöra om de är relevanta att utföra. Mål kan både syfta till projektets mål och mjukvarans mål samt funktionen som mjukvaran ska uppfylla. Generellt är målen individuella och definierade av varje medlem.</p>	<p>Mål är oftast inte utarbetade innan projektstart. Mål utarbetas vid behov och är upprättade av enstaka utvecklare. Mål dokumenteras och diskuteras på en nivå där de är möjliga att påbörjas. En fullständig plan för att hantera målet fullt ut redovisas ej.</p>

Tabell 5 - Planering (*planning*)

Teori	Analys	Teori vs Analys
<p>Planering avser att beskriva utförandet av händelser och klargöra hur mänskliga och icke-mänskliga resurser ska hanteras för att genomföra aktiviteter och uppnå mål. Planering syftar till att beskriva projektet så att det är identifierbart för varje projektmedlem, d v s ge en fullständig bild av projektets omfång, tid, kostnad, kvalitet, kommunikation, risk och anskaffning.</p>	<p>Planering sker utifrån en utvecklarens perspektiv, d v s en plan innefattar en eller flera utvecklare som individuellt bestämmer sin planering. Planering av framtida aktiviteter, liksom planering av datum, utarbetas utifrån diskussioner. En plan blir realitet först efter diskussion och utvärdering i grupp. Alla medlemmar har rätt att uttala sig om projektets olika planeringar. Planering avser att beskriva och informera om utveckling.</p>	<p>Planeringen sker inte på ett övergripande plan och en central plan för projektet existerar inte. En fullständig bild av projektet ges inte. Planering sker efter eget initiativ och speglar bara den egna utvecklingen. Planer är flexibla och inte avgörande för ett lyckat projektutförande. Planeringen utarbetas i diskussioner och planering inträffar när som under projektets gång.</p>

Tabell 6 - Resurser (*resources*)

Teori	Analys	Teori vs Analys
<p>Projektets resurser avser att bidra till projektet så att målen kan uppnås. Resurser kan vara i form av fysiska resurser, mänskliga resurser och/eller tjänster. Organisationen bistår med resurserna.</p>	<p>Resurser (resources) är inte relaterat till projektets resurser i OS-utvecklingssammanhang. Resurser speglade inte projektets resurser när temat diskuterades i grupp. Resurser beskriver mjukvarans resurser, d v s mjukvaran har resurser i form av andra mjukvaror.</p>	<p>Resurser är inte avgörande för projektets utförande. Med flexibiliteten att inskaffa resurser eller att lösa uppgifter med motsvarande resurser står sig planering inom OS friare i förhållande till traditionell utveckling. Att säkra resurser är inte heller avgörande eller nödvändigt för att ett OS-projekt ska lyckas. Resurser är oftast i form av tjänster och andra mjukvaror men dessa behöver inte planeras eller utarbetas så som teorin förespråkar.</p>

Tabell 7 - Schema (*schedule/time/date*)

Teori	Analys	Teori vs Analys
<p>Aktiviteter och resurser måste synkroniseras för att projektmålen ska kunna genomföras. Det är också en viktig aspekt då schemaläggning möjliggör kontroll samt övervakning av prestation och kostnader. Projektet måste utföras på ett bestämt manér i förhållande till tid. Det är viktigt att veta när aktiviteter ska startas och avslutas och när resurser är tillgängliga.</p>	<p>Det framgår att schema avser att bestämma datum för framför allt “release-date”.</p>	<p>Att utforma scheman inom OS existerar, t ex är datum (“release-date”) en viktig faktor vid koordination av arbetet. Scheman utformas inte för att övervaka projektet utan är utarbetade på individnivå och valfria att använda. Vikten av att projektets aktiviteter avslutas och påbörjas enligt scheman är inte avgörande inom ett OS-projekt. Det är inte viktigt att ha ett schema som koordinerar alla aktiviteter i projektet.</p>