

Slutrapport Tieto

Sammanfattning

Tieto är ett finsk-svenskt IT-konsultföretag. Företaget arbetar med konsultverksamhet och utveckling och har totalt drygt 14 000 anställda. Vi i grupp 6 fick i uppgift att bygga en prototyp för en ny typ av e-handel åt en av Tietos kunder. I den här rapporten så beskrivs arbetsprocessen inom gruppen, tekniker vi jobbat med, slutresultatet samt en del funderingar som uppkommit under projektets gång.

Innehållsförteckning

Sammanfattning	1
Innehållsförteckning	2
Inledning/bakgrund	2
Syfte och mål	3
Projektorganisation	3
Genomförande: metodik, teknik	4
Utvecklingsmetodik	4
Kommunikation	4
Versionshantering	5
Teknik	5
Utvecklingsmiljö	5
Utvecklingstekniker	5
Resultatbeskrivning/måluppfyllelse	6
Avvikelser/efterkalkyl	6
Slutsats	6
Förslag på vidareutveckling	7
Eventuell övertagande organisation	8
Litteraturförslag/dokumentationshänvisning	8
Dokumentation	8
Tekniker	8
Förslag till förbättringar inför kommande projekt	9
Reflektion	9

Inledning/bakgrund

Samtliga i gruppen har skrivit under ett NDA (non disclosure agreement).

Slutrapporten innehåller därför inga hänvisningar till källkod eller bilder av produkten. Det gör det också svårt att detaljerat beskriva applikationen.

Tieto har utfört en förstudie om en ny typ av e-handel. De stora e-handelsföretagen har försvårat för lokala butiker. Den här affärsidén är tänkt som ett alternativ till dagens e-handel som främjar lokal handel.

Syfte och mål

Syftet med projektet har varit att ta fram en prototyp på en mobilanpassad webb-applikation som Tietos kund ska kunna visa upp för sina intressenter. Prototypen är en del av förstudien och ska simulera en ny typ av e-handel.

Det slutgiltiga målet har varit att ta fram en tillräckligt bra fungerande applikation som vår slutkund kan presentera för sina intressenter. Tanken är också att den här prototypen ska kunna användas som en startpunkt för den slutgiltiga produkten.

Projektorganisation

Projektorganisationen bestod av fem studenter på programmet Webbprogrammerare på Linnéuniversitetet. Tre av oss studerade på campus i Kalmar och två studerade på distans.

Vi delade upp ansvaret för olika huvudområden mellan oss tidigt, men det var inga skarpa gränser och vi arbetade alla inom de flesta områden.

- Martin Lindahl - projektledare
- Boris Grubescic - git-ansvarig
- Lena Eriksson - testansvarig
- Edvin Larsson - teknikansvarig

- William Robertsson – kund/krav-ansvarig

Tre veckor in i projektet tillkom även tre studenter från linjen Interaktionsdesign. De har parallellt utvecklat design för applikationen som vi tagit inspiration ifrån.

Genomförande: metodik, teknik

Utvecklingsmetodik

Vi har använt en agil utvecklingsmetod där projektet har varit uppdelat i fyra faser, inception, elaboration, construction och transition. Produkten har tagits fram iterativt i sprintar enligt SCRUM.

Projektet har drivits i 2-veckorssprintar. Varje sprint inleddes med ett sprintplaneringsmöte där gruppen och produktägaren närvarade. Funktionskraven för projektet togs fram i form av user stories som placerades i en produkt-backlog. På sprint-planeringarna plockade gruppen user stories från produkt-backloggen i den mängd som antogs hinnas med under sprinten och placerades i prioritetsordning i sprint-backloggen. Under sprintarna implementerade gruppmedlemmarna sedan de user stories som fanns.

I slutet av varje sprint demonstrerade vi applikationen och den funktionalitet vi implementerat. Kunden gav återkoppling på vad de ville ha åtgärdat, eller kom med förslag på nya ideér som borde implementeras.

Vid två tillfällen under projektet blev sprint-backloggen tom före sprinten var färdig. Då tog vi det högst prioriterade kortet från produkt-backloggen.

Kommunikation

Nästan all kommunikation mellan oss i gruppen har skett över Slack, där vi skapade vår eget workspace. Varje morgon hade vi daily stand up via Skype enligt SCRUM-modellen där vi i turordning berättade enskilt vad man har gjort under gårdagen, och vad man planerade att göra under arbetsdagen. Om det var längre diskussioner som behövde hållas så fördes den diskussionen efteråt.

Vi använde [Trello](#) för att hålla koll på vad som skulle göras. Vi använde två stycken olika bräden. Ett bräde användes för att organisera projektet. Här hade vi kolumner för product backlog, sprint backlog, in progress och done. I kolumnerna hade vi kort för olika uppgifter. Tieto och kunden hade också tillgång till det här brädet. Vi hade också en intern bräda för t.ex. kursrelaterad dokumentation.

Skype användes som kommunikationsverktyg under samtliga möten med Tieto och kunden.

Versionshantering

Vi använde [Git](#) för versionshantering. Huvudgrenen innehåller koden som har levererats till kunden. Som utvecklare skapade man en egen gren från huvudgrenen för den funktionalitet som skulle utvecklas. När funktionaliteten var klar så begärde man en ”pull request” på GitHub. Git-ansvarig utsåg granskare som gjorde code review (med undantag för vissa väldigt små ändringar och buggfixar). När funktionaliteten ansågs vara klar gjorde Git-ansvarig en merge med huvudgrenen.

Utvecklingsmiljö

- Operativsystem: Windows 10, MacOS, Arch Linux och Ubuntu
- Kodlagring: [GitHub](#)
- Versionshantering: [Git](#)
- Kodredigering: [Visual Studio Code](#)
- Kodstandard: [Prettier](#) och [TSLint](#) används för gemensam kodstandard

Utvecklingstekniker

BackEnd

- NodeJS
- Express
- TypeScript
- MongoDB

FrontEnd

- React
- Material UI

Deployment

- Docker
- Google Cloud Platform
- Amazon Web Services

Kunden gav oss fria händer att välja tekniker för projektet. Vi valde tekniker som vi trodde kunde passa projektet och som några av oss hade erfarenhet av. React hade inte alla använt, men parprogrammering gjorde att kunskapen kunde spridas. TypeScript var nytt för de flesta, men steget från JavaScript kändes litet.

Resultatbeskrivning/måluppfyllelse

Målet med projektet var att ta fram en prototyp som kunden kan visa upp för sina intressenter. Även fast applikationen inte är helt färdig så kommer den att kunna användas för vidare diskussioner och idéer. Kunden uttryckte ett stort nöje med resultatet på sista sprintdemot.

Avvikelser/efterkalkyl

Designmässigt har vi fått gå en medelväg mellan den ursprungliga design som vi fick från kunden och de nya designidéerna från interaktionsdesign-studenterna. Vi hann inte implementera de nya förslagen fullt ut eftersom vi jobbade parallellt, men vi tog en hel del inspiration från deras arbete.

Slutsats

Det var roligt och lärorikt att göra ett projekt mot en verklig kund. Man märker att det bara är en del av tiden som går till faktiskt kodande, mycket av tiden går till att samordna arbetet.

Det blir också tydligt att det är en verklig tillgång med olika typer av kompetens inom gruppen, man kan hjälpas åt på ett bra och effektivt sätt.

Vi känner att vi hann med att få ihop en bra prototyp som kunden kommer kunna använda för att gå vidare med i sitt projekt.

Väldigt nyttigt med insikt i hur uppstarten av projekt går till ute i arbetslivet. Kommunikation är en central del i det hela, både inom gruppen som utvecklar en mjukvara samt mot stakeholders.

Förslag på vidareutveckling

Hur kunden väljer att gå vidare med projektet avgör var fokus för utvecklingen av applikationen bör läggas. Det vi har tagit fram i detta stadiet är en prototyp. Antingen kan de välja att bygga vidare det riktiga systemet på vår prototyp vilket säkert underlättar då många delar redan är på plats, eller så börjar de om helt efter att samtliga synpunkter från deras intressenter kommit in. I vilket fall som helst så bör fokuset nu ligga på att samla så många intressenter som möjligt och ta del av deras synpunkter.

Skulle kunden vilja fortsätta med att ta fram en prototyp anser vi att fokus bör läggas på att:

- Designa prototypen enligt ID-studenternas arbete
- Göra applikationen mer dynamisk beroende på vem som är inloggad (just nu är en del hårdkodat)
- Göra det enklare för butiksägare att justera information gällande t.ex. öppettider, varulistor etc

Skulle kunden vilja fortsätta med att ta fram den slutgiltiga applikationen anser vi att fokus bör läggas på att:

- Ett riktigt och säkert inloggningssystem
- Ett system för att dynamiskt kunna få in varor
- En mer avancerad sökfunktion

- Utveckla eller använda ett rekommendationssystem

Eventuell övertagande organisation

Produkten är produktionssatt hos en molnleverantör och kunden har fått en publik URL.

Kunden får tillgång till GitHub-repot med källkoden samt all dokumentation. De kommer ha tillgång till produkt-backloggen på Trello som innehåller user stories och krav som vi inte har hunnit implementera, men som man kan gå vidare med i en utvidgad prototyp eller i en skarp produkt.

Kunden får också en manual som beskriver hur de kan demonstrera prototypen för sina intressenter.

Litteraturförslag/dokumentationshänvisning

Dokumentation

Alla dokumentation som rör projektet finns tillgänglig här:

<https://github.com/1dv611-2019-group6/1dv611-2019-group6-wiki/wiki>

Tekniker

- **NodeJS** används som plattform för applikationen.
<https://nodejs.org/en>
- **Express** används som ramverk för servern.
<https://expressjs.com>
- **TypeScript** används som programspråk på serversidan för att få fördelen av ett typsäkert, kompilerat språk.
<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
- **MongoDB** dokument-databas används för hantering av persistent data för applikationen.

<https://www.mongodb.com>

- **React** JavaScript-bibliotek används för klientsidan.

<https://reactjs.org>

- **Material UI**, React UI-ramverk används för design av webbgränssnittet.

<https://material-ui.com>

- **Docker** används för virtualisering.

<https://www.docker.com>

- **Git** används för versionshanteringen av applikationen.

<https://git-scm.com>

Förslag till förbättringar inför kommande projekt

- Jobba bort stora risker tidigt i projektet. Skjut absolut inte fram det. Vi hade en del problem med deploy-biten under projektets gång och bytte molnleverantörer fram och tillbaka.
- Se till att vara effektiva på morgonmötena (Ifall ni bestämmer er för att ha sådana). De kan lätt dra ut på tiden.
- Lägg inte onödig tid på “feature creeping”. Om ni är osäkra på om kunden vill ha ett visst utseende, eller en viss funktionalitet, gör först ett grovt “utkast”, gärna hårdkodat, och ta feedback efter nästa möte med kunden.
- Om interaktionsdesigners ska vara med i projektet, involvera dem så tidigt som möjligt, för att maximalt kunna utnyttja dokumenten som de tar fram.

Reflektion

Vad gjorde vi bra respektive vad gjorde vi mindre bra?

Dagliga möten med gruppen var ett väldigt bra sätt för alla att veta vilka som gjorde vad samt vem man kunde kontakta om något skulle strula. Hade man inte möjlighet att vara med på daily standup så framfördes det i Slack.

Hög transparens. Då nästan alla körde en till kurs parallellt så var det full förståelse om någon var tvungen att fokusera på den andra kursen under dagen.

Parprogrammering var något som vi gjorde flitigt i början av projektet. Det var ett bra sätt för att sprida kunskaper i gruppen. Efter några veckor blev det mer självständigt arbete, och effektiviteten blev högre. Vi parprogrammerade även vid behov, när någon fastnade och behövde hjälp.

Alla i projektgruppen har jobbat med samtliga delar av applikationen, vilket gjort att vi fått en bättre helhetsbild och mer erfarenhet av olika tekniker. Det gjorde oss också mindre sårbara för t.ex sjukfrånvaro.

(Mindre bra)

I början brukade våra daily standups dra ut mycket på tiden och var inte särskilt strukturerade. Vi löste det genom att turas om i bokstavsordning. Det gjorde våra daily standups mycket mer tidseffektiva. De gick från att vara runt en timme till uppskattningsvis 15-20 minuter.

Ibland fick vi lite olika uppfattningar om flödet i applikationen. Vi hade kunnat gå igenom sådant tidigare och sett till att alla i gruppen hade samma uppfattning.

ID-gruppen anslöt sig till projektet tre veckor in, det här var dock utom deras kontroll och något som universitet borde ha styrt upp. Men det var ändå värdefullt, framför allt mot slutet med de designexempel vi fick av dem. Vi tror också att kunden uppskattar deras arbete väldigt mycket eftersom utseendet på deras mockup är snyggare än prototypen, och möjligtvis något som kommer att implementeras i den riktiga applikationen.

Utvärdering av tidplanen – höll den? Vad beror det på? Hur avvek tidsskattningar? Blev skattningarna bättre med tiden? Varför/varför inte?

I många fall var det svårt att göra tidsskattningar eftersom flera av de tekniker vi valde att jobba med var nya för oss. Det här är givetvis något som blev bättre längs vägen men de första 4-5 veckorna kunde utförandet av en task inkludera inlärningsbiten också, och den är svår att tidsskatta. I överlag har tidsplanen hållit.

Vi har jobbat med verktyget Trello och enligt vår kontaktperson på Tieto som varit med på alla Sprint-möten bör varje kort som placeras under Sprint Backlog tidsskattas på 8 timmar. Det här har vi inte hållit under projektets gång. Ibland har vi delat upp större kort i 3-4 mindre kort och i vissa fall har vi jobbat på ett större kort i mer än 8 timmar.

Blev dokumentationen tillräckligt bra? Kunde vi gjort annorlunda?

Vi jobbade med dokumentationen kontinuerligt och vi fick under projektets gång bra feedback från vår referentgrupp och vår handledare. Det gjorde att vi kunde förbättra vår dokumentation efter varje granskning.

Hur fungerade informationsflödet mellan projektgrupp och kund? Vad kunde vi ha gjort annorlunda?

Vi hade möte med slutkunden vid varje sprint-demo, dvs. varannan vecka. Vi bjöd även in kunden till Slack om det var några akuta frågor. I en perfekt värld så hade kunden varit lite mer involverad i arbetet, och man hade tidigare kunnat få feedback på funktionalitet som kanske i slutändan inte togs med ändå. Detta hände som tur var bara ett fåtal gånger under projektets gång.

Vi borde från början ha sett till att kontinuerligt uppdatera en publikt tillgänglig applikation så att kunden kunde klicka runt. Då hade det nog varit enklare för kunden att ge feedback.

Under sprintdemonerna när kunden diskuterat önskemål har det ibland varit oklart vad som borde prioriteras för vår prototyp och vad som bara rör den slutgiltiga applikationen.

Hur fungerade samarbetet inom och utom projektet?

Vi tycker att samarbetet inom gruppen har fungerat väldigt bra. Vi gick tidigt igenom alla personliga ambitionsnivåer. Alla i gruppen var tydliga med vilka mål och förväntningar som fanns.

Var projektmöten, styrgruppsmöten med flera lagom långa? Tillräckligt effektiva? Hölls mötena tillräckligt ofta? Kunde vi gjort på annat sätt?

Vi bestämde oss tidigt för att ha dagliga möten inom gruppen. Dessa är enligt SCRUM-modellen tänkta att pågå i ungefär 15 minuter. Till en början var det mycket att diskutera i projektet, och dessa möten kunde ofta springa iväg och bli onödigt långa. Vi förbättrade detta allt eftersom i projektet, och vi såg till att vara överens om att övriga diskussioner skulle hållas efter daily stand up med involverade personer, så att övriga gruppmedlemmar kunde komma igång med arbetet. Skulle det uppstå några oklarheter under dagen, så brukade vi annonsera detta i Slack, och de som var tillgängliga och berörda kunde delta i mötet.

Vi är alla överens om att beslutet av att ha dagliga möten var riktigt bra.

Hur var riktlinjerna vi fick? Var de tydliga nog?

Kunden gav bra input i början av projektet om vilken funktionalitet de ville ha och hur de ville att den slutgiltiga applikationen skulle se ut i stora drag. Dock var de mindre tydliga med vad som borde ha prioriterats för prototypen.

Motsvarade vi kundens krav?

Ja, vi levererade en fungerande prototyp. Kunden fick idéer under projektets gång om hur prototypen skulle fungera och vi lyckades implementera ändringarna under varje sprint.

Gav vi realistiska förslag på lösningar? Kunde vi gjort annorlunda?

Tieto har utfört en närmare undersökning parallellt med utvecklingen av vår prototyp och ofta under möten med kunden har både vi och Tieto gett förslag på lösningar för den eventuellt “riktiga” applikationen som ännu inte är utvecklad. Många av dessa lösningar har inte varit realistiska för vår prototyp men diskussionerna har varit värdefulla och intressanta för alla delaktiga parter och kommer helt klart att vara till stor nytta i ett senare skede.

Har vi haft ett system för tidiga varningar om planerna inte följs? Hur har det fungerat?

Vi har haft en risklista där vi försökt förutse eventuella problem och också skrivit ner strategier för att hantera dem.

Genom våra daily standups och kontinuerliga kommunikation via Slack har vi kunnat diskutera och lösa problem som uppstått.

Hur fungerade testningen? Litar vi på att systemet fungerar?

Vi lade fokus på manuell testning och Postman-tester. Efter att ett krav hade implementerats, kodgranskats och mergats in i huvudgrenen på GitHub sattes kravet till "redo för test". Nya tester för kravet skapades och kördes då innan kravet ansågs klart. På så vis fyllde vi på med tester kontinuerligt. Efter varje sprint körde vi igenom alla tester och skapade en testrapport. Problem som hittades fördes in som issues i GitHub.

Det har varit svårt att skriva detaljerade testplaner och testrapporter eftersom vi inte har fått avslöja vad projektet går ut på. Kanske kunde vi ha haft mer detaljerade manuella tester internt.

Idealiskt borde vi ha haft automatiska enhetstester, speciellt för frontend. Kollektionen av Postman-testerna testar API:ets funktionalitet rätt bra. Men för frontend har vi i stort sett bara manuella tester. Det kan vara svårt att skapa bra enhetstester för ett användargränssnitt, men det hade varit guld värt.

Vi kan inte svara på ifall applikationen fungerar som den ska vid väldigt många användare. Inga stresstester har utförts. Detta är egentligen inte ett problem eftersom den prototyp som vi har utvecklat kommer att användas av ett fåtal användare.

Levererade vi kontinuerligt fungerande versioner till kund och slutanvändare?

Om inte; varför och hur påverkades projektet av detta?

Vi visade en fungerande lokal version varannan vecka på sprint-demot. Vi fick kontinuerlig feedback på vad som var bra, vad de skulle vilja ändra, prioritera osv.

Vi borde dock från början ha sett till att kontinuerligt uppdatera en publikt tillgänglig applikation så att kunden kunde klicka runt.

Vad blev resultatet på kort och lång sikt?

På slutdemot så var de nöjda med vad vi hade åstadkommit. Slutkunden har en lång önskelista på vad som de skulle vilja ha i den slutgiltiga applikationen, alltså ej den prototyp vi utvecklade. Eftersom vi hade möten med kunden varannan kunde vi också ge dem en realistisk bild på vad vi var kapabla till att leverera efter en sprint.

Kunden hade inte från början en exakt bild av hur prototypen skulle fungera, projektet har hjälpt dem med att börja fundera mer på konkret funktionalitet som de vill ha i sin slutgiltiga produkt.