

Crunch supportsag portal

Indhold i gruppe Gs repository på Fronter:

Afleverede artefakter

- 01 Gantt / Burndown chart
- 02 Definition of done
- 03 Functional requirements
- 04 Use cases
- 05 Domain
- 06 Wireframes
- 07 State-machine
- 08 Risk analysis
- 09 ER-diagram
- 10 DataFlow
- 11 Function points / tid og pris
- 12 Kodestandarder
- 13 Test plan
- 14 Maintenance
- 15 Tid og pris
- 16 Trello-board

Arbejdsmetode

Vi har i gruppen ønsket at arbejde agilt, og har stræbt efter at arbejde efter Scrums principper med vægt på "Product backlog", "Sprint backlog", "Sprint planning" og "Sprint review". Vi har forsøgt at arbejde med scrum elementerne så vidt muligt, men grundet sideløbende anden undervisning, og det at vi er et lille team, der har arbejdet på forskellige tidspunkter, har vi skiftet til at agere scrum master og product owner under vores daily scrum.

Vi har brugt Trello til at styre backlogs og løbende følge det aktuelle sprint. I vores Trello har vi givet alle elementerne story points og sat dem ind i vores burndown chart for løbende at holde styr på, hvor godt vi er med efter tidsplanen.

Crunch ApS

Vores kunde Crunch er et bureau som har speciale i rådgivning, sparring og eksekvering i det digitale univers. Virksomheden blev etableret af Adam Kischinovsky tilbage i 2010. Crunch er et webbureau med mange års erfaring når det kommer til opsætning af hjemmesider, SEO optimering samt annonceringer til forskellige online medier såsom Facebook, Instagram og Google – herunder Google Ads og Google Display.

Kundens scope

Crunch ønsker at få udviklet en portal til styring af supportsager. Portalen skal benyttes af Crunch og deres kunder, og skal gøre det nemt og overskueligt at behandle deres kunders support behov. Det er ønsket, at portalen laves i PHP da Crunch selv vil overtage opdateringer og vedligeholdelse af systemet.

Kundens første scope formulering sendt pr. mail

Vi vil lave en portal for eksisterende kunder, hvor de kan logge ind.
De skal kunne følge efter de er logget ind:

- Oprette supportsager
- Se hvem der arbejder på deres supportsag
- Se status på supportsag
- Se tidligere købte produkter
- Se tidligere kontrakter og fakturaer
- Se login detaljer til deres hjemmeside og FTP
- Tidsregistrering så de kan se hvor lang tid der er brugt på supportsagen
- De skal kunne købe flere ydelser derinde fra

Tilføjelser til scope efter fysisk 22-Apr-2021 møde, hvor vi præsenterede wireframes, der illustrerede vores opfattelse af ovenstående og stillede uddybende spørgsmål

- Søgning i faktura på datoer
- Oversigt skal søges gennem produkter

Noter til artefakter:

01 Gantt-skema:

Vi udarbejdede tidligt i processen et Gantt-skema, som vi løbende har revideret, så vi har haft et fint overblik over vores proces.

02 Definition of Done:

Vores Definition of Done blev også vedtaget tidligt i projektet og skulle danne en standard for, hvad der kan betegnes som Done. Her har vi besluttet, at hvis man mener noget arbejde, man har lavet, er færdigt, skal der være mindst ét andet gruppemedlem, som godkender arbejdets færdighed. Vi holder styr på, hvad der er Done på vores Trello board.

03 Functional requirements:

De funktionelle krav har dannet grundlag for det videre arbejde med især Use-cases og domæne-model.

04 Use cases:

Vi har valgt at udarbejde to use cases i en fully dressed version, og en casual. Det helt centrale element i vores projekt er, at en kunde skal kunne oprette en supportsag, så det var oplagt at bruge dette eksempel som den ene, og fra firmaets side har vi vurderet, at det centrale er at kunne oprette en ny kunde i systemet, så det har vi valgt som det andet. Derudover var et andet centralt element at oprette nye produkter, og hvor disse kunne redigeres, så vi har også valgt at lave en casual use case for opret produkt.

05 Domain model:

Ud fra de funktionelle krav har vi identificeret de centrale elementer i systemet, som vi har brugt til at udarbejde en domænemodel, som vi bl.a. har brugt som udgangspunkt for første udgave af Data flow diagrammet.

06 Wireframes:

Vi udarbejdede wireframes som de første artefakter, da vi brugte dem på første møde med kunden for at sikre os, at vi havde forstået kundens ønsker til systemet. Vi har så løbende opdateret vores wireframes i takt med vores State-machine, hvor vi fik bedre overblik over systemet, og de har været helt centrale i vores systemudviklingsproces, fordi de har været et godt udgangspunkt til kommunikation og diskussioner om systemets funktionalitet og begyndende design.

07 State-machine:

Som sagt er vores State-machine diagram udarbejdet i takt med vores wireframes, da det var essentielt at finde hvilke funktioner, der skulle være tilgængelige for brugeren i det stadie brugeren befandt sig i. Her bliver det også tydeliggjort, hvorhen brugeren lander efter at have foretaget en bestemt handling. For eksempel når en ny sag oprettes, kastes brugeren hen til den nyoprettede sags side.

08 Risikoanalyse:

Vi har gjort brug af risikoanalysen for at opdage og forebygge de risici, vi under projektets løb kan støde ind i.

09 ER-Diagram:

ER-diagrammet skal bruges, når databasen skal designes, og så har vi også brugt det til udregningen af Function points, samt i udarbejdelsen af testkode.

10 Dataflow diagram:

Første udgave af Data Flow Diagram udarbejdede vi med udgangspunkt i vores domænemodel. Vi lavede fem versioner af diagrammet, inden vi landede på det endelige resultat, som vi har brugt som grundlag for at udregne Function points.

11 Function points:

Vi har udregnet Function points ved at tage udgangspunkt i Data Flow diagrammet og ER-diagrammet. Optællingen af EI, EQ, EO, ILF, EIF, DET, RET og FTR har vi ført ind i et excel-ark. Den endelige optælling har vi brugt til at estimere tid og pris.

12 Kodestandarder:

Vi har opsat en række kodestandarder, som skal skabe enighed om, hvordan koden skal udvikles, og sikre, at andre udviklere senere kan udbygge systemet.

13 Test plan:

Vi har opsat en test plan som skal bane vej for en mere fejlfri og optimeret udviklingsprocess. Vi regner med at der skal Unit-testes og Integration-testes hvor alt skal fungere korrekt før System-testning påbegyndes.

Ved testning af systemet skal vi gøre brug af Usability-testing, hvor en udefrakommende sætter sig med systemet og giver feedback til systemets usability.

Her skal vi også Performance-teste systemet ved at forsøge at overloade det og generelt sørge for at det kører optimalt.

Da Crunch ønsker at overtage og vedligeholde systemet selv, efter det er udgivet, vil de også gerne følge med i processen. Derfor vil vi Acceptance-teste med Crunch, efterhånden som vi opfylder functional requirements. Dette sikre os også at vi holder os på rette spor gennem udviklingsprocessen.

14 Maintenance:

Efter systemets deployment er det naturligvis vores job at rette de bugs, der nu kan opstå eftersom systemet er i brug hos end-users. Men som sagt ønsker Crunch at overtage system og kommer også selv til at vedligeholde det. Af den grund regner vi ikke med at skulle lave en maintenance aftale.

I tilfældet af at Crunch skifter mening og vil lade os vedligeholde systemet, vil vi tilbyde en aftale der ligger på 500 kr. om måneden. Denne estimering er baseret på, at systemet er forholdsvis enkelt og det umiddelbart ikke kræver 24 timers indgriben, da crashes eller fejl ikke er kritiske døgnet rundt. Vi vil med aftalen kun tage handling inden for arbejds-/kontortiden.

I aftalen sørger vi også for webhosting, sikkerhed, backup og 30 minutters gratis support pr. måned, hvor den efterfølgende support tid følger vores timeløn.

15 Tid og pris:

På baggrund af de beregnede function points har vi beregnet tid og pris. Vi har i gruppen lavet nogle estimerings-øvelser, hvor vi enkeltvis har vurderet tiden for udvikling af forskellige mindre elementer, og vi har så fundet et gennemsnit af de forskellige estimeringer og fundet frem til, at et function point svarer til 10 arbejdstimer.

Da vi er junior-udviklere i et helt nyt start-up, der brænder for at få kunder, har vi sat timeprisen på 250 kr og så ganget den samlede pris med en faktor på 1,3, så der er dækning for vores faste udgifter og lidt profit.

I forhold til vedligeholdelse af systemet regner vi ikke med at tage noget for det, da Crunch selv har ønsket at vedligeholde det selv. Men skulle det ske, at de hen af vejen ændrer mening, ligger vores estimering på 500 kr. om måneden for vedligeholdelse.

16 Trello- board

Vi har brugt Trello til at holde styr på vores sprints gennem hele forløbet.