**Eksamen i Programvareutviklingsmetoder skal besvares i dette dokumentet!**

Sandefjord Travel Company – Case oppgave

**Min definisjon/ tolkning av case :** Få en oppklaring på case, hvor hovedfokus ligger her. Dele opp oppgaven, svare etter beste evne ved å drøfte, fordeler og ulemper å benytte ressursene riktig i forhold til prosessen og utviklingsmodellen jeg skal følge. Rollen min som konsulent vil derfor være å tilpasse nettsiden å sette den opp sammen med teamet mitt. Hvor denne blir tilpasset etter et kravspesifikasjonsdokument utvikles i tråd med kunden vår. Tar med eksempler der jeg føler det er relevant.

1. **Valg av arbeidsmetodikk**

**Lean metode:**

Er en smidig arbeidsmetode som har som mål å øke produktiviteten. Eliminere unødvendige trinn i prosessen og øke verdistrømmen til kunden. Den oppstod i bilindustrien og har blitt brukt i ulike områder som programvareutvikling. Lean fokuserer på å minimere avfall, **bruke unødvendige kostnader, eliminere bortkasted tid** og maksimere profitt. Lean er åpen for kontinuerlig endring ved tett samarbeid mellom utvikler og kunde.

**Agile metode:**

Er en tilnærming til programvareutvikling og prosjektledelse som fokuserer på samarbeid, fleksibilitet og iterativ utvikling. Det er en iterativ tilnærming til å planlegge, utføre og overvåke prosjekter ved å fokusere på samarbeid og kommunikasjon med teammedlemmer og kunder, og være åpen for endringer underveis. Agile- metoder er ofte brukt i programvareutvikling.

**XP metode:**

XP modellen **(handler om programvareutvikling).** XP modellen skiller seg fra andre utviklingsmetoder ved å ha en sterk vektlegging på kommunikasjon og samarbeid mellom teammedlemmer og kunder, samt enkelhet og kontinuerlig tilbakemelding.

**Waterfall modell:**

Dette er en lineær fasebasert arbeidsmetode som krever fullføring og godkjennelse av fasene, før man går videre. Dette gir kontroll over budsjett, og egnet for prosjekter med forståelig krav, og lite behov for tilpasning. Ulempen her er begrenset kompleksitet og langsiktig prosjekter hvor det kreves fleksibilitet og tilpasning.

**Hva jeg anbefaler av arbeidsmetoder:**

Ut ifra hva som anbefales av arbeidsmetoder og hva som er understreket bør fokuset ligge på å **spare så mye tid og penger som mulig.** Derfor virker det logisk å velge en Lean metodikk her. Prinsippene i Lean metodikken bør ikke følges i rekkefølge nødvendigvis. **Fleksibel metodikk som gjør at vi kan tilpasse å velge litt selv med tanke på prioriteringer ved utvikling av tjenesten. Basert på kravspesifikasjonsdokumentet under kan vi nå gå i gang med å utvikle nettsiden basert på kravene som er satt.**

**Jeg velger denne fremfor en annen metodikk/fordel og ulempe:**

Fordi denne arbeidsmetodikken går på å minimere **bruken unødvendige kostnader, eliminere bortkasted tid.** Og det er akkurat det Sandefjord Travel Company ser etter som et viktig krav og derfor velger jeg å utarbeide kravspesifikasjonsdokument ut ifra det. De andre metodene kan også benyttes, spesielt Agile fordi det går på utvikling av programmer. Men mangler etterspørselen av hva som bør være fokus(tid og penger). Fossefall er irrelevant, samme med XP- vanskelig med kundeinvolvering. Fossefall er allerede tjenesten ferdig når den er levert og kan ikke endres. **Under lean** kan man være fleksibel og kontinuerlig endre. F**ordelen med metodikken er:** fleksibel og tilpasningsdyktig, kunden integreres underveis og tett samarbeid med utviklingsteam og kunde. Ulempen kan være dyrere arbeidskraft- men det har jeg tatt høyde for med studenter.

**Lean inneholder 5 prinsipper:**

**1. Fokus på kundeverdi: hva trenger kunden? Hva trenger den ikke?** kunde trenger nettside, spare penger derfor billig arbeidskraft å bruke GA studenter til å utvikle nettsiden. Dette vil gi økt kundeverdi til Sandefjord Travel Company.

**2: Kartlegge verdistrømmen: hva aktiviteter gir verdi for oss og kunde? Hva er bortkastet:** kartlegge produksjonsprossessen for å identifisere bortkastet tid og arbeid. Eksempel: overproduksjon av ansatte som er bortkastet. Men i dette tilfelle hvor konsulent selskapet er meg og studenter blir det ikke noe problem. Derfor øker det lønnsomhet for oss og effektivitet. Det vil igjen gi gøyere kvalitet/verdi for kunden fordi de behøver å betale mindre for å leie inn oss.

**3. Flyt: hva fungerer, hva fungerer ikke?** Kontinuerlig forbedre prosessen ut ifra å optimalisere nettsiden. Eliminere hindringer som stjeler tid for kunden. Som kan være overbelastet nettside som gjør at den kræsjer.

**4.Pull systemer:** Gi kunden det den skal ha, ikke noe mer. Eliminere tid, bruke unødvendige kostnader som er noe de verdsetter som igjen gir kundeverdi når vi leverer å fokuserer på dette. **Eksempel:** kunde vil ha nettside, spesifisert hva den vil ha på nettsiden, derfor baserer vi utviklingsprosessen ut ifra det og ikke lager noe mer utenom hva som er spesifisert. Bidra til å redusere ressurser og bortkasted tid.

**5.Kontinuerlig forbedring:** Oppnå perfeksjon på det vi leverer. Og i dette tilfelle hele tiden forbedre oss. Be om tilbakemelding fra Sandefjord Travel Company og ha en tett dialog hele veien slik at man øker arbeidseffektiviteten, sparer tid og holder kostnadene nede.

**Kravspesifikasjonsdokument:**

**Hva det er, innebærer, fordelen, fallgruver med kravspesifikasjonsdokument:**

Kravspesifikasjon definerer nøyaktig hva som skal lages, oppnås og nødvendig for å oppnå. Dokumentet er kontrakt mellom kunde (Sandefjord Travel Company) og utviklingsteam. Sikrer man felles forståelse av hva som skal lages. Gir en god veiledning på utviklingsprosessen. **Fallgruver** kan oppstå hvis det er for detaljerte krav, eller hvis det endrer seg underveis i utviklingsprosessen.

Del av arbeidsprosessen er å utvikle kravspesifikasjonsdokument før man går i gang med selve utviklingen av nettsiden. **Dette går man i gang med før selve produktutviklingen begynner.** Dette innebærer blant annet:

**Produktbeskrivelse:** beskrive kort tjeneste, formål: en nettside som tillater kunder å finne flybilletter, hoteller og pakkereiser. Booke direkte på nettside, få verifikasjon av booking på mail.

**Funksjonelle Krav:** hva skal tjeneste inneholde, fungere?: brukere skal kunne søke etter det som er nevnt i produktbeskrivelse. Se priser, legge til i handlekurv, betale med kort.

**Ikke Funksjonelle:** det som ikke handler om tjeneste, men hvordan det funker: nettsiden skal være responsiv og rask. At det går ut automatisk verifikasjon av booking på mail når de legger inn ordre på nettsiden.

**Designkrav:** tjenestens utseende og estitikk?: nettsiden skal ha et enkelt intuitivt grensesnitt med tydelig skrift, lett å finne frem.

**Teknisk Krav:** krav til det tekniske, som programvareegenskaper og ytelse: nettsiden skal utvikles av programmerere som er nyutdannet ved Gokstad Akademiet som trainee/ billig arbeidskraft. Optimalisere nettsiden kontinuerlig

**Testing og Validering:** Sikre at produkt fungerer som det skal: Nettsiden testes grundig før man gir den ut til Sandefjord Travel Company. Så vet man at alle funksjoner fungerer som det skal. Bl. Annet betalingssystem. Når den er utgitt skal den ha oppfølging av studentene, så man kan ta tak i problemer som oppstår.

**Dokumentasjon:** brukermanual på hvordan man benytter nettsiden for kunden vår. Betaling, finne flybilletter, hoteller og pakkereiser raskest mulig. Hvordan vi har utarbeidet det tekniske med gokstad studentene som er under opplæring/trainee som billig arbeidskraft som var et av kravene. brukermanualer, teknisk dokumentasjon og hvordan benytte nettsiden(brukervennlighet):

1. **Unified Modelling Language**

Er et visuelt verktøy som brukes i programvareutvikling for å beskrive og dokumentere et system. Det gir en felles og standardisert måte å representere systemet på, og består av en rekke diagrammer som representerer forskjellige aspekter av systemet, brukstilfeller, klasser, sekvenser og aktiviteter.

**Fordeler:** gir tydelig kommunikasjon, bedre design, øke effektiviteten, designprosessen, gjenbruk av koden. Men best av alt, hjelper frontend og backend å forstå system på en strukturert og oversiktlig måte

**Ulemper:** komplekst, tidkrevende å lage, begrenset verdi i små eller enkle prosjekter.

I form av caset er Unified Moddeling Language viktig for Sandefjord Travel Company fordi de skal utvikle en nettside. Derfor har vi 3 diagrammer å gå ut ifra. Det er **Use case diagram, Sekvens diagram, Aktivitetsdiagram.** De setter krav om å få et eget booking system, formålet er å finne frem til ulike billetter, hoteller og pakkereiser på nettsiden pluss det aller viktigste, **spare så mye tid og penger som mulig.**

**Aktivitetsdiagram:** er nødvendig fordi det er relevant for målet Sandefjord Travel Company ønsker å nå i første omgang. Som kan være fint å benytte med tanke på bestillingsprosessen som er: (start) -> **velg tjeneste:** eks. *pakkereise* -> legg produkt i handlekurv -> valider bestilling -> behandle betaling -> send bestilling -> (slutt).

**Use case diagram:** viser aktører og handlinger som brukerne kan utføre i systemet som blir laget. Kan eksempelvis være nettsiden Sandefjord Travel Company vise aktører som vanlig bruker og admin. Handlinger som da gjøres er å søke etter produkter, legge til produktet i handlekurv å se produktinformasjon som er billetter, hoteller og pakkereiser. Bidrar til å dekke kravene å hva brukerne av nettsiden skal gjøre. Lean arbeidsmetodikken er fleksibel, og her kan krav endres på underveis i prosessen samt utviklingen av tjenesten i selve metodikken. Så kan være muligheter for å benytte dette diagrammet også.

Grunnen til at jeg ikke velger **Sekvens diagram:** Irrelevant nå i første omgang. Dette går mer på hvordan objekter samhandler med hverandre over tid. Identifiserer problemer med funksjonalitet og ytelse. Siden dette ikke er så relevant nå er fordi man først ser resultater over tid, og nettsiden skal utvikles **nå** med tidsbesparelse. Går på innlogging-> matcher passord er det vellykket (**if – else** hvis ikke matcher for å håndtere problemet).

1. **Git og github kommandoer**

Git er et verktøy for versjonskontroll, Github er en skybasert tjeneste for å lagre og dele koder og prosjekter. Git brukes lokalt på datamaskinen for å administrere koden, GitHub gir en plattform for samarbeid på koder og prosjekter i skyen.  
  
Kobling av Git og Github må man følge trinn som å opprette Remote repository på Github, kloning av repositoryet, legge til filer, commit, pushe endringer og være oppdatert på endringer fra utviklere (pull).

**I Terminal:** Git- kommandoer som **git init-** lager lokalt repository, **add-** legge til filer som ønskes å commite, **status-** sjekke status på prosjekt, **commit-** lager commit av endringene våre, **push-** pusher ut og sender endringene til repository (alle kan se) og **pull-** trekker til seg endringer andre gjør som vi jobber med. Den lar deg administrere koden fra lokal datamaskin og koble den til Github.

**Konklusjon:** Er viktig for et utviklerteam for å kunne oppdatere, å samarbeide tett sammen på kode man benytter for å lage programmer i ulike prosjekter.