

Din guide til

Digitale projekter der ikke må fejle

- Skab løsninger der virker i virkeligheden

Den hellige gral?

Digitale løsninger indeholder ofte et stort forretningspotentiale, men også stor risiko. Igen og igen hører du om digitale løsninger der fejler. Men hvorfor?

Alle digitale løsninger ser gode ud på papiret. Men det er afgørende, at de *virker* mindst lige så godt i virkeligheden. Den skal løse de *rigtige problemer* på den *rigtige måde* - for dig, for din organisation og for dine kunder/brugere. Før er løsningen ikke investeringen værd - og før den ikke en succes.

Komplekse digitale løsninger bliver aldrig lette at udføre - men øvelse gør mester. Og vi har øvet os rigtig meget. I denne bog forærer jeg dig frugten af mange års eksperimenter og læringer.

Vi har i mit bureau, [Reload](#), de seneste 12 år stået bag nogle af Danmarks mest komplekse digitale projekter og skabt værdi for blandt andet DR, COOP, TV2, LokalBolig, Dansk Journalistforbund, Gyldendal, Novasol, Københavns Kommune og Google for bare at nævne et par stykker. I hvert projekt har vi lært noget nyt, og i denne e-bog deler jeg essensen af, hvad vi har lært om en fantastisk effektiv, meningsfuld og givende agil proces og samarbejdsmodel.

Mit håb er, at den kan være med til at hjælpe dig med at skabe de bedst mulige digitale løsninger i din organisation.

Dette er vores "hellige gral" 🧑, og nu deler vi det hele med dig. God læselyst.

Hvem er denne bog skrevet til? (og af hvem?)

Hej! Jeg hedder Rasmus 🙌

Jeg har forfattet denne e-bog, selvfølgelig med hjælp fra mine mange dygtige kollegaer i Reload.



Rasmus Luckow-Nielsen
Adm. direktør i Reload.

Jeg drager erfaringer fra min egen baggrund som projektleder, men erfaringerne og metoderne har vi fremelsket i fællesskab. I det følgende er beskrevet vores praksis lige nu.

Inden vi for alvor går ned i materien, så vil jeg lige komme med en disclaimer.

Denne e-bog er skrevet med følgende personer og roller i tankerne:



Projektledere, product owners, forretningsudviklere, agile udviklingsteams og ledere med et digitalt ansvar. Eller i det hele taget bare én af dem, der står med ansvaret for, at dit digitale projekt lykkes og skaber værdi i virkeligheden.

I Reload arbejder vi med at udvikle digitale produkter, og vores erfaringer og metoder er derfor målrettet den verden.

En skattekasse af viden

Så er forventningerne i hvert fald sat 😊
På trods af de mange sider, så er dette ikke en detaljeret guide til alle de berørte emner og metoder. Det ville kræve flere tætskrevne bøger.

Men det er en god introduktion, som jeg håber vil inspirere, og så er det krydret med konkrete tips og ikke mindst læringer, som jeg håber kan gøre, at du og din organisation kan springe en masse bøvl over.

Afsnittene kan godt læses uafhængigt af hinanden, så du behøver ikke at læse det hele, hvis du som de fleste er en utålmodig sjæl.

Ellers find en kop kaffe eller te, find dig godt tilrette, og god fornøjelse med følgende emner:

- Når virkeligheden er kompleks 5
- Løbende planlægning og prioritering 15
 - De tre perspektiver (i design thinking) 16
 - Impact Mapping 19
- Vores agile proces 29
 - Dual track scrum 37
 - Discovery 44
 - Delivery 50
- Dynamiske budgetter 58
 - Fest-budget eksemplet 62
- Teams der kan og vil selv 67
- Aktiv interessentstyring 76
- De syv læringer 83
- Kontakt 84

Når virkeligheden er kompleks

Så har vi brug for at omfavne kompleksiteten - ikke lade som om, den ikke er der.



INGEN PLAN HOLDER NÅR DEN RAMMER VIRKELIGHEDEN.

Det er teamet og metoden, der skaber
succesen.



Vores verdenssyn har formet måden, vi arbejder på

Hvis du skal forstå, hvorfor vi arbejder som vi gør, så skal du forstå vores verdenssyn. Hvis du ikke kan identificere dig med det, så kan du næppe bruge vores erfaringer til meget.

I Reload arbejder vi agilt, fordi vi ønsker at bevare fleksibilitet i de løsninger, vi bygger. **Fleksibilitet** skal bevares for at kunne levere digitale løsninger, som tilfører værdi i virkeligheden for de mennesker, der bruger dem. Vi skal med andre ord være indstillet på at gå andre veje, når det er nødvendigt, og hvor det giver mening.

Fleksibiliteten er essentiel, fordi verden omkring os oftest bevæger sig hurtigere, end vi kan overskue.

Derfor skal vi arbejde på en måde, hvor vi løbende kan styre mod vores mål - og håndtere, at vores mål oftest ændrer sig undervejs.

Agil udvikling er en tilgang, der fokuserer på at levere konkret værdi løbende og som dermed gør det muligt at få tidlig og kontinuerlig feedback fra dem, du vil påvirke med din løsning.

Vores proces og **samarbejdsmodel** er gearret til løsninger, hvor det er svært at overskue det hele. Dvs. projekter eller løsninger hvor der er *store usikkerheder*. Det kan være usikkerhed ift. om det teknisk kan lade sig gøre, om vi kan skabe den rette adfærd hos vores brugere, eller hvor vi reelt er i tvivl om, hvilke af vores mange forretningsmål vi skal jagte.

Når tingene bliver svære eller uoverskuelige, så er det nu, du har brug for en agil tilgang.

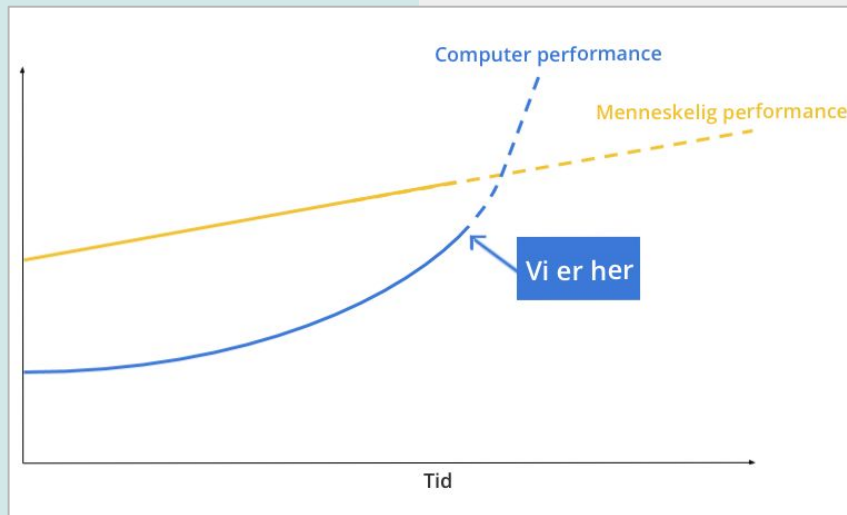
Fremtiden er usikker

Fremtiden er altid svær at spå om. Jeg forventer tilmed, at udviklingen kommer til at gå til stadighed hurtigere i fremtiden.

Særligt på det teknologiske område sker der virkelig meget - og mulighedsrummet åbnes hastigt og eksponentielt.

Det betyder bare, at vi endnu hurtigere skal kunne reagere på en verden i forandring - og det er nok én af vores bedste argumenter for at arbejde på en smartere måde; for at have et agilt mindset. Der er reelt ikke andre alternativer, vi har set fungere.

Det er den verden du, din organisation og din digitale løsning skal agere i.



Hvis computerkraft svarede til dit telefonbatteris levetid, så svarer det til, at hvis batteriet i 2012 holdt én dag, så kan det i dag [i 2018] holde i 800 år. Og om 5 år, så regner vi med at det holder 100.000.000 af år.

- Greg Brockman, CTO, OpenAI

De digitale projekter er næsten altid komplekse

Det ville være skønt, hvis du kunne gå ind i et digitalt projekt med sikker viden om omfang og omkostninger. Men det er desværre rigtig svært. I større projekter* er der simpelthen for mange parametre og bevægelige dele. Sådanne projekter er så komplekse, at man ikke fra starten kan overskue alle detaljer.

Din organisation vil naturligvis gerne vide, hvad det koster, og hvor lang tid det tager. Men det er reelt umuligt at sige, og det er der mange grunde til: Hver løsning er grundlæggende unik, når du ser på det i sin helhed. Hver linje kode er skræddersyet. Hver organisation er forskellig. Og vidensarbejde er ikke en eksakt videnskab på samme måde som at bygge en simpel fysisk genstand. En digital løsning er grundlæggende kreativt arbejde med mange ubekendte.

Der er grundlæggende ingen projekter, der indeholder:

- De samme funktionalitetskrav
- De samme mennesker
- Den samme forretningssituation
- Den samme teknologi
- De samme prioriteter og afhængigheder
- De samme omstændigheder

Der er ingen, der kan overskue alle parametre - simpelthen fordi de grundlæggende er ukendte. Du kan ikke specificere præcist nok, hvad du vil have, på et tidligt tidspunkt.

**) Et "større projekt" er for os erfaringsmæssigt når det overstiger 500 udviklingstimer.*

A black and white photograph of a hand moving a chess piece on a board. The hand is in the foreground, holding a white king piece. The chessboard is in the middle ground, with several other pieces visible. The background is blurred, showing a person's face and hands. The text 'Arbejd målstyret - ikke planstyret' is overlaid on the right side of the image.

Arbejd
målstyret - ikke
planstyret

Drop projekterne! Arbejd målstyret - ikke planstyret

De fleste organisationer forsøger at styre ting ved at pakke leverancer ind i faser og projekter. Vi oplever dog, at der ofte har været manglende klarhed om forventningerne til, hvilke mål der skulle indfries. Typisk har det været projektbudgettet, der endte med at sætte dagsordenen, snarere end omvendt. Men i en foranderlig verden, som den vi lever i, kan vi ikke vente på adskilte og projektpakketerede strategi-, design- og udviklingsfaser. Realiteten er, at små og store behov opstår, presser sig på, eller bliver irrelevante, før vi når at se os om.

Skab et produkt-fokus

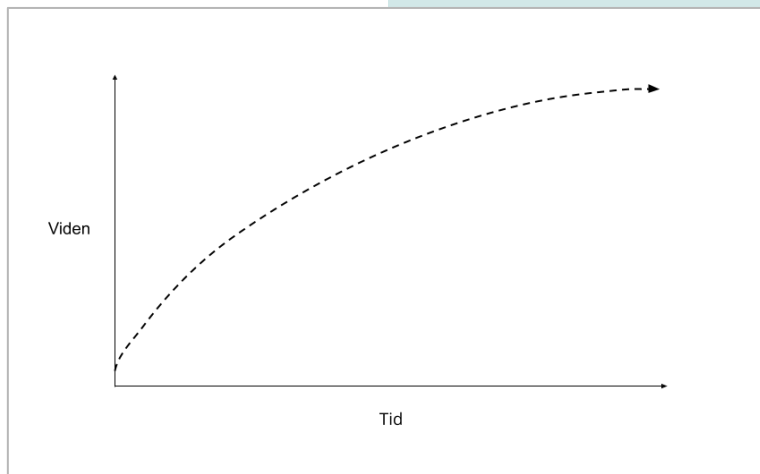
Vi oplever, at de bedste produkter udvikles, når vi udviser grænsen mellem 'drift' og 'udvikling'. Vi er på vej helt væk fra planstyret og tidsbegrænset projektudvikling.

I stedet arbejder vi med målstyret produktudvikling, med et kontinuerligt budget, der afstemmes efter cost/benefit betragtninger, og en fleksibilitet, der giver mulighed for prompte at tage handling på de behov, som dukker op efterhånden som virkeligheden omkring os skifter.

Skab klarhed og læring

Det digitale landskab er komplekst og tæt på umuligt at planlægge meningsfuldt efter. Målet er i stedet, at dit team har et fælles sprog, således I altid er fuldstændig klar over, *hvorfor* I er i gang med at udvikle hvad, i hvilken rækkefølge det skal ske, og ikke mindst holde et skarpt øje på, om I så også faktisk lykkes med at skabe den forventede værdi!

På denne måde skaber I en læringscyklus, så der hele tiden bygges produkter, der virker i virkeligheden.



Planen er død. Længe leve planlægning!

Jo, der skal stadig planlægges. Ellers vil det være kaos, men det giver mening at planlægge i mindre bidder - det, som man i den agile verden kalder for *just-in-time planning*. Og det giver mening, fordi vi over tid helt sikkert bliver klogere - på teknologi, brugerne, markedet, organisationen, forretningsbehovene osv.

Vi skal altså ikke planlægge mindre end ellers - vi skal nøjes med at planlægge, hvad der er behov for og gøre det løbende med mest mulig viden i bagagen. Den viden skal vi opbygge gennem løbende *feedback loops* over tid.

Hvordan gør vi så?

For at kunne sejle agile digitale projekter sikkert i havn, så er det vores erfaring, at det kræver følgende:

- en metode til løbende planlægning og prioritering
- løbende målbaseret budgettering
- teams der har mandat og *kan* og *vil* selv
- teams der opbygger og skaber tillid
- ekstrem høj kommunikationsbåndbredde
- aktiv stakeholdermanagement - behård forventningsstyring
- en udviklingsproces som er både afklarende og implementerende

Alle disse ting har vi eksperimenteret med, udviklet, fejlet og sejret med igennem 10 år - og på de næste sider kommer vores vigtigste erfaringer, som de ser ud lige nu.



[Se mig forklare mange af disse ting \(30 min video\)](#)

Vores agile proces

Vi kalder det for "Reloads agile proces", men det omfatter meget - lige fra værktøjer, metoder og processer til samarbejdsmodel og i sidste ende et holistisk mindset.

For det viser sig, at begynder man at arbejde på en anden måde, så skal man også begynde at tænke på en anden måde for, at det for alvor giver værdi.

Vores proces kan også blive din proces - eller i hvert fald inspirere dig til at forme din egen.

Værktøjer og metoder er relativt nemme at lære. Det svære er at ændre kultur og mindset. Men heldigvis kan værktøjer og metode være med til at forme kultur og mindset.

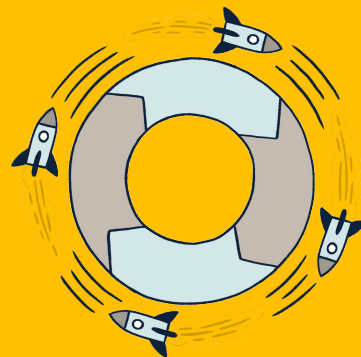
Vi starter med at arbejde målbaseret...



[Se mig forklare mange af disse ting \(30 min video\)](#)

Løbende planlægning og prioritering

Hvad skal vi lave, hvorfor, til hvem og i hvilken rækkefølge?

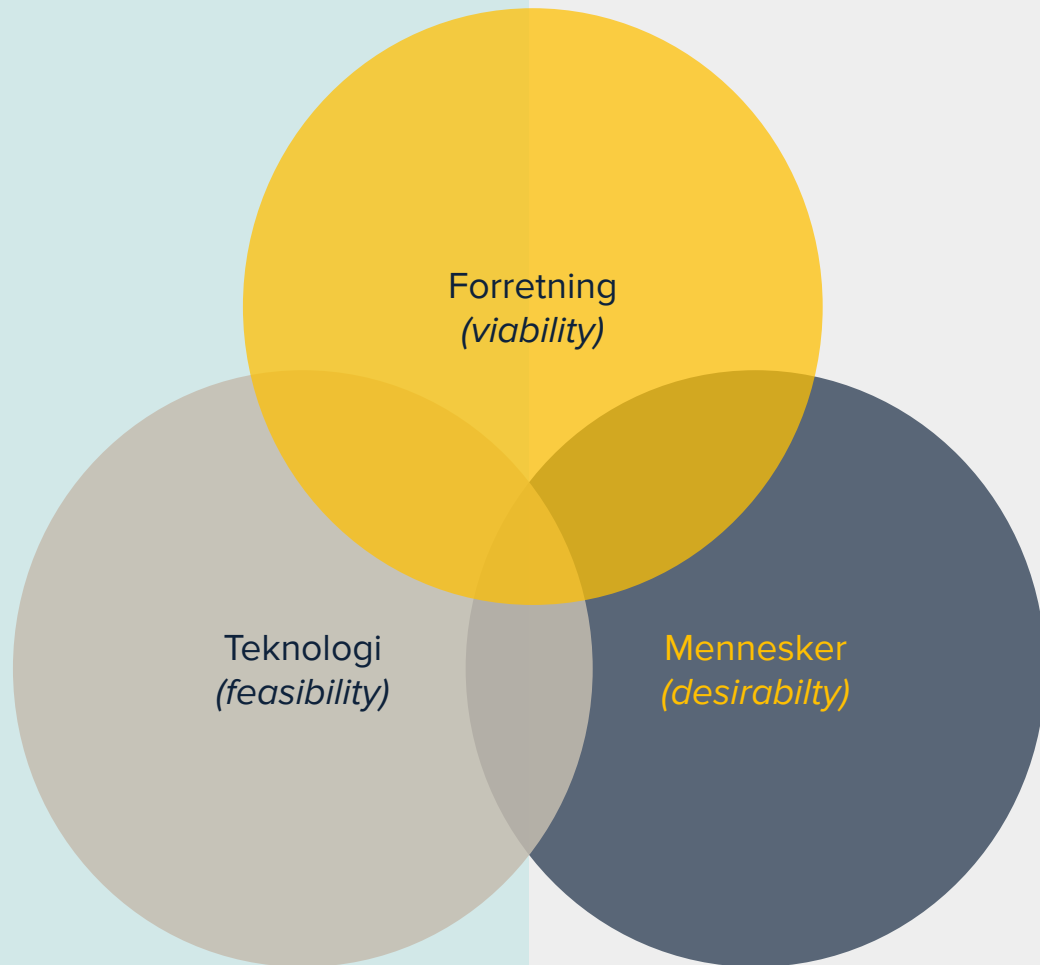


De tre perspektiver

Når vi laver digitale løsninger, så holder vi altid de tre perspektiver i Design Thinking for øje - nemlig hvilken *værdi* skal vi skabe, hvilken *menneskelig adfærd* der skal skabe denne værdi, og hvordan kan vi *bygge noget som skaber den adfærd*?

Du vil opleve, at disse tre perspektiver gennemsyrrer vores model - men det er også helt essentielt, at du sammensætter et team, hvor der er kompetencer der dækker alle disse områder.

For forstå teamet dette, så er der meget, vi ikke behøver at planlægge i detaljer - i stedet sker det af sig selv.



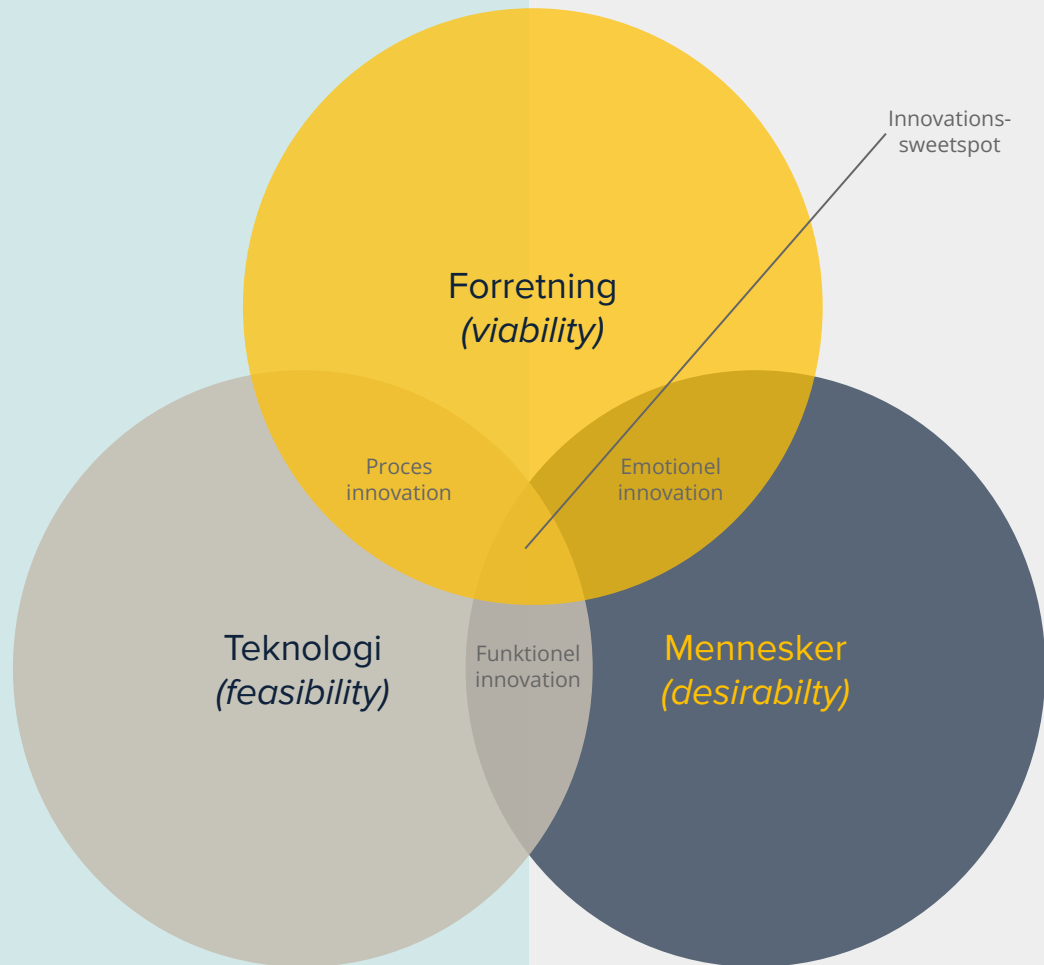
De tre perspektiver

Teamet skal forstå:

Forretningen. Hvorfor investerer vi i digital udvikling, hvilken værdi forventer vi at få ud af det? Hvilke mål er i spil? Og hvordan prioriterer vi mellem dem?

Menneskene. Hvem bruger og/eller er påvirket af den digitale udvikling, hvad er deres motivation og oplevelse, og hvordan kan vi ændre på det?

Teknologien. Hvad har vi af tekniske muligheder og begrænsninger, er der produktrammer, er der arkitektur- og designprincipper, som vi skal arbejde indenfor?



Hvordan får vi et overblik? Der er så mange mål og muligheder?

Når du begynder at kigge på, hvilke forretningsmål der er i spil, så vil du opdage, at der ofte er rigtig mange ting, du og din organisation gerne vil opnå.

I hvert fald når du har fået nedbrudt et overordnet mål (fx "tjene bunker af penge" eller "skabe en god oplevelse") til nogle mindre og mere operationelle mål (og hvis du er rigtig god, så laver du dem som SMART-mål. *Det er så ret svært i praksis...).*

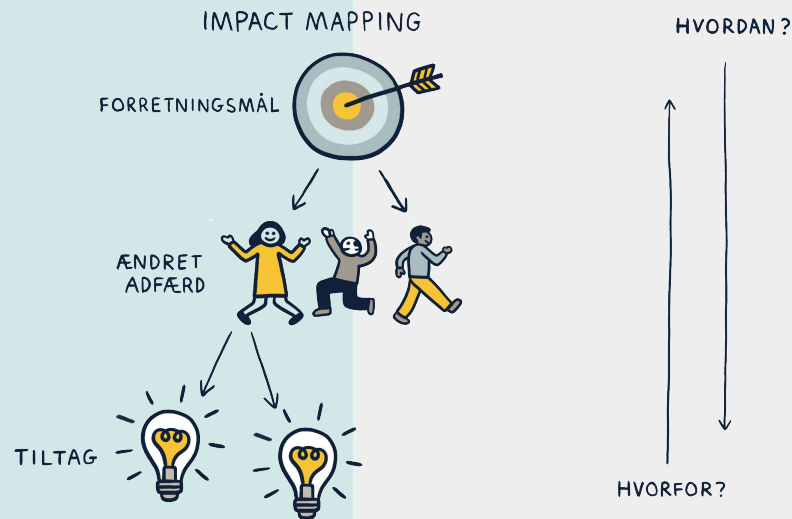
Men det er aldrig en god ide at arbejde på alle målene på én gang - og de har i hvert fald også en intern prioritering - noget *må* være vigtigere end andet. Denne prioritering skal du have de rette folk med på at lave. Og så kan vi begynde at udfolde, hvordan vi kan opnå de mål. Vi plejer at sige: Der er mange veje til Rom. Men vi skal finde ud af, hvilken vej er bedst. Til det bruger vi metoden Impact Mapping...



Impact Mapping

Et Impact Map har netop til formål at kortlægge sammenhængen mellem præcis, hvilke **forretningsmål og værdi** du arbejder henimod, med præcis hvilke konkrete **adfærdsændringer** som kan skabe den værdi og igen med præcis hvilke **tiltag/digitale løsninger**, der kan understøtte disse adfærdsændringer. I virkeligheden skaber du Impact Map strukturen ved at spørge "Hvorfor?" og "Hvordan?" tilpas mange gange.

Du kortlægger derved de tanker og din konkrete viden om de behov og idéer, som er i spil, så du og dit team sammen kan vurdere risici og opdage eventuelle svagheder i bagvedliggende antagelser.



Impact Mapping: En metode til løbende planlægning og prioritering

Jeg forudsætter, at du og dit team har en grundlæggende indsigt og forståelse for den kontekst, I arbejder i og det koncept, I ønsker at arbejde på. Humlen er nu at omsætte jeres viden til et konkret roadmap med eksekverbare to-dos, som samtidig sikrer, at I hele tiden holder et målstyret fokus. Du bruger metoden Impact mapping til at bygge det roadmap, som teamets prioritering og samarbejde tager udgangspunkt i.

Der er jo ingen der investerer i en digital løsning uden at forvente at få noget igen. Øget forretningsværdi, enten i kroner og ører eller i en mere diffus form for værdi. Det er også et faktum, at hverken kodelinier eller produktet ændrer noget i sig selv. For at produktet lykkes med at opnå mere værdi, er der i sidste ende nogle mennesker, som skal agere anderledes end de gør i dag.

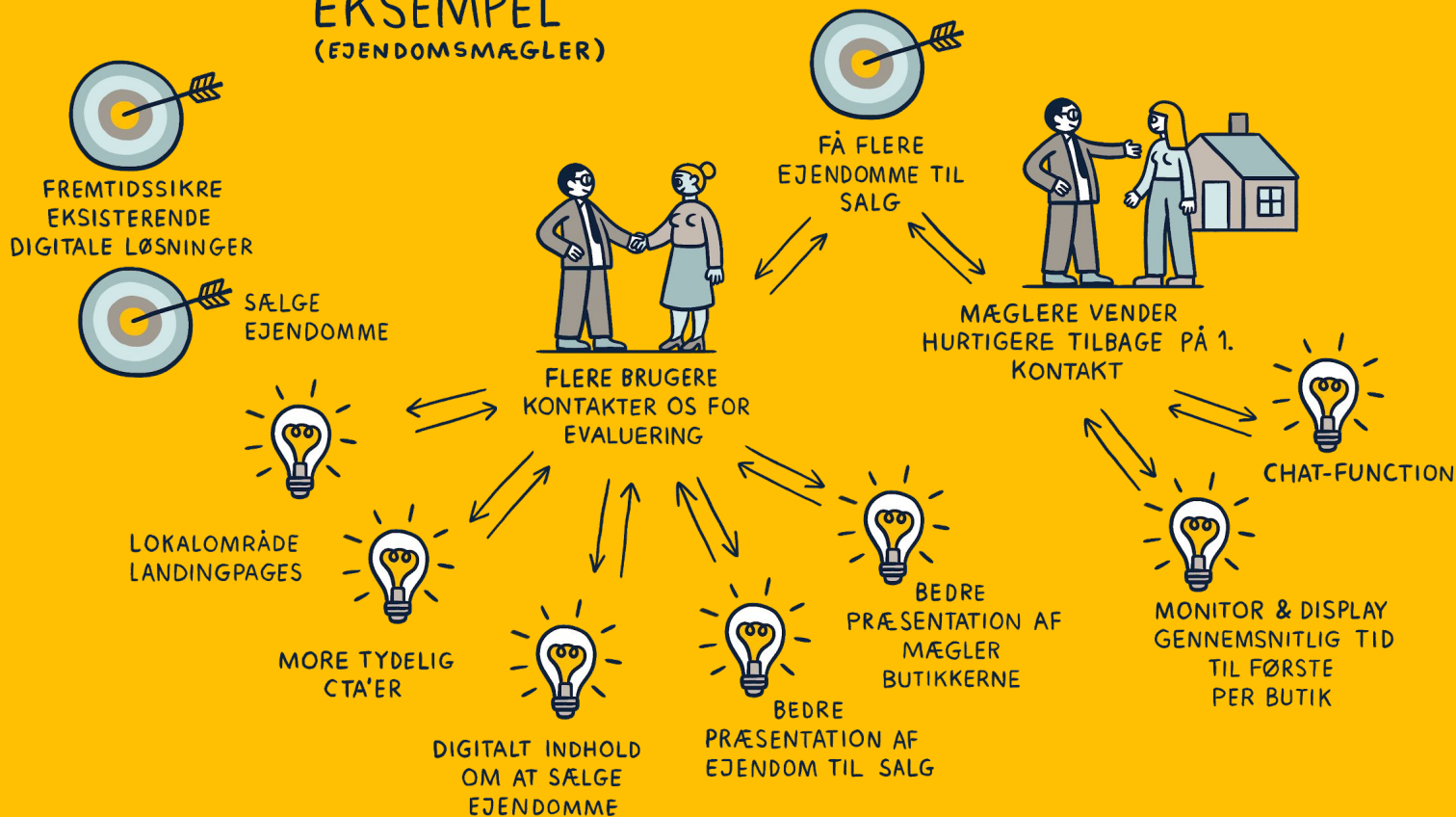
Men du har brug for at gøre tankerne eksplicitte. Og det er et Impact Map rigtig godt til. Lad os dykke ned i det...

Hvordan?



Hvorfor?

EKSEMPEL (EJENDOMSMÆGLER)



Et Impact Map hjælper dig til at arbejde eksplorativt. Et digitalt produkt skal typisk imødekomme en række situationer og består typisk af en lang række delløsninger, måske kaldet "touch points" eller "features". Frem for at strukturere arbejdet i projektfaser, der afgrænser, hvornår du er afklarende og hvornår du er i udvikling med hele produktet på én gang, så kan du arbejde mere fornuftigt med løbende at undersøge de ønsker og idéer, som kalder på det nærmere, mens du går i gang med at udvikle andre.

Vi oplever, at udgangspunktet for et digitalt produkt varierer meget. Nogle kommer med helt overordnede behov og forretningsmål - uden overvejelser om mulige digitale koncepter og løsninger. Andre kommer med tæt på færdige koncepter beskrevet med en



række funktionelle krav og ønsker. Hvis det er idéer med rimelig valide bagvedliggende antagelser, og de understøtter de rette mål, så er det bare at komme i gang med at udvikle. Men er antagelserne rene mavefølelser, så skal der nok arbejdes mere med afklaring, før du sætter i gang med dyr implementering.

Ikke sjældent oplever vi at blive mødt med en række meget konkrete løsningskrav og idéer, men med uklare forventninger til ændret adfærd og værdi - og med uklar prioritering. Selv når alle disse ting er klare og på overfladen ser fornuftige ud, kan Impact Mappet alligevel blotlægge idéer der er bundet op på antagelser, som er fyldt med risici, hvilket bør undersøges nærmere, så du ikke risikerer at spilde en masse penge.

Gør antagelserne eksplicitte

Ved at gøre antagelserne eksplicitte kan du altså bedre vurdere, om I er nødt til at afklare mere for at minimere risikoen for at arbejde ovenpå gale antagelser. Ellers vil det medføre kostbare fejl i, hvordan der investeres i udviklingen af løsningen.

Antagelser kan afklares på mange måder. Det kan være gennem kvalitative metoder som interview, eller ved at opsamle og kigge på data fra eksisterende kilder.

Ofte er det dog først, når løsningerne er ude over rampen, at vi bliver helt klar over, om vores antagelser så faktisk holdte. Derfor advokerer vi næsten altid for, at enhver nye idé skæres helt ind til benet til første release, men til gengæld, at vi har et fleksibelt budget til løbende at trimme og udvikle løsningen ud fra reel feedback.

Vi kan dog ikke validere alt, det kan være for dyrt eller svært - nogen gange må man satse. Men sørg for, at det er et bevidst sats.

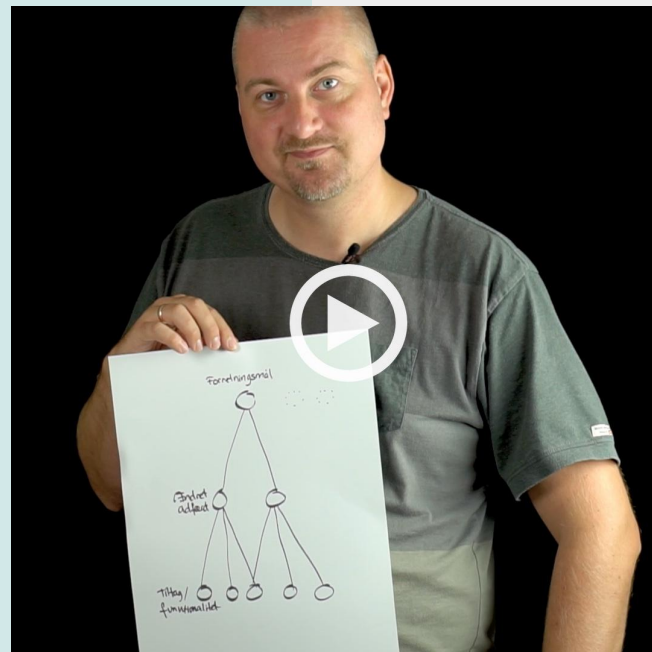
Kunsten er dog at forsøge at validere ens antagelser med den mindst mulige indsats.

Impact mappet giver dig desuden en struktur til løbende at holde øje med, om de skibe du sætter i søen også gør deres job. Medmindre du har med en e-shop eller lign. at gøre, kan det måske være svært at bevise direkte sammenhæng mellem de digitale løsninger, du udvikler, og den værdi du skaber. Men når du eksplicit har formuleret, hvad det er for en ændret adfærd, du mener vil lede til mere værdi, kan du i det mindste holde øje med, om den ændrede adfærd faktisk sker.

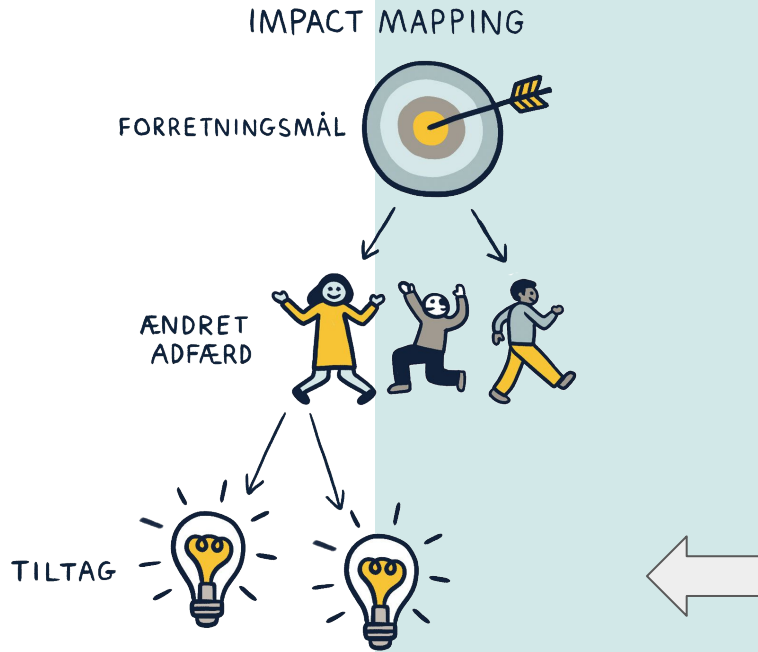
Kortet bliver aldrig komplet

Det er altså IKKE formålet at finde *alle* de gode idéer og validere *alle* antagelser på forhånd. Men blot at få indsamlet og struktureret relevant viden, og derigennem få kortlagt, hvordan landet ligger lige nu: Er der valide idéer med klare formål, som er mere eller mindre klar til at blive ført ud i virkeligheden, eller står vi overfor en mere åben proces, hvor de gode idéer først skal frem i lyset, eller hvor vi skal bruge mere indsigt til at validere dem?

Det er en øvelse, som du bør genbesøge jævnligt. I starten vil man arbejde intensivt med det. Men når et produkt først er i luften, så er det vigtigt at løbende opdatere jeres grundlæggende forståelse. Er der eksterne forandringer, I skal være opmærksomme på? Hvad har I lært siden sidst, og hvordan performer de løsninger, I allerede har sluppet ud i verdenen?



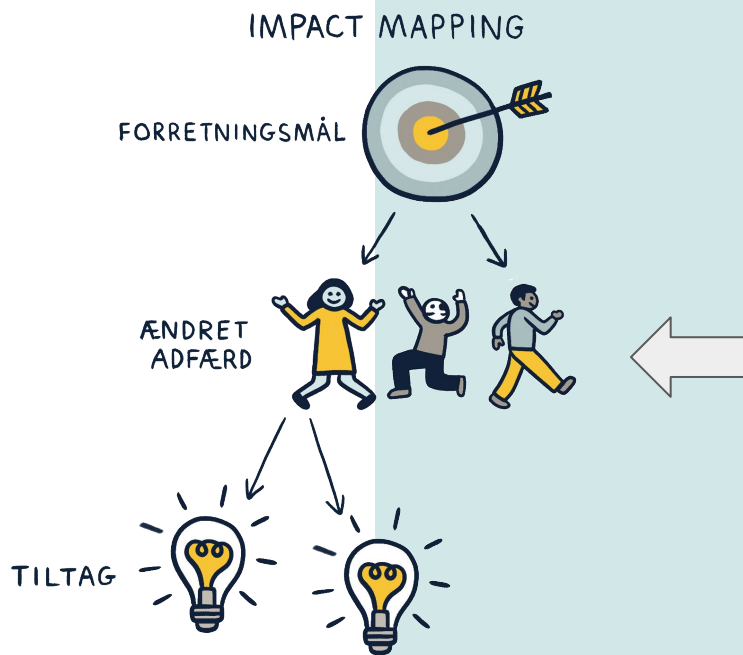
[Se metoden forklaret på video her.](#)



Eksempel fra virkeligheden #1: Features er på plads...

En kommune ønsker et nyt website om kulturtilbud i kommunen, som skal erstatte en række nuværende websites.

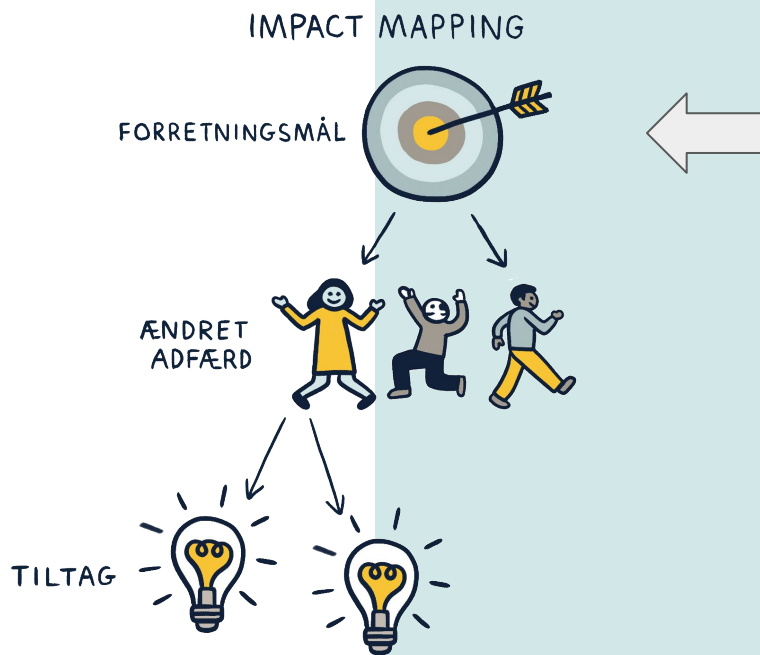
Kunden kommer med 'færdige' feature ideer. Beskrevet ved en forholdsvis detaljeret kravspec, og et designoplæg. De har løse tanker om målsætning og kontekst, men mangler synlighed og kortlægning af relationer mellem forretningsmål og features, således at de kan foretage meningsfulde prioriteringer.



Eksempel fra virkeligheden #2: Adfærd er på plads...

En fagforening henvender sig med en ide om, at deres medlemmer i højere grad skal kompetenceudvikle sig selv.

Kunden kommer med en idé der går på at ændre deres medlemmers adfærd. De har dog ikke formuleret, hvordan det hænger sammen med deres forretningsmål og strategiske vision eller, hvorfor de mener, det er det bedste, de kan gøre for styrke deres position i markedet. Ligesom idéen i sig selv heller ikke er konkretiseret, og derfor endnu er svær at føre ud i virkeligheden.



Eksempel fra virkeligheden #3: Mål er på plads...

En ingeniør virksomhed ønsker at styrke deres Employer Branding og derved få flere velkvalificerede medarbejdere.

Kunden kommer med nogle overordnede strategiske forretningsmål for de kommende år, men er ikke klare på, hvilke adfærdsændringer hos de forskellige målgrupper, der kunne have størst værdi. Skal de sætte ind ift. at få bredt kendskabet til dem ud blandt flere potentielle medarbejdere, eller handler det om at flere af dem, der allerede overvejer, skal overbevises om at tage kontakt?

Få styr på sammenhængen

Dette er tre eksempler fra vores virkelighed, hvor vores kunder typisk kommer med forskellige indgangsvinkler og mangler de to andre niveauer.

Den første kommer med feature-ideer, men har ikke gjort sig klart, hvilken adfærd der skal ændres eller hvilke forretningsmål, der reelt er i spil. Den næste har styr på en adfærdsændring, men mangler både at gøre *hvorfor* og *hvordan* klart. Og den sidste har godt styr på forretningsmålene, men har brug for hjælp til at identificere de nødvendige adfærdsændringer, og hvordan vi får dem til at ske.

Det smukke her er, at vi i alle tilfælde kan hjælpe med at udfylde de manglende brikker i puslespillet, og derved sørge for, at vi har styr på sammenhængene mellem de tre niveauer.

Og det kan du også. Et Impact Map bliver dog hurtigt komplekst at tegne i praksis, fordi niveauerne kan hænge sammen på kryds og tværs.

Men *måden at tænke på* er ekstrem vigtig og kan altid bruges.

Hvis du vil læse mere, så kan vi anbefale [Impact Mapping bogen](#)*, og dette er også et fint online (mind)mapping værktøj: www.mindmup.com

Vi rådgiver og hjælper også gerne andre med at komme godt igang.

**) Den opmærksomme læser vil bemærke, at vi har kogt Gojkos fire niveauer ned til tre. Det er i vores erfaring nemmere at arbejde med i praksis.*

Vores agile proces

En målstyret agil proces som er både
afsøgende og implementerende



A hand is shown from the left side, holding a glowing white ring. The ring has a bright white outer edge and a dark, almost black inner center. The background is dark, making the glowing ring stand out.

SKAB KORTE FEEDBACK LOOPS

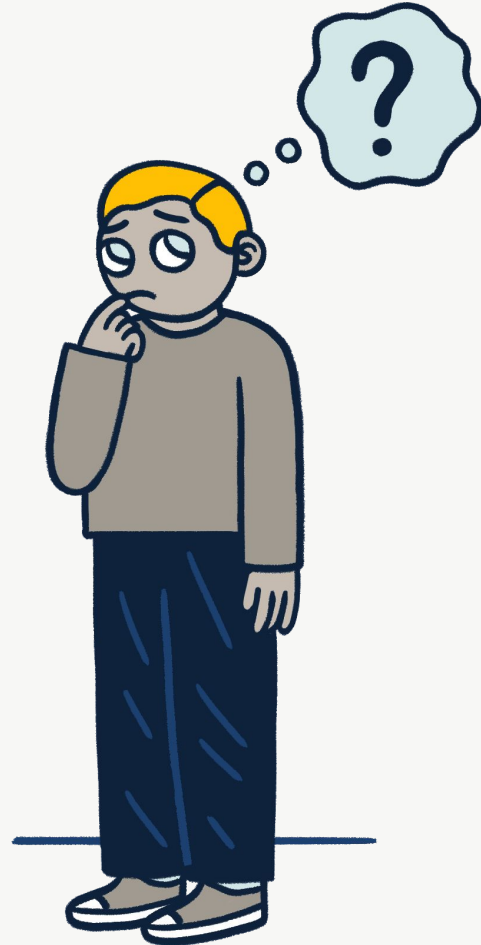
Skab tidlig og hurtig
læring

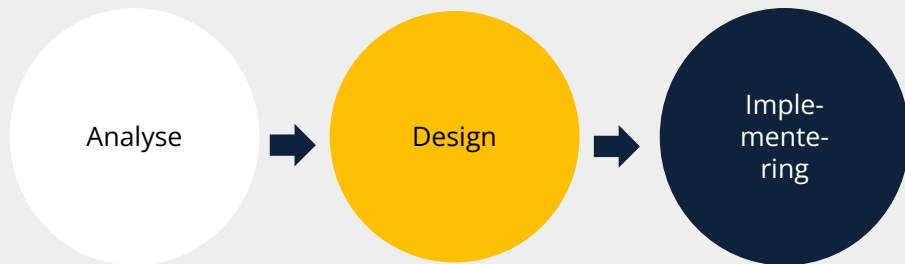
Vi kan ikke lide vandfaldsprojekter

Som du nok har fornemmet, så er vi ikke fans af klassiske vandfaldsprojekter. Dvs. projekter hvor al analyse sker på forhånd, og så aftales et *fast scope* og en *fast pris*. Så skal der bare implementeres - og så er afleveringen klar. Det er opskriften på de fleste (offentlige) IT-projekter, og mange af dem fejler hårdt. Særligt når kompleksiteten i projekterne stiger.

Hvis du fastlåser dig på både økonomi, funktionalitet og tid, så er den eneste parameter, du har at skrue på, når noget uforudset viser sig, kvaliteten! Og det betyder bare, at tingene bliver dyrere i længden.

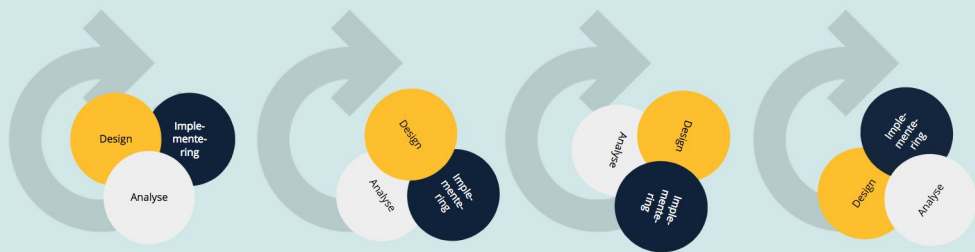
I stedet bør man fokusere på hvilke overordnede behov der skal løses - men ikke præcist hvordan. Lås dig hellere til tid og økonomi end til funktionalitet. Den kan nemt ændre sig og bør være underordnet så længe forretningsmål opnås.





Big bang lancering

Du lærer først, hvordan din målgruppe faktisk handler, når al tid og budget er brugt.



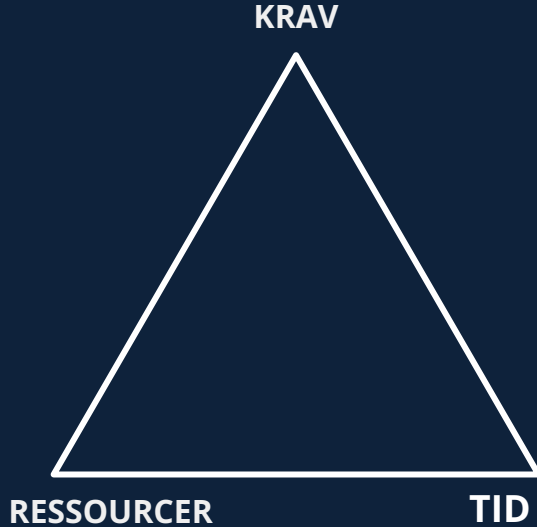
Mindre løbende lanceringer

Du lærer tidligt, hvordan din målgruppe faktisk handler og kan løbende handle på den feedback, du får.

KENDT

FORTID

MINIMERE
RISIKO VED
AT STILLE
KRAV,
ESTIMERE OG
FØLGE
PLANER

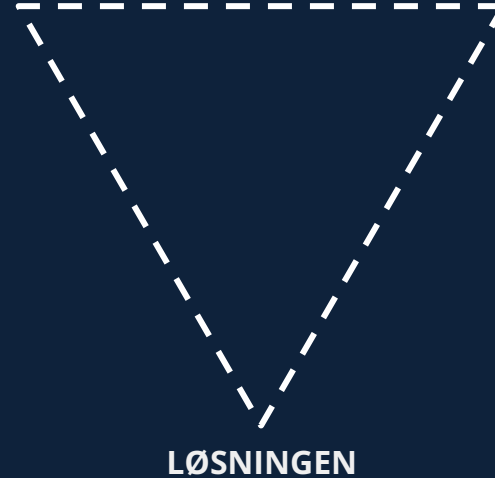


RESSOURCER

TID

FREMTID

MINIMERE
RISIKO
GENNEM
KONTINUERLIG
TILPASNING OG
VED AT LÆRE
AF FEJL



UKENDT

En lærende proces giver større chance for succes

Vi minimerer altså risiko i et projekt ved løbende at håndtere de ubekendte der dukker op. På den måde højner vi chancen for rent faktisk at ende med en succes.

Man bliver klogere undervejs

Typisk bliver I så meget klogere undervejs i et projekt, at I ender med løsninger, der er markant anderledes end først forventet. Kun hvis de relevante dele af organisationen er med til at skabe og trimme løsningen, kan I skabe det værktøj som understøtter jeres digitale virke på bedst mulig vis. Det kræver, at I kan stille med en Produktejer (Product Owner), som kan arbejde uafhængigt og på tværs af siloer, som har fuldt beslutningsmandat inden for de aftalte projektrammer, og har den nødvendige tid til at være en aktiv part i processen.

Vi håndterer risiko ved *ikke* at aftale alt på forhånd

I Reload tror vi ikke på, at løsninger, som er færdig specificerede og aftalt på forhånd, ender med det bedste resultat. Derfor arbejder vi ikke med aftaler som låser os fast på konkrete funktionalitetsleverancer. Det er angstprovokerende at gå ind i et forløb uden at vide, hvad man kommer ud med på den anden ende. Det forstår vi. Det kan virke som en stor risiko, men risikoen for at vi simpelthen ikke fokuserer rigtigt, hvis vi skulle fastlåse løsningen på forhånd, er i vores optik langt større.

Målstyring som grundlæggende koncept

Hvad du i virkeligheden skal gøre, er at styre produktudvikling. Det, at du pakker det ind i en række mere eller mindre arbitrært afgrænsede projekter må ikke styre vores beslutninger om produktet, og at projektet i sig selv lykkes giver ingen garanti for succes samlet set - tværtimod kan det komme til at skabe et alt for kortsigtet fokus.

Vi anerkender selvfølgelig, at du har en projektøkonomi at rapportere op imod og styre indenfor, men fokus er på løsningens fulde levetid og total cost of ownership. I Reload tilrettelægger vi derfor ikke vores proces rundt om det enkelte projekt og taler ikke om "projektfaser". I stedet giver det mening for os at organisere arbejdet i to typer af aktiviteter: Discovery og Delivery opgaver. Dvs. aktiviteter hvor vi afklarer og bliver klogere, og aktiviteter hvor vi bygger og laver færdige leverancer.

Dette er en kontinuerlig proces, som der kan skrues op og ned for, alt efter behov og økonomi.

Resten af dette afsnit beskriver lidt mere jordnært, hvordan en typisk agil proces fungerer i praksis og i hverdagen hos os. Dvs. lige fra møderytmer og nogle af de mange styringsredskaber vi bruger for at komme sikkert frem til målet.

Før I går i gang med udviklingen af en ny løsning, skal I have kortlagt de behov og forventninger der er i spil til jeres løsning, og herudfra evt. skitsere overordnede konceptuelle tanker.

Men først skal der nogle grundlæggende ting på plads...

Fokusområder inden I for alvor går i gang

Hvem er I?

Gennemgang af jeres organisation, vision og de strategiske tanker, der ligger til grund for en ny løsning.

Strategi og målgrupper

Gennemgang af kendte målgrupper og centrale brugerscenarier for jeres nye løsning

Roadmapping

Gennemgang af (high level) feature ønsker og heraf en Impact Mapping som tager højde for betragtninger om strategi og brugeradfærd, samt illustrerer prioriteringen af behov, ønsker og idéer.

Tekniske afhængigheder

Analyse af det tekniske setup herunder særligt fokus på centrale integrationer og særligt komplekse features.

Koncept og principper

Som udgangspunkt for design af en ny løsning udvikles typisk et koncept der beskriver de overordnede principper for arkitektur, applikationsdesign, informationsarkitektur, navigation, interaktion og look & feel.

Konceptet sikrer, at udvikling og specifikation af løsningens forskellige indholdsområder og funktionaliteter tager afsæt i en overordnet ambition og prioritering, som hele tiden tager udgangspunkt i den bærende idé og principper og derfor giver en meningsfuld og konsistent brugeroplevelse.

Vi arbejder med en variation af Dual Track Scrum

I dag er vores proces en variation af det som kaldes Dual Track Scrum. I denne metode har man to kontinuerlige sprint-spør kørende: Et *Discoveryspor* og et *Deliveryspor*.

Deliverysporet er en traditionel SCRUM-proces med løbende leverancer i færdig kode, mens Discoverysporet oftest er en mere flydende proces, idet vi kan være afhængige af udefrakommende kilder, fx interviewpersoner. Det bliver dog stadig struktureret som opgaver i en backlog.

Discovery er altså *afklaring*, mens Delivery er *implementering*.

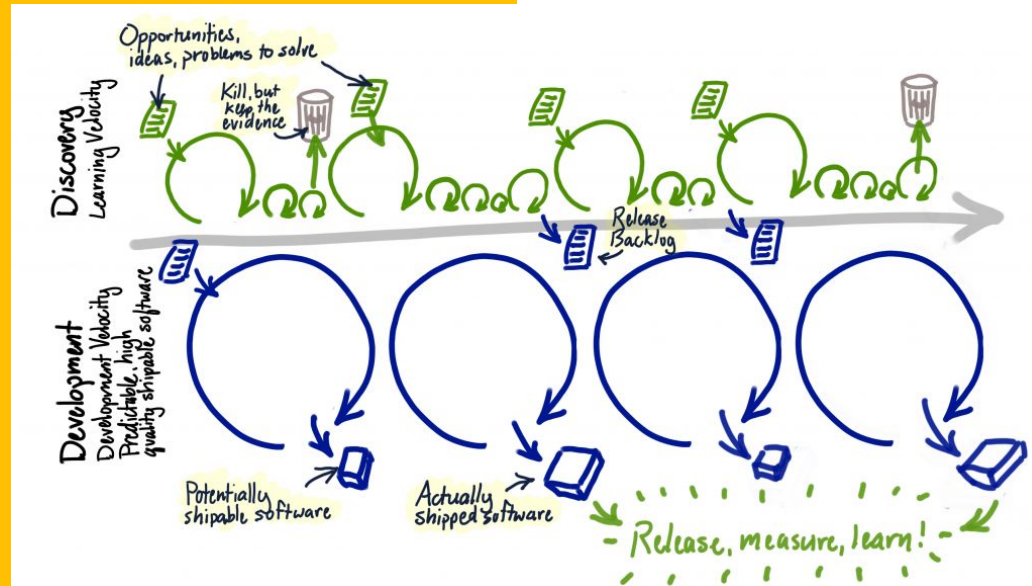


Dual hvad for noget?

Med mindre du er meget fremme i skoene, så er det sandsynligvis første gang, du hører om Dual Track Scrum. Men det er nu ikke noget, vi har fundet på - så vil du vide mere om ræsonnementet bag, så skynd dig at læs denne glimrende artikel fra Jeff Patton:

<https://www.jpattonassociates.com/dual-track-development/>

Det er ganske interessant læsning, og vi er kommet frem til mange af de samme konklusioner. Hvordan vi tænker og gør bliver forklaret på de næste sider.



Forskellige opgaver - ikke forskellige spor

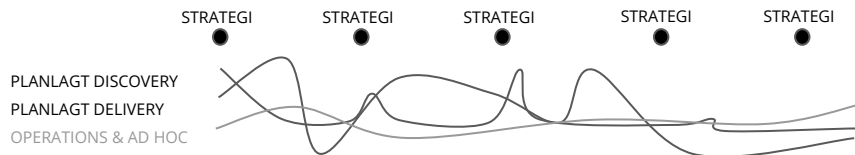
Vi har arbejdet med mange variationer af agile metoder - og eksperimenterer løbende - men for at gøre en lang snak kort, så fungerer det efter vores erfaring bedst, når de afklarende og implementerende aktiviteter *følges ad* og *informerer hinanden*.

Det nytter ikke så meget, at kun udviklingen kører agilt - hele tanken er jo, at vi skal opnå et kort feedback loop og lære af det - og så skal alle andre discipliner lige fra design, ux og strategi jo også være i spil, efterhånden som vi bliver klogere. Derfor forsøger vi at gøre det hele løbende og **deler det ikke op i to spor**, men derimod bare op i discovery og delivery opgaver.

Klassisk “water-scrum” 😞



Dual track scrum ala Reload

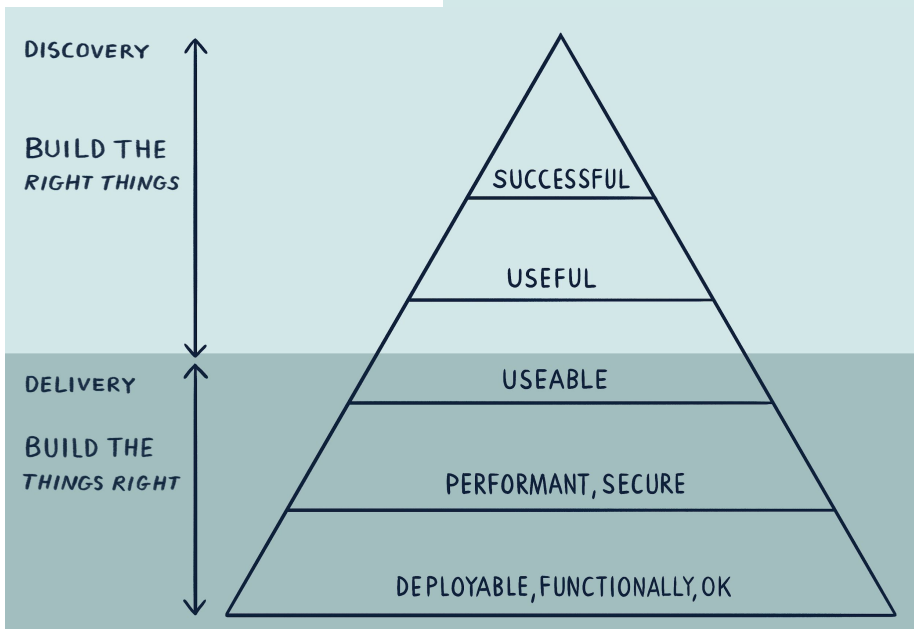




Alice fortæller om den digitale behovspyramide.

Byg de rigtige ting

Med discovery-opgaver fokuserer vi på, hvordan vi kan bygge "det rigtige". Det kan være en eksplorativ og kreativ proces, hvor vi prøver ideer af, interviewer folk i målgruppen eller kigger på eksisterende data. Vi skaber viden. Backloggen (opgavelisten) består af research spørgsmål, leverancen er *svær*, og det er oftest en iterativ proces, hvor vi løbende retter vores hypoteser til. Vi ender med nogle svar som i sidste ende sikrer, at vi fravælger de mindre gode idéer og sender de gode videre som velspecificerede user stories med validerede bagvedliggende antagelser.



Kilde: [Software hierarchy of needs, Gojko Adzic.](#)

Byg tingene rigtigt

Med delivery-opgaver fokuseres på kvalitet, performance, scalability og lign. User stories fungerer som udviklings-specifikationer, og leverancen er produktionsklare features klar til at blive deployet (udrullet) og taget i brug.

At der er to *typer af opgaver* er dog ikke at forveksle med to separate *teams*. Det er ét team, alle dem der arbejder sammen om jeres nye løsning, og de arbejder sammen i én proces, den har bare to typer af opgaver. Det er vigtigt, at alle så vidt muligt inddrages i, hvad der arbejdes på på tværs af opgaver. Det er vores erfaring, at det er en nøgelfaktor i forhold til at skabe den gensidige team-ansvarsfølelse der er et af elementerne, hvis man vil opnå et *high performance* team.



Hvad så med “design”?

Designarbejdet i traditionel forstand (som i “vi laver noget i form og farver”) har en vigtig rolle at spille i begge typer af opgaver.

I discovery-opgaver designer vi for at lære og for at skabe en fælles forståelse for løsningsrummet. Vi skaber muligheder, prøver ting af og finder ud af, hvad der virker.

I delivery-opgaver designer vi for kunne levere et færdigt produkt. Det er her designarbejdet former og indarbejdes i den tekniske løsning og der er fokus på systematik, konsistens og genbrugelighed af komponenter i brugergrænsefladen.

Sådan kan det se ud i praksis

The screenshot displays a Jira board with the following columns and tasks:

- SKAL GØRES**
 - Overvågning af lead submissions (Salgsvurdering etc.) - MindWorking integration - Priority: 5 - Assignee: LB-1086
 - I Back-Enden skal det fremgå HVEM der sidst har opdateret/gemt artiklen - Prioritet Q1: Butikker i admin - Priority: 3 - Assignee: LB-1223
 - Cache af HTTP kald til fra Node til Drupal. ~150ms til ~60ms. - Prioriterede Q3: Performance - Priority: 15 - Assignee: LB-1268
 - Projektliste i admin - vis antal sager - Prioritet Q3: Projekter - Priority: 5 - Assignee: LB-1255
 - Omdirigering til butiks subdomæner skal aktiveres - Discoverability (SEO) - Priority: 5 - Assignee: LB-1210
 - LB-1217 Producere ambassadør indhold - Teknisk galleri på "Du vil Sælge?" siden - Priority: 5 - Assignee: LB-1261
- SKAL AFKLARES**
 - CO2 Neutral website mærke - Discoverability (SEO) - Priority: 5 - Assignee: LB-1302
 - SEO - issues? - Discoverability (SEO) - Priority: - - Assignee: LB-204
- IGANG**
 - LB-1268 Cache af HTTP kald til fra Node til Drupal... - Spejling af Drupal cache tags i frontend - Priority: 0 - Assignee: LB-1372
 - Opvarm den interne Drupal cache på en bolig. Fra ~8 sekunder til ~150ms. - Prioriterede Q3: Performance - Priority: 8 - Assignee: LB-1267
 - LB-1302 CO2 Neutral website mærke - Bjarne giver et design skud der er mere spændende end statisk design - Priority: 0 - Assignee: LB-1311
 - Producere ambassadør indhold - Prioritet Q2: Indhold og CTA'er til - Priority: 13 - Assignee: LB-1217
- KLAR TIL REVIEW**
 - LB-1268 Cache af HTTP kald til fra Node til Drupal... - GraphQL PQ - Priority: 0 - Assignee: LB-1270
 - Sæt rigtige cache tags på Drupal GraphQL responses - Priority: 0 - Assignee: LB-1328
 - Hvor stor en del af brugerne rammer de ~8 sekunder? - Prioriterede Q3: Performance - Priority: 0 - Assignee: LB-1272
 - Test cache invalidering hos Fastly CDN - Prioriterede Q3: Performance - Priority: 3 - Assignee: LB-1234
- KLAR TIL RELEASE**
 - Igår, Overskriften, tusinde seperator - Boligvisning - Priority: 2 - Assignee: LB-1378
 - Eirrapporter importeres ikke til sager - Boligvisning - Priority: 0 - Assignee: LB-1375
 - WYSIWYG på alle artikelsider - Priority: 0 - Assignee: LB-1377
 - Scroll til toppen af mæglerens boliger ved næste klik - Priority: 0 - Assignee: LB-1345
 - Gør det attraktivt at tilmelde sig ved bestilling af materialer - Boligvisning - Priority: 3 - Assignee: LB-1310
 - Design og udformning - Priority: 0 - Assignee: LB-1367
 - IE-disclaimer er forsvundet igen - Priority: 0 - Assignee: LB-1342

Her er et [Jira](#) udklip fra et igangværende sprint. Her er både discovery- og delivery-opgaver repræsenteret.

Her er det et produkt, der er i produktion, og der er både nyudvikling, drift og optimering blandet sammen i én stor pærevælling.

Det repræsenterer dog også den virkelighed, de fleste oplever, og det kan sagtens lade sig gøre at styre.

Discovery

Opgaver der skaber svar



Discovery - her skal vi blive klogere

Inden et sprint starter, så har vi afholdt et grooming møde, hvor vi tager fat i de vigtigste behov og features, vi kan identificere i vores Impact Map. Vi vurderer her risikoen for, at vi mangler viden, og vores antagelser derfor ikke holder. Dvs. hvis vi er stensikre på vores business case og vores kendskab til målgruppens brugssituation, skal vi selvfølgelig ikke bruge mere krudt på at analysere og iterere over design eller koncept. Men er der områder, hvor vi er mindre sikre, giver det mening at grave lidt mere i det.

Det er selvfølgelig virkelig svært for os på dette tidspunkt at vurdere, hvor meget tid vi skal bruge på analyse og specificering af krav og ønsker for at kunne skære scopet rigtigt.

Det svære ved den slags afklarende arbejde er, at man aldrig kan blive færdig. Du kan altid sikre sig lidt bedre, undersøge lidt mere, prøve lidt flere visuelle muligheder osv. af inden du føler dig klar til udvikling. Så ligesom du skal være knivskarp på ikke at agere efter antagelser, der burde valideres bedre, skal du den anden vej rundt være lige så skarp på, at den tid du bruger her, går fra tid du kunne bruge til udvikling og læring deraf. Du skal altså kun undersøge forhold, I ikke ved nok om, og hvor det er tiden værd.

Udgangspunktet er igen, at I diskuterer behov og relevant viden ift. den givne feature, og her giver det typisk mening at trække på forskellige ressourcepersoner som ved noget om det område.

Det tværfaglige team finder sammen ud af: **Hvordan kan vi udvikle en løsning, der tester faktisk adfærd?**

- Hvordan kan en prototype, et minimumsprodukt eller anden intervention se ud, der gør det muligt at teste adfærd?
- Findes der eksisterende løsninger, data eller viden, vi kan lære af eller bygge ovenpå?
- Hvordan designer, udvikler og tester vi løsningen, så vi får hurtigst og billigst feedback om faktisk adfærd?

Lavere sygefravær

Alle medarbejdere skal drikke 2 liter vand om dagen

Personlig reminder + feedback, fx via mobilen

Det tværfaglige team undersøger løbende: **Hvordan performer løsningen ift. at understøtte den ønskede adfærd?**

- Hvor godt påvirker eller understøtter løsningen den ønskede adfærd?
- Hvad kan vi lære af kvantitative data - fx fra Google Analytics?
- Hvad kan mere kvalitative data lære os - fx optaget brugeradfærd?
- Hvor har vi evt. brug for mere dybdegående viden - fx gennem interviews eller observationer?

Vi skal stille spørgsmål

Ud fra en risikobetragtning beslutter vi, om eksisterende viden skal følges op af analyse (hvor stor risiko er der for, at vores viden er forkert eller forældet? Og hvor stort et problem har vi egentlig, hvis det viser sig at være sagen?).

Selve analysen i sprintet kan altså bestå af meget forskellige aktiviteter. Målet er at besvare de spørgsmål, vi kan have ift. om vores idé og forventet cost/benefit holder. Typisk er det en balancegang mellem analyse processer og kreative processer at finde frem til den rigtige løsning.

Typiske aktiviteter (og altså måder at besvare spørgsmål på) kunne eksempelvis være:

- *“Hvad er brugernes behov?”*
 - Eksplorativ analyse via fokusgruppe interviews og feltundersøgelser, der kan afdække behov, vi selv og vores brugere ikke ved, vi har.
- *“Kan det teknisk lade sig gøre?”*
 - Teknisk prototyping der kan verificere, om formodede teknologier og 3. parts moduler kan bruges i et specifikt scenarie.
- *“Kan vi skabe den ønskede adfærd hos brugeren?”*
 - Iterative designprocesser, baseret på visuelle prototyper, hurtige brugertests og feedback.

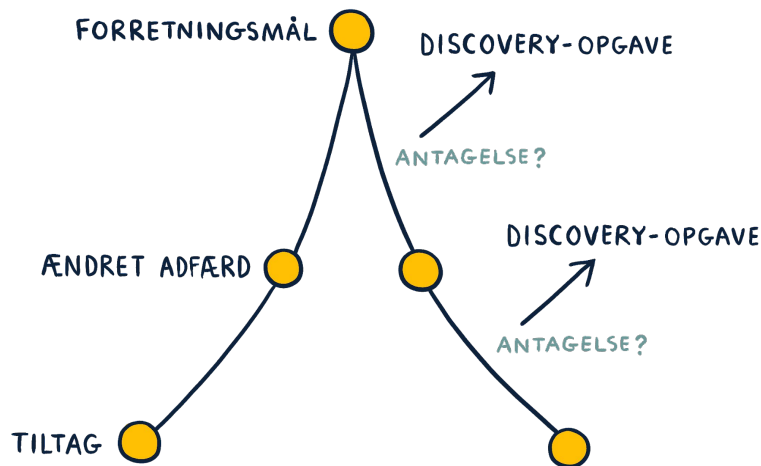
Vores antagelser giver anledning til spørgsmål

Disse spørgsmål vi skal stille udspringer oftest fra Impact Mappet. Vi *antager*, at en given funktionalitet ændrer en adfærd, og vi *antager*, at denne ændrede adfærd påvirker vores forretningsmål positivt.

Hvis dette reelt er antagelser og ikke faktisk viden, så skal vi muligvis sørge for at blive klogere her.

Og dette er særligt tilfældet, hvis prisen for at udvikle et tiltag er høj, eller risikoen for projektet er høj, hvis antagelsen er forkert (det kan fx være nogle tekniske muligheder/begrænsninger eller fx en stor økonomisk omkostning).

Det er ikke alting, vi kan undersøge og måle på forhånd, men det nemmeste er oftest at tage udgangspunkt i folks adfærd.



Discovery-opgavernes produkt er svar

Når du planlægger sprintet, så skal du estimere (eller ofte timeboxe) "hvor lang tid tager det at besvare et givent spørgsmål" og herefter udvælge de mest relevante opgaver at arbejde på. Resultatet af det arbejde du laver her er *svær*. Svær som ender med at skabe viden og ultimativt bliver til en backlog af udviklingsopgaver, som med meget høj sandsynlighed vil blive brugt og skabe faktisk værdi. I denne optik måles der altså på fremdrift, når vi bliver kloge, hvilket betyder, at det også er fremdrift, når vi bliver så kloge, at vi tør fravælge en feature, der umiddelbart virkede som en god idé.

“The most expensive way to test your idea is to build production quality software”.

- Jeff Patton

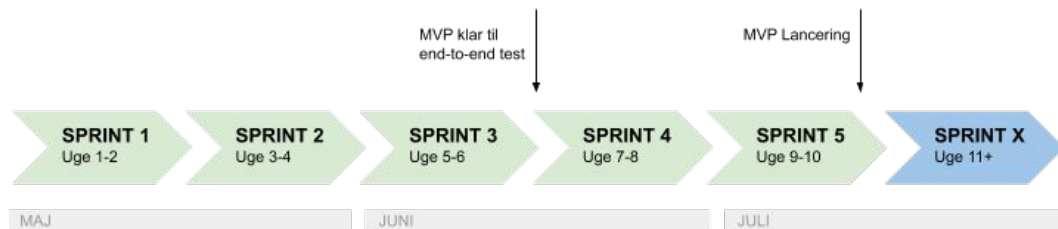
Delivery

Opgaver der bliver til fungerende løsninger





**HELLERE LILLE OG
LÆKKER END STOR
OG USPISELIG**



Møde: Sprintplanning (hver sprint)

Vi gennemgår de estimerede udviklingsopgaver og prioriterer, hvilke opgaver der har højest værdi/prioritet og dermed skal i scopet for udviklingsprintet.

Forløb: Udvikling og test (2 uger)

Vi arbejder på én opgave ad gangen per udvikler og færdiggør en ny version af løsningen, der kan afprøves ved sprintets afslutning.

Møde: Demo, retrospektiv og rapportering (hver sprint)

Vi demonstrerer sprintleverancen og vender processen og evt. forbedringer hertil. Vi leverer også en sprintrapport, der opsummerer status på fremdrift, økonomi, scope, risici og vigtige beslutninger.

Møde: Daily scrum / Standup (hver dag i sprint)

En kort, stående gennemgang af dagens opgaver og status på hvad der blev lavet siden i går morges. Bruges til at vidensdele og identificere roadblocks. Alle er velkomne på mødet.

Møde: Backlog grooming (hver sprint)

Her forbereder vi opgaver til næste sprint. Dvs. vi taler dem igennem, finder ud af hvilke opgaver der kræver yderligere afklaring og beskriver de opgaver, vi forventer, skal med i næste sprint. I veletablerede teams kan dette også ske løbende.

Forløb: Afprøvning og feedback

Vi forventer, at interessenter umiddelbart efter et sprint afprøver leverancen og giver feedback, vi kan tænke ind i de kommende leverancer.

Så skal der bygges fungerende leverancer

Delivery-opgaver lever typisk som user stories i en traditionel Scrum proces. Det betyder, at vi har en backlog (opgavelisten) som er en løst struktureret opgaveliste af "user stories" med de specifikationer, vi former og validerer med discovery-opgaver. Det er altså at betragte som en dynamisk kravspecifikation, selvom det er også vigtigt at forstå, at det ikke er meningen (eller opnåeligt), at alt i backloggen skal laves.

Derimod skal den løbende prioriteres, forfines og justeres efterhånden som tiden skrider frem. Vi vælger løbende en håndfuld opgaver og implementerer dem i et *sprint*, hvorefter de færdige funktionaliteter kan demonstreres (og per definition er produktionsklare).

Scrum processen kort forklaret

Selve Scrum-processen kan du finde masser af bøger og materiale om. Men jeg vil lige forklare helt kort, hvordan vi snitter den på en måde der erfaringsmæssigt fungerer godt for os. Dette er vores udgangspunkt, når vi starter nye projekter op.

Et udviklings-sprint varer typisk 2 uger og inkluderer forberedelse til det kommende sprint samt til leveranceafprøvning. I Reload tror vi fast på, at det er essentielt, at vi bruger tid sammen som samlet team og har mulighed for at træffe hurtige beslutninger sammen. Vi stiller for eksempel lokaler til rådighed, så vi og vores kunder kan sidde fysisk sammen minimum én dag om ugen udover de faste møder.

Jo mere teamet er fysisk sammen, desto bedre.

Vi skaber løbende læring gennem korte feedback loops

Inkrementel udvikling er trinvis udvikling. Et trin ad gangen - en feature ad gangen. Så frem for at nedbryde og udvikle scope ud fra de tekniske lag (først datagrundlag / funktionalitetslag / præsentationslag), så skal du nedbryde i fulde funktioner, små prioritérbare bidder, der giver værdi i sig selv. Det, du tager ind i sprintet bør altså være så afgrænsede features, at de kan færdiggøres og være klar til produktion ved sprintets ende. På den måde kan de komme ud og generere faktisk værdi, mens I udvikler på de næste.

Værdi er ikke nødvendigvis produktion. Oftest skal der en del sprints til, før man er nået til et produkt der kan publiceres. Men det skal ud og prøves af, om ikke andet hos interessenter i organisationen. Således skaber du feedback og dermed viden. Og det er værdifuldt!



Færdiggørelse er jo så et elastisk begreb i denne sammenhæng, og det er her, at det iterative og udnyttelse af feedback loops kommer ind i billedet. Vi anbefaler altid, at første version af en ny feature er så skrabet som overhovedet muligt, men aldrig mere skrabet, end at du (hvis den er færdig) vil kunne lukke den ud i produktion, som den er.

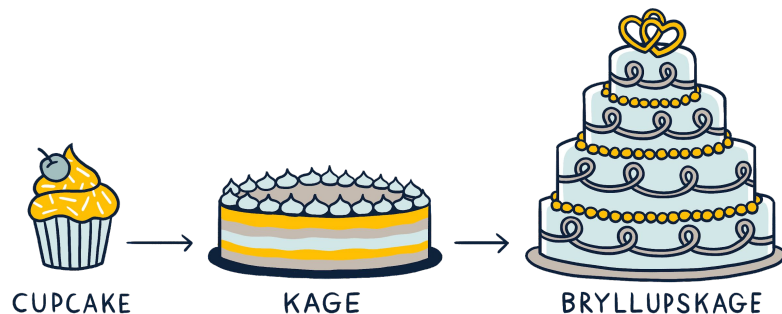
Du skal altså *bygge kagen vertikalt* - i bidder. Eller også skal du starte med at bygge en *lille* kage.

Jeg elsker kage-analogier

Typisk er det de færreste behov og features, vi helt kan undvære at komme omkring, men ofte kan vi vælge at gøre mere eller mindre ud af en given feature. Både fordi vi gerne vil have feedback og lære, før vi udvikler den dyre løsning, og fordi vi vil være sikre på, at vi kommer hele vejen rundt, er fokus i udviklingen af en ny løsning altså altid på, hvor simpelt vi kan forme og skære den. Når vi så først er hele vejen rundt, kan vi arbejde mere iterativt og backloggen kommer i højere grad til at bestå af forbedringer til eksisterende features frem for nye features.

Lille og lækker

Hellere bygge en funktionelt lille, men virkelig velfungerende løsning og siden bygge på derfra. Så får brugerne en god oplevelse, og du får gang i et rigtigt sundt feedback-loop, som kan hjælpe dig med at prioritere det videre arbejde.



Grundtanken er, at vi undgår at bygge én stor lagdelt kage, der først er spændende, når den står færdig med glasur og det hele. I stedet bygger vi en lille, men perfekt lækker kage. En fuldendt cupcake. Er dine brugere begejstrede for den, jamen så kan du bygge den videre til en stor kage. Måske kan de ende med at få en bryllupskage. Men går du efter bryllupskagen fra starten af, kan det være, at den er så dyr eller kompleks, at du aldrig bliver færdig med den. Eller også ser den flot ud, men falder ikke i nogens smag.

Vi kalder det for “cupcake modellen”.

Bliv ikke en metode-zombie

Når vi snakker agile udviklingsprocesser, så går der hurtigt rigtig meget metode snak i den. Metoder er gode, og de kan understøtte, at man gør tingene rigtigt. Men metode gør jer ikke i sig selv agile. Du kan sagtens arbejde ganske traditionelt, selvom du har både user stories og daily standups. Så pas på med blindt at følge metoder, hvis det ikke giver værdi. Det er bare svært at vide på forhånd hvad man kan vælge til og fra - det kræver erfaring. Sørg for at søge hjælp hvis I er nye i de agile metoder.

Det var introduktionen til vores metode. Vi forklarer glædeligt om vores erfaringer i dybden, så tag bare fat i mig hvis du vil høre mere (se sidste side for kontaktinformationer).



Agile is simple, but it's not easy

– T-shirt tekst på en konference

Dynamiske budgetter

Hvordan du kan bruge budgettering som aktivt
prioriteringsværktøj og
forventningsstyringsmekanisme

Hvad er dynamisk budgettering?

Hvordan bruger du bedst det budget, du har for at løse en givent forretningsmål? Og hvordan skal du balancere discovery/delivery arbejdet? Er det billigere at prøve at bygge noget, evt. noget der håndholdes manuelt, og se hvad der sker, frem for at forsøge at analysere sig frem til et svar?

Eller skal du trykprøve dine antagelser lidt mere først, måske med en prototype? Måske ved at blive klogere på det reelle indtjeningspotentiale eller ved at blive klogere på brugernes oplevelse? Det giver mening, hvis det er en betydelig investering at bygge - eller måske fordi der er risiko for at ødelægge en allerede god oplevelse.

Det er ofte svært at lave korrekte hårde regnestykker, men du kan sikre dig, at du aktivt tager stilling til de input, du har at arbejde med.

I den agile tilgang advokerer vi ofte for at prøve os frem, ud fra en anerkendelse af, at vi sjældent kan forudsige ret meget alligevel. Til gengæld er det absolut afgørende, at du gemmer den energi (budget), som du ellers ville have brugt på at validere antagelser upfront, til at opsamle og udnytte feedback og til at tilrette de initiale versioner af nye features/tiltag, så de performer i virkeligheden!

Det klassiske budget

De fleste lever i en verden, hvor man enten er tvunget til at forsøge at gætte, hvad en digital løsning koster at lave - eller hvor der kommer et budget ovenfra, som så skal bruges til at køre et projekt.

Det involverer ofte en fast budgetramme, nogle milesten og deadlines, og hvis det er rigtig grelt, så ønsker ledelsen også at se et Gantt diagram.

Selvfølgelig skal man tale økonomi, men det er grundlæggende svært at fremkomme med det rigtige økonomiske tal.

Men hvad med at vende det på hovedet? Hvad med at tage udgangspunkt i, hvad et digitalt projekt er værd for jer?

I den bedste verden vil du løbende kunne søge og få midler til at drive dine indsatsområder i et digitalt projekt. Men står du, som vi også ofte gør, i en situation, hvor der skal etableres en økonomisk ramme for at lave et "MVP" (minimal viable product), så giver det god mening at bruge det til at styre forventning og prioritering.

Hvis løsningen er meget udefineret, så kan vi bruge budgettet til at beslutte, hvor meget de forskellige områder i løsningen er værd. Det kan gøres i procent, som vi så kan fordele budgettet ud på bagefter. Tanken er, at vi løbende skal holde øje med, om budgettet holder - og rette ind, når det ikke gør.

Det kræver vist lidt ekstra forklaring...

Dynamiske målbaserede budgetter

I stedet for at samle en række mere eller mindre urelaterede features i et mere eller mindre tilfældigt "projektbudget", kan vi i stedet begynde at arbejde mere intelligent med "målbaserede budgetter".

Den værdi som du antageligt kan indfri ved at opnå et givet mål bør sætte budgetrammen for, hvor meget du vil arbejde for det mål.

Det gælder alt det arbejde som skal gøres for at nå i mål. Dvs. både udforskende arbejde (spørgsmål) og implementering, som relaterer sig til et givet mål, bør derfor estimeres og holdes op imod den værdi, som forretningsmålet, de relaterer til repræsenterer, så cost / benefit betragtningen hele tiden er i fokus.



Festbudget eksemplet

Når vi skal forklare, hvordan vi arbejder med budgetter, plejer vi at forklare det således:

Du skal planlægge en stor fest. Du har måske 50.000 kr, og det skal være suppe, steg og is - og så skal der selvfølgelig være godt til ganen, og det kunne også være fedt med en DJ til at sætte gang i festen.

Du tager turen til supermarkedet, men her begynder virkeligheden at ramme. Der er simpelthen ingen jordskokker - men i stedet opdager du at der er nogle utroligt lækre jomfruhummere på tilbud. Og det skal jo være en god fest, så *ja sgu*, det gør du. Det skal jo være lidt lækkert, ikk?



Festen udvikler sig. Hvad med budgettet?

Men hvad nu? Festen er ikke engang i gang, og du er allerede ved at sprænge budgettet, fordi du vil give den gas til forretten.

Nu står du så i den situation, at du skal til at prioritere: **Skal du droppe bøffen til hovedretten** og i stedet satse på løgsuppe? Eller skal vi **droppe 1. Cru Chablisen? Er Spotify i virkeligheden nok?**

Noget skal i hvert fald prioriteres (*konsulent-speak for "skæres væk"*), hvis budgettet skal holdes. Eller også skal du beslutte at tilføje flere midler til fest-budgettet, hvis du ikke vil gå på kompromis. Det er i hvert fald vigtigt at have en ide om: *Hvad gør en god fest?*

Det er egentligt ret åbenlyse valg.

Kunsten her er jo, at det skal være informerede valg - og ikke mindst, at det er prioriteringer, vi gerne vil **gøre på forkant** - og ikke når pengene "pludselig" er brugt.

Så ved på forhånd at have budgetteret forskellige områder (vi kalder det *epics*), så kan du - inden der er betaling ved kasse 1 - proaktivt beslutte, hvordan du vil håndtere en ændret virkelighed.

Al udvikling af digitale løsninger er én lang prioriteringsøvelse - der er altid enten mangel på tid eller penge, og det bør der også være. Kunsten her er at prioritere at udvikle de mest værdifulde ting - og løbende sørge for, at der er budget til dem.

Derfor skal der budgetteres løbende.

Dynamisk budgettering - eksempel fra virkeligheden

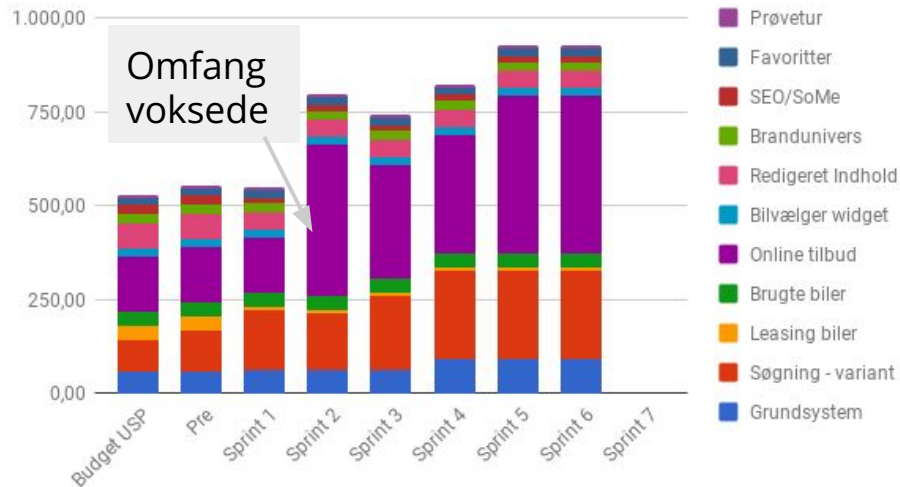
Epicoverblik

Epic	Budget USP	Budget kr.
Grundsystem	58,00	134.593,88
Søgning - variant	85,00	197.249,65
Leasing biler	38,00	88.182,20
Brugte biler	38,00	88.182,20
Online tilbud	146,00	338.805,29
Bilvælger widget	20,00	46.411,68
Redigeret Indhold	67,00	155.479,14
Brandunivers	24,00	55.694,02
SEO/SoMe	25,00	58.014,60
Favoritter	20,00	46.411,68
Prøvetur	8,00	18.564,67
	529,00	1.227.589,03

Her er et budget fra et projekt i bilbranchen. Vi har delt løsningen ind i en række epics, som vi i dette tilfælde har forsøgt at estimere i point. Det oversættes så til et budget i økonomi, som summerer op til et samlet budget. Andre gange gør vi det omvendt - vi allokerer budgettet og regner om til point.

Efterhånden som vi nedbryder opgaver og laver reelle estimater, så kan vi tracke, om budgetposten skal forøges eller bliver billigere. Dette kan vi forholde os til, inden vi starter på udviklingen, men det skal også ske løbende.

Epicoverblik



Dynamisk budgettering - eksempel fra virkeligheden

I dette eksempel kan du se at scope udvikle sig ret kraftigt. Dvs. der kom simpelthen langt flere opgaver til i løbet af projektet.

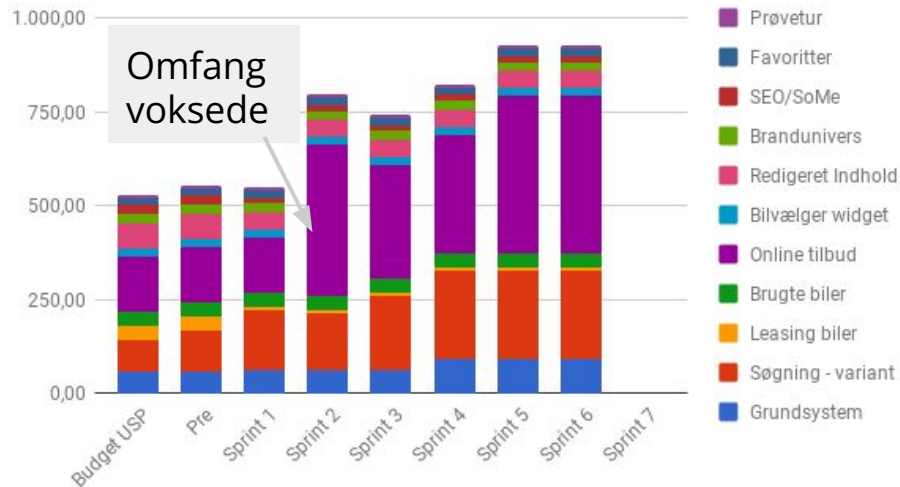
Der er mange måder vi kan håndtere det. Er det hele virkelig nødvendigt at tage med? Hvad skal være en del af MVP?

Generelt, så advokerer vi for at budgettet skal være fast, og vi i stedet prioriterer hårdt i scope - altså en hård udvælgelse af hvad vi reelt implementerer.

Det skyldes, at vi gerne vil i luften hurtigst muligt, og på den måde få reel feedback. For så ændrer prioriteterne sig alligevel.

Kan der ikke findes billigere veje eller skæres i projektets omfang, så er der reelt ikke andre muligheder end at øge budgettet - eller stoppe det helt. Men hvis du på nogen måde kan, så skær altid til i scope!

Epicoverblik



Kan man overhovedet budgettere?

Budgettering er bare svært. Det er jo mere eller mindre kvalificeret gætværk, og man må nok bare indse, at det sjældent holder. Så man slipper bedst fra det, hvis man italesætter den usikkerhed, og løbende styrer efter det. Men kan man fortælle hvilke antagelser der viste sig ikke at holde, så står man et langt bedre sted, når man enten skal omprioritere eller bede om flere midler/tid.

Når vi skal budgettere, så sker det ofte ud fra en blanding af erfaring, bottom up estimering og top-down estimering og så betragtninger omkring værdi og usikkerhed.

Ud fra det beskriver vi præmisser og antagelser, og så laver vi et budget.

Nogen gange er det baseret på, hvor lang tid vi tror noget tager at lave - og andre gange er der så mange ubekendte, at vi i stedet må afsætte et budget ud fra, hvad vi tror, det er værd at få løst et givent problem.

Er der meget store usikkerheder, så giver det mening at grave i det hurtigst muligt for at få afklaret, om budget og præmisser holder.

Nogen gange sker det, inden vi lander et første budget - andre gange er det bare en del af at udvikle løsningen.

Lige meget hvad så kræver det løbende opfølgning. Budgettet er levende og skal løbende styres og formidles.

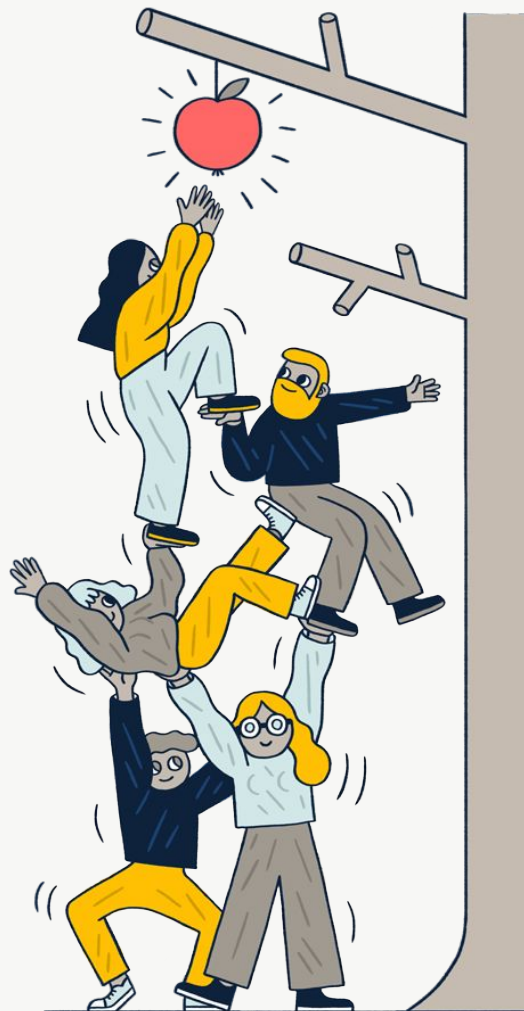
Teams der kan
og vil selv



Dyrk det gode samarbejde

Vi tror på, at digitale løsninger skal udvikle sig kontinuerligt og agilt og sammen med de mennesker, der bruger dem og har gavn af dem, for at holde trit med virkeligheden. Måden du kan opnå det på, er ved at agere som ét samlet team på tværs af afdelinger og leverandører og ved at have en holistisk tilgang. Løsningerne er levende: Nogle gange kræver de udvikling i hurtigt tempo og i en komprimeret tidsperiode (mange timer på kort tid), andre gange skal de formes over tid. Nogle gange med en åben og nysgerrig tilgang, andre gange med en effektiv og eksekverende tilgang.

Men de gode løsninger skabes af velfungerende teams, som har de rette kompetencer, indsigter og værktøjer. Og ikke mindst evnen til at spille hinanden stærke og bruge hinandens styrker.



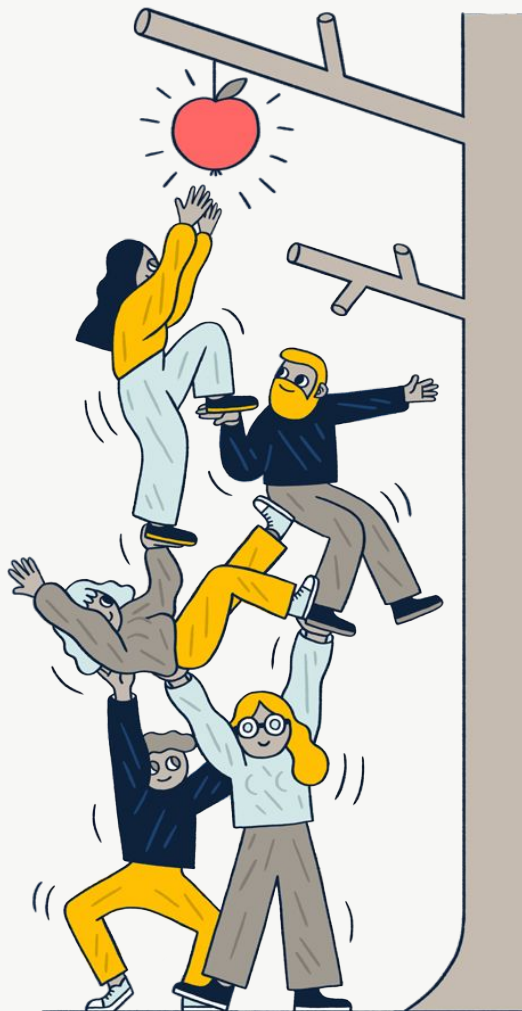
Tillid er nøglen

Det betyder, at du skal etablere et team, hvor den *psykologiske sikkerhed* er høj. Dvs. der er tillid i teamet - og tillid til at kunne fejle, erkende det og lære af det, og så komme videre.

Det er et rum, der skal skabes i teamet og er helt essentielt for, at et agilt forløb skal lykkes.

Det kræver åbenhed og ærlighed, høj kommunikation og et team der løber efter fælles mål - og har en holistisk forståelse af, hvad de skal opnå.

Der er der også mange andre der har skrevet gode artikler om, men jeg vil på de næste sider dele nogle af de vigtigste erfaringer, vi har gjort os over de sidste 10 år.





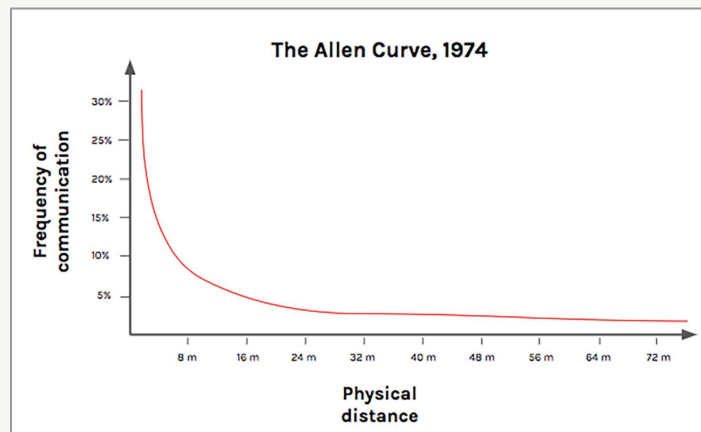
SID FYSISK SAMMEN

Det skaber tillid og høj
båndbredde i
kommunikationen

At sidde sammen er HELT VILDT effektivt

Det er faktisk næsten det bedste tip, vi kan give: *Sid sammen*. Der opstår noget særligt, når man arbejder tæt rent fysisk. I Reload har vi organiseret os efter det - vi rykker rundt og sætter os sammen alt efter, hvilken løsning vi arbejder på. Og vi beder vores kunder sætte sig sammen med teamet - gerne et par dage om ugen, når vi kører på fuld blus. Det oplever vi og vores kunder er guld værd. Det skaber nemlig stor båndbredde i kommunikationen og derved større fremdrift - er man i samme rum og får afklaret tingene undervejs, så bliver vi bare mere effektive.

Anderledes ser det ud, hvis man sidder for længe med tingene alene og begynder at arbejde ud af en (uhensigtsmæssig) tangent. Der sker også noget på et menneskeligt plan - man lærer hinanden bedre at kende, og tilliden vokser. Det samme gør forståelsen for hinandens fagligheder og vidensområder. Det gør arbejdet sjovere og mere inspirerende.



Det er ikke lang tid siden, at vi fandt ud af, at der er videnskab der underbygger vores erfaringer. Fænomenet er kendt som The Allen Curve, og det holder stadig vand på trods af vores mange digitale kommunikationsredskaber. Det er rasende interessant - hav det for øje, når du planlægger, hvordan I skal sidde og arbejde sammen.

Og det næstbedste er en velfungerende chat

Når vi så ikke sidder sammen, så bruger vi selvfølgelig en række kommunikationsværktøjer.

Når man kender hinanden godt, så er chat faktisk et virkelig godt værktøj til at holde hinanden i loopet og få hurtig feedback på ting.

Det giver mulighed for en ret fri snak, ofte på alle mulige tidspunkter af døgnet. Det er selvfølgelig et tveægget sværd og skal bruges ansvarligt, men det giver en ret klar team-følelse og responsiv dialog.

The image shows two side-by-side screenshots from a digital workspace. The left screenshot is a Slack chat conversation between three people: Arne, Mie, and Arne. The messages are as follows:

- Arne:** OK, jeg tror jeg ved hvad det skyldes. Og jeg tror det er fordi jeg var inde at rette på dem. Mistanken er det kode der knytter webforms op på grupper — og fordi jeg rettede med system-brugeren er den reference nok ikke længere validid. (det er også et ret broken koncept der er fundet på i tidernes morgen... men mon ikke de havde deres grunde)
- Arne:** Jeg leder lige efter et fix og om det kan gøres mindre sårbart fremadrettet.
- Mie:** takt! :D Og dejligt, du fandt problemet - Edited Oct 7, 2019 12:42
- Arne:** Ok, det rigtige problem er fundet... Og det var en tilfældighed!
- Arne:** Et fix er på vej.
- Mie:** hah, super godt 🍷
- Arne:** Faktisk er det en af dem hvor jeg tænker "men hvordan har det nogensinde virket????"
- Mie:** armen super 🍷🍷
- Arne:** Ser det rigtigt ud nu, @mie?
- Mie:** yes 🍷

The right screenshot is a JIRA activity feed for the 'DANJOUR-994' issue. It shows a series of updates:

- 15:25 • JIRA**
- OPEN #1037: Add "tal til øjet" group**
 - tripox cd373b7 Add group menu for tal til øjet: DANJOUR-994 15:22 • GitHub
- MERGED #1036: Show text in user menu and fix tabs ...**
 - tripox merged at 20fa3e9 and closed
 - tripox deleted branch 15:09 • GitHub
- master at reload/dj updated**
 - tripox 20fa3e9 Merge pull request #1036 from re...
 - tripox 66bdb2e Improve tabs on profile on mobil... 15:09 • 1 more message • GitHub
- OPEN #1037: Add "tal til øjet" group**
 - tripox DANJOUR-994 DID YOU REMEMBER TO EXPORT BOTH DJ AND AJKS DRUPAL SETTINGS? 14:50 • GitHub
- NEW DANJOUR-756: Find freelancer filtrering**
 - Mathias Larsen changed assignee to Mathias Larsen at 13:51 • JIRA
- NEW DANJOUR-998: Fjern SoMe-delning ikoner på b...**
 - Mathias Larsen changed assignee to Mathias Larsen at 13:51 • JIRA
- IN PROGRESS DANJOUR-1001: Synlighed på indholdsl...**
 - Mathias Larsen changed assignee to Mathias Lar...
 - Mathias Larsen changed status to In progress. at 13:51 • JIRA
- IN PROGRESS DANJOUR-994: Ny gruppe-side til "Tal t...**
 - Mathias Larsen changed status to Reviewable. at 13:50 • JIRA
- Created branch feature/DANJOUR-994 at reload/dj**
 - tripox 8de9b41 Add "tal til øjet" group: DANJOUR-994



HAV ET REELT MANDAT

Få en sponsor. Skab et
rum hvor I kan agere.

I skal have autonomi og mandat til at beslutte det meste

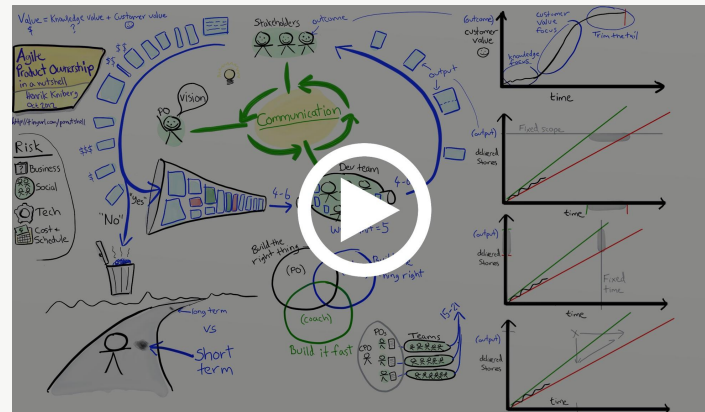
Man er ikke et velfungerende agilt team, hvis man ikke har et stort frirum til at agere. Det er lige fra metoder, værktøjer og hvad der reelt skal laves for at opnå de mål der jageses.

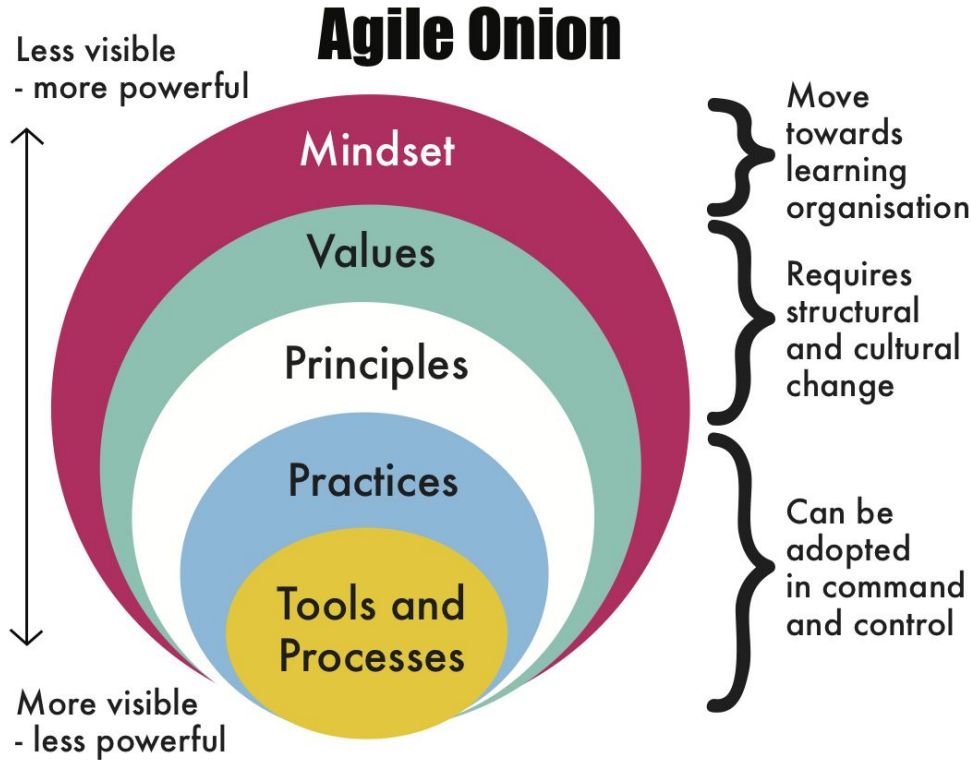
Der vil naturligt være mange, der har indflydelse på disse ting og i særlig grad de overordnede mål.

Men I skal have friheden til at jage målene, som I mener det gøres bedst - og rette ind efterhånden som I bliver klogere.

Sidder du, som de fleste, i en traditionel organisation, så er der et vist hierarki, som I skal passe ind i. I så fald bliver du nødt til at finde en "guardian angel" højere oppe i systemet - altså en chef-type som kan skabe et frirum for jer, hvor I kan få lov til at gøre tingene på andre måder end normalt.

Det skal selvfølgelig være en person, som fuldt ud tror på værdien i at gøre det anderledes og kan forsvare det overfor resten af organisationen eller en ledergruppe, som har svært ved at forstå, hvorfor al funktionalitet, økonomi og tidsplan ikke bare kan ligge fast på forhånd. Står du i den situation, så prøv at få dem til at se [denne fantastiske 15 minutters video](#), som jeg altid viser nye kunder / produkt ejere i Reload:





Når vi arbejder på en ny måde, så ændrer det os

Nye måder at arbejde på giver nye måder at organisere os på - og i sidste ende fordrer det et nyt mindset.

Og det skal der til, hvis du for alvor skal have effekt af den agile tilgang.

Fremtidens organisation er en lærende organisation.

Og det er sådan set essensen i at være agil: *Skab løbende læring og tag handling herpå.*

Det er et stort emne, men jeg kan godt lide denne artikel, hvis du er sulten på mere: [What is the Agile Mindset?](#)

Aktiv interessentstyring

... eller *stakeholder management* på nudansk 🙄

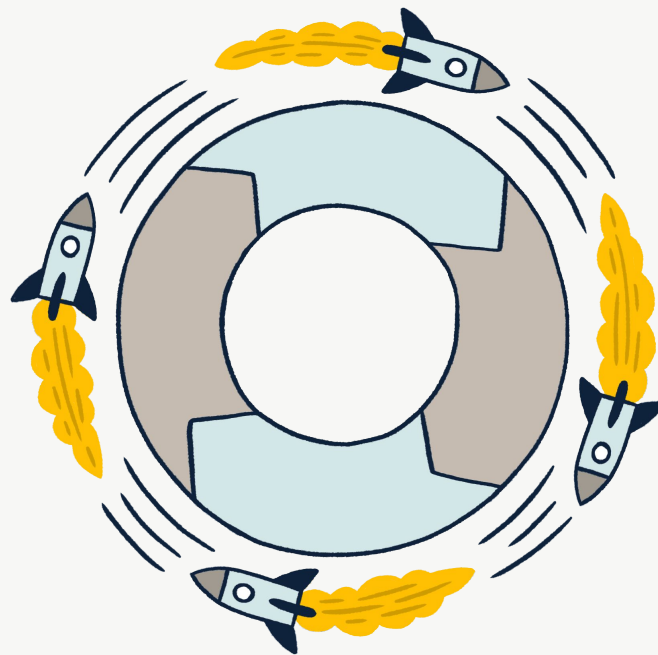
Lad os styre forventningerne

Det er jo meget fint at få etableret et agilt team, der har mod, muskler og mandat til at arbejde ud fra en lærende tilgang. Fedt! Men du kommer jo næppe til at arbejde i et vakuum - der vil altid være interesser omkring et nyt digitalt produkt.

I en større organisation vil der typisk være en chef- eller styregruppe som har stærke interesser og valide holdninger til mål og succeskriterier.

De er en stor ressource, men det er også en kunst at styre forventninger overfor dem.

Grundlæggende skal de jo have forståelse for, at vi her ikke arbejder helt traditionelt med en planstyret tilgang. Derimod skal vi have dem med til at definere og støtte op omkring målene - og så skal vi have dem med i loopet på, hvordan det går.



Forventningsafstemning er nøglen til succes. Husk på, at ligegyldigt hvor fantastisk et resultat, du mener at have skabt, vil det være en fiasko i dine interessenters øjne, hvis det ikke møder deres forventninger. Så de interne forventninger skal nøje styres, efterhånden som virkeligheden ændrer sig. Og det gør den altid.

– Dyrt betalt erfaring fra forfatteren...



VIS AT DET VIRKER

Giv hurtigt folk noget i hænderne - det trumfer mange indsigelser.

Glem ikke, at et digitalt projekt ofte er et organisatorisk forandringsprojekt

Der er også andre stakeholders - lige fra forskellige brugere i din egen organisation, samarbejdspartnere og det/de udviklingsteams der er i gang.

For det første oplever vi ofte, at vi skal bevise at processen virker. Dvs. at vi kan visе hurtige resultater, vise at viden vokser, og at tingene ændrer sig, samt at vi kan håndtere det.

Det er nemt at se for de folk, der er meget tæt på projektet - men alle de interessenter, som ikke lige er med til daglig, har brug af blive løbende inddraget.

Et demo-møde efter hvert sprint fungerer ret godt til at vise, hvad der sker. Men der er også en virkelighed der handler om risiko, økonomi og fremdrift, og den redegør vi for i noget, vi kalder en *sprintrapport*.

RELØAD
Bilag/Udlever.dk - Sprint 2 sprintrapport - 12. marts 2019

Sprint 2

"Grundlæggende opsætning og boligfagning"
Sprintperiode: 25. februar - 10. marts

Projektstatus

	Kommentar	Status
Generelt	Meget generelt går projektet godt. De udfordringer vi støder på er ikke dramatiske og er faktisk forventede i en proces som denne. Hele holdet arbejder godt sammen og der er god forståelse for arbejdet og den nødvendige pragmatisme.	Grøn
Økonomi	Økonomien i sig selv er stadig fin. Vi har brugt ca. 7% mindre end forventet og samtidig lukket flere point end forventet. Status er dog markeret som gul da ændringer i scope (se nedenfor) kan ende med at have konsekvens for det nuværende budget.	Gul
Tid	Tidsplanen lader ikke til at være et problem. Der kan potentielt ende med at mangle tid i slutforløbet, men det viser ikke særligt mange problemer.	Grøn
Ressourcer	Der er stadig allokation nok ressourcer og vi har mulighed for at booke mere. Da scope er udvidet (se nedenfor) kan det blive nødvendigt at udvide tidsplanen, men det virker ikke aktuelt med nuværende budget.	Grøn
Scope	Det samlede scope er efter grooming af boligplanen ca. 15% større end hvad det nuværende budget kan summe. Dette giver anledning til opmærksomhed og potentielt pragmatisme f.eks. de resterende opgaver/budget. Der er dog en rimelig forventning om, at vi potentielt kan simplificere andre opgaver og dermed hente point. Status er dog markeret med rød da der er en reel ændring i omfang, hvis vi fastholder alle resterende opgaver og ikke vinder fremgang på andre områder som forventet.	Rød
Risiko	De fleste identificerede risici er ikke indtruffet - dog har en større risiko (f.eks. scope) vist sig at være indtruffet. Fremadrettet er der ikke identificeret yderligere risici - der er dog markeret med gul da flere opgaver kan vise sig at være større end tidligere antaget.	Gul
Følg og Teknisk gæld	Ingen teknisk gæld vi er bekendt med.	Grøn

Bilag/Udlever.dk - Sprint 2 sprintrapport - Side 1/6

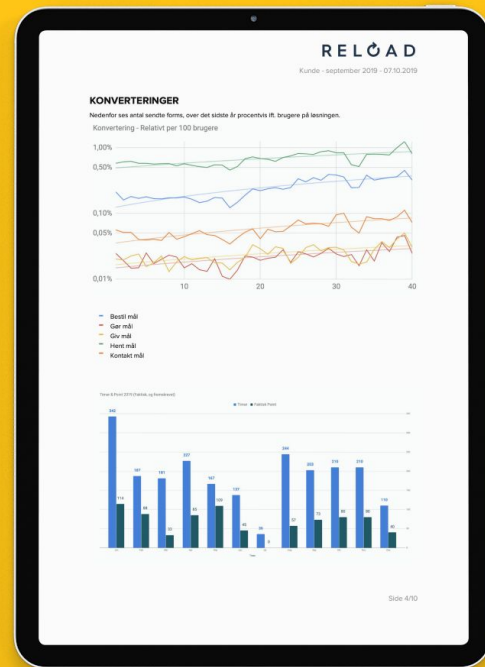
Mål på resultaterne

Det er selvfølgelig ikke nok at styre forventninger på de indre linier. Det du laver, skal selvfølgelig rent faktisk også skabe den effekt, som du sætter dig for.

Derfor er det selvfølgelig også vigtigt at måle og følge op på, om vi rent faktisk flytter noget på de ønskede mål. Og er selve målene svære at gøre målbare, så kan du i stedet måle på, at du ændrer adfærd.

Men det kræver, at du har en løsning som er i produktion, og det er sjældent her, at nye produkter starter. Men reel data er guld værd - og så er det utroligt svært at argumentere imod, så det er en stor motivation for at komme ud og virke hurtigst muligt.

Vi rapporterer også på vores mål i vores sprintrapporter. [Se eksempler her.](#)



Hvad nu?

Nu er det om at komme i gang. Jeg har her givet dig nogle af de mange læringer, vi har fået gennem årene. Men én ting er at læse sig til ting - en anden ting er at gøre det. Jeg håber, at den har givet dig mod til at prøve at arbejde på en anden måde.

Når det virker, så fungerer det exceptionelt godt

Vores oplevelse er nemlig, at det er virkelig givende, når samarbejdet og processen går op i en højere enhed.

Motivationen bliver enorm, kunden bliver glad, teamet bliver produktivt og tager et kæmpe ansvar.

Og kan vi gøre en forskel og levere værdi, mens vi rent faktisk har det fedt - så bliver det sgu ikke meget bedre. Det er værd at stræbe efter!

Hvad er så "the catch"? Det er, at det ikke er nemt at opnå dette. Det kræver mod og ihærdighed.

Klap dig selv på skulderen og fail forward

Og når I kommer igang, så husk på, at det, I får ud at arbejde agilt, er *læring*. Det handler om at acceptere, at vi ikke ved alt, og at vi ikke kan styre verden omkring os. Det agile mindset handler rigtig meget om at skabe *læring* gennem *feedback loops* og jo mere af dét, du kan indføre i dine projekter og hverdag, desto bedre.

Inviter dig selv på kaffe eller kig forbi

God vind ud på de nye farvande! Giv lyd, hvis du får brug for lidt hjælp. Jeg tager gerne en snak omkring, hvad du vil kunne gøre i din organisation.

Det er også tit noget, vi taler om til vores gå-hjem-møder, tag og kig hvad vi har på programmet her: reload.dk/events

Tak 🙏

Lad os opsummere de syv læringer

1. Skab korte feedback loops
2. Arbejd målstyret - ikke planstyret
3. Sid fysisk sammen
4. Ingen plan holder når den rammer virkeligheden
5. Hav et reelt mandat
6. Vis at det virker
7. Hellere lille og lækker end stor og uspiselig



Vil du vide mere?

Så skal du være meget velkommen til at kontakte mig. Jeg vil også rigtig gerne have feedback på, hvad der er godt, og hvad der mangler i denne e-bog.

Jeg har forfattet denne e-bog, men det er selvfølgelig med hjælp fra mine dygtige kollegaer i Reload. Særlig tak til [Alice Jensen](#), som har pioneret mange af vores agile læringer.

Du kan fange mig på rasmus@reload.dk eller på tlf 30224545.

Tak for interessen! 😊

Rasmus Luckow-Nielsen
Adm. direktør i Reload.