

## Oppg 1.

NorFly kan oppnå disse fordelene ved å innføre et system som dette, først er effektivitet som kommer av at når alle de små selvskapene har samme system så blir det mye enklere å få oversikt over hva skom skjer. De kan da gi kundene sine en mer utvidet service og for eksempel hvis en av reisene til det ene flyselskapet blir avlyst så kan kunden få beskjed og en liste av hvilke av de andre selvskapene som tilbud liknende reise, dette kan også da skje automatisk. Med et sentralt system for NorFly og alle småselskapene dens kan det blir enklere for NorFly og arbeide. Sjekking eller billettkontroll kan være en suboptimalisering slik at man ikke sniker seg inn på flyet.

Største ulempen jeg klan tenke meg blir da at hvis det er en feil i systemet som leder til at NorFly ikke får tatt imot bestillinger eller reiser ikke blir booket, så vil da også de mindre selvskapene under NorFly også bli nødt til å slutte å ta imot kunder. I motsetning til hvis alle små selvskapene beholder sitt individuelle system hvor de da kunne ha fortsatt som normalt mens problemet blir fikset.

## oppg 2.

a) En enkel og kort definisjon av hvem som kan ses på som interessente i et systemutviklingsprosjekt kan være ansatte, felleskap, aksoner, utviklere, leverandører også videre. Med andre ord de som er involvert enten indirekte eller direkte.

b) Interesser involvert i prosjektet:

-NorFly: De vil ha en full oversikt over alle sine og sine underselskap sine reiser. Enklere og bedre service til kundene av NorFly. De ønsker også kontroll over reisene, flyene og alle passasjerer, kunne bli varslet om eventuelle problemer. Men viktigst er nok ønsket om fornøyde kunder.

-Systemutvikleren: Ønsker å skape et program som skal være nærmest mulig det NorFly har bestilt, viktig for dem at det er lite til ingen feil feil i systemet som kan lede til farlige konsekvenser. Det blir også veldig viktig å utvikle et program som er sikkert mot misbruk, som f.eks bestillinger uten betalinger. Når det kommer til informasjonen er det viktig at det systemet viser og mottar er likt og uten feil.

-Passasjerene: vil ha en enkel og detaljert oversikt over reisen sin, som flyplass de reiser fra og til, tidspunkt, priser, eventuelle forsinkelser, etc.

- Datatilsynet: Interessert i at systemet ikke tar inn mer informasjon om kunden en nødvendig.
- Ansatte på flyplassen: Er interessert i informasjon om hvem og hva som kommer igjennom flyplassen. Som for eksempel mengden bagasje, hvor de skal og hvem som eier dem.
- Piloter Trenger informasjonen om hvor de skal, når de skal dit, hvem de jobber med og hvilken fly model de skal fly.

## Oppg 3.

a) De argumentene vi har når det kommer til valget av plandreven utviklingsprosess som f. Eks fossefall er at utvikleren kan informere kunde(e) at det bli lite til ikke noe kommunikasjon under prosessen, sånn at kundene skal komme på alt de vil at systemet/programmet skal kunne gjøre i motsetning til å gradvis komme med endringer til utviklerne, dette gjør da at oppgaven kan gjøres raskere.

b) Hoved fordelen som kan oppnås ved bruken av scrum er å ha en mye høyere mulighet for å gi kunden det systemet de ønsker, Siden det ikke er bare samarbeid mellom utviklerne og kunden(e) under underskrivelsen av kontrakten, men også under skapelsen av programmet. Det vil si at partene har kontakt ofte under prosessen. Det kan være noen ulemper av at kunden og utvikleren har mye kontakt, som f. Eks store endringer på systemet under utviklingen.

c)

1. Som passasjer ønsker jeg sikker betaling på terminal.
2. Som kunde ønsker jeg å ha mulighet til og bestille flybilletter.
3. Som NorFly ansatt ønsker jeg å vite flyrute informasjon.
4. Som kunde ønsker jeg å kunne se på rutetabeller.
5. Som prosjektleder så ønsker jeg å ha en oversikt over bookingsystemet.
6. Som kundeservice så ønsker jeg å kunne gi så mye informasjon til kunden som mulig.
7. Som ansatt hos NordFly ønsker jeg at kunden skal gi tilbakemelding.
8. Som systemutvikler ønsker jeg å møte mesteparten av kravene til Nordfly.
9. Som passasjer ønsker jeg muligheten til å gi tilbakemelding til NordFly.
10. Som leder av prosjektet ønsker jeg at utviklerene skal ha det enklest mulig å jobbe.

d) I løpet av den første sprint backloggen så vil jeg ha tatt 1, 2, 4, 8 og 10 ettersom det er kunden som bruker systemet oftest.

4a)

## 1) Introduksjon:

NordFly har bestilt et sentralt bookingsystem som har muligheten til kommunikasjon mellom alle de mindre selskapene de eier. Som kan gi kunden muligheten til å sjekke ledige billetter, bestille dem med enklere betaling. Skape større kundeservice muligheter. De ønsker at dette systemet skal kunne utvide markedsandelen dem. Prosjektet har et budsjett på 2-2.5 millioner kroner og vil ta mellom 2.5 og 5 mnd.

## 2) Organisasjon:

- Person1 er prosjektsleder, som tar seg av planlegging og bestemmer hvilket lag som gjør jobber på de spesifikke delene av systemet.
- Lag 1(person2 leder laget, person3, person4, person 5) jobber med tabellsystem, billetter og ruter.
- Lag 2 (Person6 leder laget, person7, person8, person9) arbeider med kommunikasjon mellom selskapene og med feilsøking.
- Lag 3 (Person10 leder, person 11-13,) tar seg av lagring, backup, servere og kravene til kunder.
- Lag 4 (Person14 og person2) lager dokumentasjon over alt de har gjort og setter sammen de andre lagenes arbeid til et system.

## 3) Risikoanalyse:

- Teknologi:
  - Databasene som brukes i systemet klarer ikke å håndtere stor mengde transaksjoner.
  - Kan være feil i gjenbrukte komponenter som leder til at det nye systemet ikke fungerer som det skal.
  - Systemet mangler programmeringskoder.
- Mennesker:
  - Opplæringer.
  - Mulige vanskeligheter å ansette personer med kompetansen som trengs for å fullføre oppgaven.
  - Sykdom eller fravær av personell under behov.
- Organisasjonen
  - finans problemer
- verktøy:
  - Programmer samarbeider ikke.
  - Ulike programmer.
- Krav.
  - Endringer forårsaker om designing av systemet.
  - Kunden skjønner ikke endringene utvikleren har gjort på kravet.
- Estimeringer:
  - Koden sin lengde og størrelse.
  - System endringer.

## 4) Hardware og software krav:

- Lagene kan blandt annet velge selv hvilke programmer de programmerer på. Evt, ta i bruk tidligere bookingsystemer får å endre firmware til oppdateringer.

5) **Work breakdown.**

- Planlegging (A): planlegge hvor mange oppgaver oppgaven skal deles opp i, hvor lang tid prosjektet skal ta. Tid: 1 dag.
- Plan (B) (milepæl) (checkpoint 1): klar gjøre programmering planen. Tid: 1dag.
- Programmering (C) = skaper programmer til systemet og passer på at det møter kravene NordFly har satt. Tid: 5 dager.
- Test 1 (D), test 2 (E) og test 3(F): Test 1 handler om å finne ut om systemet fungerer eller hvordan det fungerer i praksis. Test 2 handler om og sjekke stabiliteten til systemet. Test 3 er basert på å oppdatere systemet. Tid: test 1: 1dag, test 2: 3 dager og test 3: 1dag.
- Feilsøking (G) og (H): sjekker om om test 1 og test 2 om det er noen feil/problemer i koden og eventuelt retter disse opp. (H) utgår på det samme, men for seg selv. Dette gjøres fordi begge skal ha egne rapporteringer slik at den nye oppdateringen ikke leder til feil når det nye systemet skal brukes. Tid (G): 2 dager, (H): 3 dager.
- Programmering og testing (I) (milepæl) (checkpoint 2): Utføre en siste sjekk etter å ha samlet opp alle rapportene. Tid: 1 dag.
- Undersøke om systemet står opp kravene til NordFly. Tid: 2 dager. (J).
- Checkpoint 3: klargjøre systemet for å legges inn på server. (K) tid: 1 dag.

**6) Prosjekt planlegging**  
milepæle

	Aktivitet	Tidsforbruk (dag(er))	Avhengighet
A	Planlegning	1	Ingen
B	Checkpoint 1 (plan)	1	A
C	Programmering og kravsjekk	5	B
D	Test 1	2	C
E	Test 2	3	C
F	Test 3	1	C
G	Feilsøking 1 og retting	2	D, E
H	Feilsøking 2 og retting	3	F
I	Checkpoint 2	1	G, H
J	Krav test	2	I
K	Checkpoint 3	1	J

**7) Overvåkning og rapportering:**

- Prosjektet blir overvåket og dokumentert alt som blir gjort med systemet som f. Eks endringer, eventuelle feil, tidsbruket, feilsøking som da er til senere bruk hvis NordFly vil utvide systemet i et senere tidspunkt.

## 4b) Matrise i vedlegg 1.

5 a)  
milepæle

	Aktivitet	Tidsforbruk (dag(er))	Avhengighet
A	Planlegning	1	Ingen
B	Checkpoint 1 (plan)	1	A
C	Programmering og kravsjekk	5	B
D	Test 1	2	C
E	Test 2	3	C
F	Test 3	1	C
G	Feilsøking 1 og retting	2	D, E
H	Feilsøking 2 og retting	3	F
I	Checkpoint 2	1	G, H
J	Krav test	2	I
K	Checkpoint 3	1	J

Diagram vedlegg 2.

5b) Diagram i vedlegg 3.

5c) Fra start til A til B til C til E til G til I til J til K, også i diagrammet med ekstra tykk linje. Dette blir sett på som den mest kritiske veien siden den krever mest tid.

6a)

- funksjonelle krav
  - Sjekke ledigheten av billetter.
  - Gi kunden muligheten til å printe ut billetten sin.
  - Kunne se rutetid og rutetabell.
  - Booke og bestille over flyselskapene.
  - Kundeservice med større dekning.
  - Utvide kommunikasjonen mellom alle selskapene.

6b) ikke-funksjonelle krav til det nye systemet kan evalueres under en testfase der produktskravene, organisatoriske kravene og de eksterne kravene blir testet på forskjellige måter. Man kan teste produktskravene ved å motarbeide kravene som blir stilt. Som for eksempel å kjøre systemet med stort antall brukere. Testing av organisatoriske krav kan gjøres ved å ha et konstant syn over budsjettet, bruke en kode som det eventuelt er lett å gjøre endringer på. Testing av de eksterne kravene utføres av en advokat som passer på at systemet følger lovene.

ikke-funksjonelle krav:

- produkt krav:
  1. Må være enkel å bruke med menyer som er oversiktlige og hjelp når det trengs
  2. Datasikkerhet som brukernavn og passord beskyttet.
  3. Må kunne håndtere stort antall brukere samtidig.
  4. Har en rask responstid på mindre enn 1 sekund.
- Organisasjon krav:
  1. pålitelighet og lett å vedlikeholde
  2. At systemet blir levert innen tidsfristen.
  3. Holde seg til budsjettet.
  4. Leveres innen frist.
- Eksterne krav:
  - følger lover satt av datatilsynet.
  - Beskytter brukerinfo.

## Vedlegg

Usikkerhets- element #	Beskrivelse av usikkerhetselement	Kundens vurdering		Leverandørens vurdering		Tiltak	Ansvarlig
		Sannsyn- lighet	Konsekvens	Sannsynlighe- t	Konsekvens		
1	Sykdom eller fravær av personell under behov.	lav	tolererbar	middels	katastrofal	Ansette vikar	Person1
2	Programmer samarbeider ikke.	Lav	moderat	Lav	katastrofalt	Omprogrammere eller finne programmer som vil	Person14
3	Bugs, kræsje og hacking	høy	Katastrofal	Middels	Katastrofal	Omprogrammere eller slette deler av koden	Person14
4	Ansette personell med rett kompetanse	Lav	tolererbar	Høy	Katastrofal	Lete etter en som har.	Person1
5	Underestimere tidebruk	Høy	katastrofal	Høy	Katastrofal	Forsøke å teste raskere	Person14
6	Kommunikasjonen melleom selskapene	Middels	Tolererbar	Lav	Tolererbar	Sjekke tilkobling, putte på annen server, gjøre manuelt eller feilsøke	Person6
7	Strøm problemer	Lav	tolererbar	Høy	katastrofal	Backuping	Person10
8	Finanseproblemer	Lav	Katastrofal	Lav	Katastrofal	Ta lån	Person1

Vedlegg

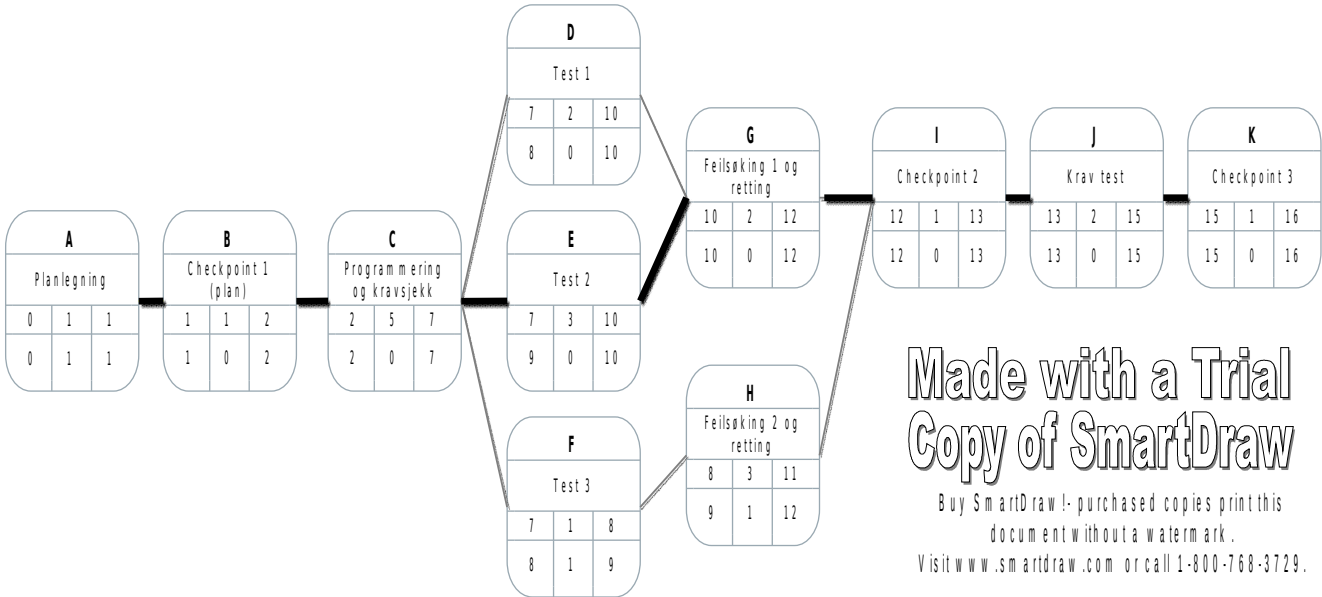
Vedlegg 2.  
Stolpediagram til oppg 5a.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A																				
B		A																		
C			B																	
D								C												
E								C												
F								C												
G										D, E										
H								F												
I												G, H								
J													I							
K																J				



Vedlegg

Vedlegg 3.  
PERT-diagram til opppg 5b.



Made with a Trial  
Copy of SmartDraw

Buy SmartDraw! - purchased copies print this document without a watermark.  
Visit [www.smartdraw.com](http://www.smartdraw.com) or call 1-800-768-3729.