

# BACHELOR OPPGAVE

## Automatisering av linuxtjenere i Nasjonalbiblioteket - Forprosjekt

---

Utarbeidet av:

Henning Fjellheim

Studium:

Bachelorstudium i informasjonssystemer

Innlevert:

13.03.2015



# Innholdsfortegnelse

## Innhold

|   |   |
|---|---|
| 1. Innledning.....  | 1 |
| 1.1 Bakgrunnen for prosjektet. ....                                 | 1 |
| 1.2 Problemstilling.....  | 1 |
| 1.3 Beskrivelse av Nasjonalbiblioteket.....                         | 1 |
| 1.4 Prosjekt mål.....   | 2 |
| 1.4 Oppgavebeskrivelse.....   | 2 |
| 1.5 Enkel beskrivelse av Puppet / CFEngine.....                     | 3 |
| 2. Prosjektmetodikk.....  | 5 |
| 2.1 Valg av prosjektmetodikk.....                                   | 5 |
| 2.2 testing av installasjon.....                                    | 5 |
| 3. Prosjektorganisering.....  | 6 |
| 3.1 Oppdragsgiver.....  | 6 |
| 3.2 Prosjektleder.....  | 6 |
| 3.3 Styringsgruppe.....   | 6 |
| 3.4 Veiledere.....  | 6 |
| 3.5 Interessenter.....  | 6 |
| 4. Kost nytteanalyse.....   | 7 |
| 4.1 Kost nytteanalyse.....  | 7 |
| 5. Prosjektgjennomføring.....                                       | 8 |
| 5.1 prosjektets avgrensninger.....                                  | 8 |
| 5.2 Arbeidspakker.....  | 8 |
| AP1 Utrede.....   | 8 |
| AP2 Installasjon.....   | 8 |
| AP3 Sammenligningsstudie.....                                       | 8 |
| AP4 Utarbeide anbefaling.....                                       | 8 |
| AP5 Sluttrapport.....   | 8 |
| 5.3 Milepælsplaner:.....  | 9 |
| Milepæl 1 avdekking av omkringliggende kjøremiljø.....              | 9 |
| Milepæl 2 avdekke sammenligningskriterier.....                      | 9 |
| Milepæl 3 estimert tidsbehov for installasjon og konfigurasjon..... | 9 |
| Milepæl 4 ferdig installerte kjøremiljø.....                        | 9 |

|   |    |
|---|----|
| Milepæl 5 installasjon og test av Puppet og CFEngine..... | 9  |
| Milepæl 6 Ferdigstilling av anbefaling.....               | 9  |
| Milepæl 7 Ferdigskrevet sluttrapport. ....                | 9  |
| 5.4 Ansvarskart.....                                      | 10 |
| 5.5 Gantt Skjema .....                                    | 11 |
| 6 Risikoanalyse .....                                     | 12 |
| 6.1 Risikoanalyse .....                                   | 12 |
| 7. Konklusjon .....                                       | 13 |
| Appendix A: Referanseliste .....                          | 14 |
| Appendix B: gantt diagram .....                           | 15 |
| Appendix C: Teknisk ordliste .....                        | 16 |

## 1. Innledning

Dette prosjektet skal ta for seg automatisering av linuxtjenere i Nasjonalbiblioteket. Jeg vil her (i henhold til bestillingen fra Nasjonalbiblioteket) for å se hvilke gevinster automatisering av linuxtjenere kan gi. For å hjelpe meg med dette skal jeg evaluere Puppet og CFEngine (automatiseringsverktøy). Jeg skal skrive en anbefaling til Nasjonalbiblioteket basert på det jeg har funnet ut om disse verktøyene om hvilken av disse to verktøyene som passer best til Nasjonalbiblioteket, pluss en anbefaling om hvordan man burde implementere dette verktøyet. Hvis det ikke lar seg gjøre å konkludere hvilket av disse verktøyene som passer best for Nasjonalbiblioteket, så skrive en anbefaling på hvordan Nasjonalbiblioteket burde gå fram for å komme seg videre med automatisering.

### 1.1 Bakgrunnen for prosjektet.

Bakgrunnen for dette prosjektet er at jeg skal utføre en bacheloroppgave ved Høgskolen i Nesna i forbindelse med dette, så tok jeg kontakt med min arbeidsgiver med forespørsel om det var mulig å få ett bachelorprosjekt som involverte automatisering. Da dette er noe som interesserer meg i stor grad, siden jeg synes mye av tiden min blir brukt på oppgaver som føles repetitive og det kan være forskjell på hvordan forskjellige folk utfører den samme oppgaven. Det kan være vanskelig å sette opp en identisk maskin for testmiljø etc. å få på plass automatisk utrulling og patching av maskiner er noe som jeg tror vil lette arbeidsdagen min og ønsker å se på om dette er tilfellet. Siden man da i teorien skal kunne rulle ut nye systemer automatisk og ha bedre oversikt med at systemene er satt opp likt i test og produksjon.

### 1.2 Problemstilling

Nasjonalbiblioteket har en stor park av linuxtjenere som pr. i dag har flere ulike kjøremiljøer. Der det er flere miljøer som skal være uniforme som ikke er det. Ikke uniforme systemer har ulempen at de kan introdusere uventede feil. For eksempel hvis man har ulik installasjon mellom testmiljø og produksjonsmiljø.

### 1.3 Beskrivelse av Nasjonalbiblioteket.

Nasjonalbibliotekets sin organisasjon(Mandat, 2015) er forankret i lov av 9. Juni 1989 om avleveringsplikt for allment tilgjengelige dokumenter. Formålet med loven er å sikre levering av dokument med allment tilgjengelig informasjon til nasjonalbiblioteket slik at vitnesbyrd om norsk kultur og samfunnsliv blir bevart og gjort tilgjengelig som kildemateriale for forskning og dokumentasjon.

Som man kan se av det mandatet som har blitt gitt til Nasjonalbiblioteket, så får vi og behandler ganske store mengder med data. (Film, musikk, bøker, aviser etc.) vi tar vare på det som er født digitalt og det blir og digitalisert en stor mengde materiale som ikke er født digitalt. Nasjonalbiblioteket har derfor en maskinpark som inneholder over 100 linuxtjenere som gjør forskjellige jobber knyttet til det å digitalisere og å vise fram materiell som er digitalisert via blant annet tjenester som bokhylla (bokhylla, 2015) der Nasjonalbiblioteket viser fram bøker og tjenester som man har digitalisert som er offentlig tilgjengelig.

## 1.4 Prosjektmål

Målet med dette prosjektet er å finne ut hvilke gevinster og kostnader som vil komme med innføring av automatisering for linuxtjenere ved Nasjonalbiblioteket.

Det skal utarbeides en rapport som beskriver hvilken gevinster som kan realiseres ved å innføre automatisering av nevnte linuxtjenere, i rapporten vil det også bli beskrevet hvilke kostnader som vil komme med en eventuell innføring av automatisering.

Det vil bli forsøkt å avdekke risikoer ved innføring av automatisering og se på hvordan disse 2 automatiseringsverktøyene vil møte på disse risikoene.

Det skal i prosjektet utprøves 2 automatiseringsverktøy for å besvare dette. Fordeler og ulemper med det enkelte verktøy skal beskrives.

Delmål vil være å få på plass en anbefaling om innføring av verktøy for automatisering, og i så fall hvilket verktøy Nasjonalbiblioteket burde velge. Hvis det ikke lar seg gjøre å konkludere, skal det skrives en anbefaling om veien videre.

## 1.4 Oppgavebeskrivelse

Her er bestillingen fra Nasjonalbiblioteket. Skrevet av prosjektgruppen ved Sverre Bang og Marianne Drotninghaug.

Nasjonalbiblioteket har et stort antall Linux-tjenere i sin maskinpark. Det er knyttet betydelige utfordringer til installasjon og vedlikehold av disse. Noen av utfordringene som oppleves er:

- Tidkrevende installasjon
- Tjenere som skal være uniforme opptrer i ulike varianter av OS, patchnivå og andre konfigurasjoner
- Vedlikehold og patching er i stor grad uavhengig utført på hver av tjenerne og med varierende frekvens
- Ulik praksis fra person til person ved både installasjon, oppsett, vedlikehold og patching

Nasjonalbiblioteket ønsker svar på hvilke gevinster som kan realiseres ved å innføre automatisert installasjon og patching av Linuxtjenere. Et av virkemidlene for å besvare dette vil være utprøving og evaluering av verktøyene Puppet (Puppet, 2015) og CFEngine (CFEngine, 2015). Fordeler og ulemper med det enkelte verktøy skal beskrives.

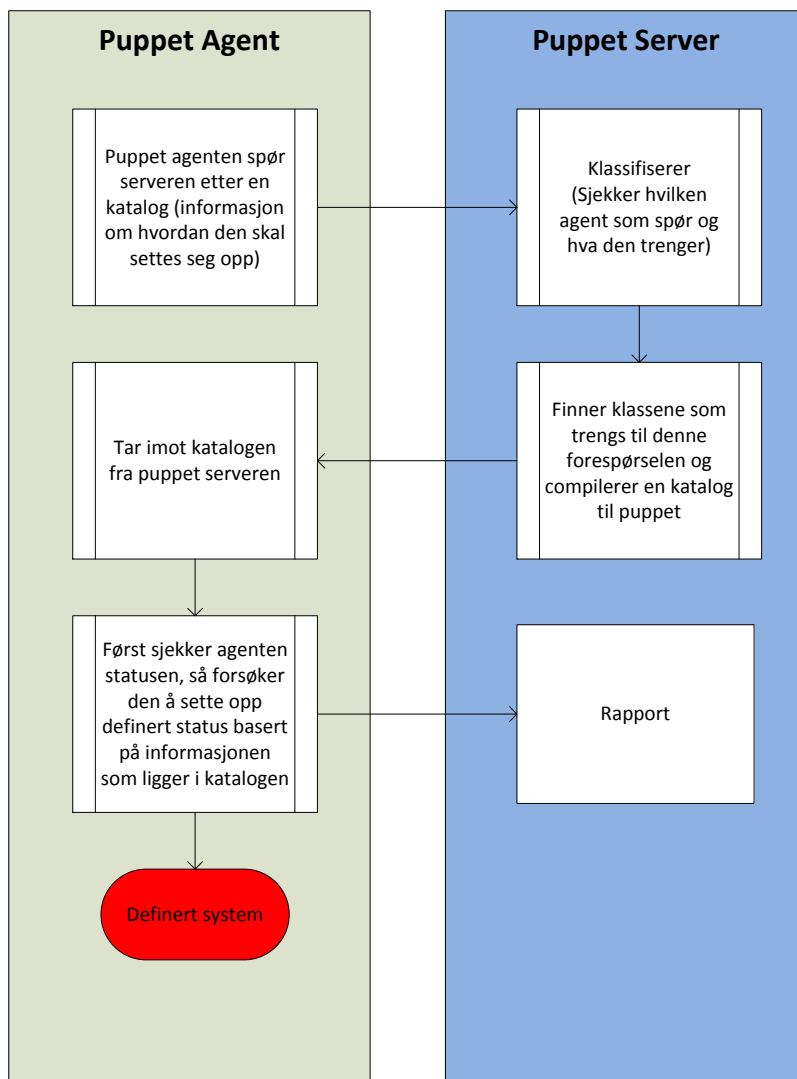
Nasjonalbiblioteket ønsker en anbefaling om innføring av verktøy for automatisering, og i så fall hvilket verktøy som bør velges. Om det i prosjektet ikke lar seg gjøre å konkludere ønskes det en anbefaling om veien videre.

## 1.5 Enkel beskrivelse av Puppet / CFEngine

Hovedarkitekturen til Puppet og CFEngine fungerer omtrent på samme måte, begge har en tjener og en klient. Tjeneren inneholder informasjon om hvordan klienten(e) skal være konfigurert. Dette er beskrevet i filer på tjeneren. Klientene spør så tjeneren etter hvordan de skal se ut, tjeneren svarer klienten med info om hvordan den skal se ut og klienten setter seg opp. Jeg tar det litt mer i detalj for hvordan Puppet og CFeninge fungerer under.

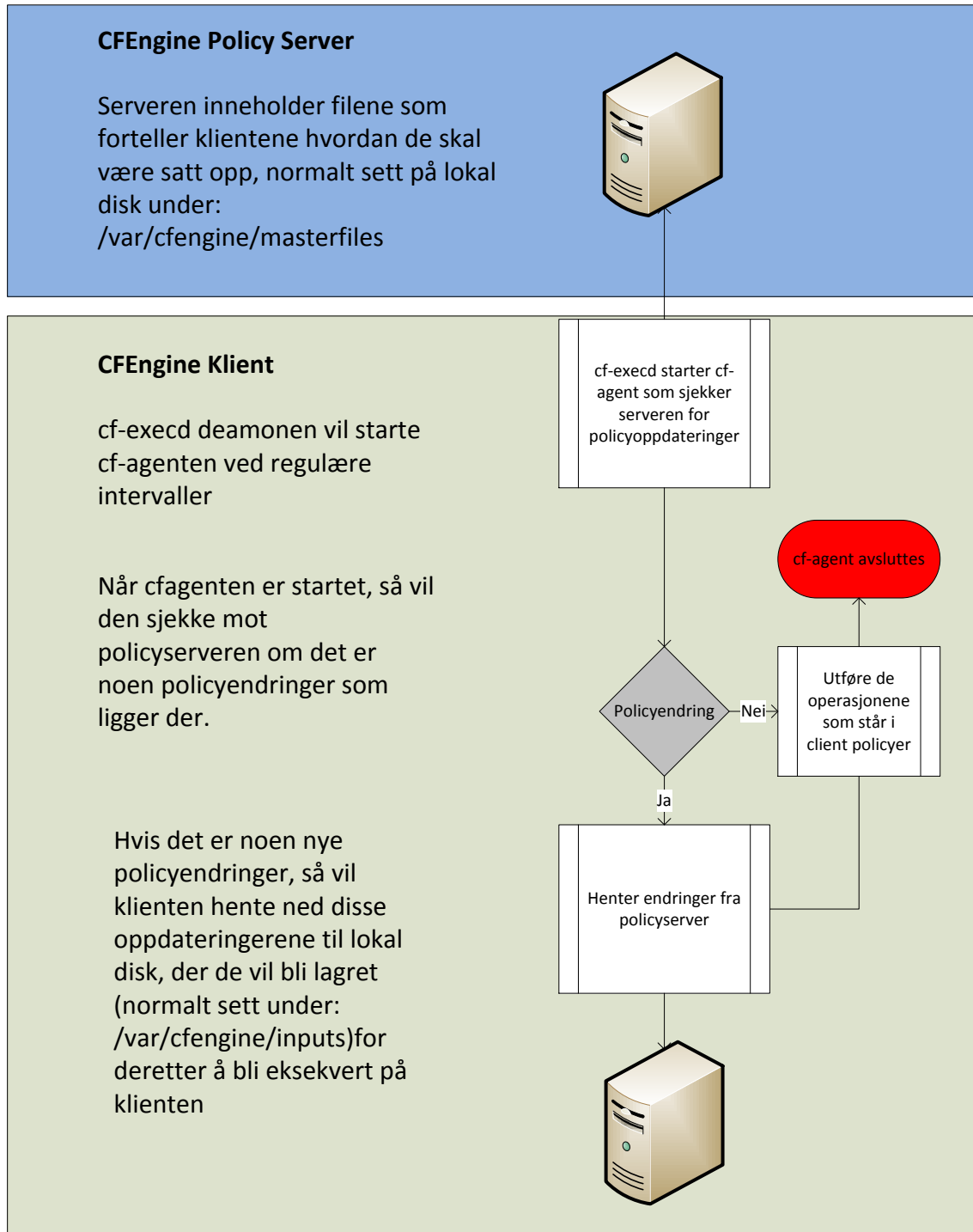
### Puppet

Det man her ser, er at klienten sender informasjon om seg selv til tjeneren (tjeneren og klienten kommuniserer via HTTPS med host-verifisert SSL) tjeneren sjekker informasjonen den får fra klienten, finner ut hvilken informasjon den har som gjelder klienten. Kompilerer denne til en katalog og sender katalogen tilbake til klienten. Klienten tar så tak i denne katalogen, den sjekker hvordan den skal se ut og forsøker å sette seg opp i forhold til hvordan den skal se ut, deretter sender den en rapport til tjeneren, før den avslutter.



## CFEngine

CFEngine bruker offentlige og private nøkler for å snakke mellom tjener og klient, den er basert på OpenSSH (OpenSSH, 2015) for å utveksle nøkler. Det som skjer når CFEngine skal kjøre en oppdatering er at cf-execd (dette er en skedulerings-agent for cf-agenten) starter cf-agenten på klienten. CF-agenten vil så sjekke tjeneren (policyserveren) om det ligger noen oppdateringer der. Hvis det ikke ligger endringer der, så vil den utføre de endringene som skal utføres lokalt og avslutte agenten på tjeneren. Hvis det ligger endringer på tjeneren, så vil tjeneren hente ned de endringene som ligger der, før tjeneren utfører de endringene som skal utføres lokalt.



## 2. Prosjektmetodikk

Beskrivelse av hvilken prosjektmetodikk som ble valgt til dette prosjektet og hvorfor denne ble valgt, dette prosjektet blir drevet fram ved hjelp av målrettet prosjektstyring.

### 2.1 Valg av prosjektmetodikk.

Til dette prosjektet har jeg valgt Målrettet Prosjektstyring (Andersen, 2004), mye grunnet at jeg er alene og denne metodikken synes jeg passer godt til den type arbeidsoppgave som jeg skal utføre. Den er som mange arbeidsmetodikker egentlig skrevet for team på mer enn en person, men hovedpunktene i metodikken mener jeg kan brukes.

Grunnen til at jeg synes denne metodikken passer bra, er at den lar meg sette milepæler som jeg skal nå. Det at man kjører delprosjekter mener jeg passer min målsetning bra, da det passer bedre for min del å kjøre ett delprosjekt med å sette opp puppet-miljøet, oppnå de milepælene jeg skal der, for deretter å sette opp CFEngine for deretter å forsøke å nå de milepælene som blir satt på det prosjektet. Her vil jeg anta at milepælene blir det samme. Det kommer til ett annet punkt som passer meg og denne oppgaven ganske bra: Det er ikke nødvendigvis så viktig i starten å definere hva man trenger og hvordan man skal sette det opp, bare hva ønsket resultat er. Deretter kan jeg finne ut mer eksakt hva jeg trenger og hvordan jeg skal sette det opp og få testet det.

### 2.2 testing av installasjon

Jeg vil benytte meg av en komponent av Test Driven Development (TDD, 2015) det jeg mener med dette, er at jeg kommer til å skrive tester for hva som skal oppnås før jeg utfører det. Dette for å forsikre meg om at jeg oppnår samme resultat når jeg tester Manuell installasjon og automatisk installasjon ved hjelp av Puppet eller CFeninge. Disse testene kommer til å bli skrevet i ett eksternt program siden Nasjonalbiblioteket allerede bruker Nagios (Nagios, 2015) for å overvåkning (Nagios er ett program for overvåkning av maskiner/tjenester). Så blir jeg å skrive testene i Nagios, hvis dette viser seg å være for tidkrevende. Så vil testene skrives manuelt i Python (Python, 2015) for å kunne bruke de samme testene på alle 3 installasjonsmetodene.

Ett interessant aspekt er jo tidsforbruk, så tidsforbruket fra jeg starter med å sette opp en maskin til alle testene er ferdig er jo ett interessant for rapporten som skal skrives når jeg skal komme med en anbefaling.



## 3. Prosjektorganisering

Oversikt over hvordan er prosjektet organisert.

### 3.1 Oppdragsgiver

Oppdragsgiver for dette prosjektet er Nasjonalbiblioteket i regi av styringsgruppen (se punkt 3.3)

### 3.2 Prosjektleder

Prosjektleder: Henning Fjellheim

Prosjektleder har ansvar for:

- Utarbeiding av prosjektplan
- Administrasjon av prosjektet
- Utarbeiding av milepælsplan
- Administrere endringer i prosjektet hvis behovet skulle oppstå
- Ansvarlig for at tidsfrister blir fulgt

### 3.3 Styringsgruppe

Styringsgruppe: Sverre Bang, Marianne Drotninghaug fra Nasjonalbiblioteket.  
Styringsgruppen har møter med prosjektleder ved behov.

Styringsgruppen har ansvar for:

- Utarbeide begrensninger for prosjektet i samråd med prosjektleder.
- Hjelpe til med utarbeiding av mål for prosjektet i samarbeid med prosjektleder
- Fremskaffe budsjettpris på Enterprise versjon av Puppet og CFEngine.

### 3.4 Veiledere.

Veiledere: Rune Bostad og Øyvind Hanssen fra Høgskolen i Nesna. Vi har avtalt å ta møter ved behov, i begynnelsen av prosjektet. Så forsøkes det å få til ukentlige møter. Dette for å få til progresjon og hjelp med utarbeiding av oppgaven.

### 3.5 Interessenter.

Her er en liste over mulige interessenter i prosjektet og en oversikt over hva de får ut av prosjektet.

Oppdragsgiver:

- Får se hvilket automatiseringsverktøy som vil være best egnet til oppgaven
- Få avdekket fordeler og ulemper ved automatisering.

IKT Drift:

- Enklere utrulling av systemer etter at automatisering er startet
- Vet at systemer er identiske.
- System som skal være identiske er identiske.

Meg:

- Bedre opplært i automatiseringsverktøy
- Bachelorutdanning

## 4. Kost nytteanalyse

Kost nytteanalyse for å se omtrentlig hvor mye prosjektet kommer til å bruke i tid og vil hjelpe til med å bedømme om det tidsmessig burde utføres.

### 4.1 Kost nytteanalyse

For å finne tidskostnadene så henviser jeg til gantt skjemaet i kapittel 5.5 hvor jeg har skrevet inn antall dager jeg kommer til å jobbe med prosjektet. Etter forprosjektet så er det beregnet at jeg kommer til å jobbe 70 dager hvis jeg da beregner 7,5 timers arbeidsdag, så blir det 525 timer. (Har utelatt forprosjektet da dette er tid som egentlig ikke inngår i prosjektet)

Nytteverdien av dette prosjektet vil være å se hvilke fordeler Nasjonalbiblioteket kan få av automatisering. Det som vil bli nytteverdien av dette prosjektet er at det vil utrede hvilket oppsett Nasjonalbiblioteket får best nytte av. Målsetningen for prosjektet er å definere hvordan Nasjonalbiblioteket best mulig kan få nytte av automatisering, fremheve fordeler, ulemper, kostnader og risiko.

Men hvis man tar ett røft estimat og sier at dette prosjektet kan spare inn 2 arbeidstimer pr. person på mitt team i uken, så vil dette utgjøre ~8 timer pr. person pr. måned, siden vi er 6 personer på mitt team. Så kan man da ta utgangspunkt i at det blir spart inn 48 timer i måneden.

Hvis dette estimatet stemmer, så vil det ta  $525/48=10,9375$  Så hvis vi da runder oppover til 11 måneder, så vil prosjektet være kostnadseffektivt i løpet av ett år.

Selv med ett enda mer forsiktig estimat, så ville prosjektet være kostnadseffektivt innen 2 år.

## 5. Prosjektgjennomføring.

I dette kapittelet skal selve prosjektplanleggingen og hva som skal gjøres komme frem.

### 5.1 prosjektets avgrensninger

Her er de avgrensningene som er satt for dette prosjektet, tatt i samråd med styringsgruppen.

Det skal sammenlignes open source versjonen av Puppet og CFEngine.

- Det skal utarbeides prisoverslag av hva kostnadene på å gå over til en support versjon av verktøyene. (Dette kunne styringsgruppen være behjelpelig med)
- Miljøet som skal settes opp, er i samråd med styringsgruppen bestemt at det skal være en maskin med standard OS. Vi legger oss i første omgang på Centos(Centos 2015) som er «gratisversjonen» av RedHat(RedHat 2015). Det skal settes opp to maskiner, en med versjon 6.6 og en med versjon 7 av Centos, disse skal meldes inn i IPA(freeIPA 2015) settes opp slik at man kan bruke sudo(sudo 2015), slå av brannvegg og SELinux(SELinux 2015) deretter installere NginX(NginX 2015), sette NginX til å starte automatisk ved oppstart for deretter å vise en enkel nettside.
- Omkringliggende infrastruktur settes opp manuelt.

### 5.2 Arbeidspakker

Jeg har valgt å dele inn prosjektet i 5 arbeidspakker. Disse er som følger.

#### AP1 Utrede

- Avdekke behov for kjøremiljø og omkringliggende infrastruktur.
- Beslutte sammenligningskriterier Puppet og CFEngine.
- Estimere tidsbehov for installasjon og konfigurasjon.

#### AP2 Installasjon

Sette opp kjøremiljøene

#### AP3 Sammenligningsstudie

Sjekke hvordan CFEngine og Puppet måler seg opp mot de kriteriene som er satt for dem.

#### AP4 Utarbeide anbefaling

Utarbeide en anbefaling om innføring av automatisering i Nasjonalbiblioteket. Basert på de kriteriene som er satt og den dataen jeg har samlet inn ved å teste ut Puppet og CFEngine.

#### AP5 Sluttrapport

Skrive sluttrapporten basert på det jeg har funnet ut om fordeler og ulemper ved innføring av automatisering i Nasjonalbiblioteket. Basert på sammenligningen jeg har gjort mellom Puppet og CFEngine.

### 5.3 Milepælsplaner:

Arbeidspakkene er delt opp i disse milepælene. Da har hver milepæl en enkel og klar målsetning.

#### Milepæler for arbeidspakke 1:

##### Milepæl 1 avdekking av omkringliggende kjøremiljø

Avdekket kjøremiljø for omkringliggende infrastruktur.

##### Milepæl 2 avdekke sammenligningskriterier

Ferdige sammenligningskriterier for Puppet og CFEngine

##### Milepæl 3 estimert tidsbehov for installasjon og konfigurasjon

Ferdig estimert tidsbehov for installasjon og konfigurasjon.

#### Milepæler for arbeidspakke 2:

##### Milepæl 4 ferdig installerte kjøremiljø

Ferdig oppsatt kjøremiljø.

#### Milepæler for arbeidspakke 3:

##### Milepæl 5 installasjon og test av Puppet og CFEngine

Dokumentasjon på oppsett og tester av Puppet og CFEngine mot de mål og sammenligningskriterier som ble utarbeidet i milepæl 2.

#### Milepæler for arbeidspakke 4:

##### Milepæl 6 Ferdigstilling av anbefaling

Ferdig anbefaling for innføring av automatisering av installasjon av linuxtjenere, ferdig anbefaling av valg av eventuelt automatiseringsverktøy.

#### Milepæler for arbeidspakke 5:

##### Milepæl 7 Ferdigskrevet sluttrapport.

Ferdigskrevet sluttrapport.

## 5.4 Ansvarskart

Oversikt over hvem som har ansvar for at hver enkel milepæl blir nådd

|          |                                     |
|----------|-------------------------------------|
| <b>A</b> | Arbeidsledelse med framdriftsansvar |
| <b>U</b> | Utfører arbeid                      |
| <b>B</b> | Hovedbeslutning                     |
| <b>b</b> | Delbeslutning                       |
| <b>R</b> | Må rådspørres                       |
| <b>r</b> | Kan kontaktes for råd               |
| <b>I</b> | Må informeres                       |
| <b>D</b> | Må drøftes                          |

| Arbeidspakke             | Milepæl                       | Prosjektleder | Styringsgruppe | Veiledere |
|--------------------------|-------------------------------|---------------|----------------|-----------|
| A1 Utrede                | M1 Kjøremiljø                 | A/U/B         | R/D            | r         |
|                          | M2 Sammenligningskriterier    | A/U/B         | R/b/D          | r         |
|                          | M3 Estimere tid               | A/U/B         | r              | r/I       |
| AP2 Installasjon         | M4 Oppsatte kjøremiljø        | A/U/B         | r              | r         |
| AP3 Sammenligningsstudie | M5 Installasjon og test       | A/U/B         | r              | r         |
| AP4 Utarbeide anbefaling | M6 Anbefaling                 | A/U/B         | r              | r         |
| AP5                      | M7 Ferdigskrevet sluttrapport | A/U/B         |                | r/I       |

## 5.5 Gantt Skjema

Vedlagt ligger ett Gantt diagram for prosjektet, her er en oversikt over de arbeidspakkene som skal utføres og milepælene som ligger i og etter arbeidspakkene.

En oversikt over milepæler og arbeidspakker finner man under 5.2 og 5.3.

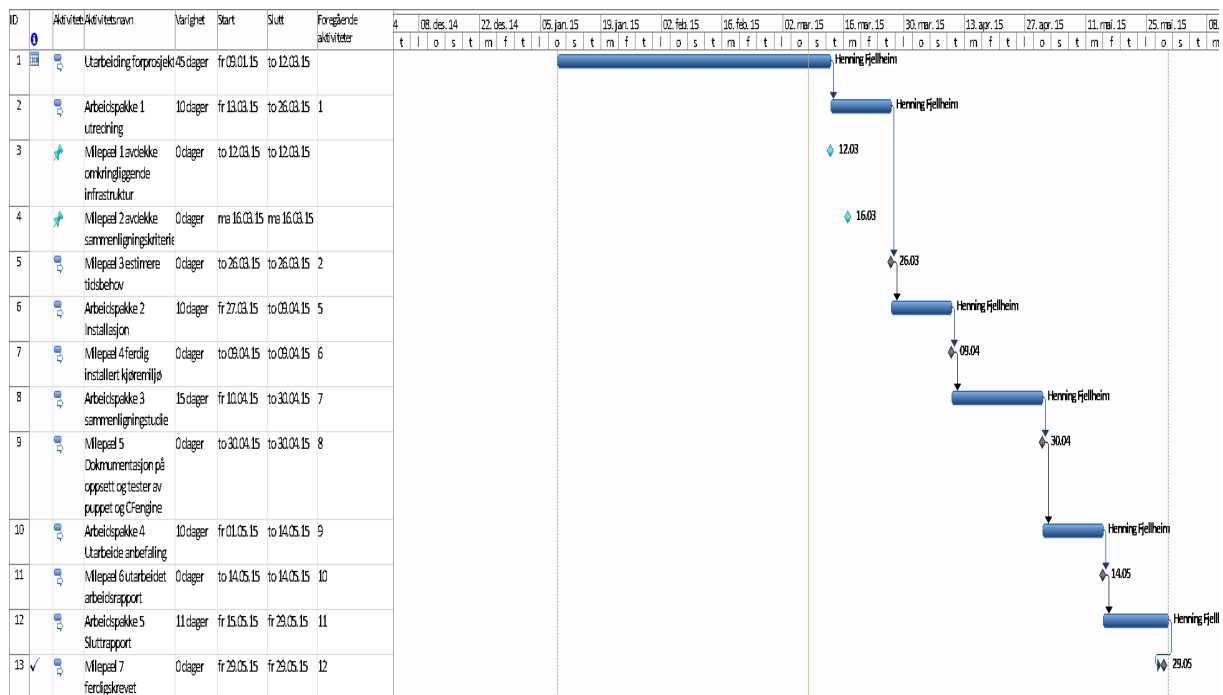
Arbeidspakke 1 har jeg beregnet til 10 dager, inne i den har jeg 3 milepæler, jeg har beregnet 3 dager pr. milepæl.

Arbeidspakke 2 har jeg beregnet til 10 dager, her har jeg bare en milepæl.

Arbeidspakke 3 har jeg beregnet til 15 dager, her har jeg bare en milepæl.

Arbeidspakke 4 har jeg beregnet til 10 dager, her har jeg bare en milepæl.

Arbeidspakke 5 har jeg beregnet til 11 dager, her har jeg bare en milepæl.



## 6 Risikoanalyse

Det er her tatt opp de punktene som er viktige for prosjektet, har også satt opp hvilke tiltak som blir utført for å minimere risikoen for at problemene skal oppstå.

### 6.1 Risikoanalyse

I risikoanalysen bruker jeg rangeringen fra 1 til 3. under er de risikoene som er for prosjektet:

1 er lav risiko/konsekvens 3 er høy risiko/konsekvens, jeg multipliserer risikoer, så 1 blir lav risiko/konsekvens, 6 blir middels risiko/konsekvens og 9 blir høy risiko/konsekvens

| Risiko   | Sannsynlighet | Konsekvens | Sum | Tiltak   |
|--|---------------|------------|-----|--|
| 1. Uforutsette problemer som kan ta lengre tid enn planlagt                                      | 2             | 3          | 6   | Hvis dette skulle bli for vanskelig, søke ekstern hjelp utenfor prosjektet.  |
| 2. Dårlig disponering av tid   | 2             | 3          | 6   | Følge tidsfristene som er satt og ikke falle for fristelsen å utføre flere oppgaver enn de som er planlagt                                       |
| 3. Sykdom  | 2             | 3          | 6   | Har planlagt med litt "luft" i prosjektet (litt ekstra tid) slik at det skal være mulig å være litt syk, lang tids sykdom vil være katastrofalt. |
| 4. Tap av viktige data (dokumenter eller kode)   | 1             | 3          | 3   | Kode lastes opp i GIT. Dokumenter legges på server som det tas sikkerhetskopi av   |
| 5. Systemer feiler hardwarefeil eller feilkonfigurasjon som gjør at system må settes opp på nytt | 2             | 1          | 2   | Sørge for å ha god dokumentasjon på hva som er satt opp og hvordan slik at det går fort å gjenopprette systemer.                                 |

## 7. Konklusjon

Dette kapittelet skal avgjøre om prosjektet burde gjennomføres, det er ikke ett poeng å starte med ett prosjekt som ikke kan gi ett ønsket utfall.

Etter å ha utført forprosjektet, så er konklusjonen at dette prosjektet burde gjennomføres. Det er funnet at prosjektet kan tilføre Nasjonalbiblioteket en gevinst ved å utføre prosjektet ved at man får en bedre forståelse av automatisering og man da har en faglig forankring om automatisering vil gi den gevinsten som er ønskelig.

Ved å utføre dette prosjektet så vil Nasjonalbiblioteket bli bedre rustet, da man har fått vurdert fordeler og ulemper med automatisering og også fordeler og ulemper ved å bruke forskjellige automatiseringsverktøy.

Man vil også finne ut hvilke gevinster de forskjellige automatiseringsverktøyene vil gi. Da forskjellige verktøy for å gjøre samme jobb, gjerne har forskjellige fordeler og ulemper.

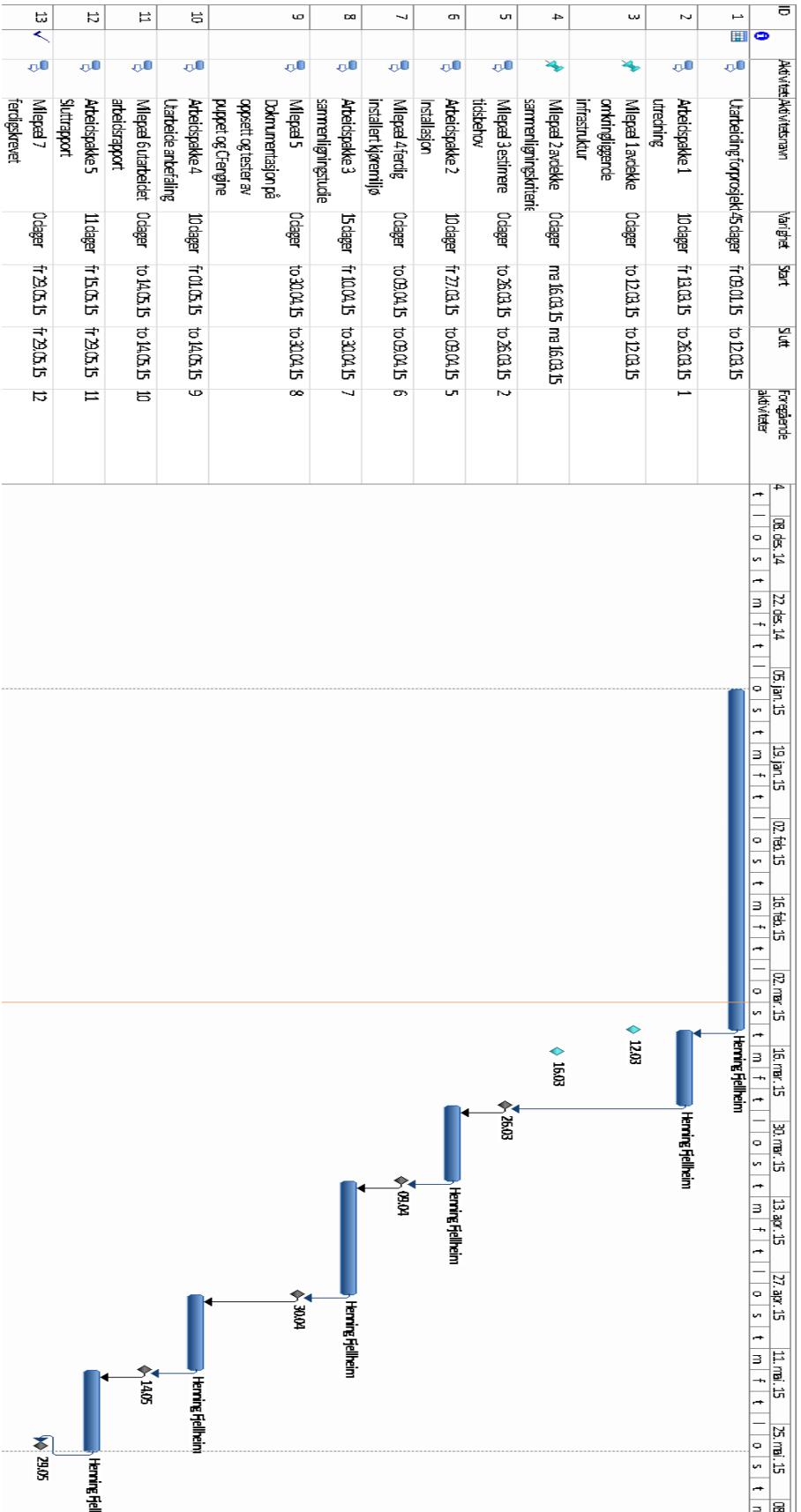
Det vil også kunne gi nasjonalbiblioteket svar på om både patching og vedlikehold kan utføres tilfredsstillende med de verktøyene som er valgt eventuelt så vil prosjektet forsøke å avdekke om det er andre verktøy som må inn i tillegg til Puppet/CFEngine for å få til den måloppnåelsen som er ønskelig for prosjektet.



## Appendix A: Referanseliste

- Andersen, Erling S. (2004) Målrettet prosjektstyring. (5. utgave)  
Bokhylla (2015) hetet den 15.02.15 fra <http://www.nb.no/nbsok/>  
Centos (2015) hentet den 17.02.15 fra <http://www.centos.org/about/>  
CFEngine (2015) Hentet 15.01.2015 fra <http://cfengine.com>  
freeIPA (2015) hentet den 03.02.15 fra <http://www.freeipa.org/page/About>  
Nagios (2015) hentet den 05.03.15 fra <http://www.nagios.org/>  
NginX (2015) Hentet 15.02.2015 fra <http://nginx.org/en/>  
Mandat (2015) Hentet den 15.02.2015 fra <http://www.nb.no/Om-NB/Fakta/Mandat>  
OpenSSH(2015) Hentet den 04.03.2015 fra <http://en.wikipedia.org/wiki/OpenSSH>  
Puppet (2015) Hentet 15.01.2015 fra <http://puppetlabs.com>  
Python (2015) Hentet 27.02.2015 fra <https://www.python.org/>  
Redhat (2015) Hentet 15.02.15 fra <http://redhat.com/en>  
SELinux (2015) Hentet 17.02.15 fra [http://selinuxproject.org/page/Main\\_Page](http://selinuxproject.org/page/Main_Page)  
Sudo (2015) hentet den 18.02.15 fra <http://www.sudo.ws/>  
TDD (2015) hentet den 27.02.2015 fra [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)

## Appendix B: gannt diagram



## Appendix C: Teknisk ordliste

**Bokhylla:** Nettside der brukere kan lese elektroniske publikasjoner på nett. (Dette kan være aviser, bøker, tegneserier med mer.)

**Centos:** "gratisversjon" av RedHat. Bortimot det samme som RedHat, men man har ikke support.

**CFEngine:** Automatiseringsverktøy for å sørge for automatisk utrulling av tjenere.

**freeIPA:** Dette er ett identitets, policy og revisjonsverktøy. Her legger man inn brukere, grupper og hvilke rettigheter disse har for å enklere kunne styre hvilke rettigheter brukere har ute på klientmaskiner som er tilkoblet denne.

**Nagios:** Verktøy som brukes til overvåkning av servere, klienter.

**NginX:** Er en webserver, brukes for å vise fram websider.

**OpenSSH:** En pakke med nettverks sikkerhetsprodukter basert på SSH protokollen, brukes til å hjelpe til med å sikre nettverkskommunikasjon.

**Puppet:** Automatiseringsverktøy for å sørge for automatisk utrulling av tjenere.

**Linuxtjener:** Her referert til en datamaskin, enten fysisk eller virtuell som det er installert Linux operativsystem på.

**Python:** Objekt orientert programmeringsspråk

**RedHat:** En linuxversjon der man betaler for support.

**SELinux:** Dette er en sikkerhetsmodul for Linux, skal sørge for bedre sikkerhet på linuxmaskiner.

**Sudo:** Er en kommando man bruker for å kjøre superbruker kommandoer på en datamaskin som det er installert ett \*nix operativsystem på.