Kleans bog om scrum

Introduktion til scrum

Ordet "scrum" er hentet fra rugby. Dér bruges det om den måde, som spillet sættes i gang på. Hvert hold samler sig tæt sammen og går som fuldt team ind i den opgave det er et erobre bolden og flytte den frem af banen.

Scrum til realisering af webprojekter

Scrum er en udviklingsmetode, der blev opfundet som modvægt til de traditionelle it-projekter efter "vandfaldsmodellen".

Idéen med vandfald er at definere meget præcist, hvad man skal gøre, før man fortsætter til næste trin. Til gengæld er der ingen vej tilbage, når man først er på vej nedad.

Den perfekte specifikation
I praksis skriver man derfor en "perfekt" kravspecifikation, før man går igang. Det lyder allerede hult, ikke?

Scrum er baseret på brugerdeltagelse, forandring og iteration. I vores projekter er der aflevering hver anden uge. Ved hver aflevering har kunden fuld ret til at skifte mening. Det er faktisk helt okay at blive klogere undervejs.

Hvad kan man skrue på?
Hvordan kan det lade sig gøre at holde budgettet, vil nogle spørge? Det var også den udfordring, vi selv stod overfor, da vi besluttede os for at lave scrum-baserede projekter med stor involvering af kunder.

I virkeligheden er det ikke så svært. Et projekt består af fire ting:

- Et budget og deadline
- En række features, der skal laves
- Et team af mennesker, der skal lave dem.
- Prisen afhænger af, hvor mange timer man bruger på at lave features, og antallet af features er bestemt af, hvad man vil med sit website.

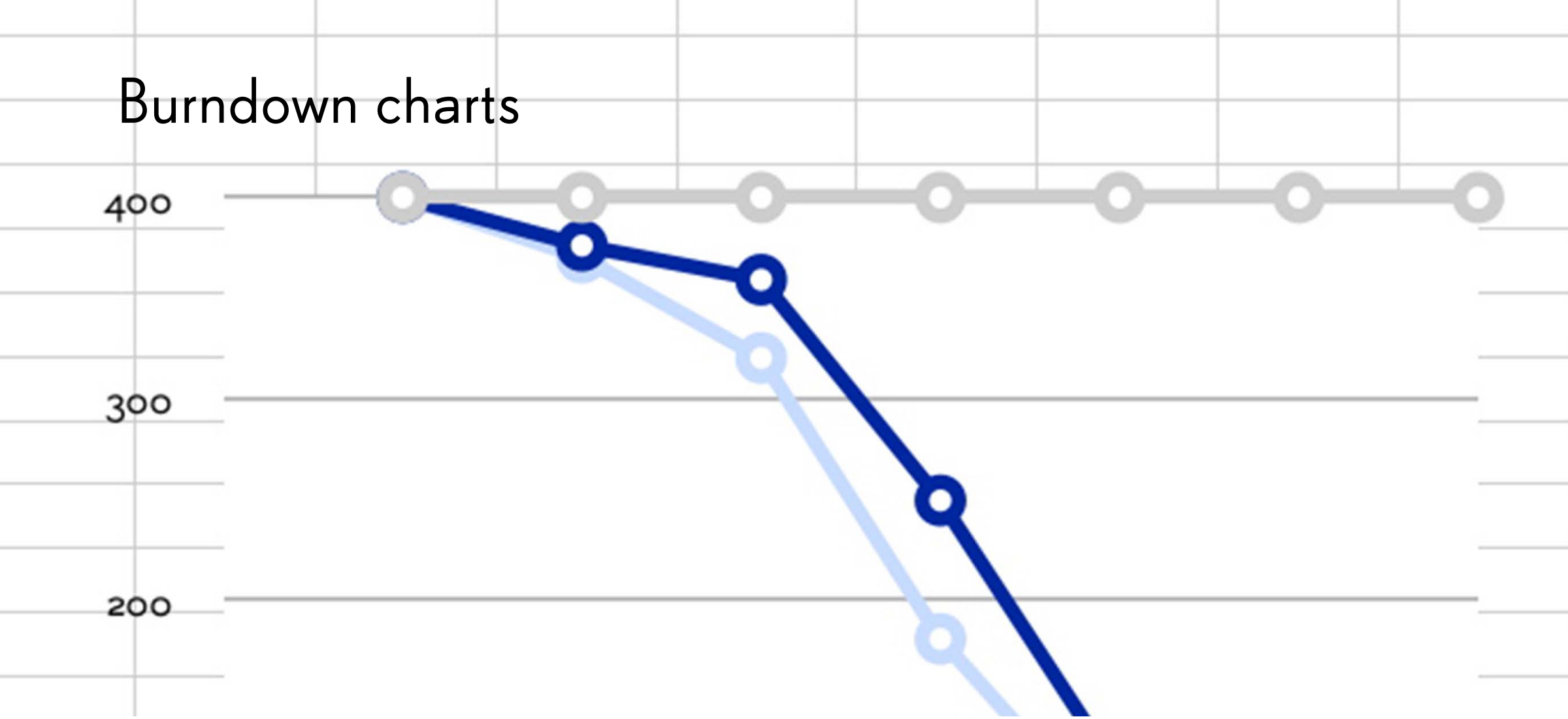
Vil man levere et projekt til fast pris og fast deadline, skruer man på antallet af features.

Det lyder jo meget simpelt, og det er det faktisk også.

Ud fra et økonomiske synspunkt er scrum en fordel, fordi det er en lettere måde til at ramme deadline og budget.

Undervejs er det også sjovere, for produktiviteten er højere, når projektet opdeles i overskuelige opgaver.

Burndown charts



Projektets økonomi overvåges i et burndown chart. Populært sagt oversættes budgettet til et antal timer, og burndown viser, hvor mange timer vi brænder af i projektet.

Al tid registreres undervejs, og produktejeren kan se enhver deltager over skulderen og se, hvor mange minutter der bruges på hver enkelt opgave.

Et burndown chart er åbent overfor fortolkninger. Det viser data om projektet, men forklarer ikke hvorfor.

Ofte er det godt at bruge post mortem på sprintmødet til at diskutere, hvad vi får ud af burndown chart.

I eksemplet her er det faktiske tidsforbrug (den mørkeblå linie) mindre end det forventede (den lyseblå linie).

Mulige forklaringer er:

1. Vi er efter tidsplanen og må sætte foden på speederen for at nå projektet til tiden.

2. Vi har nået de features, vi skulle, så vi er klar til at levere før tiden og under budget.

Tro det eller ej, men det sidste kan også lade sig gøre. Som produktejer har man til enhver tid retten til at sige, at man er tilfreds med projektet, og da hver aflevering er fuldt funktionsdygtig og i fuld kvalitet, kan man stå af toget før man når den planlagte endestation.

Workshops

En god metode til at finde den korteste vej til det rigtige resultat er workshops. Det tager normalt mellem to og tre timer. Det er meget intensivt, det er sjovt, og resultatet er masser af tegninger, mindmaps og inspiration.

Er den lige linje mellem A og B den rette at følge?

En workshop tager normalt mellem to og tre timer. Det er meget intensivt, det er sjovt, og resultatet er masser af tegninger, mindmaps og inspiration.

Det er mere reglen end undtagelsen, at et projekt ender et helt andet sted end planlagt.

Det er som regel også det bedste. Hvis projekter blev udført som beskrevet i den oprindelige specifikation, ville vi ende med en bil med tre hjul, uden rat og med blinklys i taget.

En god metode til at finde den korteste vej til det rigtige resultat er workshops.

Ofte holder vi tre typer workshops:

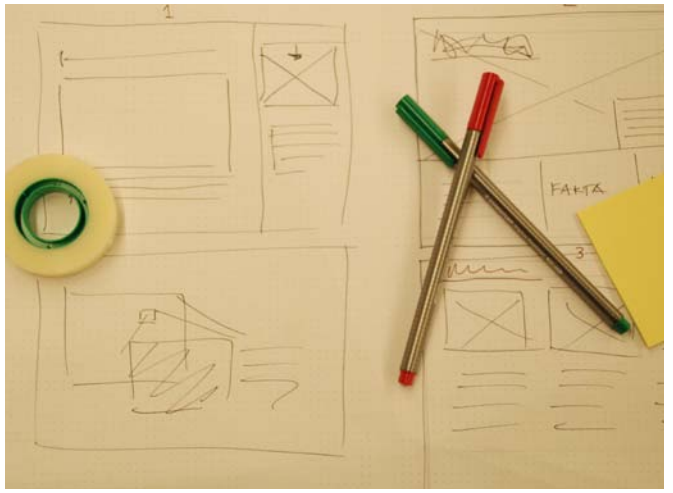
1. En workshop på strategisk niveau
Her diskuterer ledelse, web-redaktion og medlemmer af et scrum-team, hvad de forretningsmæssige mål er samt hvordan et website eller intranet kan opfylde strategien.

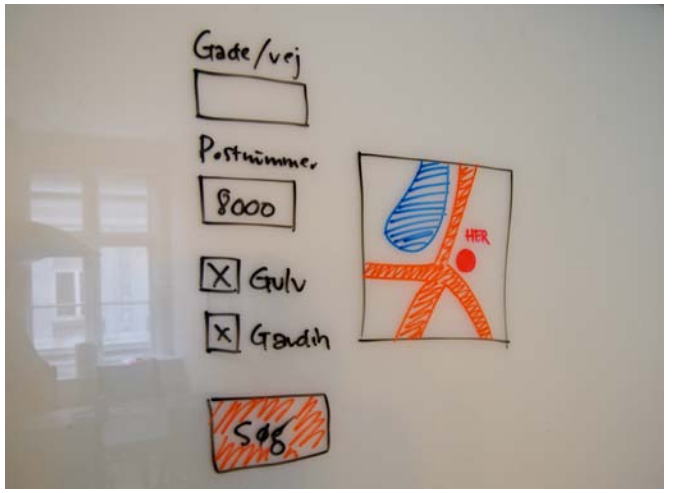
2. En workshop om et forretningsområde
Som efterfølger laver vi en mere operationel workshop, hvor vi udvælger et forretningsområde og prøver at være konkrete. Det er altid godt med et realistisk eksempel, og "hele projektet" er som regel for stor en størrelse. Ledelsen deltager ikke på denne type møder, men der skal være repræsentanter for forretningsområdet med.

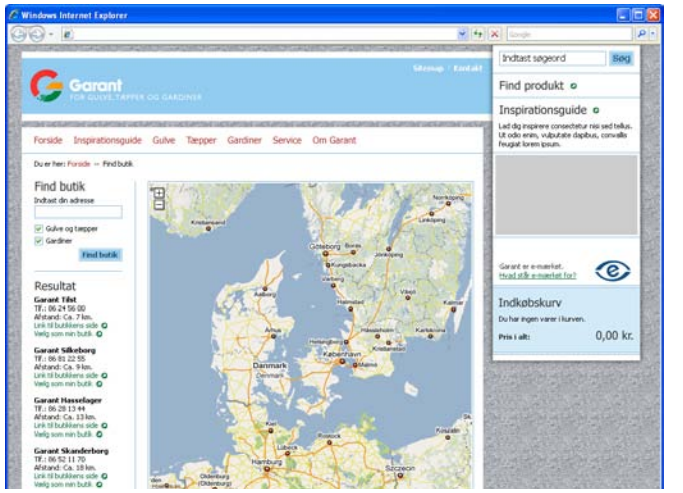
3. En workshop om projektet
Nu, hvor vi har en konkret plan, starter vi arbejdet. Hvem har ansvar for hvad, hvilke opgaver skal løses først, hvor mange sprint skal vi bruge og hvor starter vi? Denne workshop er ofte orienteret om detaljer, og som vi siger, så er "djævelen gemt i detaljen". Deltagere er web-redaktionen og medlemmer af scrum-teamet.


QUADOR
Coca-Cola light

Prototyper er fantastiske







Det er utroligt, så god software man kan bygge med prototyper.

Den bedste kravspecifikation er ofte tegnet på en tavle eller en serviet. I mødelokalet har vi tusser, et kamera, papir, notesblokke, lim og papirklips i et omfang, der ville have gjort Jørgen Clevin glad.

Her er, hvordan vi byggede en butikssøgning i tre raske trin:

1. Prototype

Vi holdt et møde på 45 minutter, hvor vi fik tegnet fem skærmbilleder til et modul, der kan finde nærmeste butik ud fra en adresse, man indtaster på websitet.

Moduler skal integreres med Google Maps, og det skal levere en kørselsvejledning.

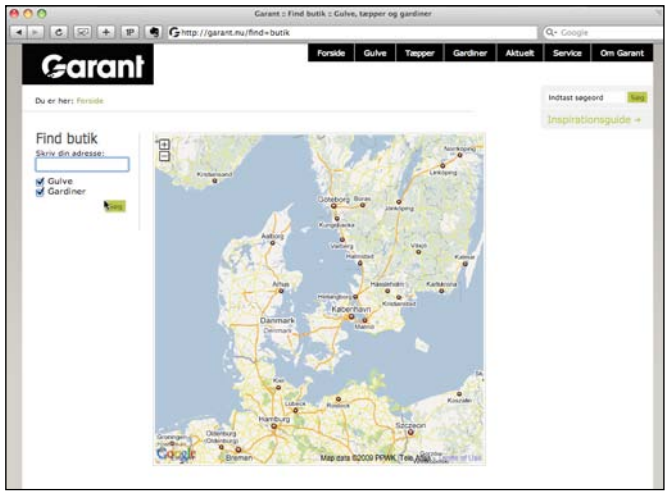
Vores skærmbilleder viser alle trin, også hvordan det færdige modul kan konfigureres i content management systemet af en almindelig web-redaktør.

2. Interaktionsdesign

Den godkendte prototype blev sendt videre til en interaktionsdesigner, der lavede detaljerede skærmbilleder.

Websitets design var stadig ikke godkendt, så det gamle website-design blev brugt som ramme.

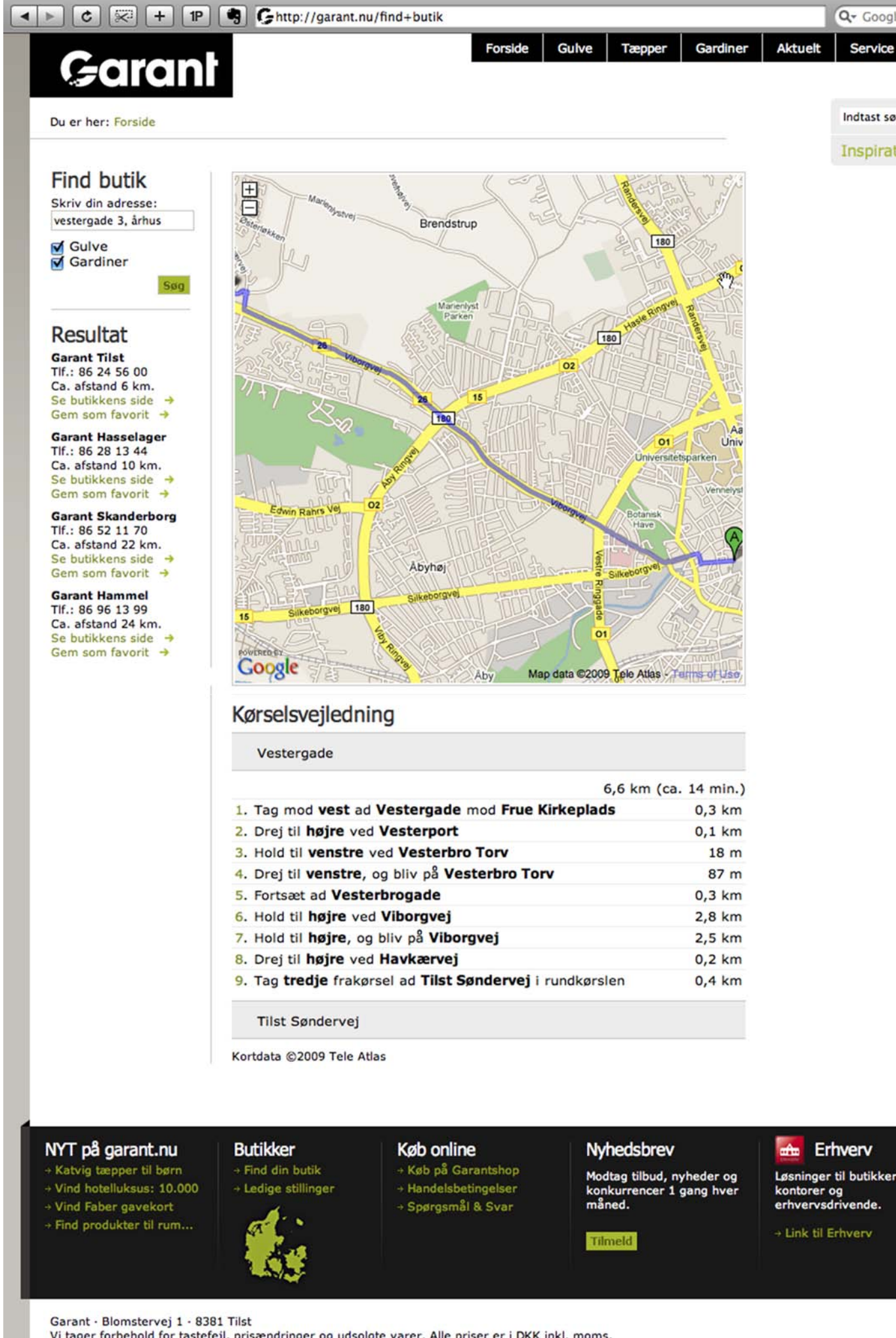
Elementer fra prototypen blev tegnet helt ned i detaljen. Sådan finder man fejl, før de bliver kodet.



3. Færdigt website

Så blev modulet sat i produktion, og det endelige resultat er forbløffende tæt på den oprindelige prototype.

...og vi slap for at skrive en kravspecifikation på tyve sider. Herligt.



Teamet

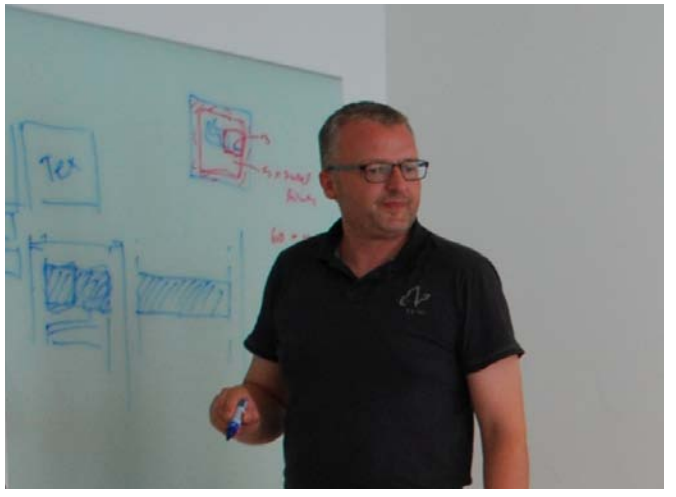
Et scrum team er den arbejdsgruppe, der skal gennemføre projektet.

Når vi laver projekter her i huset er der et par klare regler:

- Alle er lige
- Al kommunikation er fri
- Dem, der har viden, skal inddrages i beslutninger
- Dem, der har ansvaret, skal være enige i beslutninger

Inden vi ser på et praktisk eksempel, kan vi lige få et overblik over et scrumteam.
Hvem er med, og hvad laver de?

Roller



Produktejer
Produktejeren er kunden, og i et scrum-projekt er man med i hele projektet. Modellen med en "kunde" og en "leverandør" er forældet. Projekter, det er noget, vi laver i fællesskab.

Fordelen er, at projektet er godt forankret i den organisation, der i sidste ende skal anvende systemet.

Som produktejer har man ansvaret for alle prioriteter. I de første projekter, vi lavede, betød det også et ansvar for hele product backlog. Vi fandt ud af, at det ikke var helt perfekt.

Derfor er vores team nu opdelt, så vi har en produktejer og en ansvarlig for product backlog. De deler opgaven med at prioritere, så de kan hjælpe hinanden.

Produktejeren bestemmer, og den ansvarlige for backlog har både ansvaret for at alle opgaver står på listen, men også for at opgaverne er specificeret og estimeret.

Det er beslutningsgrundlaget for produktejeren. Selv om det er et team, der leverer, er det i sidste ende produktejeren, der beslutter.

Backlogejer
Som ejer af backlog ejer man ikke så meget. Til gengæld har man et stort ansvar.

I traditionelle scrum-projekter ejer produktejeren samtidig backlog. Scrum er opfundet til interne udviklingsafdelinger, så i projekter hvor "rigtige kunder" er med, har man typiske en produktejer, der "repræsenterer" kunden.
Det princip kan vi ikke lide. Vi vil hellere have kunden med i projektet. Direkte.

Derfor opfandt vi rollen at eje en backlog. Det er stadig produktejeren, der prioriterer. Det var forhistorien - nu til opgaverne.

Som ejer af backlog sørger man for at alle opgaver er oprettet, beskrevet og estimeret.

Man kan uddelegere, men er ansvarlig for overblikket. Sager, der er estimeret, sendes til produktejeren til godkendelse.

Efter sprintmøder opretter ejeren af backlog nye sager og sørger for, at de bliver behandlet som de skal. Når opgaverne er på plads, tager scrummasteren over.

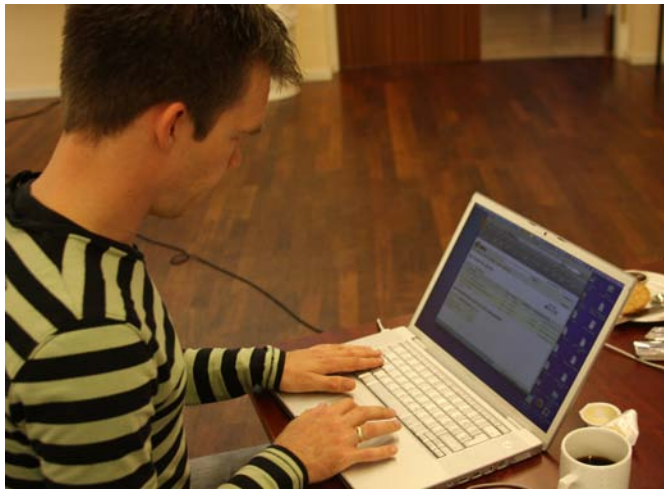
Scrummaster
En scrummaster har ansvaret for processen. Det er ikke nødvendigvis et fuldtidsjob, men det er en vigtig opgave.

Som scrummaster fjerner man forhindringer for de andre, holder produktiviteten i top og driver processen fremad.

Man er ikke mor for de øvrige deltagere, men nogle gange er det tæt på.

En scrummaster planlægger sprintmøderne, laver burndown-charts, tager initiativ til det daglige stand-up møde og tager affære, hvis et område i projektet er gået i stå.

I vores projekter har scrummaster en betydelig andel i den samlede succes. En god scrummaster er dygtig til at kommunikere og motivere, og har et øje på hver finger når det gælder om at spotte problemer inden de opstår.



Udviklere

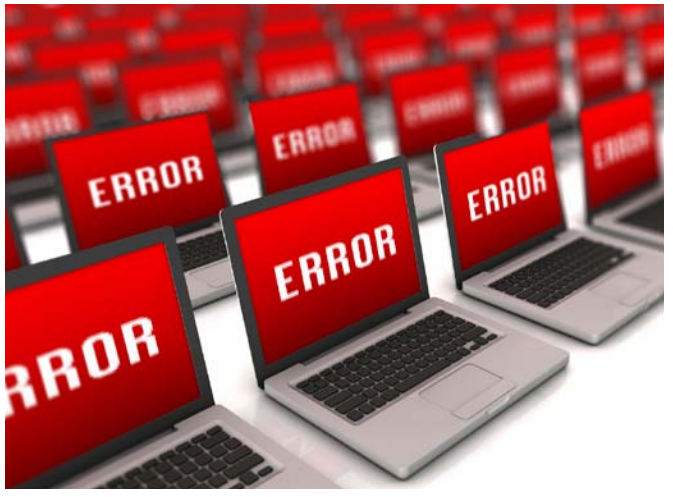
Er alle udviklere ens? Er alle kvinder ens? Heldigvis ikke (her er der så plads til fortolkninger).

Medlemmerne i vores team har ofte forskellige specialer, som er meget forskellige:

- Grafisk designer. Laver websitets stil og tone.
- Interaktionsdesigner. Sørger for at det arbejde, den grafiske designer har lavet, også kan bruges af rigtige mennesker.
- Front-end udvikler. Transformerer interaktionsdesign til kode, der ligner det, det skal.
- Back-end udvikler. Sørger for at front-end udviklerens kode ikke kun ligner, men også fungerer.

Andre roller omfatter søgemaskineoptimering, brugervenlighed, test og dokumentation.

Et team sammensættes efter behov, og vi sørger for at ethvert team er besat af eksperter indenfor alle roller.



Testere
Som tester er det legitimt at efterstræbe nedbrydningen af eksisterende eller netop udviklet materiale. Fejl er godt og testeren elsker dem - altså når de identificeres før udgivelsen.

Testeren godkender eller afviser løbende det testede, og er på den måde ansvarlig for, at et område er klar til at indgå i et givent sprint eller endeligt kan udgives.

Selvom testeren kan betragtes som udviklerens ærkefjende, er det god skik som tester, at supplere fundne fejl med eventuelle løsninger.

Så faktisk er testere slet ikke så slemme endda.

Budget

Vi stiler mod faste budgetter, ufravigelige deadlines, overholdte aftaler, tilfredse kunder og glade udviklere. Scrum kan hjælpe til med at nå et langt stykke af vejen, men det kræver en rolig og bestemt hånd at holde projektet på ret kurs.

Sprintmøder: Se fremad og bagud

Et helt projekt består af en række af små forløb kaldet sprints. Den typiske længde af et sprint hos os er 14 dage.

Hvert sprint afsluttes med et møde, hvor det der er lavet i sprintet bliver afleveret, og der bliver prioriteret opgaver til det næste sprint.

I slutningen af sprintet sørger ejeren af product backlog og scrummasteren for, at nye sager er beskrevet og estimeret.

Scrummaster sørger for et burndown chart til sprintmødet, så alle kan se, om tidsforbruget er på sporet.

Produktejeren kan prioritere inden sprintmødet, når det går rigtig godt.

I næste sprint gennemføres test af brugervenligheden af det sprint, som netop er afleveret.

Dage

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Leverance i 100% kvalitet
Færdig installation på testserver

Burndown chart
Usability-test

Specifikationer
Estimater

Ret til at skifte mening?

Normalt giver store ændringer i projektet relativt langt henne i udviklingsprocessen anledning til Kina-syndrom i både budgetter og projektdeltagernes nattesøvn.

Scrum tillader i langt højere grad, at man kan tillade sig at skifte mening undervejs. De korte sprint giver muligheden for at reagere hurtigt, hvis projektet af den ene eller anden grund skifter idegrundlag og/eller forudsætninger.

Retningsskiftet i projektet udspringer som oftest af ét af 4 nedenstående områder.

1. Vi er lige kommet på noget smart, som vi gerne vil have med...

Produktejeren i projektet tilføjer nye elementer til den endelige løsning. Det kan for eksempel skyldes resultatet af en brugertest, eller at elementerne simpelthen er blevet overset i dataopsamling i produktejerens organisation. For at undgå at budget og deltagere lider overlast, kan man enten udsætte deadline eller tilføre flere ressourcer til projektet. Det er også en mulighed at fjerne andre features for at gøre plads til de nye elementer. Egentlig meget sund fornuft...

2. Ryk tilbage til start

Forudsætningerne for projektet ændrer sig. En leverandør til projektet kan af den ene eller anden grund ikke opfylde sine forpligtelser.

Igen kan man hurtigt reagere, for da ingen delopgave i projektet bør være større end ca. 6 timers arbejde, kan leverandørskifte som oftest foretages med et relativt lille fald i produktiviteten. Selvfølgelig forudsat at andre ressourcer er tilgængelige.

3. Det ligger et eller andet sted imellem 3 og 50 timer

Et eller flere estimater holder ikke i praksis. Det betyder igen, at projektejeren skal afgøre, hvilke at de resterende elementer der skal overleve, eller om budgettet skal revideres eller deadline udsættes. Hovedsagen er, at spildtiden ikke bør være meget over 6 timer.

4. Vi går altid over budget

Et projekt bør altid estimeres med et overhead i budgettet. Det gør det muligt at agere i forhold til nye forhold og muligheder.

Et forventet overhead på ca. 20% er efter vores erfaring passende for de fleste projekter. Det giver frihed til at træffe de rigtige beslutninger, selvom det ikke altid er de nemmeste.

Et projekt med 5% overhead kunne lige så godt være lavet efter vandfaldsmodellen. På den anden side kunne et projekt med 50% overhead lige så godt være estimeret af Stein Baggers revisor...

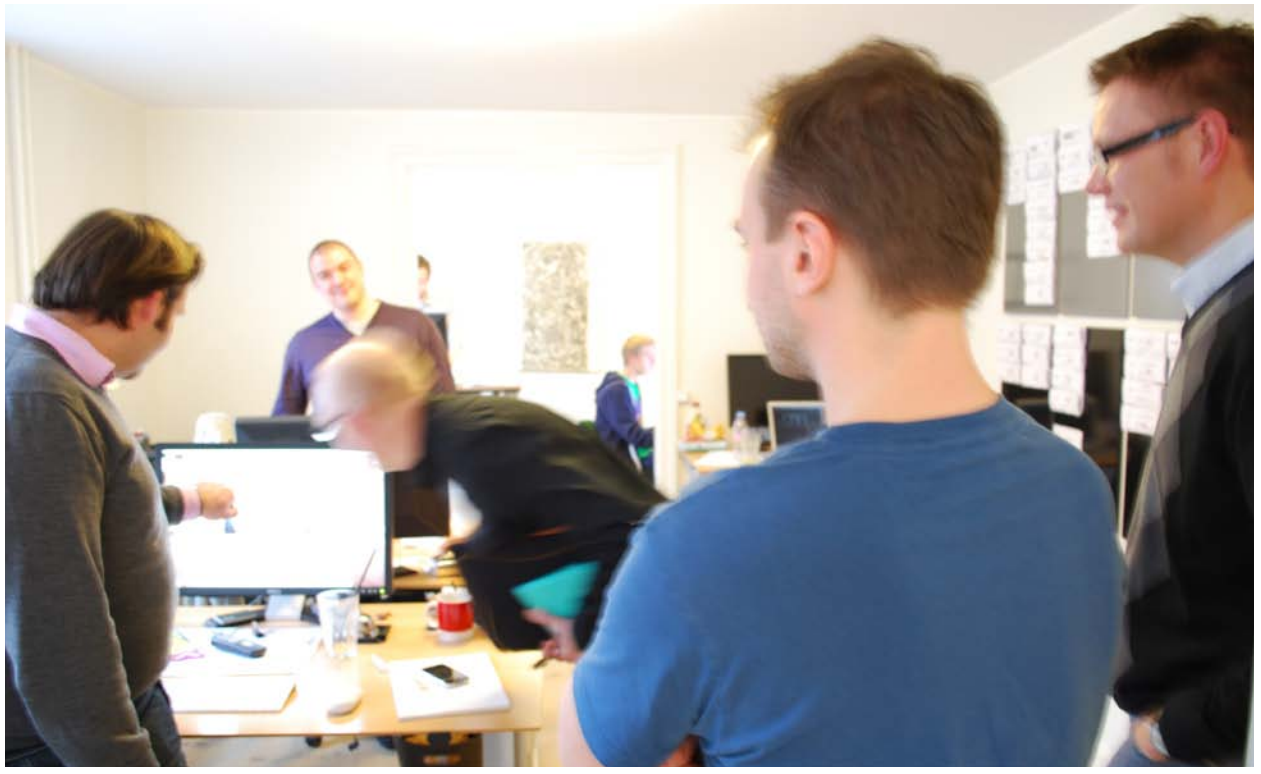
Arbejdsdagen

Der er brug for to redskaber, til at holde styr på alle trådene i et projekt:

1. Et kort, dagligt møde til overblikket
2. Et system til det løbende arbejde med detaljerne.

Hos os klares overblikket i det daglige stand up-møde, og detaljerne styres i FogBugz.

Dagligt stand up-møde



Under det daglige stand up-møde, holder deltagerne i projektet hinanden orienteret om, præcist hvor den enkelte befinder sig i projektet. Mødet skal være kort. Det må ikke være en belastning for nogens arbejde. Derfor står man op. Møder, hvor man sidder ned, er altid længere.

Stand up afholdes hver dag på samme tidspunkt af dagen. Hvert teammedlem har 3 minutter til at svare på 3 spørgsmål:

- Hvad har jeg lavet siden i går?
- Hvad skal jeg lave til i morgen?
- Hvad forhindrer mig i at komme videre?

Og så tilbage til arbejdet.

Ti tip om oprettelse af cases i FogBugz

1. Der skal oprettes en case, når der er noget, der skal tages hånd om i projektet.

2. Den generelle regel er én opgave én case. Det er vigtigt at den enkelte case ikke bliver for lang. Man bør begynde at være opmærksom, når en case breder sig over tre skærmfulde eller mere.

3. Et estimat skal omfatte såvel kodetid som test og evt. behov for dokumentation.

4. Estimater må ikke overskride 6 timer. Breder en opgave sig til mere end det, skal den brydes op i delopgaver. Det er bedre med to cases med estimat på 3 timer end en med et estimat på 6.

5. Der må ikke arbejdes på cases uden estimater.

6. Når et estimat hæves, skal det forbi produktejeren til godkendelse.

7. Bagatelgrænse: Hvis noget kan fixes på under to timer, kræver det ikke godkendelse fra produktejeren.

8. Det er vigtigt at linke mellem relevante cases. Skriv f.eks. "Case 3000" (uden anførselstegn), så linkes der automatisk.

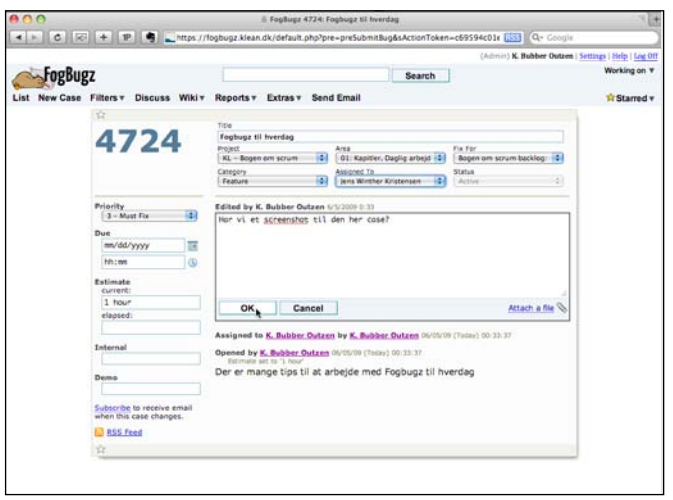
9. Når man får assignet en case skal man håndtere den i forhold til, hvordan den er prioriteret. Man skal ca. en gang i døgnet gennemgå hele listen af cases, der er assignet til én.

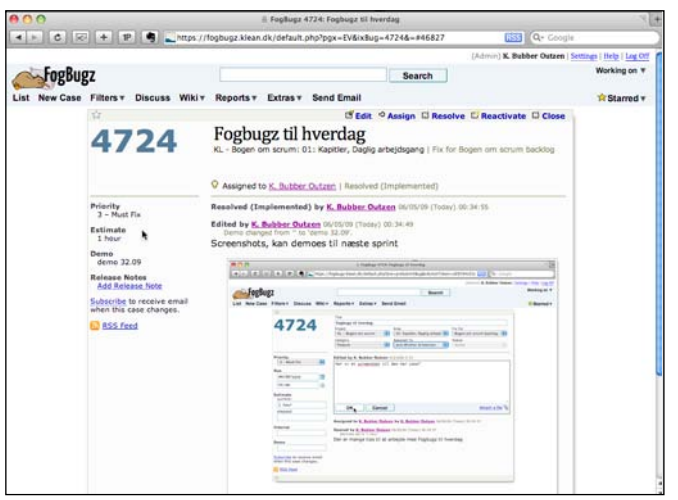
10. Hvis man ikke kan forholde sig konkret til en case - f.eks. hvis man har for mange andre ting kørende - skal man melde dét tilbage til den person, som har assignet casen til en.

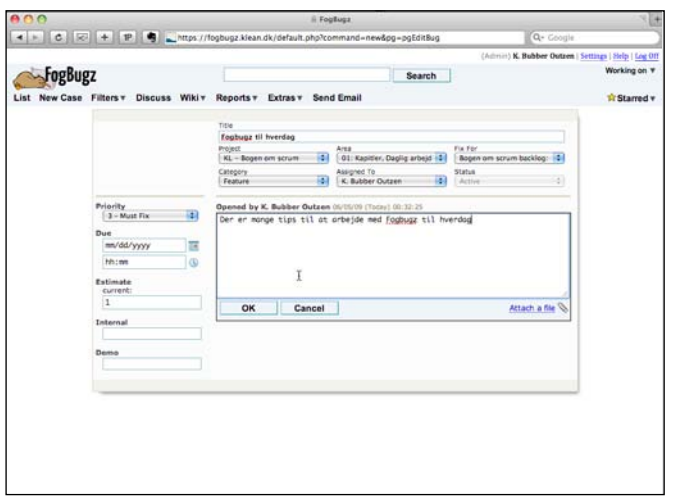
Fem tip til indholdet i en case

1. En meningsfuld titel
2. Et diagnosticeringsestimat på 1 time.
3. En beskrivelse af steps to recreate.
4. Beskrivelse af forventet opførsel.
5. Link til den side, hvor fejlen findes eller Screen shot.

FogBugz til hverdag







Opret en ny sag

1. Man opdager som medlem af teamet et område, som trænger til forbedring.
2. Sagerne i projektet kigges igennem for om fejlen allerede er rapporteret.
3. Så opretter man en diagnosecase.
4. Med mindre andet er aftalt, eller hvis man er i tvivl, skal diagnosecasen nu assignes til scrummasteren i projektet.
5. Når man assigner skal det ske med en bemærkning om, hvad man forventer, der videre skal ske. Et spørgsmål er altid godt at komme videre fra.

Du får assignet en case til dig

- Kan du svare på det, der bliver spurgt om?
- Ja? Svar og send tilbage.
- Nej? Bed afsenderen om flere oplysninger og send tilbage. Eller giv et bud på, hvem der så kan hjælpe, og send tilbage til afsenderen.

Sende en case til test

- Det er udvikleren, der sender en case til test hos den test-ansvarlige, når opgaven er løst og løsningen er efterprøvet.
- Når en case sendes til test skal det fremgå, hvordan den mest optimalt kan testes.

Godkendelse til fremvisning i et sprint

- En feature kan først godkendes til fremvisning, når den er kodet, testet og dokumenteret.
- Det er den testansvarlige som afgør, om kvaliteten er god nok til fremvisning.

Lukning af en case

- Når en sag er afsluttet, skal den sendes til lukning hos scrummaster.

FogBugz

List New Case Filters▼ Discuss Wiki▼ Reports▼ Extras▼ Send Email

Filter: All open cases in KL - Bogen om scrum

PRIORITY 1 – MUST FIX CASES

Case	Title	Status	Opened By	Priority
4713	Final edit: Workshops intro: En god metode til at finde den korteste vej	Active	Christian Tilsted	1 – Must
4634	Layout af indholdsfortegnelse Indholdsfortegnelsen skal layoutes. Mockup	Active	Anders Ellersgaard	1 – Must
4648	Tekst: Kolofon	Active	Christian Tilsted	1 – Must
4649	Tekst: Indholdsfortegnelse	Active	Christian Tilsted	1 – Must
4720	Final edit: Introduktion case 4650 HVORFOR UDVIKLE MED SCRUM? Sc	Active	Christian Tilsted	1 – Must

Add Case

PRIORITY 2 – MUST FIX CASES

Case	Title	Status	Opened By	Priority
4691	Tekst: Intro "Kolofon" Kort introtekst til kapitelside.	Active	Anders Ellersgaard	2 – Must
4606	Projektteam Jens 22759023	Resolved	K. Bubber Outzen	2 – Must

Add Case

PRIORITY 3 – MUST FIX CASES

Case	Title	Status	Opened By	Priority
4708	Final Edit: Nedbrydning af opgaver * Intro * Nedbrydning * Estimering '	Active	Christian Tilsted	3 – Must
4709	Final edit: Om Klean Vi startede Klean for at være med til at lave bedre	Active	Christian Tilsted	3 – Must
4711	Final Edit: Sprintmøder SPRINTMØDER: Der er sprintmøde hver anden u	Active	Christian Tilsted	3 – Must
4715	Final edit: Kvalitetsstyring Værsgo' her er den samlede..	Active	Christian Tilsted	3 – Must
4718	Final Edit: Daglig arbejdsgang Introtekst: Der er brug for to redskaber,	Active	Christian Tilsted	3 – Must
4721	Final edit: Termer Ill. Post Mortem INTRO Scrum er en udviklings- og	Active	Christian Tilsted	3 – Must
4609	Afklaring af deadline for aflevering af indhold til layout	Active	Søren Erland Vestø	3 – Must
4639	Tekst: 10 bedste FogBugz tricks Kanon. Kim, til layout. Søren og jeg kan	Active	Søren Erland Vestø	3 – Must

Sprintmøder

Der er sprintmøde hver anden uge. Alle medlemmer i et team deltager, og mødet tager maksimalt seks timer. Ofte går det langt hurtigere. Vores mål er, at mødet tager 1-2 timer, men det afhænger af projektets type.

Aflevering af et sprint

På sprintmødet afleveres opgaver, der er løst 100%.

En opgave skal være kodet, testet, fejlrettet og dokumenteret.

Er en udvikler 99% færdig, rundes der altid ned til nul. Det er en hård virkelighed, vi lever i.

Post Mortem

Vi tager en snak om det afdøde sprint.

Var det en god proces, har alle en god mavefornemmelse, hvad har vi lært og hvad skal vi gøre bedre i næste sprint?

Ja, der er behov for at tale om processen og hvordan projektet og arbejdsglæden kan blive bedre.

Priotering

Vi planlægger de næste to ugers arbejde i fællesskab.

At løse opgaverne er et fælles anliggende, men der er altid naturlige måder at fordele de første opgaver på.

Resten tager vi i løbet af sprintet, når vi kan se, hvem der har tid og overskud til at hjælpe andre i mål.

Links om scrum

http://www.klean.dk/metode/scrum
http://en.wikipedia.org/wiki/Scrum_(development)
http://jeffsutherland.com/oopsla/schwapub.pdf

Nedbrydning

Nedbrydningen af opgaver i overskuelige underopgaver er en vital del af scrum-processen.

Nedbrydningen resulterer i produktets backlog og skal give produktejeren et realistisk billede af arbejdets omfang.

Samtidig giver den teammedlemmerne en "arbejdsseddel", idet underopgaverne så vidt muligt skal være atomare enheder.

Nedbrydning

Nedbrydning af et delområde i projektet foretages af bedste mand på teamet. "Bedste mands bedste bud" er en god retningslinje.

Til de tekniske nedbrydninger kan man med fordel tage udgangspunkt i en samling userstories, der beskriver det, der skal implementeres. En userstory er et par sætninger, der i uformelle vendinger beskriver en isoleret funktionalitet.

Eksempel: "Brugeren udfylder adressefeltet og trykker 'Søg butik'. Kortet opdateres med de 5 nærmeste butikker".

Med udgangspunkt i mængden af userstories kan man nu identificere de konkrete opgaver, der skal implementeres.

Estimering

Estimeringen af de enkelte opgaver danner udgangspunktet for hvor mange opgaver, der kan lægges i et sprint.

En næsten ufravigelig regel for estimater er, at de ikke må overstige 6 timer for en enkelt opgave. Hvis estimatet er større end 6 timer, skal opgaven nedbrydes yderligere.

Hav i tankerne, at et estimat er præcist dét; en vurdering af, hvor lang tid en given opgave tager at fuldføre. Man kan være fristet til at estimere for højt for at undgå at løbe tør for tid, men i den forbindelse er det vigtigt at bemærke, at estimatet skal være så realistisk som muligt.

Hvis (når..) det sker, at estimatet på en opgave viser sig at være for lavt, er det vigtigt, at det bliver meldt ud, således at produktejeren har en mulighed for at vurdere konsekvenserne og eventuelt omprioritere opgaverne.

Prioritering

Produktejerens prioritering danner udgangspunkt for, hvilke opgaver, der skal tages med i et sprint, og er dermed hans bedste mulighed for indflydelse på implementeringsforløbet.

I sprintet løses opgaverne så vidt muligt i prioritetsrækkefølge ovenfra og ned. Dog kan det forekomme, at opgaver med en naturlig sammenhæng har fået forskellig prioritet. Så er det selvfølgelig tilladt at omgå prioritetsrækkefølgen.


HELLO

Kvalitetsstyring

Dårlig kvalitet betyder svær vedligeholdelse. Svær vedligeholdelse betyder DYR vedligeholdelse. At satse på go' kvalitet er derfor den bedste investering for dit projekt;
- Du sparer penge i længden og resultatet er bedre for alle parter - OGSÅ slutbrugeren!

Hvorfor skal man have et testmiljø?

Det er dyrt at teste

Hvis man lader være med at teste, vil omkostningerne til test blive den største post på budgettet.

Det lyder skørt, men manglende test resulterer i dårlig kode, og dårlig kode vil give projektet en rædselsfuld økonomi. Normalt siger man, at "Total Cost of Ownership" er fem gange prisen af fase 1, men tro os når vi siger, at det kan gå langt værre.

Man kan teste funktioner, brugervenlighed og koncepter.

Det rigtige er at teste det hele. Lad os starte med den funktionelle test.

Funktionel test

I et godt projekt installerer man tre servere. En server til udvikling, en server til test og en server til produktion. Det sidste er det, ikke-nørder kalder drift.

Udviklerne skriver kode på deres udviklingsserver og den ansvarlige for test godkender al kode i slutningen af hvert sprint.

Derefter må det flyttes til testserveren. Denne proces kalder vi deployment.

Efter deployment er der sprintmøde, og her får produktejeren adgang til nyeste version på testserveren. Det er kun kode, der er 100% godkendt, der må flyttes til testserveren. 99% godt nok er en ommer.

Et par fordele:

- Udviklernes små tests og halvfærdig kode ender aldrig på det færdige website ved en fejl.
- Udviklerne er opmærksomme på at levere 100% færdig kode hver anden uge.
- Test af brugervenlighed kan foregå i fred på en testserver, der ikke forstyrres af udviklingsprojekter.
- Produktejeren ved, hvad der er færdigt.

Værdien af en god server.
Uden en server, intet website.

En webserver kan placeres flere steder. De fleste vælger et hosting-center, men man kan også have serveren stående hos sig selv.

Uanset sted, skal driften være i top. Serveren skal overvåges, den skal være bedst muligt sikret mod angreb og den skal være dimensioneret korrekt (læs: Den kan aldrig blive stor nok).

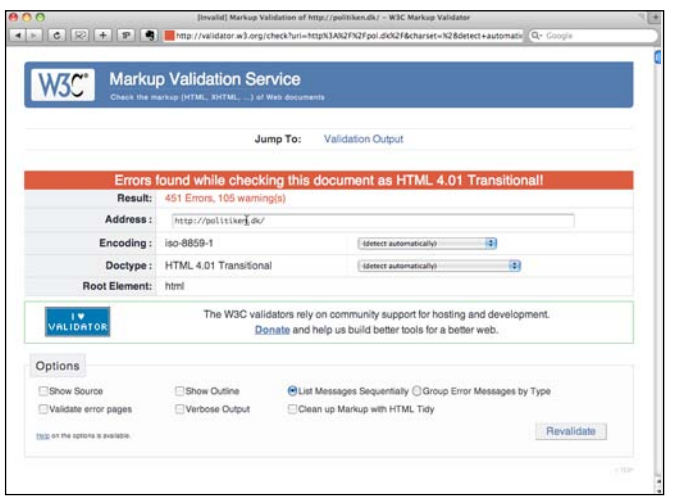
Serveren skal sættes op først. Punktum.

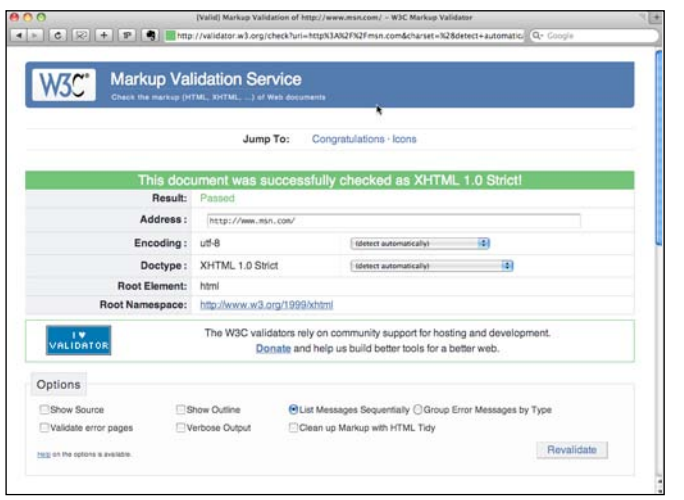
Desuden skal alle i teamet oprettes som brugere og det skal sikres at produktejeren og andre interessenter har adgang til den tilhørende testserver.

Tjekliste for webservere

- Er der 24/7/365 overvågning?
- Er der fuld adgang til driftsrapporter?
- Tages der dagligt backup?
- Er backup-proceduren beskrevet i dokumentationen?
- Er serveren stor nok?
- Har vi licenser til al nødvendig software?
- Har vi en licensnøgle til content management systemet?
- Er der nok båndbredde i hostingcenteret?

Værdien af valid kode





At skrive valid kode handler om kunsten at skrive kode, der overholder de standarder, der er defineret af World Wide Web Consortium.

Det lyder meget simpelt, og det er det også.

Til gengæld er det helt normalt, at kode er skrevet i dårlig kvalitet. Vi skulle hilse at sige, at det ikke tager længere tid at skrive god kode, og heldigvis bliver det også mere almindeligt.

Kode, der ikke validerer, ender ofte med at blive en bombe under budgettet.

Dårlig kode mangler ofte struktur. Det bliver uoverskueligt at rette fejl, problematisk at estimere og kan sidestilles med en tur ud over vandfaldet.

Det behøver ikke være en uoverkommelig opgave at skrive overskueligt HTML, XHTML og CSS-kode som validerer fra starten, og det er en indsats der betaler sig i længen.

Man sikrer ensartethed på tværs af browsere, bedre kompatibilitet med mobile enheder, bedre placering i søgemaskiner, nemmere vedligehold og sansynligvis hurtigere visning af siden.

Hvis man fra starten arbejder imod ensartet, kompatibel og overskuelig kode, vil der være langt færre potentielle faldgruber for udviklingsteamet.

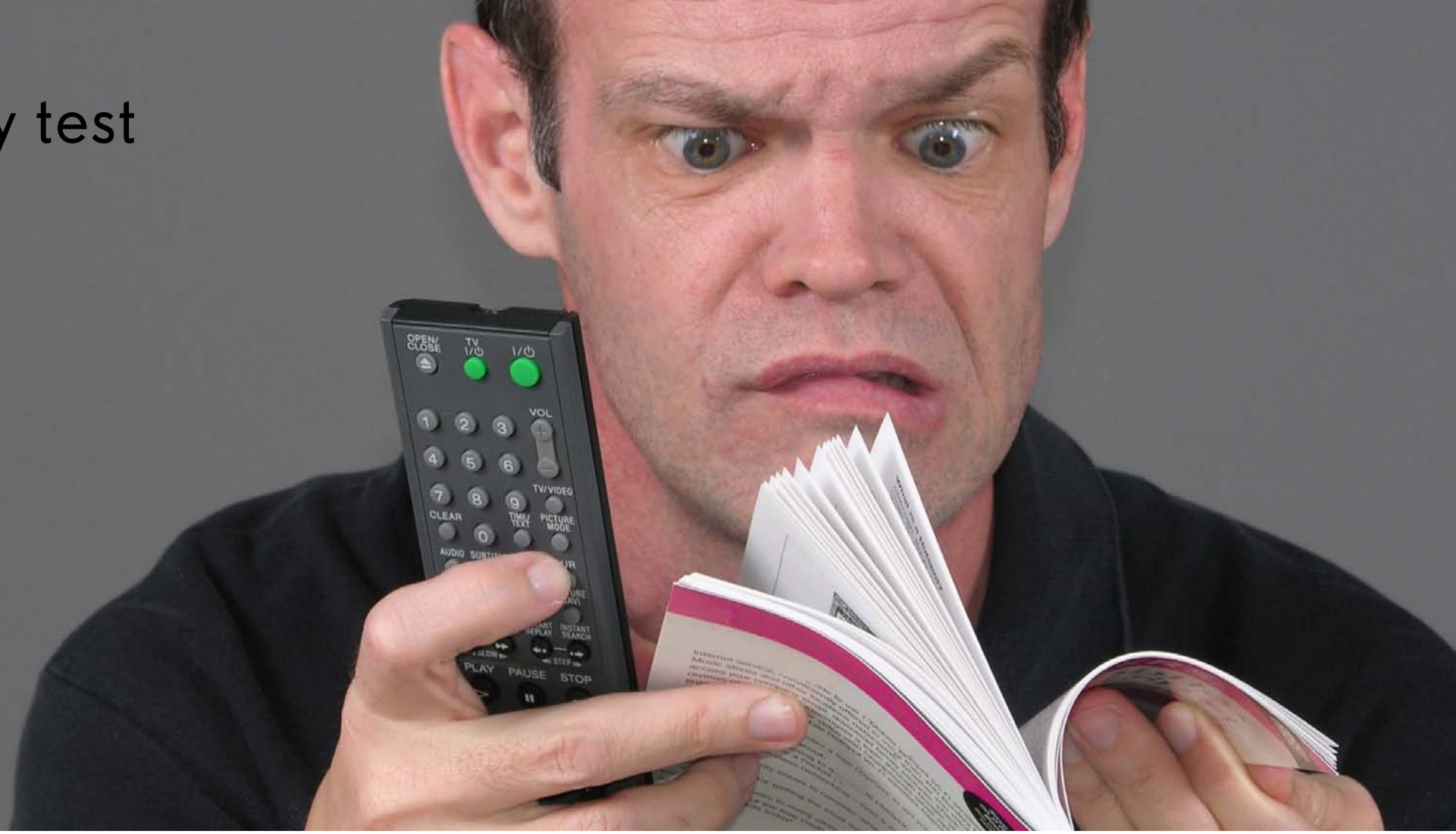
Sansynligheden for at nå deadline og budget stiger kraftigt. Hvem synes ikke godt om det?

Valid kode:

- Ensartet visning på tværs af browsere
- Bedre kompatibilitet med mobile enheder
- Bedre placering i søgemaskiner
- Hurtigere visning af siderne
- Lavere omkostninger til vedligeholdelse af koden
- Lavere omkostninger til nyt design
- Lavere risiko.

På http://validator.w3.org får du hurtig en oversigt over hvad der skal rettes for at din side validerer

Usability test



Usability-test og efterfølgende optimering/bug fixing må ikke foregå som to massive, adskilte processer.

Gør man det, ender man med at grave oversete fejl dybere ned og desuden tvinger man programmørerne til at håndtere fejl eller uhensigtsmæssigheder on-the-fly. Det giver kun sjældent det bedste resultat.

Vi tester i stedet så hurtigt som overhovedet muligt - og vi tester løbende.

Når en ændring er implementeret, skal det nye resultat testes. Det giver den bedste kvalitet og minimerer på lang sigt omkostningerne, da alternativet vil være mere omfangsrig kode(om) skrivning.

Usability test som iterativ proces kræver, at test brydes op i brudstykker - eller cases. En god regel er at oprette én case til hver fejl eller uhensigtsmæssighed.

Det giver overblik og effektivitet for pengene.

Deployment og launch

Deployment

Deploymentprocessen er en vigtig del af kvalitetssikringen i et projekt. Deplyment vil sige, at man flytter en afgrænset klump af funktionalitet fra udviklingsmiljøet til test- eller produktionsmiljø. På den måde sikres det, at der vitterligt er tale om en afgrænset klump, der kan lægges på som en selvstændig "pakke".

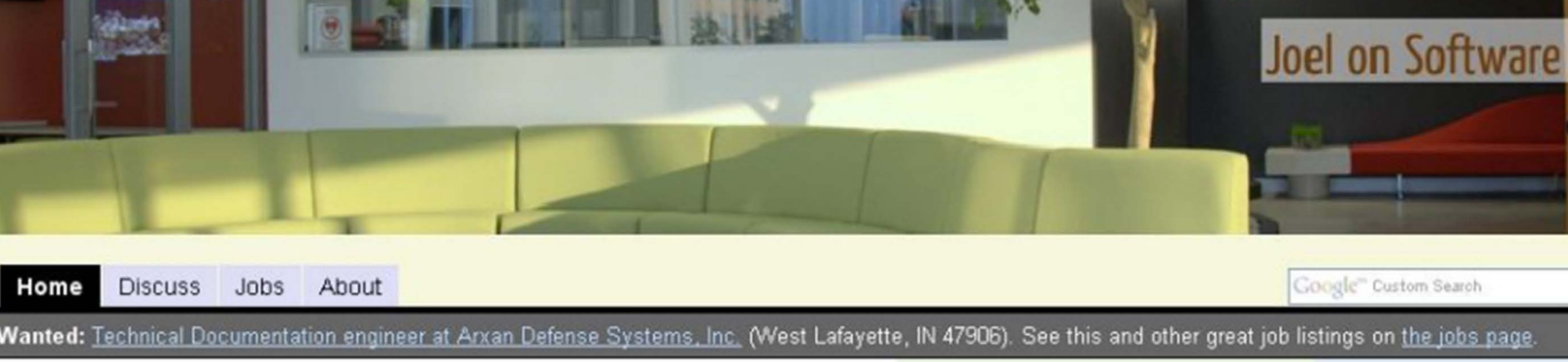
Deployment er en løbende proces i et projekt. Når én opgave er løst, bliver resultatet deployet til testserveren hvor funktionaliteten testes igennem. Efter succesfuld test deployes igen til produktionsmiljøet, som i praksis er den server, der skal drive websitet efter launch.

Launch

Når sitet går i luften, skal resten af verden kunne gå ind på sitet. Det gøres i praksis ved at den ansvarlige (muligvis en ekstern partner) peger adressen ind på serveren (DNS-styring). Efter et par timer har resten af verdens servere opdaget ændringen, og besøgende på websitet bliver betjent af produktionsserveren.

Tjekliste ved launch

- Adgang udefra (Hul gennem firewall etc.)
- DNS - Hvem er ansvarlig
- Favicon
- Title, metadata
- Cross-browser-check
- Proof read
- Døde links
- Funktionality check
- Validering
- Statistik
- Sitemap
- 404-side
- 301 redirect



Hvorfor vi hader Internet Explorer 6

Hvis dit website er optimeret til Internet Explorer 6 er det ikke optimeret til noget som helst andet. Det er et stort problem, og det er skabt af Microsoft, som altid har været langsomme til at følge de standarder, der er lavet til alles fordel.

Heldigvis har Microsoft skiftet mening, så Internet Explorer 7 er i langt bedre kvalitet. Det er stadig en langsom browser, men det er en anden sag.

I dag bruger under 30% af dine læsere Internet Explorer 6. Det er desværre nok til, at noget kode skal skrive to gange. Vi hader at gøre det, men det er nødvendigt.

Hvis du synes, at standarder ikke betyder så meget, har vi i øvrigt nogle Betamax-bånd til salg. Billigt.

Termer

Scrum er en udviklings- og projektmodel.

I modellen findes forskelige roller, nogle helt centrale begivenheder og nogle standardmåder at håndtere ting på; ritualer, om man vil.

Som alt andet i it-branchen findes der også her en herlig liste af trebogstavsforkortelser (TBF) og fagudtryk, man skal kende.

Modellen er opbygget omkring et teamsamarbejde mellem en produktejer, en scrummaster og et udvalgt team.

Her er listen over ord, som er gode at kende.


class
Token
char
char

Bug
En fejl kan bestå af mange ting og kan prioriteres forskelligt. Bugs af prioritet Must Fix bør rettes hurtigt og kan eksempelvis betyde, at brugeren strander eller data tabes.

Bugfix
En rettelse af en bug.

Case
En delopgave i forbindelse med projektet. Alt hvad der skal laves, skal nedbrydes i delopgaver, som er beskrevet i cases.

Feature
Et afgrænset stykke funktionalitet i en løsning.

FogBugz
Det system vi bruger til at håndtere alle opgaverne i vores projekter.

Post Mortem
I forbindelse med sprint reviews gennemgår teamet, hvad der har været godt skidt i sidste sprint.

Product Backlog
Alle de opgaver i projektet, som endnu ikke er prioriteret.

Produktejer
Kunden har ansvaret for at oprette de opgaver, der skal udføres, samt at prioritere dem.

Scrummaster
Afholder sprintmøder og daglige scrum møder. Scrummaster skal sørge for at fjerne alle forhindringer for teamet.


.us
.mx
No froth.

Sprint
En periode på 2-4 uger hvor der udføres de opgaver, der er prioriteret til det pågældende sprint.

Sprint Backlog
Listen over de opgaver der er løst eller skal løses i det givne sprint.

Sprintmøde
Et møde for teamet hvor der planlægges hvilke opgaver, der skal med i næste sprint.

Sprint Review
Teamet præsenterer de opgaver, der er udført i sprintet.

Stand up-møde
Dagligt møde for teamet, som gerne afholdes stående. Det er ofte passende, at hver deltager har 3 minutter til rådighed.

Team
Består ofte af 5-8 personer, som tilsammen har de kompetencer, der skal til for at udføre opgaverne i sprintet.

Ressourcer

Hvis du vil vide lidt mere om scrum og de fordele du kan høste i dit team, kan du finde flere informationer her.

Links
http://www.klean.dk/metode/scrum
http://en.wikipedia.org/wiki/Scrum_(development)
http://jeffsutherland.com/oopsla/schwapub.pdf

Bøger
Agile Software Development with SCRUM, Ken Schwaber and Mike Beedle


The End
NEXT EXIT


etnies
etniesskateboard

Om Klean

Vi startede Klean for at være med til at lave bedre it-projekter.

Vores baggrund er udvikling af web-baseret software. Vi er blandt de mest rutinerede i branchen, hvilket betyder at vi gennem tiden har begået flere fejl end de fleste.

Nu hvor vi selv mener, vi er blevet klogere, vil vi gerne dele ud af det, vi har lært.

Vi arbejder sammen med vores netværk, og derfor inviterede vi nogle gode venner fra C-Tilsted, Codecompany, Mindlab og Webwize til at lave en bog sammen med os.

Det er den, du sidder med her. God fornøjelse. Send dine kommentarer til info@klean.dk. Vi glæder os til at høre din mening.