

Er I klar til at få udviklet software til jeres virksomhed?

Den bedste vej væk fra et begrænsende system

Din lille eller mellemstore virksomhed har allerede et etableret IT-system, men det er begyndt at virke begrænsende, forældet eller ineffektivt. Hvad er de bedste skridt for at komme videre?

IT er en væsentlig del af jeres forretning, for hvis det ikke var for jeres IT-system ville jeres arbejdsgange tage markant længere tid. Men over tid bliver IT-systemer også forældede, f.eks. når jeres arbejdsgange ændrer sig, og systemerne ikke nødvendigvis følger med. Hvis I er endt med et system der føles gammeldags og langsomt, eller som ikke understøtter jeres arbejdsgange så godt som det kunne, er denne udgivelse til jer. Her dækker vi hvordan man bedst kommer ud af netop sådan en situation.



Den bedste vej væk fra et begrænsende system

1. udgave, 11. november 2018.

Udgivet af deranged, CVR 35695583.

Web: <https://deranged.dk>. Email: get@deranged.dk.

Coverbillede af Wendy Scofield (<https://unsplash.com/@gypsycompassrose>).

Om serien

Deranged er en konsulentvirksomhed, der har alle de kompetencer der skal til for hjælpe din virksomhed med at få mere effektive IT-systemer. I denne serie af dokumenter forsøger vi at samle de overvejelser din virksomhed bør gøre sig, før den kaster sig ud i specialudvikling af software.

Dokumentet er udtryk for vores anbefaling som professionelle udviklere og er baseret på vores erfaring såvel som den seneste forskning i hvordan man bedst koordinerer softwareudvikling.

I andre udgivelser i serien dækker vi også andre typer af virksomheder og situationer, f.eks. startups i opstartsfasen, eller udvikling af første interne IT-system i en lille eller mellemstor virksomhed. Du kan finde de andre udgivelser på deranged.dk/klar-til-software.

Fanget af et system

Det er meget normalt at se software som noget der udvikles i projekter. I et typisk softwareudviklingsprojekt indsamler man først krav, derefter finder man en leverandør, og efter en (ofte forholdsvis lang) udviklingsperiode modtager man et produkt, som lever op til de krav man opstillede. Der er rigtig mange hyppigt forekommende problemer med denne proces.

Det første problem er identificering af krav, og hvordan krav til et IT-system normalt indsamles og struktureres. Ens forretning er ofte meget mere kompleks end den umiddelbart virker, og derfor kan det være en noget nær umulig opgave at beskrive krav i en sådan detaljegråd at en udefrakommende leverandør kan forstå og implementere passende software. Man kan ikke vide præcis hvordan software kommer til at indgå i en virksomhed før man prøver det, og dermed lærer mere om virksomhedens processer. Derudover er identificering af krav ofte en dyr og langtrukken proces.

Det andet store problem er den lange ventetid involveret i udviklingen af et system, og det faktum at systemet leveres i én stor leverance. I den tid der går, kan forretningsgange og praksis nå at ændre sig, så kravene bliver forældede. Medarbejdere i en virksomhed bliver hele tiden klogere på deres arbejde, og hvordan det bedst struktureres. Hvis et IT-system tager 5 år at udvikle, vil det være 5 år forældet set fra de forretningsgange det gerne skulle understøtte. Efter et endt udviklingsprojekt modtager man et kæmpe softwaresystem, som det tager lang tid at træne medarbejdere i brugen af, og det tager lang tid at undersøge om systemet faktisk lever op til de krav der oprindeligt blev stillet.

Projektudvikling er den mest gængse tilgang til softwareudvikling, og de ovennævnte problemer medfører at det er meget normalt at ende med en forretning der er fuldstændig afhængig af et system der understøtter forretningsgange dårligt, eller helt mangler kritisk funktionalitet. Samtidig ses videreudvikling af systemet som nye projekter, der ligeledes skal kravspecificeres og tager lang tid at udføre, og dermed medfører præcis de samme problemer. Man bliver fanget i et system som det er dyrt og langsomt at lave ændringer til...

*

Den vigtigste observation at gøre sig om IT-systemer, er at de er tæt integreret med resten af jeres forretning, og at hele forretningen i virkeligheden er et levende system der konstant ændrer sig. Derfor kan man ikke nøjes med at lave en stor projektudvikling af et softwaresystem, og så regne med at det opfører sig optimalt i forhold til resten af forretningen. Software må ses som et levende system, der skal udvikle sig sammen med forretningen, for at blive ved med at give mening.

Når softwaren først er adskillige år bagud kan det være fristende at erstatte systemet med et nyt, men hvis man vælger samme rute som hidtil går man blot en gentagelse af historien i møde. Problemet er ikke nødvendigvis det eksisterende system eller dets tænkte funktionalitet, men dets mislighold og manglende overensstemmelse med forretningens faktiske behov.

Den bedste måde at undgå de samme problemer på i fremtiden er at opbygge kompetencer i virksomheden til løbende at vedligeholde og videreudvikle på systemer, så de aldrig bliver alt for forældede. I stedet for at bruge sit IT budget på store projekter med års mellemrum, bør man sprede sit budget ud over årene, så man løbende vedligeholder sit system.

Når først problemet er opstået kan det dog være svært at finde ud af hvor man skal starte henne, for at gøre sig fri af problemerne med det nuværende system, hvis ikke løsningen er at erstatte det. De følgende sider dækker nogle af de mest væsentlige overvejelser man bør gøre sig i denne situation.

Løs de vigtigste problemer først

Når et IT-system giver anledning til frustrationer kan det være fristende blot at ønske sig et nyt system. Et mere nuanceret perspektiv vil dog ofte synliggøre bedre og mere effektive løsninger. Ønsket om et nyt IT-system kommer af nogle konkrete problemer, og ved at identificere disse problemer kan man blive bedre i stand til at finde ud af hvad det bedste næste skridt at tage er.

I en sådan proces er det vigtigt at skelne mellem ønsker til løsninger og konkrete problemer. Ønsker til løsninger kommer ofte af et problem, men det er nemmere at finde bred enighed blandt medarbejdere i virksomheden om problemer end om konkrete løsninger, for forskellige folk kan ønske sig forskellige løsninger på det samme problem. Denne del kan I foretage før I involverer en leverandør, men en leverandør vil ofte også kunne hjælpe med at køre processen på en mere effektiv måde, da de ofte har erfaring med typiske faldgruber.

Når der skal vælges en konkret løsning giver det ofte bedst mening at inddrage en leverandør, da de har den nødvendige tekniske viden til at kunne vurdere hvordan et problem løses mest effektivt, så jeres penge er bedst givet ud.¹

For at undgå at bygge software i store, uoverskuelige omgange med alle de relaterede udfordringer kan man starte med det problem der giver størst værdi, og samtidig er nemmest at løse. Dette vil være en vurdering som både leverandør og forretning bør være involveret i: forretningen ved hvor meget værdi det har at løse de forskellige problemer, og leverandøren kan bedst estimere hvor lang tid det tager at løse problemet. Hvert problem bør løses på en sådan måde at løsningen med det samme kan tages i brug i forretningen. På den måde kan leverandøren og forretningen sammen finde ud af detaljer i løsningen, og hvordan den bedst tilpasses den konkrete situation, og dermed giver optimalt afkast. Når det første problem er løst, fortsætter man ganske simpelt med det næste.

¹ Denne observation illustrerer samtidig et væsentligt problem med kravspecifikationer, da der ofte beskrives konkret ønsket funktionalitet frem for arbejdsgange man ønsker understøttet eller problemer man ønsker løst. Leverandører giver et tilbud baseret på virksomhedens ønsker, og projektet kan derfor blive markant dyrere end den billigste løsning der egentlig ville løse de oplevede problemer.

Denne proces fører automatisk til træning af medarbejdere, idet deres brug af systemet giver værdifuld feedback til løsningen, som derefter kan tilpasses og gøres endnu bedre. Dine medarbejdere er eksperter i jeres forretningsprocesser i en grad som en leverandør aldrig kan forventes at være, og derfor er det kritisk at involvere dem i design af systemer der skal understøtte jeres forretning. Ellers ender man med løsninger der på papiret er korrekte, men ikke fungerer i praksis.

Sidst men ikke mindst resulterer en proces med løbende leverancer automatisk i bredere accept blandt medarbejderne, der får lov til at give konstruktiv input i processen. Indførelsen af det nye system vil derfor ske mere gnidningsløst end leveringen af et stort softwareprojekt lang tid efter man oprindeligt identificerede krav. Processen med at levere løbende undgår alle de tidligere identificerede problemer med at bygge software som store projekter.

Har man et stort IT-system vil denne tilgang ofte resultere i at man begynder at bygge et eller flere nye, mindre systemer, som langsomt udbygger eller erstatter funktionalitet i den gamle løsning. Dette kaldes Strangler Pattern opkaldt efter slyngplanter, der vokser op ved hjælp af en tidligere plantes struktur, før de helt overtager. På samme måde erstattes det gamle system lidt efter lidt, men hele processen er drevet af forretningsbehov, så dele af det gamle system der allerede fungerer kan blive ved med at køre.

Det kan selvfølgelig være at der er nogle udgifter til det gamle system som man efter noget tid ønsker at komme af med, og i sådan en situation vil det give mening at erstatte de sidste dele af systemet med noget nyt. Dette skridt bør dog først komme relativt sent i processen.

Når der løbende leveres mere funktionalitet giver det også jer som forretning en mulighed for at opdage problemer eller reagere på ændringer i jeres interne prioriteter, forretning eller arbejdsgange. Fordi der arbejdes på ét (eller få) problem(er) ad gangen, kan I hurtigt ændre prioriteringen og dermed hvad der arbejdes på. Alt hvad der hidtil er leveret vil fortsætte med at give jeres forretning værdi, og tabet er derfor minimalt.

Det betyder også at man har muligheden for at slippe af med en leverandør der efter en periode viser sig ikke at leve op til de forventninger man har. Hvis leverandøren løbende har leveret, vil man ikke stå på bar bund.

Du får mere for pengene ved små, tidlige leverancer

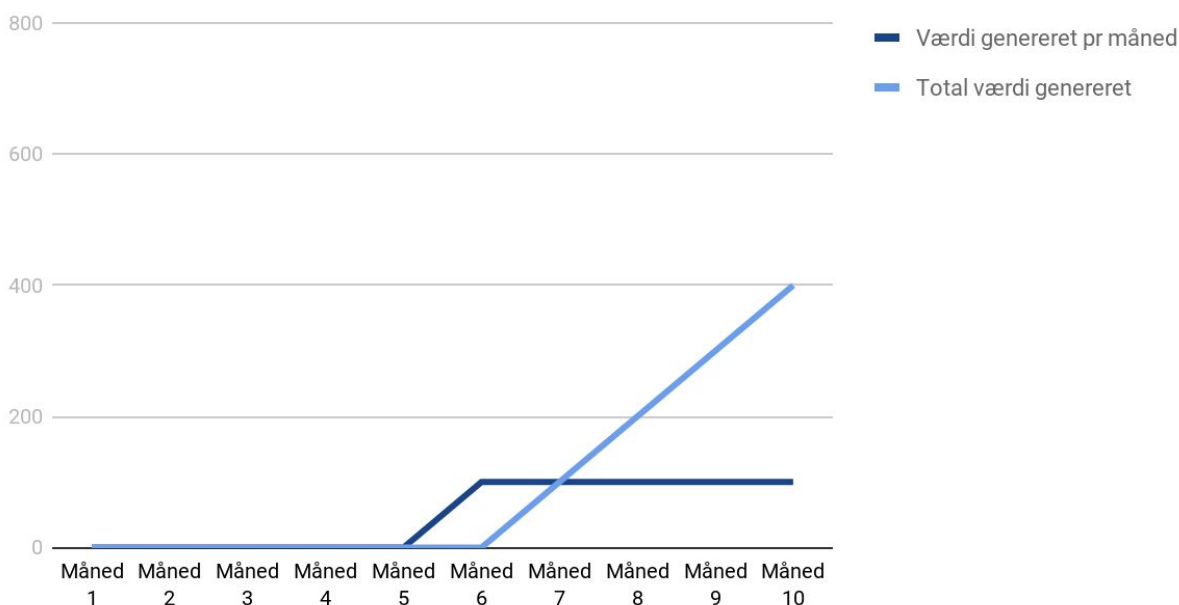
Hver feature i et stykke software skulle gerne give værdi til din forretning. For eksempel vil en bedre søgemekanisme spare tid for dine medarbejdere, når de leder efter information i systemet. Hvis den nye søgemekanisme sparer 10 timer om ugen, er det 10 timer der er frigivet til andet arbejde. Ved at bruge en gennemsnitlig timeløn kan man finde et cirkatal at sætte på værdien for en feature (f.eks. Vil en gennemsnitlig timeløn på 250 kr betyde at værdien er omtrent 2500 kr om ugen).

Hvis søgemekanismen leveres ved afslutningen af et projekt vil de 10 timer sparet om ugen først begynde derfra. Efter én måned har man sparet 2500 kr, efter to har man sparet 5000 kr, osv. På denne måde kan man regne ud hvor lang tid det tager for søgemekanismen at have tjent sig selv ind, altså hvornår besparingerne er lig med den investering man lagde i udviklingen.

Hvis der i stedet for én stor leverance løbende leveres features, kan søgemekanismen leveres tidligt i projektet, hvilket betyder at besparelsen løber over længere tid. Dermed bliver værdien leveret på et hvilket som helst givent tidspunkt markant større, og udgifterne til softwareudvikling tjenes hurtigere ind.

Vi kan sammenligne de to modeller. Ved én stor leverance er der er en lang periode hvor udviklingen ikke genererer nogen værdi for forretningen, hvorefter projektet leveres, og værdien genereret pr. måned bliver den totale værdi projektet leverer.

Værdi leveret ved én stor leverance

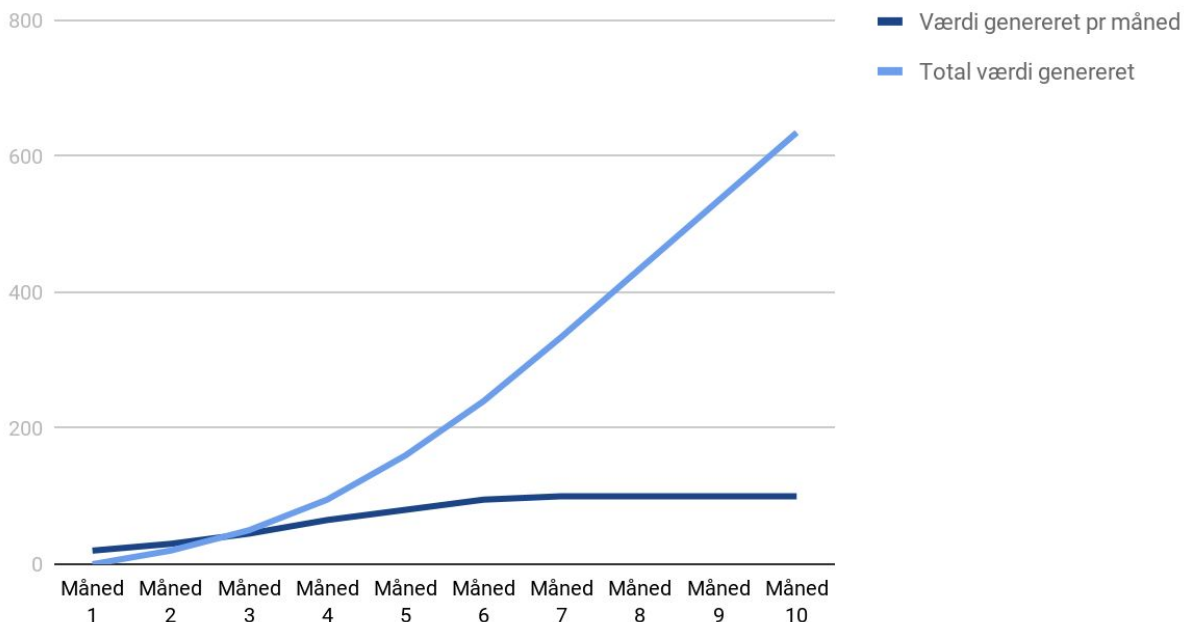


Hvis projektet i stedet leveres i små bidder, får forretningen allerede tidligt værdi. Selvom mængden af værdi leveret efter først måned er mindre end det totale projekt, betyder løbetiden at den totale værdi over tid bliver højere. Forskellen kan illustreres ved søgemekanismen, som vi introducerede som et eksempel tidligere.

Hvis hele projektet tager 5 måneder at udvikle, men søgemekanismen kan leveres allerede efter én måned, er det en konkret forskel på 12500 kr i værdi, som man får ved leverancer i små bidder.

Forskellen er så stor, at selv hvis hele projektet tager længere tid med små leverancer, bliver den totale værdi leveret større.

Værdi leveret ved små, tidlige leverancer



Vælg den rigtige leverandør

I de tidligere afsnit har vi beskrevet hvordan et sundt og gavnligt samarbejde med en leverandør fungerer, og hvad I kan gøre før I kontakter en leverandør. Når I skal finde en leverandør er det vigtigt at de også passer til de behov I har.

Det første valg er mellem leverandører der giver en fast pris på et projekt og leverandører der fastsætter en timepris. Hvis en leverandør giver en fast pris er det en indikation af at de normalt arbejder i projekter, og her gælder de samme farer som nævnt i tidligere afsnit. Det vil ofte være bedre at finde en leverandør der fastsætter en timepris, og aftale et timeantal der passer til jeres budget, og dermed lade værdien I får leveret være variabel. Det gør det muligt for jeres leverandør at reagere på jeres feedback, og sikre at de løsninger I får lavet faktisk løser jeres problemer, og er af høj nok kvalitet til at gøre jeres medarbejderes liv nemmere.

Hvis en leverandør er meget billig, altså tilbyder en lav timepris, vil det ofte være forbundet med en langsommere arbejdshastighed. Vi observerer tit at kunder har brugt mange flere penge på leverandører der er billige i timen, end de ville have hvis de havde brugt en leverandør af højere kvalitet (og dermed højere timepris) til udvikling af det samme produkt. Man kan dog også komme til at betale for meget i timen, så det rette niveau er en balancegang.

Det er vigtigt allerede tidligt at diskutere mulige leverandørers tilgang til softwareudvikling igennem, for at finde ud af om de er kompatible med jeres tilgang. For eksempel kan det være en god ide at undersøge om de er vant til at reagere på løbende feedback, og høre om deres tidligere erfaring med at levere små funktionelle produkter løbende. Det er også vigtigt at undersøge om leverandøren har erfaring med at arbejde med kunder for at identificere krav til software, således at ansvaret for deres løsnings korrekthed bliver fornuftigt fordelt mellem jer, i stedet for at hele ansvaret ligger hos jer.

Hvis man har behov for at ens systemer lever op til juridiske krav udover de gængse krav der findes i dansk lovgivning (f.eks. kunne man have krav til at den amerikanske HIPAA-regulering overholdes, hvis automatiseringen behandler patientdata og skal bruges i sundhedsvæsenet i USA) skal man sikre sig at leverandøren har de nødvendige kompetencer. Det vil ofte blive dyrere, og involvere timer brugt på juridiske vurderinger udover selve automatiseringsprocessen.

Samtidig kan du have behov for leverandører der har eksperter i dit arbejdsområde. Erfarne leverandører vil tit være stærke til at oversætte dine medarbejders forklaringer af deres opgaver til software, men det kan være at dit arbejdsområde kræver specialister der forstår området endnu bedre end dine medarbejdere. Det er sjældent tilfældet, men det er væsentligt at overveje om I kunne falde i denne kategori. Oftest vil automatisering løses bedst af et hold af generalister, der kan gå til mange forskellige opgaver, og levere produkter der fungerer godt hele vejen fra brugeroplevelse til softwareinfrastruktur.

Det kan være relevant at undersøge leverandørens tilgang til løsning af et problem, f.eks. deres valg af programmeringssprog og framework. Et framework sætter rammen for hvordan software kan bygges, men begrænser også hvor fleksibel man kan være i sin udvikling. Det er væsentligt at sikre sig at man ikke unødigt låses fast til et kommercielt framework, som det kan være svært at komme ud af senere i processen. Det kan også være sværere at skifte udvikler, hvis man er afhængig af et mere obskurt framework, som færre udviklere kender til.

Sidst men ikke mindst er det vigtig at sikre sig at leverandøren har den nødvendige kapacitet for at kunne leve op til dine forventninger om budget og leveringstid. Her er det især væsentligt at spørge ind til hvordan de vil håndtere opgaven, hvor mange personer de kan engagere i arbejdet, og hvor lang tid der går før de kan gå i gang.

Er deranged de rette for dig?

Vi er et lille firma der fokuserer på at udvikle software i tæt samarbejde med kunder. Vi er især gode til at gå fra problem til løsning, og finde den hurtigste og billigste løsning for kunder, så de slipper for at spilde penge på unødvendig kompleksitet.

Vi har eksperter i en bred vifte af emner inden for softwareudvikling, fra brugeroplevelser, til effektiv kode, til opsætning af IT-infrastruktur, og leverer specialdesignede løsninger til kunder i mange forskellige industrier, fra tech startups til sundheds- og teleindustrien.

I din virksomhed kan vi hjælpe med at lægge en god og effektiv strategi for at komme videre fra gamle systemer, med fokus på hurtigst muligt at løse de kritiske problemer I står overfor. Vi kan integrere med og rådgive eksisterende IT-afdelinger, og inddrage dem i udviklingsprocessen. Vi kan hjælpe jer med at optimere jeres IT-afdeling og hjælpe jer med at blive bedre til at tage højde for jeres forretningsbehov i den software I bygger eller vedligeholder.

Vi går meget op i at vores kunder skal have en bæredygtig vej frem. Derfor tilbyder vi også hjælp med at etablere en sund kultur i udviklingsafdelinger, samt at ansætte og træne flere medarbejdere, eller træne og oplære eksisterende IT-afdelinger.

Hvis det lyder som om vi ville være en passende leverandør for jer, kan vi kontaktes på email get@deranged.dk. Fortæl os lidt om jeres virksomhed, og hvilke problemer I står foran, så kan vi aftale et uforpligtende møde og snakke nærmere om potentielle løsninger. I kan også læse mere om os på <https://deranged.dk>.