



PROCESSRAPPORT

Deviations and Reporting

FORORD

Vi ønsker å takke vår veileder Simen Hasselknippe for veldig god veiledning gjennom hele prosjektet, resultatet hadde umulig blitt så bra uten hans innspill og veiledning. Vi ønsker også å rette en stor takk til CO2focus for tillitten de har gitt oss og for tiden de har avsett til oss. Vi ønsker spesielt å takke Jan-Willy og Per-Otto for å finne oppgaven til oss og god kommunikasjon.

I denne rapporten vil vi beskrive prosessen bak utviklingen av CEMAsys mobil applikasjon for iOS og Android. Noen kapitler forutser at leseren er kjent med ulike teknologier og tekniske aspekter.

Vi beskriver hvordan arbeidsprosessen har vært gjennom hele prosjektet for vår gruppe. Den består av fire kapitler i følgende rekkefølge.

- **Planlegging og metode:** Forteller om planleggings prosessen. Kommunikasjon med oppdragsgiver, arbeidsfordeling og prosjektstyring. En oversikt over hvilket verktøy som ble brukt og hvorfor. Hvordan samarbeidet i gruppa har vært igjennom hele prosjektet.
- **Utviklingsprosessen:** Forteller om utviklingsfasene prosjektet har hatt, og hvordan vi har reflektert over de faglige utfordringene. Valg vi har tatt i forhold til funksjon i programmet. Hvordan forholdet til oppdragsgiver har utviklet seg under prosessen.
- **Kravspesifikasjon og dens rolle:** Hvilket endringer kravspesifikasjonen har hatt underveis. Hvordan kravspesifikasjonen har vært viktig for utvikling av design og implementering. Hvordan kravspesifikasjonen samsvarer med produktet som beskrives i produkt-dokumentasjonen.
- **Avsluttende del:** Har forteller vi om eget utbytte av prosessen og hva vi ville gjort annerledes hvis vi skulle begynt på nytt. I tillegg til en oppsummering og konklusjon med arbeidsgivers tanker om hvordan produktet tas i bruk.

Den som skal vurdere rapporten, bør legge spesielt fokus på utviklingsprosessen. Det er her bakgrunnen og underlaget for produktet vi har laget kommer fram.

INNHOLDSLISTE

FORORD	1
1.1 Gruppen	4
2 PLANLEGGING	5
2.1 FORLØPER TIL PROSJEKTSTART	6
3 ARBEIDSTEKNIKKER OG UTVIKLINGSMETODER	7
3.1 SMIDIG UTVIKLING	7
3.1.1 Scrum	7
3.2 KOMMUNIKASJON	7
3.2.1 Gruppen	7
3.2.2 Oppdragsgiver	7
3.2.3 Veileder	7
3.3 PROSJEKTSTYRINGSDOKUMENTER	8
3.3.1 Arbeidsplan	8
3.3.2 Fremdriftsplan.....	8
3.4 ARBEIDSFORHOLD	8
3.4.1 HiOA	8
3.4.2 Hjemmekontor	8
3.4.3 CO2focus	8
4 BENYTTETE TEKNOLOGIER OG VERKTØY	9
4.1 Git med Github	9
4.2 Prosjekthåndtering	9
4.2.1 Trello	9
4.3 Dokumentasjon	10
4.3.1 Word online	10
4.3.2 One Drive	10
4.3.3 Dropbox.....	10
4.4 Native eller hybrid utvikling	10
4.5 Utviklingsverktøy	11
4.5.1 Xcode.....	11
4.5.2 Eclipse	11
5 OM UTVIKLINGSPROSESSEN	12
6 KRAVSPESIFIKASJON OG DENS ROLLE	14

7 AVSLUTTENDE DEL	16
REFERANSELISTE	18

1 INNLEDNING

CO2focus er blant verdenslederen når det kommer å levere systemer for regnskapsføring av miljøavtrykk. Deres produkt CEMAsys, et rapportering og analyseverktøy brukes av mange store bedrifter i Norge og selges også til aktører i utlandet. For å bli attraktive for enda flere kunder og øke tilfredsheten til eksisterende kunder ønsker CO2focus å utvide CEMAsys med en mobilapplikasjon. Applikasjonen skal kunne brukes av alle til å sende avvik til CEMAsys, avvikene som sendes vil være både av miljø og helse, miljø og sikkerhet type avvik. Målet med applikasjonen er at flere avvik blir rapportert slik at bedriftene som bruker CEMAsys lettere kan bedre arbeidsforholdene for de ansatte i bedriften og redusere sitt totale klimaavtrykk.

1.1 Gruppen

Gruppen bestod av Odd Fredrik Holen fra dataingeniørstudent, og informasjonsteknologi studentene, Jaune Reinola og Kim Robert Bravie. Jaune og Kim startet samarbeidet allerede i andre klasse i faget applikasjonsutvikling, mens Odd Fredrik kom inn i webapplikasjoner i følgende høstsemesteret. Gruppen har lært hverandre og kjenne og vet hvilket kvaliteter hver enkelt har og bidra med.

2 PLANLEGGING

Å planlegge utviklingen av applikasjonene var en forholdsvis lang prosess. Det ble diskutert mye om vi skulle utvikle kun en Android applikasjon, som alle på gruppen var kjent med å utvikle. Eller om vi også skulle lage en iPhone basert iOS applikasjon. Vi endte med å utvikle til både Android og iOS.

Det var også diskusjon om hvordan vi skulle utvikle applikasjonene. Vi begynte i utgangspunktet med "fossefall"-metoden, men fant fort at arbeidsgiver ikke var hundre prosent sikker på akkurat hva han ønsket. Dette førte til at vi ble enige om å basere utviklingsmetoden vår på scrum modellen. Vi ble enige om fast møtetid hver mandag. Møtene bestod av tre deler. Den første delen var å vise fram hva som var gjort siden sist, og hvorfor. Den andre var å diskutere aktuelle ting i forhold til utviklingen, hovedsakelig ønsker og presiseringer fra arbeidsgiver og problemer i forhold til utviklingen. Den siste delen var planlegging av hva som skulle gjøres til neste uke.

Grunnet at vi alle hadde valgt forskjellige fag og har jobb ved siden av skolen var det vanskelig å møtes oftere enn en gang i uken selv om dette ofte var ønskelig. For å ha bedre oversikt over framgangen valgte vi derfor å bruke Github, både til kodedeling og å holde orden på oppgaver og milepæler. Vi bestemte oss også for å teste Trello for oppgaver og milepæler, og byttet etter en ukes testperiode over fra Github.

Til utvikling av Android applikasjonen ble Eclipse (Android-bundel versjonen til Google) benyttet da vi var godt kjent med denne editoren og at den fungerer både på Windows baserte maskiner og Mac OS X.

Utvikling av iOS applikasjonen foregikk i Xcode. Som er det eneste utviklingsverktøyet for Apples Cocoa Touch utvidelse av Objective-c som kreves for å utvikle iOS. Xcode finnes pr. dags dato kun til Mac. Dette gjorde at kun en fra gruppen hadde mulighet til å kode til iPhone. Etersom vi ikke hadde kjennskap til iOS utvikling fra før av måtte dette læres fra grunn av i utviklingsfasen.

Vi fikk i de to første månedene av prosjektet mange tilbakemeldinger fra arbeidsgiver både angående design og funksjonalitet i applikasjonene. Dette innebar både at applikasjonen ble mer omfattende og tidkrevende å utvikle. I starten av prosessen var det snakk om å levere en ren HMS-avviksrapporterings applikasjon. I starten av mars ble vi enige om at vi heller

lagde en applikasjon for å rapportere både HMS og miljøavvik. Hovedgrunnen til at vi gikk med på dette var at vi så at applikasjonen ville bli mer brukt og at vi følte vi leverte et bedre produkt.

2.1 FORLØPER TIL PROSJEKTSTART

Det første gruppemøtet fant sted tidlig i oktober 2012. Møtets hensikt var å få kartlagt tanker og ideer, hvilke teknologier vi ønsket å benytte og hva vi generelt så for oss som aktuelt å jobbe med. Vi var alle enige om at mobil applikasjons utvikling hadde vært spennende. Vi tok kontakt med ulike bedrifter vi kunne tenke oss å jobbe for på telefon og fikk beskjed om å sende søknad og CV på e-post. Så ble vi kalt inn til intervju. Intervju prosessen var både spennende og lærerikt. Vi var på to andre intervjuer før vi kom i kontakt med CO2focus.

Første møte med vår kontaktperson og CO2focus fant sted 28.11.2013 i firmaets lokaler i Hegdehaugsveien. Da fikk vi vite mer om bedriften og deres tjenester. Vi fikk velge mellom et sett av oppgaver tilpasset vårt kompetanse nivå. Vi valgte prosjektet for enkel og effektiv rapportering av HMS avvik til bedriftens *cloud* system.

3 ARBEIDSTEKNIKKER OG UTVIKLINGSMETODER

3.1 SMIDIG UTVIKLING

Vi valgte en smidig utviklings form fordi vi forutså at prosjektet vårt vill gjennomgå hyppige oppdateringer og endringer av kravspesifikasjon.

3.1.1 Scrum

Er et rammeverk laget med henblikk på å utvikle komplekse informasjonssystemer. Det brukes i hovedsak til å utvikle programvarebaserte systemer og benyttes gjerne i kombinasjon med Extreme Programming (XP), men kan i prinsippet brukes til å håndtere alle slags prosjekter av en viss kompleksitet. Teorien er basert på empirisk prosesskontroll og fordrer at man jobber inkrementelt og iterativt og at selve utviklingsjobben utføres av tverrfaglige, selvstyrte team.

3.2 KOMMUNIKASJON

3.2.1 Gruppen

Vi ble enige om fast møtetid hver mandag. Grunnet at vi alle hadde valgt forskjellige fag og har jobb ved siden av skolen var det vanskelig å møtes oftere enn en gang i uken selv om dette ofte var ønskelig. Det var naturlig for gruppa og ta i bruk sosiale medier for å opprettholde daglig kommunikasjon innad i tillegg til telefonsamtaler og SMSer.

3.2.2 Oppdragsgiver

Vi hadde god kommunikasjon med CO2Focus igjennom hele prosjektet. Vi fikk instruksjoner angående design og funksjonalitet i applikasjonene som var i stadig endring i starten av prosjektet. Vi møttes ved flere anledninger i firmaets kontor lokaler i Hegdehaugsveien fem minutters gange fra Pilestredet 35 for oppdatering og diskusjon om videreutvikling.

3.2.3 Veileder

Simen Hasselknippe var vår veileder i prosjektet. Vi har hatt god kontakt med Simen og fått konstruktive og gode råd igjennom hele prosessen. Han hjalp oss også med å sette oss milepæler som var viktig for jevn flyt i prosjektutviklingen. I tillegg fikk vi råd om hvordan vi burde prioritere i forhold til programmering og dokumentasjon, samt konkrete forslag til hvilke dokumenter vi burde ha med i dokumentasjonen.

3.3 PROSJEKTSTYRINGSKORT

3.3.1 Arbeidsplan

Det ble tidlig skrevet en arbeidsplan med delmål og milepæler for prosjektet. Denne ble ikke arbeidet videre med, etter vi begynte å jobbe i scrum. Milepælene har vært fastsatt fra starten av, mens arbeidsoppgavene var mer flytende etter vurdering av fremdriften i prosjektet.

3.3.2 Fremdriftsplan

For å ha bedre oversikt over framgangen valgte vi å bruke Github, både til kodedeling og å holde orden på oppgaver og milepæler. Vi bestemte oss også for å teste Trello for oppgaver og milepæler, og byttet etter en ukes testperiode over fra Github. Fremdriftsplanen ga oss en oversikt over hvor vi burde være på hvilket tidspunkt og ble kontinuerlig justert for endringer, og var et nyttig verktøy i både planleggingen og utføringen av prosjektet.

3.4 ARBEIDSFORHOLD

3.4.1 HiOA

Ukentlige møter på HIOA foregikk som regel i kantina eller på datatorget i 4.etg. i Pilestredet 35. Etter møtene fortsatte arbeidet her.

3.4.2 Hjemmekontor

Grunnet at vi alle hadde valgt forskjellige fag og har jobb ved siden av skolen var det vanskelig å møtes oftere enn en gang i uken selv om dette ofte var ønskelig. Vi har som regel hatt hjemmekontor og jobbet selvstendig hver for oss med kommunikasjon via prosjektgruppa vår og Skype.

3.4.3 CO2focus

Vi var på samtlige møter med arbeidsgiver under utviklingen av design av funksjonalitet i applikasjonen. Vi har gjort små oppdatering i prosjektet mens vi har vært der.

4 BENYTTETE TEKNOLOGIER OG VERKTØY

4.1 Git med Github

Ettersom vi var flere personer som utviklet applikasjonene og ønsket tilgang til å se og redigere filer samtidig var det nødvendig med godt versjonskontrollverktøy. Vi valgte Git fordi vi hadde erfaring med dette fra tidligere og visste at dette ville dekke våre behov. Når en fil oppdateres eller forandres, slettes ikke den gamle versjonen, men blir lagret i en database som inneholder tidligere versjoner av filene. Man kan hente frem gamle versjoner, og se på forskjellen mellom versjonene. Git tillater også at flere brukere kan jobbe på filene samtidig. For hver gang man har implementert noe nytt, rettet *bugs* eller gjort andre ting man ønsker å ha en versjon av utfører man et «commit». Dette gjør at du har muligheten til å rulle tilbake prosjektet til den versjonen. Dette fungerer uavhengig om man har internett tilgang eller ikke. For at vi alle skulle få tilgang på filene var det også nødvendig å laste opp prosjektet til en server. Vi valgte å bruke Github til dette, se mer under.

Git kan installeres og brukes gratis på alle servere med internett tilgang. Ettersom at vi ønsket litt mer en standard server med Git valgte vi å bruke Github. Github er gratis for alle, men tar betalt dersom man ønsker private «repositories», forskjellen på disse og vanlige er at ingen andre enn de du har gitt tilgang kan se verken repository'et, eller filene i det. Både Android og iOS applikasjonene våre ble utviklet i private reposetories etter ønske fra oppdragsgiver. En av hovedgrunnene til at vi ønsket å bruke Github er at det gir mulighet til å se og redigere kode online, samt at web-grensesnittet tilbyr mye god statistikk om utviklingen av prosjektet. Det er blant annet mulig å se hvor mange *commits* hver person er utført samt antall kodelinjer.

Videre tilbyr Github også prosjektstyringsverktøy, med «issues» og «milestones». Dette var også en veldig viktig grunn til at vi valgte Github, men vi valgte senere å bytte til Trello.

4.2 Prosjekthåndtering

For å holde styr på oppgaver som skal gjøres, hvem som gjør hva og hva som har blitt gjort i et prosjekt. Er det viktig meg god prosjekthåndtering. Da kan det være greit og ta i bruk et prosjekt håndterings verktøy. Vi valgte å bruke Trello for å håndtere vårt prosjekt.

4.2.1 Trello

Trello er et prosjekthåndterings verktøy utviklet med tanke på Toyotas samlebånd produksjon kalt Kanban. Verktøyet fungerer fint i andre smidig utviklingsformer som for

eksempel scrum. Brukergrensesnittet i Trello består av et brett med flere lister som inneholder mange kort. I denne sammenhengen representerer brettet selve prosjektet, hver liste en tilstand og hvert kort oppgaver som har oppnådd denne tilstanden. Alle prosjektdeltakere kan legges til brettet og på den måten erklære sin avhengighet og sitt ansvar til arbeidsoppgaver i de forskjellige listene. Det er også mulighet for å kommentere på kort, sette opp sjekklister, legge ved filer med mer. Siden Trello er en nettbasert tjeneste oppdateres prosjektfremgangen i sanntid.

4.3 Dokumentasjon

4.3.1 Word online

En lettere versjon av Microsoft Word. Brukeren kan lese eller redigere dokumenter i nettleseren. Endringer i et dokument lagres automatisk med brukerens OneDrive konto.

4.3.2 One Drive

Microsoft sin skylagringstjeneste. Gir brukeren muligheten til å laste opp og synkronisere filer i nettskyen, hente filer ved hjelp av nettleser eller på en lokal enhet. Brukeren kan lagre filer, dele dem med kontakter eller gjøre dem offentlig. Offentlige filer kan leses uten Microsoft konto.

4.3.3 Dropbox

Dropbox tilbyr skybasert lagring gratis til alle brukere. Gratisabonnementet har begrenset lagringsplass, men er tilstrekkelig for enkle tekst og regnearkfiler med mer. En av de store fordelene til Dropbox er at den forenkler prosessen med å laste opp filer ved at man installerer en klient på sin maskin. Denne klienten oppretter en ny mappe der brukeren måtte ønske og synkroniserer denne automatisk. I forhold til andre konkurrerende tjenester syntes vi Dropbox håndterte deling av mapper langt bedre, derfor falt valget naturlig. Dropbox sin klient finnes til alle de vanligste plattformene som innebærer Android, iOS, Linux, Mac OS X og Windows.

Dropbox tar også automatisk backup av alle filene som ligger i mappen og det er mulig å gjenopprette slettede filer og tilbake stille til tidligere tidspunkt. Vi har i utgangspunktet brukt Dropbox til deling av filer, som bilder, små videosnutter og dokumenter.

4.4 Native eller hybrid utvikling

Når vi skulle utvikle prosjektet stod vi med to valg. Det ene var å utvikle en hybrid applikasjon, den andre var å utvikle to native applikasjoner uavhengig av hverandre.

Hybride applikasjoner kan utvikles ved hjelp av forskjellige utviklerverktøy, f. eks. Visual Studio eller Xamarin studio.

Native utvikling vil si å utvikle plattform spesifikt. For Android vil dette si Java, mens for iOS er det Objective-c.

Hybrid applikasjoner har mange fordeler. Det er lett å tilpasse forskjellige operativsystemer, bruker kjente teknologier som html5 og javascript (Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options, 2014). Det er også mulig å oppdatere, i alle fall deler av applikasjonen uten å gå gjennom Apple sin AppStore eller Google sin Play Store.

Ulempen med hybrid utvikling er at det er mer krevende å designe applikasjonen, og at det i forhold til Native utvikling gir dårligere ytelse. Det er også vanskeligere å bruke funksjoner bygd inn i operativsystemet (Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options, 2014).

Vi valgte å utvikle native fordi veldig mye av utviklingsprosessen vår ville være designbasert og vi viste vi ville trenge å ha tilgang til kamera.

4.5 Utviklingsverktøy

4.5.1 Xcode

Utvikling av iOS applikasjonen foregikk i Xcode. Som er det eneste utviklingsverktøyet for Apples Cocoa Touch utvidelse av Objective-c som kreves for å utvikle iOS. Xcode finnes pr. dags dato kun til Mac.

4.5.2 Eclipse

Til utvikling av Android applikasjonen ble Eclipse (Android-bundel versjonen til Google) benyttet da vi var godt kjent med denne editoren og at den fungerer både på Windows baserte maskiner og Mac OS X.

5 OM UTVIKLINGSPROSESSEN

Vi fikk i de to første månedene av prosjektet mange tilbakemeldinger fra arbeidsgiver både angående design og funksjonalitet i applikasjonene. Noe som gjorde at applikasjonen ble mer tidkrevende å utvikle. I starten av prosessen var det snakk om å levere en ren HMS-avviksrapporterings applikasjon, men i starten av mars ble vi enige om at vi heller lagde en applikasjon for å rapportere både HMS og miljøavvik. Hovedgrunnen til at vi gikk med på dette var at vi så at applikasjonen ville bli mer brukt og at vi følte vi leverte et bedre produkt.

CO2focus hadde ikke hatt erfaring med mobilapplikasjoner før, i tillegg hadde de lite datateknisk kunnskap. Vi hadde lite erfaring med styring av prosjektet som ikke hadde klare krav og forventinger (skoleoppgaver har alltid oppgaveteksten som forteller hva som skal gjøres). Med dette prosjektet måtte vi finne ut arbeidsoppgavene våre underveis. Vi synes det var utfordrende men samtidig veldig spennende å oppleve hvordan oppdragsgiveren stolte på oss og spurte om å finne beste løsningen for dem uten at de visste hva det var selv. Vi er glade for ha bevist dem at vi er dyktige og fått kundens tillit.

Det var utfordrende å finne den beste løsningen på utseende av produktet. Oppdragsgiveren hadde nye ønsker og nye ideer om hvordan det skulle se ut hver gang vi hadde møte med dem. Akkurat der og da følte vi at det var mye frem og tilbake, men nå ser vi at det var verdt det: CO2focus er veldig fornøyd med utseende.

Frem til begynnelsen av april var vi sikre på at vi skulle få CEMAsys sitt API og vi kunne begynne å teste applikasjonen «live». Vi var skuffet da vi fant ut at dette ikke skulle være mulig likevel. Det var en stor utfordring for oss å komme på «plan B», siden det var helt uaktuelt å ikke få testet og demonstrert applikasjonene til oppdragsgiveren. Vi bestemte oss for å lage en «test CEMAsys»: en test-database og API. Sånn fikk applikasjonene et sted å hente data fra og lagre data i (mer om test-databasen og testAPI i produktrapporten). Vi var veldig fornøyd med resultatet.

Det var også mange utfordringer i forhold til selve kodingen. I Android applikasjonen slet vi veldig med å få bildestien til valgt bilde til å vises i tekstboks. Med etter mange nettsøk og gjennomgang av stort antall stackoverflow spørsmål og svar, fikk vi det til. Svaret var enkelt, vi måtte bare finne oss frem til det. Det var heller ikke lett å få SharedPreferences - som har viktig oppgave å huske og lagre brukernavn og passord sikkert - til å fungere i Android

applikasjon slik det skulle. Vi endte opp med å lage privat lytterklasse for SharedPreferences som fungerte som vi ville ha den (man kan lese mer om SharedPreferences i produkt rapporten).

Det tok lang tid både i iOS og Android applikasjonen å kode den delen av «DAL» som laster opp det bildet bruker legger ved rapporten. Det tok noen uker, men vi klarte å løse også det problemet. I iOS applikasjonen hadde vi også litt problemer med å hente data asynkront, da det er delegat metodene som mottar data og ikke metoden man kaller. Dette ga oss en utfordring i forhold til hvordan vi skulle returnere data til brukerobjektet som kaller metoden for å sende forespørselen. Vi løste dette ved å kreve brukerobjektet i konstruktøren av klassen, på denne måten kunne metoden som kalles når tilkoblingen til server er fullført skrive til brukeren. Det skal nevnes at dette ga et nytt problem fordi begge klassene importerte hverandre, noe som gjorde det umulig å bygge prosjektet (Build-dependency error) vi løste dette ved å heller ha en «class»-tag i .h filen og senere importere brukerobjektet i .m filen.

6 KRAVSPESIFIKASJON OG DENS ROLLE

Den første kravspesifikasjonen besto av et enkelt oppsett av punkter oppdragsgiveren ville ha med i applikasjonen etter første møte med oss. Denne ble utviklet i samarbeid med Jan Willy (vår kontaktperson i CO2focus). Neste møte var med Jan Willy og Per Otto, en kollega av Jan Willy som tilhører fagavdelingen for avviksrapportering, etter dette møte kom det fram betydelige forandringer og utdypning i kravspesifikasjonen.

Den største endringen var at vi ikke skulle lage webapplikasjon for å styre HMS-avvikene, kun mobilapplikasjoner. Vi diskuterte også mye frem og tilbake om hvordan layout og utseende skulle være. Det tok et par møter før det ble avklart akkurat hvilke tekstbokser applikasjonene skulle ha, hvilken data vi trengte å hente fra CEMAsys og hvordan er den dataen knyttet til hverandre (f. eks. ansvarlige personer er knyttet til både kategorier og enheter) for at vi skulle kunne lage riktig kode. Det sto i kravspesifikasjonen at kategorier kommer fra CEMAsys, men det sto ingenting mer om CEMAsys om hva slags databasedesign som er implementert der. Etter februar hadde vi et veldig godt bilde av hvordan applikasjonen skulle se ut og hva vi trengte av funksjonaliteter vi trengte. På siste møtet, kom det noen siste instruksjoner og forslag om forbedring av layoutet og utseende. Uten at dette forandret hvilke tekstfelter, men heller organisering og fonter.

Under utviklingen fant vi ut at en del funksjonalitet som kravspesifikasjonen forventet av oss å implementere, ikke var fornuftig å ha i mobilapplikasjoner likevel. Vi skjønnte at en «mobilapp» trenger ikke å kunne sende en PDF-fil av innsendte rapporten. Siden avviket skal bli lagret i CEMAsys, virket det mer fornuftig at CEMAsys lager PDF-filen og sender den på e-post til både avvikets avsender (hvis senderen har huket av at han/hun ønsker å få rapporten på e-post) og personen ansvarlig for kombinasjonen av kategori og sted. En annen funksjon, som vi ikke implementerte, er lagring av avvik i applikasjonen når mobile enheten mangler internett-tilkobling. Det hadde vært veldig fint å ha den med, men på grunn av tidspress valgte vi å droppe det, og heller ha feilmeldinger om manglende internett-tilkobling istedenfor. I sluttproduktet, når enheten mangler internett, får bruker beskjed om mislykket innlogging eller innsending av rapport, og brukeren bes om å fikse koblingen og/eller prøve igjen senere, som var akseptabelt for oppdragsgiveren.

Bortsett fra de to funksjonene som ble nevnt ovenfor, klarte vi å implementere alt som sto i endelige kravspesifikasjonen (for endelige kravspesifikasjonen, se vedlegg). Gruppen vår og

CO2focus hadde avtale på allerede første møtet at vi skal lage detaljert og gjennomtenkt kravspesifikasjon som vi kan basere hele utviklingen vår på. Og det klarte vi - vi avviket veldig lite kravspesifikasjonen og oppdragsgiveren var veldig fornøyd med produktet.

7 AVSLUTTENDE DEL

Vi har i løpet av oppgaveprosessen blitt mange erfaringer rikere og lært mye om utvikling. Vi startet på prosjektet med uten erfaring med å løse faktisk arbeidsoppgaver. Vi hadde tidligere løst mange skoleoppgaver med gitt problemstilling og krav. Men å utvikle for en skikkelig arbeidsgiver var både nytt og spennende for oss. Det ga oss også en del ekstra utfordringer, men alt i alt er vi veldig takknemlig for hver og en av utfordringene som vi har fått og løst.

I det vi startet utviklingen av prosjektet hadde vi heller ingen erfaring med Objective-c, eller andre aspekter ved utvikling av iOS applikasjoner. Det ble en bratt læringskurve vi føler vi har lært mye av.

Vi er veldig fornøyd med hovedoppgaven vår, men det er en del ting vi hadde gjort annerledes nå, når produktet er ferdig og alt ser mye enklere ut enn det gjorde under utviklingen. For eksempel burde vi begynt å spørre etter CEMAsys API allerede i begynnelsen av prosjektet, men vi var uerfarne og antok/stolte litt for mye på kunden som regnet med at et API allerede eksisterte, og vi kunne få det den dagen vi trengte å ta det i bruk. Nå vet vi at man må bekrefte antakelsene sine først og be kunden dobbeltsjekke alt de ikke er hundre prosent sikre på. Heldigvis klarte vi å finne en annen løsning og manglende API ble en lærerik lekse for oss, både hva vi burde gjort og hva vi fikk til som «test CEMAsys».

En annen ting vi hadde gjort annerledes, hvis vi skulle gjort alt dette på nytt, er å skaffe oss en teknisk veileder som vi kunne kommet til med våre kodeutfordringene. Vi klarte oss uten denne gangen - internett var vår viktigste lærer, men vi føler at vi hadde rukket å implementere mer funksjonalitet til applikasjonene dersom vi hadde hatt en teknisk veileder.

Nå på etterkant tenker vi at vi skulle kanskje prøvd å få til også Windows Phone applikasjon, siden vi er godt kjent med C# (programmeringsspråket til Microsoft/Windows Phone). Det hadde vært helt perfekt for oppdragsgiveren å få applikasjonen deres for alle de tre største plattformene. Målet med en sånn HMS-avviksapplikasjon var jo at flest mulig ansatte hos CO2focus sine kunder skulle begynne å sende inn avvik, og bedrifter skulle få bedre oversikt over problemområder i arbeidsmiljøet. Hvis denne applikasjonen hadde vært tilgjengelig i alle telefoner, kunne enda flere ansatte registrert avvik. Det kan hende at CO2focus skal nettopp gjøre dette om en liten stund – utvikle samme applikasjon også for Windows Phone - når de har fått testet populariteten til Android og iOS applikasjonen.

CO2focus kommer til å fokusere på å utvikle API til CEMAsys i sommer for at de skal kunne ta i bruk applikasjonene vi har utviklet for dem. Da vi var på siste møtet hos oppdragsgiveren og demonstrerte applikasjonene, var de veldig spent på å vise dem til kundene side. De må også utvikle tilleggs funksjonalitet for CEMAsys for å håndtere mobilapplikasjon-brukere. Når det er gjort, APIet til

CEMA sys er ferdig og applikasjonene tilpasset til det, er HMS- og miljørapporteringsapplikasjonen vår klar til å bli lansert.

CO2focus var veldig fornøyd med vårt innsats. Vi fikk mye ros og en flott attest (se vedlegg) på at de satte veldig mye pris på vårt arbeid. Vi som gruppe har lært mye om team-arbeid, samarbeid med kunde, planlegging og koding, dvs. blitt (litt) mer modne utviklere enn vi var for et halvt år siden. Vi gleder oss til mange sånne opplevelser i arbeidslivet til høsten.

REFERANSELISTE

Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options.

(2014, 21. mai). Hentet fra Salesforce:

https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options