BOOTSTRAP

Bootstrapping in business means starting a business without external help or capital.

Fra Wikipedia

Introduksjon

De senere årene har antall tjenester på nett nærmest eksplodert. Årsaken er at det aldri før har vært enklere og billigere å starte noe nytt. Alt du trenger får du mer eller mindre gratis på nett. Gratis e-post. Gratis programvare. Billig infrastruktur. Billig utstyr. Og, som med denne pamfletten, gratis kunnskap. Jeg håper den kommer til nytte.

Selskapene som for få år siden var helt ukjente for de fleste, har benyttet seg av en kombinasjon av gratis verktøy og kunnskap med trender i samfunnet. Noen har også drevet trendene. Uber har blitt et av verdens største transportselskaper, uten å eie et eneste kjøretøy. Airbnb har blitt en av verdens største «hotellkjeder», uten å eie et eneste rom. Fellesnevneren er at de har benyttet seg av verktøy, infrastruktur og kunnskap som er lett tilgjengelig for dem, noe som har gjort det mulig å tenke globalt og skalere fra liten til stor med tilnærmet ingen ressurser. Snapchat startet som to personer som bygde sin løsning i Google sin infrastruktur, uten at Google var klar over det i starten.

Videre finnes det crowdfunding-plattformer som Kickstarter og Indiegogo som formidler kontakt mellom folk med ideer og folk med penger, og som gjør det enkelt å komme i kontakt med markedet man retter seg mot. Aldri før har det vært enklere å komme ut i markedet.

Det er snart 20 år siden jeg startet i min første stilling som IT-konsulent. Jeg har arbeidet for store konsulentselskaper og små konsulentselskaper. Jeg har hatt store kunder og små kunder. Jeg har arbeidet for offentlig sektor og privat næringsliv. Jeg har hatt kompetente kunder og inkompetente kunder. Jeg har vært konsulent og nå er jeg selv kunde. Jeg har arbeidet med vannfallsmetodikker og smidige metodikker. Å være konsulent innebærer vel så mye læring som det å lære vekk. I løpet av alle disse årene har jeg lært masse, og hva jeg vil lære i løpet av de neste 20 har jeg overhodet ingen anelse om.

Denne pamfletten er en slags komprimert utgave av erfaringene jeg har med å starte nye initiativer og prosjekter, enten internt i noe som allerede er etablert, eller helt nye. Det er ikke sikkert de gir mening for deg. Det er ikke en gang sikkert de vil fungere neste gang jeg selv måtte starte noe nytt. Kan hende, eller skal vi si sannsynligvis, gjør jeg noe feil. Og helt sikkert må det jeg her skriver oppdateres igjen om ikke lenge, for erfaringene endrer seg naturligvis kontinuerlig.

Tipsene er skrevet for deg som ikke nødvendigvis har masse erfaring med det å starte noe radikalt nytt fra før, men kanskje kan også du som har erfaring lære noe nytt. Kanskje er du ansatt i en bedrift. Kanskje du vil starte for deg selv. Kanskje er du mellomleder. Kanskje du er toppleder.

Uansett hvem du er, vil jeg gjerne høre fra deg. Jeg vil høre fra deg hvis du gjennomfører noe av det jeg omtaler i denne pamfletten og jeg vil høre fra deg hvis du lar være. Hva fungerte? Hva fungerte ikke? Hvorfor lot du være?

Dette er ikke en fagbok. Den bygger ikke på eksakte vitenskapelige metoder og analyser, men den bygger på empiri, erfaringer, god praksis fra anerkjente metoder og kunnskap som er kjent i bransjen. Jeg har vektlagt at den skal være faglig nok til at den gir verdi, samtidig som at den skal være kort og lettlest, lettforståelig og gi konkrete nok råd til at du som leser den selv kan teste tipsene du får. Temaet jeg skriver om er den vanskeligste delen av jobben min, men samtidig også den viktigste.

Har du innspill til innholdet, må du gjerne sende meg en e-post til martin@bekkelund.net.

1

Kostnaden ved å gjøre ingenting

Kostnaden ved å gjøre ingenting

Kan hende lurer du på hva du egentlig skal gjøre. Hva skal du gjøre med denne pamfletten? Hva skal du gjøre med det du lærer?

Sannsynligvis vet du svaret, men du vet kanskje ikke hvor du skal begynne.

I disse dager snakker alle om digitalisering. Dette enkle ordet som egentlig er ganske enkelt å forstå og gjøre, men som har blitt tømt for mening de siste årene. Hva er du redd for? Hva er det verste som kan skje din bedrift? Slik digitalkameraet tok livet av Kodak. Eller smarttelefonen livet av Nokia. Du klarer helt sikkert å komme på noe for deg og din bedrift. Hva skal du gjøre for å imøtekomme utfordreren? Du skal teste ut en bedre tjeneste eller et bedre produkt, og det er dette du lærer i denne pamfletten.

Alternativet er å vente å se. Å håpe at ingenting skjer. Å vente til det går over. Men å gjøre ingenting kan bli kostbart, slik som for Kodak og Nokia.

2

Hva må du som leder gjøre?

Hva må du som leder gjøre?

Dersom du som ansatt i en bedrift har tro på at det å teste ut ideer slik det er beskrevet i denne pamfletten er en riktig måte å arbeide på, hjelper det lite dersom ledelsen i bedriften din tror noe annet. Ledelsen må kjøpe ideen om at eksperimentering og testing av ideer i markedet er overlegent planlegging, utredning og analyse. Siden endringene i markedet skjer så raskt, risikerer et for omfattende planleggingsarbeid å være utdatert før man i det hele tatt rekker å teste ut ideen i markedet. Mange vil nok føle en viss trygghet i å gjøre grundig planleggingsarbeid, men all erfaring tilsier at slikt arbeid bare gir en følelse av trygghet og ikke reell trygghet. Det er denne erfaringen som dannet grunnlaget for The Agile Manifesto som ble skrevet for over 20 år siden, og som i dag er gjeldende praksis for hvordan man bedriver produkt- og tjenesteutvikling i dag.

3

Hva må du som leder gjøre?

Et av de vesentligste skillene mellom gode og dårlige ledere i dag, er mellom dem som aksepterer slik eksperimentell arbeidsmetodikk og dem som ikke gjør det.

Som leder må du akseptere at verden er uforutsigbar og ukjent og at ingen finner svarene bak et skrivebord. Du må genuint ønske at bedriften din skal arbeide slik og du må gjøre nødvendige tiltak for at den gjør det. Du må skaffe den riktige kompetansen og du må organisere bedriften på en måte som gjør det mulig. Du trenger mennesker som har kompetanse i det som omtales i de følgende kapitlene og du må sørge for at de får organisere seg slik det beskrives. Eller enda bedre, du må sørge for at de får organisere seg slik de selv mener er best mulig. Erfarne ledere blander seg sjelden opp i detaljene. Dulting, kaller jeg dette, hvor man dulter medarbeiderne i den retningen man skal. Man leier dem ikke. Man dytter dem ikke. Man dulter dem.

3

Hva er smidig?

Hva er smidig?

Det snakkes mye om smidig. Ikke bare i IT-bransjen, men innen produktutvikling generelt. Smidig er en slags samlebetegnelse på prosjektmetodikker som ikke har noen fast definert prosess de følger. Essensen i smidig er godt oppsummert i The Agile Manifesto. Selv om manifestet begynner å dra på årene, holder det seg forbausende godt. Små leveranser, tidlige leveranser, som alle leverer verdi til kunden. Skriv gjerne ut prinsippene i manifestet, heng dem opp og les dem om og om igjen.

Smidig er motparten til såkalte vannfallsmetodikker, der man kartlegger, planlegger og gjennomfører gjennom en prosess som i liten grad tar inn over seg læring man tilegner seg i prosessen.

3

Hva er smidig?

For meg handler smidig om leveransene og kundene, ikke om prosessen og metoden. Om det heter Scrum eller XP er ikke viktig. Det handler om læring og forbedring og verdiskaping, om og om igjen.

Samtidig kan smidig kritiseres for flere ting. Smidig tar gjerne utgangspunkt i det å bygge noe. At man med en viss rimelighet vet hva som skal bygges, i det man starter å bygge. En av de store utfordringene med å starte noe nytt, er at man ikke må starte å bygge noe før man vet med rimelig sikkerhet at det man bygger er det riktige. Det er derfor det med tiden har kommet bøker og metoder som The Lean Startup og design sprint som beskriver nettopp viktigheten av læring og innsikt før man bygger, samt viktigheten av å bygge noe lite som kan testes og justeres samtidig som man får mer innsikt. Samtidig kan man med rimelighet hevde at dagens måte å arbeide smidig på tar inn over seg nettopp denne kritikken.

3

Hva er smidig?

Hvem må du overbevise?

Jeg har i et tidligere kapittel skrevet om hva man som leder må ha tro på. Men hva gjør du som medarbeider? Hvem må du overbevise for å teste det som står i denne pamfletten? Hvis du driver for deg selv, er dette punktet uproblematisk. Hvis du derimot arbeider i en større bedrift, er sannsynligheten stor for at ikke alle deler entusiasmen for smidig og eksperimentell produktutvikling i samme grad som deg selv. Det vil alltid være noen som ikke vil gi slipp på prosjektrapporter og milepælsplaner og kravspekker og fremdriftsplaner og endringsanmodninger og business case og analyser og presentasjoner og andre artefakter som de tror gir dem kontroll og styring.

Det er ikke et universelt svar på hva som vil overtale beslutningstakere til å arbeide smidig. Kan hende hjelper det å peke på verden utenfor. Hvordan arbeider selskapene som var ukjente for bare få år siden, men som i dag er blant verdens største? Heller ikke her finnes det et universelt svar, men det finnes noen fellesnevnere: alle benytter smidige metoder, fokuserer på å levere verdi for sine kunder og verdsetter læring og eksperimentering fortløpende.

3

Hva er smidig?

Autonomi er også et viktig stikkord, noe vi kommer tilbake til i et senere kapittel. Når man har dyktige medarbeidere må man også gi dem frihet til å utøve sitt fag. De må ha autonomi til å ta selvstendige valg, både når det gjelder hvordan man arbeider, hva som skal lages og hvilke verktøy man bruker til jobben. Det er ledelsens ansvar å beslutte og anerkjenne slik autonomi.

Samtidig må du ha respekt for at det ikke bare er å beslutte at man skal arbeide smidig. Kultur og erfaring er innarbeidet og krever innsats hvis man skal endre noe. Det viktigste er uansett erkjennelsen av at digitalisering truer etablerte aktører i alle bransjer, og krever nye måter å arbeide på. Dette er gjerne inngangen for smidig. Hvem vil vel ende opp som Kodak?

4

Hvem er kunden?

Hvem er kunden?

Det er ufattelig lett å forelske seg i egne ideer, spesielt hvis det er noe man selv synes er nyttig. Men å lage et produkt basert på deg selv som kunde er en dårlig start. Du trenger noen kunder som kan utfordre ideen din. Og noen trøbbelmakere som kan stille de kjipe spørsmålene. Jo raskere du får utfordret ideen din, jo raskere kan du forbedre den.

Glem aldri kundene, for det er de som skal betale lønnen din.

Hvis du synes at noen kunder er teite fordi de ikke vil ha produktet ditt, kan det være flere årsaker til det. Her er det viktig å forstå hvorfor kunden sier nei, i stedet for å bli irritert over å bli avvist. Ta med kunnskapen du får og forbedre produktet. Hva skal til for at kunden sier ja? Det heter at kunden har alltid rett, men det følger også med en bisetning «men ikke for enhver pris». Når du lager løsninger i tett samarbeid med kunder, må du også passe på at det du lager ikke blir for spesialisert, slik at det ikke skalerer og fungerer for andre kunder.

4

Hvem er kunden?

Det er mange måter å definere kunder og markedssegmenter på. Ideelt sett bør du finne en nisje ingen andre opererer i, men hvor lett er egentlig det? Og hvis ingen andre opererer i denne nisjen, hvorfor gjør de ikke det? Det finnes også forskjellige strategier for hvilket markedssegment du bør gå inn i. Der elbilprodusentene Think og Buddy gikk inn i low-end-segmentet, gikk Tesla inn i premium-segmentet. Forskjellen er at det i premium-segmentet antas å være høyere betalingsvilje, noe som i sin tur kan finansiere utvikling av produkter for low-end-segmentet, uten at det nødvendigvis alltid er sant. Det er uansett viktig å definere hvilket segment man bør inn i, noe som fremgår av neste kapittel.

Uansett hva du gjør, må du være villig til å innse at ideen din kanskje er dårlig, at ingen vil ha produktet ditt. Ha, så langt det er mulig, et rasjonelt forhold til den innsikten kundene dine gir deg og vær villig til å ta livet av ideen din tidlig, om nødvendig.

5

Visualiser forretningsmodellen

Visualiser forretningsmodellen

En fellesnevner jeg ser på tvers av mange startups jeg har snakket med, er at de sliter med å forklare forretningsmodellen sin skikkelig. Det betyr ikke at de ikke har en, eller at den er dårlig, kun at den trenger å tydeliggjøres for at alle skal forstå den.

De som kjenner meg vet at jeg får utslett av metoder eller prosesser som er for omfattende. En venn av meg uttalte at «PRINCE2 er som et atomkraftverk: alt må måles og overvåkes kontinuerlig, hvis ikke sprenger noe i lufta». Derimot vender jeg ofte tilbake til Business Model Canvas. På ett eneste A3-ark kan du visualisere hele forretningsmodellen din, så enkelt og selvforklarende at du ikke trenger å bruke timesvis på å forstå hvordan det skal gjøres. Men det lønner seg å lese litt om hvordan du bør bruke det.

Du trenger ikke ha svar på alle delene av Business Model Canvas, men sørg i det minste for å ha dekket «Value Propositions», «Customer Segments», «Revenue Streams» og gjerne også «Customer Relationships» og «Channels». Regn også litt på inntekter og kostnader, så får du vurdere den omfattende jobben med et fullstendig business case, det blir sjelden som du tror uansett.

6

Visualiser verdiforslaget

Visualiser verdiforslaget

Når du har en idé om hvem som er kundene, hva du skal levere og hvordan du skal tjene penger, bør du se på verdiforslaget. Et verdiforslag er et mye misbrukt ord, men er enkelt forklart hvilken verdi ditt produkt skal levere til de kundene du har definert. Hvilke problemer løser du og for hvem?

Akkurat som med Business Model Canvas, benytter du her Value Proposition Canvas. Som du vil se er Value Proposition Canvas kun en detaljering av Business Model Canvas.

Gjør det tydelig hva du skal løse for kundene og gjør det enkelt å forstå.

7

Hva med en design sprint?

Hva med en design sprint?

Flere av elementene jeg omtaler, inngår i en såkalt design sprint, en femdagersprosess utviklet av Google Ventures for å prototype og teste ideer. Design sprinter har blitt svært populære og har en del til felles med ideene i The Lean Startup, men min erfaring er at design sprinter er mer konkrete, enklere å forstå og forholde seg til og ikke minst å gjennomføre. På den annen side har de blitt kritisert for ikke å være forretningsnære nok. Det vil si at de kan helt fint løse reelle problemer, men det ligger ikke i metoden å prioritere mellom mange forskjellige forretningsproblemer. Hva som gir mest verdi gir en design sprint deg ingen åpenbare svar på.

En av de åpenbare fordelene med en design sprint, er absolutte time-box-er. Én dag per oppgave. Til slutt sitter man igjen med en testbar prototype, og noen testbare hypoteser. Ingen grunn til å kaste vekk kostbar tid. Kjøp boken og prøv det selv. Jeg kjenner sågar til design sprinter som har vært gjennomført på én dag, visstnok med tilfredsstillende resultat. Det avhenger mye av oppgaven.

8

Test hypotesen din

Test hypotesen din

Så til noe litt mer vanskelig, og det er her mange (meg selv inkludert) har feilet. Mange hopper her rett på utviklingen og starter å bygge produktet før de har fått verifisert at kundene faktisk vil betale for det. Hvem kan klandre dem? Det er mye enklere å selge et produkt som faktisk eksisterer, fremfor å selge noe basert på en idé. Problemet er bare at du har kastet bort tid og penger hvis du har laget feil produkt, det vil si et produkt som ikke løser reelle behov og som ingen vil betale for.

Jeg har skrevet om det før, at det er ingen vits i å dra på jakt hvis ingen vil kjøpe skinnet. Det gamle ordtaket om at man ikke skal selge skinnet før bjørnen er skutt er mer til skade enn til nytte. Selg skinnet, dra på jakt!

8

Test hypotesen din

Det er ingen grunn til å sette i gang med utvikling hvis du ikke har testet hypotesene og fått bekreftet at noen vil betale for det du har tenkt å lage. Hvis dette feiler er alt annet du finner på bortkastet, for da er det bare et spørsmål om tid før du går tom for penger. Det er her en Design sprint hjelper deg.

Dette er en helt reell måte å arbeide på, og den krever is i magen og at man er litt kynisk. Jeg har selv levert tilbud på store kontrakter til offentlig sektor, uten å ha noe som helst annet enn papiret tilbudet er skrevet på. Og jeg har fått kontraktene. Og deretter bygget løsningene. Alternativet hadde vært mye verre: at jeg hadde bygget løsningene først, skrevet tilbud men ikke fått kontraktene. Jeg kjenner mennesker som oppriktig og i god tro mener at «build it, and they will come» er en foretrukken måte å utvikle produkter på. Dessverre er det ingen sammenheng mellom det å ha det beste produktet og det å få kunder, noe historien er full av eksempler på, for eksempel VHS versus Betamax.

9

IT-prosjekter finnes ikke

IT-prosjekter finnes ikke

Det er ingenting som heter et IT-prosjekt. Produktet du skal lage skal være under utvikling til det dør. Hvis utviklingen av produktet stopper opp, dør produktet. I stedet for IT-prosjekter, snakker vi om prosjekter. Slike prosjekter er for viktige til å overlates til IT-avdelingen, men skal i stedet være drevet i og av forretningsenhetene.

10

Drit i dokumentasjonen

Drit i dokumentasjonen

Løsninger som er i kontinuerlig endring har ikke behov for dokumentasjon. Hva er poenget med dokumentasjon? Hvem er den for? Kommer noen egentlig til å lese den? Hvor lang tid tar det før den er utdatert og dermed bortkastet? Har kundene bedt om den?

Dokumentasjon gir ofte et feilaktig inntrykk av å ha kontroll, men for å trekke frem et mye brukt sitat i smidig-sammenheng:

> *If everything's under control, you're going too slow!*
> – Mario Andretti

Og så er det selvfølgelig forskjell på FAQs eller hjelpesider og dokumentasjon.

11

Etabler et lite team

Etabler et lite team

Når alle forutsetningene for å arbeide smidig og eksperimentelt er oppfylt, begynner det morsomme arbeidet. Mennesker skal organiseres og finne sin arbeidsform.

Det viktigste man trenger å forstå i denne fasen, er at man ikke trenger å skaffe alle tenkelige ressurser fra første dag. Start med et lite team. I første omgang trenger du kun noen som visualiserer forretningsmodellen, gjør innsiktsarbeid og tester hypoteser og skisser. Senere trenger du et par utviklere som også kan drift (DevOps). Bruk en av dem som var med fra starten som produkteier. Senere trenger du kanskje noen selgere? Hold teamet lite! Skaler opp når økonomien tillater det og behovet melder seg. «Think big, start small, scale fast» heter en annen smidig floskel.

En interessant observasjon gjennom mange år i IT-bransjen, er at arkitekter ofte gjør mer skade enn nytte. Arkitektur er viktig, arkitekter er det ikke. Rollen har det med å fungere som en hviskelek mellom forretningsmenneskene på den ene siden og utviklerne på den andre; ikke har de direkte kontakt med kundene og har forretningsforståelse, ikke har de direkte kontakt med hva som utvikles.

12

Etabler en kultur

Etabler en kultur

Hold hele organisasjonen samlet på ett sted. Alle skal sitte sammen og jobbe sammen, men ha en kultur hvor man også kan jobbe konsentrert. Bygg den kulturen du vil ha hos deg.

Spis lunsj sammen og ta en øl etter jobb.

13

Før du bygger

Før du bygger

Uansett hva du skal lage, så må det sikres. Forstå standarder og metoder og lovverk og begreper som ISO/IEC 27001 og BSIMM 6 og GDPR og hva som enn måtte være gjeldende når du starter. Bygg kultur, rutiner og standarder først, ikke etterpå. Det trenger ikke være vanskelig eller komplisert eller ta lang tid.

Gjør det, hvis ikke er det bare et spørsmål om tid før du driter deg ut.

14

Finn en smidig metodikk

Finn en smidig metodikk

Når du skal bygge trenger du en utviklingsmetodikk. Kommer du fra vannfallsmetodikker er Scrum en fin avrusningsmetodikk. Har du erfaring fra Scrum kan du droppe Scrum helt og gå for XP uten iterasjoner (sprinter). Prioritere heller arbeidsoppgavene kontinuerlig; hva gir mest verdi for minst mulig innsats? Bruk Kanban for visualisering, som beskrevet i et senere kapittel.

Lag små brukerhistorier, små releaser, ha stand-ups og demo. Test det du lager med kundene.

Selv ikke etter at du har fått kunder trenger du å ha alt bak kulissene klart heller. Det er ingen grunn til å lage helautomatiserte prosesser før du har fått en kritisk masse kunder. Det holder lenge med manuelle prosesser bak kulissene, inntil arbeidsmengden har blitt et luksusproblem. Dette kalles et Wizard of Oz-eksperiment. Du merker gjerne at det er på tide å automatisere noe når du drukner i de manuelle prosessene. Følg nøye med på dem.

15

Bygg det i skyen

Bygg det i skyen

Aldri før har det vært enklere og billigere å bygge løsninger uten å investere i masse dyrt utstyr som servere og infrastruktur og programvare på forhånd.

I dag er det ingen grunn til å bruke en eneste kalori på infrastruktur, når det finnes selskaper som har det som sin eneste oppgave. Det er så enkelt at selv undertegnede klarer å bruke dem.

Gi alle utviklerne alle rettigheter (root) i alle miljøer, slik at de kan produksjonssette og gjøre endringer fortløpende. Etabler et sikkerhetsregime for godkjenning av endringer som skal gjøres i produksjon, dersom det er nødvendig.

Det finnes massevis av fantastisk programvare som er gratis og fritt tilgjengelig når du skal bygge løsningene dine. Fri programvare er en av byggestenene for internett og er tilgjengelig også for deg. Se etter programvare med en reelt fri lisens, slik som liberale lisenser i kategorien BSD, MIT eller Apache. GitHub lar deg sågar søke etter programvare med disse lisensene.

16

Lag minimumsløsninger

Lag minimumsløsninger

Lag minimumsløsninger, få dem i produksjon, lanser dem i markedet, få tilbakemeldinger tidlig, forbedre produktet og gjenta. Mål kun det du kan gjøre noe med. Mål inntjening. Forbedre prosessen kontinuerlig

Forbedring innebærer vel så gjerne å fjerne rutiner fremfor å innføre dem. Ha fokus på salg og utvikling av produktet, ikke på merkantile rutiner som krever bruk av dyrebar tid. Automatiser rutiner når du bruker for mye tid på å gjøre dem manuelt. Utviklere er en samvittighetsfull gjeng. De jobber så fort og godt de kan. Hvis du synes utviklingen går for sakte bør du se på hvordan du kan forbedre utviklingsprosessen fremfor å mase på utviklerne.

17

Bruk Kanban

Bruk Kanban

Kanban er mer et verktøy enn en metode. Det er derfor jeg liker Kanban, fordi du står friere til å definere prosess og metode selv, og Kanban fokuserer på hva du gjør og ikke hvordan du gjør det. Bruk Kanban til å visualisere arbeidet, for eksempel med Trello. Ha et Kanban-board som visualiserer backlog, i arbeid, verifisering og akseptanse på samme board. I tillegg kan det være smart med et board for pre-utvikling (grooming) og et for post-utvikling (releaser).

18

Gi folk frihet

Gi folk frihet

Det snakkes mye om autonome team. Når du har dyktige mennesker rundt deg, må du gi dem frihet til å utøve faget sitt, selv om det innebærer at oppgavene som skal gjennomføres løses på andre måter enn hvordan du ville ha løst dem selv. Dyktige mennesker liker ikke å bli detaljstyrt. Det gjelder sannsynligvis også deg selv.

Forklar *hva* som skal løses, så får andre finne ut *hvordan* det skal løses.

19

Begrense flaskehalser

Begrense flaskehalser

Begrense alle flaskehalser i utviklingsprosessen. Ha for eksempel ikke flere oppgaver i arbeid enn det er utviklere på teamet. Har du 3 utviklere på teamet skal det aldri være mer enn 3 oppgaver i arbeid om gangen. Ha også en begrenset back-log. Da holder du også fokus på de viktigste oppgavene.

20

Få viral spredning

Få viral spredning

Snakk om produktet ditt. Få viral spredning (om mulig). (No shit, Sherlock!)

21

Ha slack

Ha slack

Slack er betegnelsen på å ha rom til å gjøre andre oppgaver enn bare de som til enhver tid er prioritert. Slack til å ta unna teknisk gjeld. Slack til å ta unna design-gjeld. Slack til å lære seg nye ting. Slack til å jobbe på noen arbeidshobbyer. Slack til å stoppe opp og reflektere litt. Slack er for øvrig også et utmerket arbeidsverktøy.

At man ikke belaster all kapasitet 100 % er bra for mange ting, både for flyt og motivasjon og trivsel.

22

Ha det moro

Ha det moro

Ta det med ro. Du har dårlig tid, men hvorfor stresse?

23

Noen ord til slutt

Noen ord til slutt

For mange er disse punktene radikale, selv om de selvfølgelig er spissformulert. Noen vil sikkert til og med kalle det uforsvarlig. Mindre rutiner? En prototyp på bare fem dager? Lansere uferdige produkter? Gi utviklerne alle rettigheter i produksjon? Ingen dokumentasjon av det som utvikles? Release fortløpende?

Det stemmer. Det gir mening den dagen du prøver det ut.

24

Del med dine venner

Del med dine venner

Ønsker du å dele disse ideene med andre? Send dem denne pamfletten! Print og del den gjerne, men gjør ikke endringer i den eller selg den for penger.

Kunnskapen du får i denne pamfletten er gratis, men hvis du føler at du har lært noe verdifullt, kan du gjerne donere et valgfritt beløp via Vipps til 930 555 10. Takk!

Om forfatteren

Martin Koksrud Bekkelund (f. 1977) er til daglig direktør for produkt- og forretningsutvikling i en av Norges største bedrifter, hvor han arbeider med digitalisering og teknologiledelse.

Han har en høyere IT-utdannelse fra Westerdals Oslo School of Arts, Communication and Technology (tidligere NITH/DPH).

Siden 1999 har Martin arbeidet i IT-bransjen med store og små teknologiprosjekter, både som konsulent, prosjektleder og leder. Han har arbeidet i alt fra tradisjonelle virksomheter med vannfallsmetodikk til DevOps-orienterte enheter med smidig metodikk. Han har arbeidet med å omstille tradisjonelle virksomheter til å være bedre organisert i møtet med digitaliseringen som alle bedrifter står ovenfor.

Du treffer ham på bekkelund.net

2018-07-08