

Institutt for datateknikk og informasjonsvitenskap (IDI)

**Eksamensoppgave i TDT4140 - Programvareutvikling**

**Faglig kontakt under eksamen:**

Stipendiat Mohsen Anvaari, tlf.: 405 65 642

Faglærer Carl-Fredrik Sørensen kan kontaktes på tlf.: 951 19 690

**Eksamensdato: 21.05.2015**

**Eksamenstid (fra-til): 09:00 – 13.00**

**Hjelpemiddelkode/Tillatte hjelpemidler:** Kode C: Pensumbok: Ian Sommerville, Software Engineering 9. Bestemt, enkel kalkulator tillatt.

**Annen informasjon:** Eksamen utviklet av Faglærer Carl-Fredrik Sørensen og kontrollert av Per Håkon Meland og Jørgen Austvik

**Målform/språk: bokmål**

**Antall sider: 13**

**Antall sider vedlegg: 3**

**Kontrollert av:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dato Sign

Dokumenter og begrunn eventuelle forutsetninger eller antakelser.

# Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på den ene selv).

Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig.

For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller mer enn ett kryss gir

−1/2 poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne

oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1. **Hva er essensen i programvareutvikling (software engineering)?**
	1. Kravdefinisjon, designrepresentasjon, samle kunnskap og kvalitetsfaktorer
	2. Vedlikeholde konfigurasjoner, organisere team, kanalisere kreativitet og planlegge ressursbruk
	3. Tid/rom avveininger, optimalisere prosess, minimalisere kommunikasjon og dekomponering av problem
	4. **Håndtere kompleksitet, håndtere personellressurser, håndtere tid og penger, og produsere nyttige produkter**
2. **Hvilken faktor er mest utfordrende i forhold til ‘Software Crisis’?**
	1. Utviklingsomgivelsene
	2. Manglende kundeinvolvering
	3. **Upresise og ukomplette krav**
	4. Dårlig testing
3. **Aktører for et system identifiseres lettest gjennom**
	1. Organisasjonskartlegging
	2. Stillingsbeskrivelser
	3. Navn på brukeren som i hovedsak skal benytte systemet
	4. **Rollebeskrivelser**
4. **Du skal lage systemet som håndterer pengeflyt i en bank. Hvilken metode vil du bruke?**
	1. Inkrementell
	2. Vannfallsmetoden
	3. **Gjenbruksorientert**
	4. Clean room
5. **Hva er hovedårsaken til programvarefeil (software failure):**
	1. Manglende og dårlig testing
	2. **Manglende forståelse av hva skal utvikles**
	3. Gjenbruk av komponenter
	4. Manglende sporbarhet fra krav til produkt
6. **Et tekstlig use case beskriver i hovedsak**
	1. **Alle steg i en interaksjon mellom bruker og system for å få fullført et funksjonelt mål, inkludert avvik**
	2. Alle steg som systemet må ta for å få fullført et funksjonelt mål, inkludert avvik
	3. Alle steg som brukeren må ta for å få fullført et funksjonelt mål, inkludert avvik
	4. Målsetningen med systemet
7. **En ulempe med tekstlige use case er at**
	1. De er vanskeligere å forstå enn komplekse use case-diagrammer
	2. De er vanskelig å benytte for videre modellering i f.eks. interaksjonsdiagrammer
	3. De spesifiserer ikke aksjonssekvenser
	4. **De krever mye arbeid å utvikle**
8. **Hvilke er eksempler på hhv Change Avoidance og Change Tolerance**
	1. System Validation og System Refactoring
	2. **System Prototyping og System Refactoring**
	3. System Requirements og System Framework
	4. System Prototyping og System Validation
9. **Hva er hovedoppgaven til "change control board" i konfigurasjonsstyring?**
	1. Beskrive hvilke endringer som bør gjøres med programvaren
	2. Dokumentere endringer gjort i implementeringsfasen
	3. Foreslå hvilke endringer som bør gjøres i programvaren
	4. **Bestemme om man skal gjennomføre en foreslått endring**
10. **Hva er fordelen med eksplisitt arkitektur?**
	1. Interessentkommunikasjon
	2. Systemanalyse
	3. Storskala gjenbruk
	4. **Alle er korrekte**
11. **En sekvens av versjoner for en enkelt systemkomponent kan vi kalle en:**
	1. Baseline
	2. **Codeline**
	3. Configuration Item
	4. Mainline
12. **Test case utvikles vanligvis best...**
	1. Etter man er ferdig med design
	2. Under implementering
	3. **Gjennom alle faser i prosjektet**
	4. Når man vet at produktet ikke oppfører seg riktig
13. **Hvilken av de følgende er relatert til livssyklus til et system?**
	1. Testing
	2. Portabilitet
	3. Programmering
	4. **Planlegging**
14. **Hvilke er hovedaktivitetene i spiralmodellen i programvareutvikling?**
	1. **Planning, Risk Analysis, Engineering, Customer Evaluation**
	2. Defining, Prototyping, Testing, Delivery
	3. Requirements and Testing
	4. Quick Design, Build Prototype, Evaluate Prototype, Refine Prototype
15. **Hvilke faktorer må det tas hensyn til når man planlegger innsats i programvareutvikling?**
	1. Performance, Problem, Product, Planning
	2. **People, Problem, Product, Process**
	3. People, Problem, Productivity, Performance
	4. People, Problem, Product, Portability
16. **Hvilken testmetode benyttes for å konfrontere et program med unormale situasjoner?**
	1. Recovery testing
	2. **Stress testing**
	3. Performance testing
	4. Usage testing
17. **Empiriske estimeringsmetoder er som regel basert på**
	1. Ekspertuttalelser basert på tidligere prosjekterfaringer
	2. Raffinering av den nåværende prosjektestimering
	3. **Regresjonsmodeller utledet fra historiske prosjektdata**
	4. Prøving og feiling på parametere og koeffisienter
18. **Er kravvalidering og kravanalyse distinkte kravaktiviteter?**
	1. **Ja, fordi kravvalidering behøver et komplett utkast av kravdokumentet mens kravanalysen kan benytte ukomplette krav**
	2. Ja, fordi kravvalidering kan benytte ukomplette krav mens kravanalysen behøver et komplett utkast til kravdokument
	3. Ja, fordi validering bare kan utføres med testing mens analyse kan gjøres med prototyping
	4. Nei, de er ikke distinkte aktiviteter siden de begge jobber med krav og hjelper til I utforming av dem
19. **Gå ut fra at et team utvikler et web-basert billettdistribusjonssystem. Hvilken av de følgende avgjørelser ble mest sannsynlig gjort i systemdesignet?**
	1. **Systemet vil inkludere et delsystem for brukergrensesnitt**
	2. Systemet vil følge standarder for universell utforming
	3. Systemet vil tilby brukeren on-line hjelp
	4. Systemkravene har blitt møtt og tilfredsstiller kundebehovet
20. **Lehmans lover med hensyn til evolusjonsdynamikk (program evolution dynamics) beskriver prinsippene bak systemendringer. Lehmans lover om evolusjon av store programmer og organisasjonsstabilitet (Large Program Evolution and Organizational Stability) går ut fra at programevolusjon av store systemer i stor utstrekning er uavhengig av ledelsesbeslutninger. Dette kan ha følgende konsekvens:**
	1. Strukturelle faktorer har liten effekt på evolusjonsprosesser for store systemer
	2. Å lage store endringer fremfor små endringer, er en fornuftig strategi for store systemer
	3. Store utviklingsteam er ikke altfor påvirket av overhead i kommunikasjon
	4. **Endringer i ressurser eller bemanning har stor sett en veldig liten effekt på den langsiktig evolusjonen av store systemer**

## INNLEVERING

## Svarskjema: Oppgave 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Oppgavenr** | **A** | **B** | **C** | **D** |
| 1.1 |  |  |  | X |
| 1.2 |  |  | X |  |
| 1.3 |  |  |  | X |
| 1.4 |  |  | X |  |
| 1.5 |  | X |  |  |
| 1.6 | X |  |  |  |
| 1.7 |  |  |  | X |
| 1.8 |  | X |  |  |
| 1.9 |  |  |  | X |
| 1.10 |  |  |  | X |
| 1.11 |  | X |  |  |
| 1.12 |  |  | X |  |
| 1.13 |  |  |  | X |
| 1.14 | X |  |  |  |
| 1.15 |  | X |  |  |
| 1.16 |  | X |  |  |
| 1.17 |  |  | X |  |
| 1.18 | X |  |  |  |
| 1.19 | X |  |  |  |
| 1.20 |  |  |  | X |

# Oppgave 2 – Programvareutvikling (50%)

*Det skal utvikles et elektronisk sporingssystem for en produksjonsbedrift av mat for å imøtekomme krav til høyere kvalitet, bedre overvåking og større gjennomsiktighet i produksjonen. Vedlegg A beskriver en del egenskaper ved løsningen og presenterer et tilstandsdiagram som inneholder alle hovedtilstander og overganger mellom disse fra man mottar råstoff til man sender ut ferdige produkter til kjøpere i f.eks. detaljvarehandel eller restauranter. Anta at alle sensorer og avlesere av RFID har veldefinerte grensesnitt.*

1. (5%) Spesifiser og analyser interessenter og aktører av løsningen.

Interessenter til løsningen vil være relativt mange:

* Produksjonsansvarlig (overvåker produksjonsprosess)
* Logistikk/lageransvarlige (håndtere mottak og sending av varer, samt uttak fra lager og og overvåking av lagerbetingelser (temperatur etc.), kan også holde kontroll på datomerking ifht siste lovlige forbruksdato).
* Produksjonspersonell (produserer og overvåker produksjonen, finne årsak til alarmer)
* Kundeansvarlig (motta klager/informasjon om hendelser)
* Forbruker (har kjøpt en vare som ikke holder riktig kvalitet eller som har konsumert et produkt som er helseskadelig)
* Rengjøringspersonell (ansvarlig for å holde produksjonsutstyr og lokaler i henhold til standarder for renhold)
* Helsevesen og -myndigheter (diagnosering av skade og årsak til denne)
* Mattilsynet (ansvarlig for tilsyn av produksjon og matvarekvalitet)
* Bedrift og merkevareeier (tapt omdømme, tapt salg, mulig kassering av matvarer for store verdier)

Aktører (brukere og andre som har et forhold til løsningen):

* Produksjonovervåker
* Lagerovervåker
* Lagerarbeider: Mottar, stuer, plukker ut, og avsender varer
* Produksjonsarbeider: Veier, Blander, Splitter, behandler, emballerer
* RFID-ansvarlig: Ansvarlig for å holde alle sporbare enheter RFID-merket
* Rengjører
* Sporingsansvarlig

I tillegg vil det være systemaktører:

* RFID-avlesere
* Rom/miljøovervåkningssystem
* Overvåkningssystem av produksjonsutstyr
* Oppskriftssystem
1. (5%) Lag et UML Use case-diagram for sporingssystemet.

Vi kan se bort fra miljøovervåkning i denne oppgaven siden det bes spesielt om hvilke grensesnitt som sporingssystemet skal ha. Sporingssystemet kan deles i to: En datainnsamlingsdel (all avlesning av sensorer, inkludert RFID) og en del for å spore varer både oppover i kjede og nedover i kjede. Alle avlesninger av RFID skjer automatisk ved tilstandsskifte. RFID-avlesere er knyttet til alle tilstander, normalt vil verdier avleses ved inngang til tilstand. RFID-avlesere og alle sensorer har en kjent og registrert posisjon

Sporingsansvarlig:

* Trace a traceable unit (bakover fra et ferdigprodusert produkt til hvilke varer denne består av (oppskrift) og prosesser den har gjennomgått (fullstendig produkthistorie)
* Track a traceable unit (fremover fra innkomst til bedrift og til alle produkter som inneholder.
* Create tracking report when event
* Create tracing report when event

RFID-ansvarlig:

* Koble sporbar(e) enhet(er) med en RFID
* Fjerne RFID fra sporbar enhet

RFID-avleser: (systemaktør)

* Lese av passering

Overvåkningssystem (passiv):

* Motta sensoravlesninger fra alle sensorer
* Alarmere dersom begynnende avvik (gul) eller avvik (rød)

Produksjonsmedarbeider/produksjonsansvarlig:

* Sjekke avvik
* Nullstille alarm

Det vil også være mange andre use case i en matproduksjonsbedrift, men disse vil normalt sett ikke være en del av et sporingssystem, f.eks. lagerstyring, overvåking av varer som kan være kontaminert eller utgått på dato, oppskriftssystem. Alle disse vil være kilder til en sporingsrapport, men ikke være en del av sporingssystemet som i hovedsak vil forholde seg til flyt av varer gjennom bedriften og hvordan disse varene har blitt behandlet.

1. (5%) Lag et tekstlig use case: «Lag sporingsrapport ved hendelse». Dersom det baserer seg på extends/use relasjoner med andre use case, lag også tekstlige use case for disse. Input til use caset er et sporingsnummer på et produkt som bedriften har produsert. Rapporten skal kunne være i et valgfritt format (tekst, XML, e.l). Bruk malen i vedlegg B.

**Use case name:** Lag sporingsrapport ved hendelse
**Goal:** Lag sporingsrapport for en spesifikk sporbar enhet enten registrert ved utskiping fra bedrift eller ved ankomst til bedrift>
**Primary Actor:** Sporingsansvarlig
**Stakeholders and Interests:** < Produktansvarlig, produksjonsmedarbeidere, lagermedarbeidere, kundeansvarlig, helsemyndigheter, matsikkerhetsmyndigheter>
**Precondition:** Aktør er autentisert og autorisert til å lage rapport
**Success End Condition:** Sporingsrapport er laget og kan bli sendt til og analysert av der forskjellige interessentene
**Failed End Condition:** Sporbar enhet er ikke funnet i systemet, sporbar enhet har ingen historie knyttet til seg
**Trigger:** Det er rapportert problemer med en spesifikk sporbar enhet
**Description:**

* + - 1. Bruker: Skriver inn/scanner en ID av en sporbar enhet til rapport-view
			2. View sender ID til systemet når bruker trykker OK
				1. Systemet (controller) ruter ID til grensesnitt for å hente produktinformasjon om ID
				2. Systemet returnerer produktinformasjon til bruker
			3. Bruker sjekker om produkt er et innkommende eller utgående produkt basert på informasjon og velger enten å lage en rapport, Spor bakover, eller en rapport Spor fremover, velger valgt format, og hvor denne skal lagres, og trykker på OK.
			4. Hvis Spor bakover, vil systemet finne produksjonshistorie for produktet, eller hvis Spor fremover, så vil systemet finne ID på alle produkter som har relasjon til sporbar enhet fra mottakstidspunkt til nå.
			5. Systemet lager en sporingsrapport basert på formatvalg og lagrer denne hvor brukere har valgt å gjøre det.

 **Extensions:** ID på sporbar enhet eksisterer ikke i systemet, send feilmelding

1. (5%) Lag et sekvensdiagram for use caset:”Lag sporingsrapport ved hendelse” basert på use case-beskrivelsen i 2c).

I denne oppgaven vil det være naturlig å se et MVC-mønster der V kommuniserer med en C og som går videre til en M-grensesnitt. Det forventes ikke at det vil være mange nivåer under dette. Normalt vil det ligge en DB i bunnen som kan motta queries basert på hvilke relasjoner en ID har til andre.

1. (5%) Definer og grunngi de viktigste ikke-funksjonelle kvalitetskrav/arkitekturkrav til løsningen.

Usability: Systemet skal tilby en brukervennlig løsning der operasjoner skal være enkel å lære og operere. Mye av systemet vil være automatisert gjennom bruk av RFID og annen sensorteknologi for å samle informasjon gjennom produksjonskjeden. Det kan imidlertid være hendelser som oppdages i produksjonen som medfører at andre enn sporingsansvarlig vil benytte løsningen dersom det oppdages avvik i produksjonslokalet eller på lager.

Portability: Håndscannere og håndholdte enheter vil være viktig, spesielt ved oppsplitting av en sporbar enhet i visse produksjonssteg.

Reliability: Totalsystemet for bedriften for overvåkning av matvaretryggheten vil være kritisk. Manglende avlesninger og feil på sensorer må identifiseres raskt slik at enheter som ikke fungerer kan bli erstattet. Selve sporingsløsningen må ha presise registreringer av hendelser, både i tid og rom. Systemet må kunne finne avvik på sporbare enheter som forsvinner fra systemet uten en avlest «check-out» eller som mangler en «check-in».

Scaleability: Systemet må skalere i forhold til målinger fra et stort antall sensorer og kunne håndtere avlesning av mange RFID samtidig.

Performance: I store matvarebedrifter kan et råstoff spres over mange produkter eller inngå i mange produkter. Dette kan omfatte tunge spørringer mot DB-systemet. En komplett sporingsrapport skal kunne genereres i løpet av 10 minutter. Produktinformasjon om et spesifikt produkt (kjent ID) skal kunne finnes i løpet av 3 sekunder.

Maintainability: Systemet består både av programvare og maskinvare. Maskinvare skal kunne erstattes uten at dette påvirker driften av systemet.

Safety: Feil i sensorer kan medføre at målinger av miljø blir feil. Sensorer skal dupliseres og synkroniseres, og operatører skal alarmeres dersom feil eller avvik blir oppdaget. Alarmer som gis i produksjonslokale, men som blir oversett, skal etter fem minutter eskaleres til kontrollrom/produksjonsansvarlig. Safety-aspektet er her knyttet til sluttprodukt og/eller overvåking av produksjonslokale eller –utstyr.

Sikkerhet: Sensormålinger og RFID-avlesninger skal ikke eksponeres utenfor bedriften. Bruk av informasjon skal kun være knyttet til autoriserte personer og prosesser.

Det kan også være andre attributter som kan være relevante.

1. (10%) Lag en arkitekturskisse for sporingssystemet. Hvilke hovedkomponenter og koblinger/grensesnitt må tilbys?

Sporingssystemet vil i sin enkleste form bestå av tre hovedkomponenter: GUI (V), controller (C) og en sporingskomponent (M). Sporingskomponenten vil være delt inn i to deler, et forretningslag og et datalag (DB). Andre relevante komponenter som kan være knyttet til forretningslaget, kan være tjenester fra andre systemer som gir informasjon om de sporbare enhetene knyttet til innhold (lagersystem, oppskriftssystem), miljø (sensorsystem), prosess, personell, etc. knyttet til den sporbare enheten (tid, sted, hendelse).

Grensesnitt og koblinger kan ved gode tekstlige use case/sekvensdiagram, hentes direkte derfra. Det forventes ikke at alle typer systemer som en slik bedrift kan ha, skal være med i denne løsningen.

1. (5%) Hvilken eller hvilke arkitekturmønstre (patterns) bør et slikt system benytte? Beskriv hvordan og hvorfor et valgt mønster bør benyttes.

Denne løsningen vil trolig benytte et av disse arkitekturmønstrene:

* Layered architecture – Klient/tjener: Ved å skille presentasjon fra modell gjør det systemet mer portabelt (forskjellige GUI og plattformer).
* Repository: Det vil være hensiktsmessig å benytte et felles repository for alle hendelser knyttet til sporbare enheter. Det vil ivareta historikk og også tilby registrering av RFID-hendelser. Sensorsystemet vil være uavhengig av sporbarhetssystemet, men må kunne motta queries knyttet til tid og sted.
* SOA: Vil være nyttig for å kunne knytte sammen informasjon fra de forskjellige kildene: Oppskrift, sensorsystem, sporingssystem, personalplan, lagersystem, regelsystem for lagring og bruk av ingredienser etc.
1. (5%) Definer de viktigste klassene og lag et UML klassediagram for sporingssystemet. Inkluder de viktigste metodene på formen <response> metodeNavn(<parametere>). Argumenter hvorfor disse er de viktigste metodene.

Viktige klasser:

TraceableItem – kan ha flere subklasser (FoodItem, TransportItem). 1:N relasjon med seg selv. Hovedattributt: ID

TracingEvent – kan ha flere subklasser (QuantityEvent, ReadEvent, etc.)

Attributter: Location, Time. Relasjon 1:N med TraceableItem

1. (5%) Myndighetene ønsker å etablere en nasjonal sporingsløsning for å sikre rask adgang til informasjon om flyt av varer i tilfelle det skjer en hendelse der visse matvarer eller ingredienser er mistenkt. Anta at alle bedrifter har kjøpt sporingssystemet som du har utviklet. Hvordan kan man best etablere en infrastruktur for å gjøre søk både oppover og nedover i næringskjeden? Beskriv gjerne med eksempler. Hva vil kreves av bedriftene av grensesnitt og informasjon?

Alle identifikatorer benyttet må kunne inneholde nok informasjon til å finne produsent og lokasjon på produsent. Identifikator kan inneholde en sporingslenke til bedrift. Bedrifter kan basert på sporingslenke, gi informasjon til spørrende aktør om produktinnhold, næringsinnhold, produkthistorie hvis relevant. Ved et sporingsbehov, kan bedriftene tilby en tjeneste som gir sporingsinformasjon til registrerte innkommende produkter benyttet i produksjon av gjeldende produkt, samt kunne gi informasjon om hvor produktet som inngår i samme parti er sendt eller solgt. Det vil da være mulig å etablere en sporingskjede både oppover og nedover i matkjeden, fra bord til jord/fjord.

Bedriftene må derfor tilby et standardisert grensesnitt som både kan knyttes til innkommende og utgående produkter.

# Oppgave 3: Risikostyring og prosjektplanlegging (25%)

1. (10%) Beskriv risikoer og strategier for å håndtere disse i utvikling av sporingssystemet beskrevet i oppgave 2. Strategien skal sikre påliteligheten (dependability) til systemet for å unngå at en ulykke eller hendelse skjer når systemet er i operasjon. Beskriv hver strategi med passende eksempler og diskuter potensielle problemer som er assosiert med hver strategi:

Strategier: Skadebegrensning (damage limitation), Oppdagelse og fjerning av risiko, Risikounngåelse

Risikoer:

* Endrede krav/lite kravforståelse (damage limitation/risk avoidance – f.eks. smidig utvikling og mange møter med kunde knyttet til produksjonslokale, prosesser etc.)
* Stor kompleksitet med mange systemer, både programvare og maskinvare – Inkrementell integrasjon av nye delsystemer (damage limitation/monitoring risks)
* Pålitelighet (kontinuerlig testing og potensielt ha flere parallelle systemer)
* Manglende standarder eller standardsystemer – Bruke tilgjengelige standarder når de finnes (risk avoidance) og utarbeide forslag til standarder der de mangler (damage limitation)
* Integrasjon med gamle systemer – Lage Proxy-tjenester/innkapsle løsningene. Kopiere data fra gamle systemer til det nye ved gitte tidsintervall
1. (15%)
	1. Definer en grov prosjektplan for utvikling av sporingssystemet. I prosjektplanen må det defineres hovedaktiviteter, milepæler og avhengigheter.

Hovedaktiviteter: Forstudie, kravspesifisering, arkitekturdesign, design og implementering, verifisering og validering. Disse kan gå i flere iterasjoner.

* 1. Definer en WBS (work breakdown structure) for utvikling av sporingssystemet

Prosjektplanlegging – Lag plan, lag risikoplan, oppdater plan

Kravspesifisering – Kundemøter, søk etter standarder, finne informasjon knyttet til gamle systemer, intervjuer, presentasjon av mock-ups, teknologievaluering av eventuell standardteknologi (for eksempel RFID og systemer knyttet håndtering av RFID hendelser)

Arkitekturdesign – Lag skisse til arkitektur, lage UML/archimate modell, kundemøter, oppdater arkitektur.

Design og implementering – Lage UML design (sekvensdiagrammer/klassediagrammer etc.), lage DB design (ER), lage GUI design (mock-up skisser), implementere design. Dette kan gjøres inkrementelt.

* 1. Skisser hvordan prosjektet eventuelt kan deles inn i delprosjekter.

Prosjektet kan deles inn i delprosjekter, spesielt knyttet til etablering av sensorsystem og sporingsystem. Det kan også være et delprosjekt knyttet til integrasjon av andre IT-løsninger som skal benyttes

* 1. Argumenter for hvilken utviklingsprosess som vil være mest hensiktsmessig.
	2. Beregn kostnader relatert til utvikling og test av løsningen, se bort i fra utstyrskostnader, anta en fast timepris på NOK 800.

Et slikt system vil kunne ha relativt store kostnader, spesielt dersom det er mange gamle systemer som må integreres, eller dersom det finnes få eksisterende datakilder. Selve sporingsdatabasen vil antakelig være relativt enkel, men «housekeeping» kan være kompleks. Det vil derfor være viktig å kunne hente informasjon knyttet til sporbare enheter fra leverandører for å oppdatere lagersystemet og at lagerbeholdning kan oppdateres automatisk ved uttak.

Det vil være naturlig å ta utgangspunkt i WBS-en, antall delprosjekt og avhengighet mellom disse.

# Vedlegg A – Matsporing

Figuren nedenfor viser et generelt tilstandsdiagram av matproduksjon i en produksjonsbedrift på en enkel lokasjon. Bedriften mottar råstoff fra andre bedrifter, behandler, blander og videreforedler disse gjennom forskjellige produksjonsprosesser før produktene blir pakket og utskipet til butikker og restauranter. For å øke matvaresikkerheten ved bedriften, ønsker bedriften å etablere en maskinvare og programvare infrastruktur som kan overvåke produksjonsmiljøet og lagringsmiljøet. Trusler mot matvaresikkerheten er f.eks. kontaminering (forgiftning) i innkommende råstoff, dårlig varme- eller kuldebehandling internt i bedriften, dårlig renhold, og bruk av råstoffer som har gått ut på dato eller som har blitt dårlige pga. f.eks. for høy lagringstemperatur.

For å oppnå bedre sporbarhet når det opptrer en hendelse (f.eks. utbrudd av sykdom knyttet til matkonsum), ønsker man derfor å samle inn informasjon om miljøstatus både ved mottak, lagring, produksjon, pakking og utsending. Datainnsamling kan skje i alle overganger mellom tilstander i produksjonslokalet samt når man befinner seg i en spesifikk tilstand (f.eks. lagringstemperatur). Bedriften har anskaffet moderne sporingsutstyr som RFID og annen elektronisk merking som benyttes for å identifisere alle sporbare enheter i hele produksjonsanlegget. Disse kan automatisk avleses ved alle overganger mellom tilstander. I tillegg har bedriften anskaffet elektronisk overvåking av temperatur på vann, luft og i produksjonsutstyr, luftkvalitet (fuktighet, gass f.eks. ammoniakk og støvinnhold) og vannkvalitet. Innsamling av data skal kunne benyttes til forskjellige formål:

1. Beregne holdbarhetsdato
2. Beregne forbruk av råstoffer for å minimalisere svinn
3. Etablere en sporbarhetskjede fra en vare blir transportert/mottatt, gjennom produksjon og foredling til transport ut.
4. Etablere en tjeneste der et sporingsnummer på et produkt kan benyttes til å finne produksjonshistorien til produktet, inkludert miljødata og hvilke råstoffer som ble benyttet i produksjonen.
5. Finne ut hvordan en spesifikk råvare har blitt benyttet og hvilke sluttprodukter som inneholder denne. Husk at alle råvarer også er merket med informasjon og at det eksisterer et sporingsnummer for hver sporbare enhet (f.eks. en sekk med krydder).

Noen tommelfingerregler for beregning av holdbarhet er:

1. Produkter som inngår i en kjølekjede skal ikke utsettes for høyere temperaturer over en lengre tidsperiode før holdbarhetstiden reduseres. Eks. høyere enn 4°C mer enn 10 minutter
2. Produkter som skal varmes opp, må ha en oppvarming over en gitt temperatur over en viss tid. Eks. høyere enn 70°C i lengre enn 10 minutter.
3. Ved nedkjøling av produkt, så må kjernetemperatur være lik overflate- og måltemperatur etter maksimum 30 minutter.
4. Lang tids eksponering mot sollys og/eller temperaturer mellom 5°C-40°C kan medføre høy bakterievekst og gjøre maten uspiselig eller giftig.

Bedriften ønsker å etablere alarmer knyttet til produksjonen slik at avvik på temperatur og luftfuktighet, samt uttak av utdaterte produkter fra lager inn i produksjon, medfører at det kommuniseres en alarm til både produksjonsmedarbeider og til produksjonsansvarlig at et mulig avvik er under utvikling. Alarmen skal ha tre faser: Grønn – Ingen avvik, gul – begynnende avvik er registrert, og rød – avvik er registrert. Alarmtilstanden bestemmes av kobling mellom temperaturer og tid, vann- og luftkvalitet samt avvik fra rengjøringsprosedyrer.

Hvert enkelt råstoff vil ha en del attributter knyttet til seg:

* Produktnavn, produsent (en annet bedrift eller en annen produksjonsfabrikk i samme bedrift)
* Sporingsnummer inn (kan gjelde for en hel pakkeenhet, f.eks. pall, sekk, kartong)
* Produksjonsdato/tid og produksjonsnummer
* Anbefalt lagringstemperatur og temperaturhistorie (dersom temperaturovervåking)
* Dato mottatt
* Mengdeangivelse
* Siste forbruksdato

Tilsvarende vil produkter som skal sendes ut av bedriften ha data registrert på seg:

* Produktnavn, produsent
* Sporingsnummer ut (kan gjelde for en hel pakkeenhet, f.eks. pall, sekk, kartong)
* Produksjonsdato/tid og produksjonsnummer
* Anbefalt lagringstemperatur
* Dato pakket
* Mengdeangivelse
* Siste forbruksdato
* Dato sendt

Resepter/oppskrifter benyttes for å lage nye, foredlede produkter. I oppskriftene vil det stå mengdeanvisning av forskjellige råstoffer (inkludert salt og krydder), behandlingsmåte (koking, røyking, grilling, steking, blandingsmetode).

Forskjellig utstyr vil benyttes i de forskjellige prosessene, og alle kar, kasser og andre artikler som flytter seg mellom de forskjellige prosesserings/produksjonsstedene, vil være merket. Benyttet utstyret skal rengjøres/desinfiseres etter gjeldende forskrifter og regler før og etter hver produksjonsrunde.

All informasjon som samles gjennom sensorene og gjennom avlesing av RFID og vekter, blir samlet inn i et internt produksjonssystem. Sensorene og avlesningspunkter har en fast posisjon i produksjonsprosessen og produksjonslokalet. Sensorene avleses hvert minutt, mens RFID blir avlest ved uttak og inntak fra en prosesstilstand til en annen. Ved avlesing av RFID kan det også bli avlest hvor store mengder av råstoff eller produkt som er knyttet til registreringen.



# Vedlegg B – Mal for tekstlige use case

**Use case name:** < >
**Goal:** < >
**Primary Actor:** < >
**Stakeholders and Interests:** < >
**Precondition:** < >
**Success End Condition:** < >
**Failed End Condition:** < >
**Trigger:** < >
**Description:** < >
**Extensions:** < >