|  |
| --- |
|  |
| **Obligatorisk oppgave 3.** |
| INF1050-Systemutvikling |
|  |
| **Krister Borge og Nina Høegh-Larsen** |
|  |

|  |
| --- |
| Estimering, Arkitektur, Aktivitets og tilstandsdiagrammer |

**Oppgave1: Estimering**

1. Foreslå egnet måte (eller egnede måter) for å måle størrelsen eller mengden funksjonalitet til systemet for leie av markasykler. Begrunn svaret og gjør nødvendige antakelser.

Det å finne størrelsen på systemet er altså å finne et antall på funksjonene systemet har. Hvis vi antar at utviklingen skjer plandrevet etter en fossefallsmodell kan vi anvende en «spitt og hersk»-metode for å anslå denne størrelsen.

Vi kan også anta at en del av systemet er gjenbruks, og ikke trenger mye utvikling:

1. Markasyklersystemet skal integreres Ruters betalingssystemer.
2. Stasjonsutstyr er definert på forhånd og har en klar API .

Vi kan nå begynne å se på hvilke funksjoner markasyklersystemet kan ha. Når vi anvender en ”splitt og hersk”-metode deles systemet inn i mindre og mer håndterlige undersystemer for så å dele hver del inni aktiviteter blir det å anslå størrelsen på systemet lettere. Hvis vi deler opp systemet i fire systemkategorier:

1. Betalingssystem
   1. Betale gjennom app/nett
   2. Betale på automat
   3. Endre/legge til betalingsinfo
   4. …
2. Kundedatabase
   1. Legge til ny kunde
   2. Administrere personopplysninger
   3. Hente ut bruksdata
   4. Finne ledige sykler
   5. Rapportere feil
   6. …
3. Ansattsystem
   1. Administrere kunder
   2. Hente ut brukerdata
   3. Hente ut bruksdata.
   4. …
4. Teknisk/vedlikehold
   1. Transport av sykler
   2. Hente inn sykler på overtid
   3. Hente inn ødelagte sykler
   4. …

Vi har lagt inn noen aktiviteter som vil hjelpe oss i å anslå størrelsen på systemet for leie av markasykler.

Videre i estimeringsprosessen vil det å ha erfaring hjelpe oss i å se om våre estimat blir riktige. Hver del kan deles opp i aktiviteter. Betalingssystemet vil inneholde funksjoner som har med håndtering av betaling å gjøre men også funksjoner som sammenkobler betalingssystemet med de andre systemkategoreiene. Disse sammenkoblingene beskriver kompleksiteten av systemet.

Ved bruk av «splitt og hersk» metoden vil vi sannsynelig bruke innholdet i kravspesifikasjonen for å finne aktivitetene. Hvis kravspesifikasjonen er underutviklet eller inneholder feil vil også estimatet bli feil.

Hvis vi antar at utviklingen av utleie av markasykler skjer ved en smidig prosess skjer estimeringen etter hver del som blir utviklet. Utviklingen skjer i etapper og størrelsen anslås per etappe.

1. Foreslå egnet måte (eller egnede måter) for å måle kompleksiteten til systemet for leie av markasykler. Begrunn svaret og gjør nødvendige antakelser.

Det som påvirker kompleksiteten til systemet er både hvordan systemet bygges opp og hvilke krav det er satt til systemet i krav spesifikasjonen. Det at systemet har flere betalingsalterenativ vil øke kompleksiteten til systemet betraktelig. Det samme vil det at kunden har flere sykkelalternativ. Disse vil øke kompleksiteten for hele systemet. Disse to er kostnadsdrivende siden det vil ta mer innsats å utvikle et systme der en har flere alternativer enn et der det ikke er noen valg å ta.

I tillegg vil det at systemet må kommunisere med eksterne systemer øke kompleksiteten. Markasyklersystemet må kommunisere med ruter, verksteder og betalingstjeneste. Igjen vil det å estimere kompleksiteten handle om erfaring fra tidligere prosjekter være viktig.

**Oppgave 2: Arkitektur**

Innen arkitektur bruker man ofte ulike “views” for å illustrere systemet fra ulike perspektiver.

1. I forelesningen om arkitektur ble 4+1 view-modellen nevnt. Forklar kort hva de fire ulike viewene innebærer. (Lærebok kap. 6.2)

Denne innsynsmodellen er designet av Philippe Kruchten for å beskrive arkitektur i mjukvaresystemer, basert på flere samkjørende synsvinkler

1. Logisk view

I 4+1 beskriver den logiske synsvinkelen funksjonene et system vi tilby sluttbrukeren. Sekvensdiagrammer og klassediagrammer vil ofte bli brukt for å repressentere denne synsvinkelen

1. Prosessview

Prosess-synsvinkelen beskriver hvordan forskjellige prosesser av et system kommuniserer. Spennet til prosessene, skalerbarhet, effektivitet og ytelse.

1. Utviklingsview

Denne synsvinkelen er fra utviklerens ståsted. Omhandler utviklingsprosesser og programvareadministrasjon .

1. Fysisk view

Den fysiske visningen viser systemet fra en systemingeniørs synspunkt. Omhandler topologien i det fysiske laget og hvordan disse samarbeider.

1. Hvorfor er det nyttig å benytte seg av ulike views for å beskrive arkitekturen for et system?

Det er forskjellige måter å forstå systemet på ut i fra hva man skal gjøre i systemet.

Uansett hvilket synsvinkel man har så er målet å få et velfungerende system. Noe informasjon som er relevant for en ingeniør er ikke like relevant for en økonom. Prosessene i de forskjellige synsvinklene vil være relevant for rett person og på den måten får alle den informasjonen som er relevant for sin egen oppgave.

1. Gi et eksempel på hvordan systemet for markasykler kan settes opp som et:
2. trelags logisk arkitektur

[NINA]

1. firelags fysisk arkitektur

[NINA]

1. Hva er fordelene ved å benytte seg av lagdelt arkitektur?
2. Kan det være en ulempe å benytte seg av lagdeling? Begrunn svaret.
3. Redegjør for de viktigste karakteristikkene for de arkitektoniske stilene:
4. Klient-server

In a client–server architecture, the functionality of the system is organized into services, with each service delivered from a separate server. Clients are users of these services and access servers to make use of them

1. MVC – Modell View Controller

Description Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model. See Figure 6.3.

1. SOA
2. Pipe and filter

*This is a model of the run-time organization of a system where functional transformations process their inputs and produce outputs. Data flows from one to another and is trans- formed as it moves through the sequence.*

**Oppgave 3: Aktivitets-og tilstandsdiagram**

1. Hva er karakteristisk for et tilstandsdiagram, og hvorfor kan det være nyttig å benytte et slikt diagram? Begrunn svaret og kom med et eksempel på når det lønner seg.
2. Hva er karakteristisk for et aktivitetsdiagram, og hvorfor kan det være nyttig å benytte et slikt diagram? Begrunn svaret og kom med et eksempel på når det lønner seg.

Assda

Systemet for markasykler skal som nevnt tidligere benytte seg av Ruters betalingsautomater. Her har man mulighet til å fylle opp et reisekort med penger

eller en billett, og man kan også kjøpe enkeltbilletter uten reisekort.

1. Modellér et tilstandsdiagram for betalingsautomaten.
2. Modellér et aktivitetsdiagram for betalingsautomaten.

**Oppgave 4: Testing**

1. Forklar hva de ulike testfasene innebærer, og få med hva som skiller dem fra hverandre:
2. Enhetstesting

Enhetstesting er det det å kunne teste en spesifikk funksjon for seg. Man kan ha forhåndsbestemte kriterier for hva vellykkethet er.

1. Integrasjonstesting
2. Systemtesting
3. Akseptansetesting
4. Gi forslag på hvilke deler av systemet for markasykler som kan testes i hver av de ulike testfasene nevnt over. Sørg for at dere får med minst ett eksempel for hver av dem.