
Eksamensprojekt 2013

Automatisk patch management modul til Microsoft System Center Configuration Manager 2007/2012

Rapport nummer: *IMM-B.Eng-2013-49*

Vejleder: *Stig Høgh*

Dato:

Underskrift:

Skrevet af:

John Gade Birkholz Petersen

S083423

Forord:

Denne rapport er skriftelig dokumentation for det afgangprojekt, som er blevet lavet ved afslutningen af uddannelsen Diplom-IT ved Institut for Informatik og Matematisk Modellering, Danmarks Tekniske Universitet.

Projektet omhandler udviklingen af et patch management modul til SoftwareCentral, som er frontend til Microsoft System Center Configuration Manager. Projektet er udført i perioden 4. februar – 10. juni 2013.

Rapporten er afleveret i to forskellige eksemplarer. En hvor alt indhold er med, som er afleveret til DTU vejleder Stig Høgh, samt én til censor. Denne rapport er grundenet dens indhold hemmeligholdt ét år frem, startende fra afleveringsdatoen. Dette skyldes at rapporten indhold er omhandlende implementering på et kommercielt produkt, og kan være til gavn for konkurrerende produkter. Derfor er afsnit 5,6 og samtlige bilag taget ud af det andet eksemplar, som er blevet afleveret elektronisk, så rapporten kan blive offentliggjort.

Der er ikke vedlagt nogen kildekode, da det som oplyst i ovenstående tekst, tilhører et kommercielt produkt. Det er heller ikke muligt at kunne teste uden der er et fungerende test miljø.

Der opsat en demoside hvor den udviklede kode kan testes. Denne side kræver et login, som kun vil være at finde på de to udskrevne rapporter.

www.softwarecentral-demo.com

Brugernavn: _____

Password: _____

Inden der logges ind på denne hjemmeside, bedes der tages hensyn til at det er et produktionssite hvor fysiske computere er koblet op til, hvorfor den vedlagte brugervejledning, som findes i bilag C, skal læses igennem. Denne er på engelsk da produktet også skal benyttes af internationale kunder. Ydermere skal det oplyses at siden er optimeret til at køre i Internet Explorer.

Til sidst skal der gives et tak til min virksomhedsvejleder, Max Mayn Petersen, for hjælp med diverse problemer, samt medarbejdere i SoftwareCentral og Mansoft A/S med hjælp til opsætningen af demomiljøer samt brugerguides til patch management. Ligeledes en tak til Bent fra KMD, for at låne deres guide til Patch Management i SCCM 2012.

Resume:

Dette projekt omhandler udviklingen af et nyt patch management modul til SoftwareCentral. Dette er et webinterface til Microsoft System Center Configuration Manager, også kaldet SCCM. Der er to versioner af SCCM som SoftwareCentral understøtter. Dette er SCCM 2007 og SCCM 2012, hvorpå patch management modulet skal understøtte dem begge.

Rapporten består først af et prestage afsnit, der gennemgår de forskellige produkter som ligger til grund for dette projekt. Patch management er en vigtig del i en større organisation, og grundet efterspørgsels fra flere kunders side, er dette modul blevet udviklet. Ud fra denne analyse blev der fundet frem til de delkomponenter som der skulle udvikles før projektet kunne realiseres.

Der blev herefter opstillet en række krav til hver af disse komponenter, samt krav til det samlede produkt, og hvad der kunne betegnes som en succes. Derudover blev der lavet en risikoanalyse, da der var flere ting som kunne sætte produktionen af produktet på pause, fordi der igennem det meste af projektet er blevet arbejdet på store virtuelle server systemer, som først skulle konfigureres så det ønskede modul kunne udvikles.

Rapporten indeholder også et stort implementationsafsnit som gennemgår en del af den store mængde kode som er blevet skrevet igennem projektperioden. Dertil er der også en række test som skal redegøre for, om den kode som er blevet lavet, virkede som den skal. Der er heriblandt en fuld systemtest på et virtuelt miljø, som gennemkører et helt udrulningsforløb af en række opdateringer.

Hele projektet endte ud i at opfylde de succeskriterier som der blev opsat ved start, og derfor kan dette projekt antages for at være en succes.

Indhold

FORORD:	2
RESUME:	3
1. INDLEDNING:	6
2. PRESTAGE:	7
2.1 HVORFOR PATCH MANAGEMENT?.....	7
2.1.1 Sikkerhed.....	7
2.1.2 Datatrafik.....	7
2.1.3 Tjek på opdateringer.....	7
2.2 PRODUKTER:.....	8
2.2.1 SoftwareCentral	8
2.2.2 Microsoft System Center Configuration Manager	9
2.2.3 Windows Server Update Services	9
2.3 KOMPONENTER.....	10
3. PROJEKTPLANLÆGNING.....	11
3.1 MÅLSÆTNINGER:.....	11
3.2 SUCCESKRITERIER:.....	11
3.3 KRAVSPESIFIKATION:	12
3.4 TIDSPLAN:	14
3.5 RISIKOANALYSE:	15
3.5.1 Tidsmangel:.....	15
3.5.2 Udviklingsmiljø:.....	16
3.5.3 Datamangel:	16
3.5.4 Datatab:	16
3.5.5 Uopfyldte succeskriterier:	17
3.6 UDVIKLINGSMETODER:	17
4. AFGRÆNSNING:.....	18
5. IMPLEMENTERING:.....	19
6. TEST.....	19
7. VIDEREUDVIKLING:.....	20
7.1 TREDJE PARTS PRODUKTER.....	20
7.2 EULA MANUEL ACCEPTERING	20
7.3 HÅNDBTERING AF FEJLENDE OPDATERINGER I SOFTWARECENTRAL	20
7.4 SUPPORT	20
8. DISKUSSION	21
9. KONKLUSION:.....	22
10. KILDEHENVISNING:.....	23
11. BILAG:.....	24

1. Indledning:

Dette projekt omhandler udviklingen af et modul, som skal håndtere automatisk udrulningen af opdateringspakker igennem Microsoft System Center Configuration Manager (SCCM) 2007 og 2012. Denne automatiske udrulning vil blive administreret og håndteret igennem SoftwareCentral. En yderligere beskrivelse af disse to stykker software kan ses i afsnit 2.2.1 og 2.2.2

Håndtering af disse opdateringer vil blive omtalt med dets faglige term Patch Management.

Eftersom SCCM selv har et indbygget modul som kan håndtere opdateringer af klient computere, var det først nødvendigt at undersøge hvordan denne funktionalitet var integreret, og hvordan denne kunne blive forbedret. Ydermere skulle modulet kunne køre på begge versioner af SCCM. Det var derfor nødvendigt at undersøge begge versioner grundigt først, så koden nemt kunne overføres fra den ene version, og over til den anden. Dokumentationen fra Microsoft side var på dette punkt meget sparsom. Der har dermed været en stor mængde af manuelt arbejde forbundet med at finde ud af hvordan den bagvedliggende kode i SCCM fungerede. Derudover stillede firmaet, projektet er udviklet hos, servereksperter samt SCCM eksperter til rådighed, der med deres faglige kompetencer kunne give råd og vejledning.

SoftwareCentral som produktet dette modul er udviklet til, er et web frontend til SCCM. Dette produkt er installeret hos over 50 af Danmarks største virksomheder, som heriblandt tæller Region H, Velux, Siemens, Vestas, Superfos, Synoptik, Lundbeck og ikke mindst DTU. Patch Management modulet vil blive lavet som et tillægsprodukt til SoftwareCentral.

2. Prestage:

I dette afsnit vil der blive gennemgået hvorfor der bruges patch management i større organisationer, og de produkter som modulet skal køre på, samt den analyse der blev gennemgået for at finde de komponenter som var nødvendige for det færdige modul. Denne analyse blev lavet igennem en research på hvad SCCM skal bruge, for at sende opdateringer ud, samt kommunikation med flere SCCM eksperter som var tilknyttet firmaet, der kunne forklare hvordan dette normalt ville blive gjort i et produktionsmiljø.

2.1 Hvorfor Patch Management?

Følgende er en forklaring på hvorfor et firma med 20.000 computere burde investere i en løsning der understøtter patch management.

2.1.1 Sikkerhed

Med en hjemmecomputer med Windows på, har de fleste brugere sat den op til at downloade opdateringer og installere disse. Dette er hensigtsmæssigt så der herved vil være de nyeste sikkerhedsopdateringer fra Microsofts side. Dog kan det ikke vides sikkert at opdateringerne er gjort på alle maskiner i det fiktive firma, hvilket skyldes at mange opdateringer kræver at computere bliver genstartet før de virker. Da genstarten kan udsættes, vil der kunne risikeres at være computere i firmaet som indeholder sikkerhedsfejl, hvorved der nemt kan blive angrebet med virus eller lignende. Der vil heller ikke være overblik over hvilke maskiner som mangler hvilke opdateringer.

Et patch management system vil derfor være fordelagtigt, da der herigennem kan sikres, at alle computere har de nyeste opdateringer. Ydermere i tilfælde af vigtige opdateringer som kræver computerne genstartet, kan dette opsættes så computere i firmaet automatisk henter, installere og genstarter inden for et givet tidsrum.

2.1.2 Datatrafik

En normal pakke af opdateringer fylder ikke meget og vil ofte være under 50Mb af data, men da Microsoft udrullede deres første service pack til Windows 7, var den på lidt over 1GB. Dette er en stor mængde af data, og ville ikke være noget problem hvis der kun er én maskine stående. Problematikken opstår hvis et firma med 20.000 computere alle skal opdateres. Herved vil den lille opdatering på de 50 Mb, løbe op i 1 TB (terabyte) af data, som skal hentes ned via internettet, og ved den store service pack, er det 20TB af data. Dette er en stor mængde som ville belaste netværket, og mange firmaer har opsat server som henter opdateringerne ned og sender dem ud via det lokale netværk. Dette gøres i eksemplet med de 20.000 computere, at firmaet nedsætter mængden af data som skal hentes med 99,995%¹.

2.1.3 Tjek på opdateringer

Det antal af firmaet i det givne eksempel, har mange regler imod store ændringer i deres miljøer, og at en kommende opdaterings pakke giver sådanne ændringer. Derfor ønskes der at tjekke alle opdateringer før

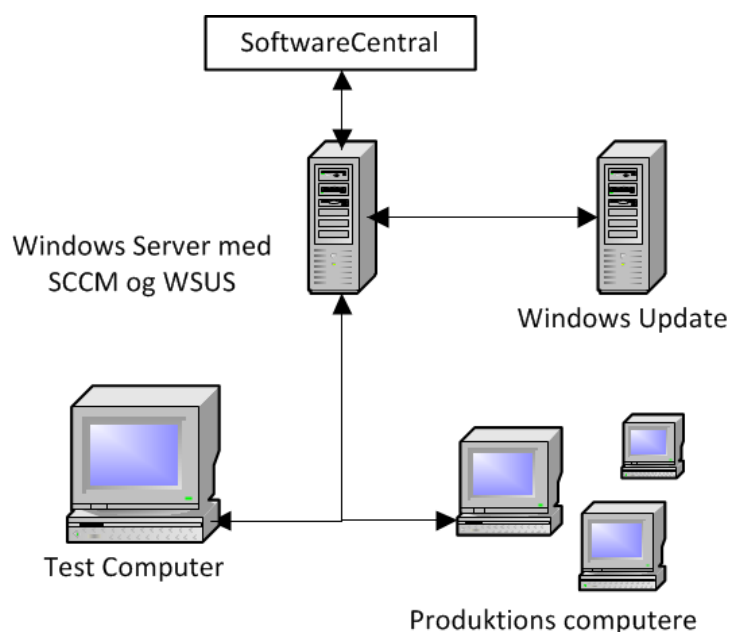
¹ Hvis der tages udgangs punkt i alle computere henter fra serveren er det kun denne som skal hente opdateringerne, så i stedet for de 20.000 download kommer der ned på én, altså $1/20.000 = 0.005\%$ af den oprindelige datatrafik.

de bliver sendt ud på alle de 20.000 computere i firmaet, både for at tjekke efter store ændringer, men også for at tjekke om opdateringer giver fejl.

En fejl kunne være at firmaet har et specielt produkt der kører oven på Windows, og at små ændringer i en opdatering ville medføre at produktet ikke virkede mere. Det er derfor vigtigt at der kan tjekkes en opdatering, inden den bliver sendt ud. Dette vil kunne gøres med et patch management system, hvor der først ville sendes opdateringerne ud til et antal test computere, inden alle computere modtager dem.

2.2 Produkter:

For at kunne forstå sammen hængende mellem de forskellige produkter som skal bruges til at dette modul kører, er der lavet en mindre skematik over opsætningen på figur 1



Figur 1: Skitsering af opsætningen

2.2.1 SoftwareCentral

SoftwareCentral er et web interface til Microsoft SCCM 2007 og SCCM 2012. Migreringen til 2012 version er blevet lavet af tidligere DTU studerende samt min vejleder under praktikken, Max Mayn Petersen.

SoftwareCentrals grundide er at det skal være nemt, hurtigt og ikke mindst sikkert at benytte SCCM 2007 og 2012. Da SCCM er et meget avanceret program som kræver en lang og kostbar uddannelse i at mestre, ville det være ønskeligt at kunne have et interface som kunne bruges af mindre erfarne IT-folk. Derfor er SoftwareCentral opbygget af meget simple og ikke mindst selvforklarende interfaces, hvor de vil kunne lave ellers komplicerede opgaver på SCCM serveren uden at vide noget om alt det der ligger bagved. Selve undervisningen og træningen i at benytte sig af SoftwareCentral tager omkring en time. På den måde vil en potentiel virksomhed kunne spare penge på løn til disse højuddannede eksperter, da der også igennem SoftwareCentral kan oprettes roller, så helpdesk brugere kan stå for den daglige administration.

Som det kan ses på ovenstående figur, arbejder SoftwareCentral oven på SCCM serveren og sender data frem og tilbage mellem interfacet i SoftwareCentral og databasen på serveren.

2.2.2 Microsoft System Center Configuration Manager

På figur 1, kan det ses at der centralt er en server i midten som er en Windows Server, hvor der er installeret SCCM på. SCCM eller Microsoft System Center Configuration Manager, er et Microsoft produkt der gør det muligt at administrere store grupper af computere der kører på Windows platformen, dog skal det siges at der i den sidste service pack til SCCM 2012 er kommet en support til MAC computere, dog ikke i samme grad som Windows computere.

Der er to forskellige versioner af SCCM, en 2007 og en 2012 version. Den helt store forskel på disse er at Microsoft i 2007 versionen havde baseret hele systemet på enheder, dvs. at der knyttes software og styresystem til computeren. Til forskel er der i 2012 versionen blevet valgt, fra Microsofts side, at software kan knyttes til en bruger, i stedet for en computer.

Den måde at kommunikationen og ned til klient computerne sker på, er ved at der er installeret et mindre klient program. Dette program står for at sende alt information ned i databasen på serveren så det kan hentes i SoftwareCentral.

Som det også kan ses på figuren er computerne delt op. Dette skyldes at computere normalt opdeles i grupper, såkaldte collections. Herved gøres det muligt at have mindre grupper af test computere, så der sikres at de opdateringer som sendes ud ville virke efter hensigten.

2.2.3 Windows Server Update Services

Windows Server Update Services også kaldet WSUS, er en rolle som installeres på serveren som kører SCCM. Denne rolle står så for at snakke sammen med Windows Update. Windows Update er den samme som der kendes fra normale Windows computer når der køres updates. WSUS er programmet der gør det muligt, at håndtere opdateringer i ens system.

Måden dette fungere på er at opsætte WSUSen til at hente metadata ned omkring opdateringerne. Der vælges herefter hvilke Microsoft produkter der ønskes at hente metadata på. Det er vigtigt at der kun vælges dem som der skal bruges, da der er mere end 200 forskellige produkter, og de har til sammen mange tusinde opdateringer.

Det metadata opdateringen har, indeholder også et link til hvor opdateringen kan downloades. Det er dette link WSUS bruger til at hente opdateringerne fysisk ned på serveren, så den kan sendes ud i systemet

2.3 Komponenter

Der blev i starten af projektet brugt meget tid på at undersøge hvilke komponenter der skulle udvikles for at modulet ville køre som det skulle.

Til denne analyse blev der både brugt internettet til at finde ud af hvorledes der normal blev udrullet opdateringer, men også guides lavet af helholdvis en medarbejder i firmaet selv, samt en guide lavet af en kunde til SoftwareCentral. Denne analyse gav følgende komponenter

1. Måde til at synkroniser opdateringer på.
2. Search folder
3. Package
4. Template

Efter som flere af disse ting vil blive gennemgået senere i denne rapport, vil der kort her blive beskrevet hvorfor de forskellige komponenter skal bruges.

Synkronisering:

Det skal være muligt at hente opdateringer for et nyt produkt direkte fra SoftwareCentral, hvis der i firmaet bliver købt et nyt produkt, f.eks. Windows 8 eller Office 2012.

Search folder:

Det skal være muligt at udvælge opdateringer ud fra en masse kriterier.

Package:

Det skal være muligt at have en pakke (package) med en masse distributions punkter. På den måde uddeles arbejdet når der skal hente opdateringer fra klienterne, så alle ikke henter direkte fra serveren.

Template:

Det skal være muligt at kunne angive hvorledes en installation skal foregå, og det skal ligeledes være muligt at sætte det forskelligt på de to kørsler af opdateringerne.

3. Projektplanlægning

I dette afsnit vil der blive gennemgået den planlægning, som er blevet lavet igennem projektet, for at sørge for projektet vil blive en succes. Derudover vil der blive gennemgået de risici, som kunne opstå for at projektet blev forsinket eller fejlede.

3.1 Målsætninger:

Følgende målsætninger blev sat op i starten af projektperioden:

- Projektet følger den opsatte tidsplan samt overholder alle milepæle som blev opsat efter analysen af nødvendige komponenter:
 - Dette mål er sat for at kunne få et flydende arbejdsflow, og undgå tidspres i slutningen af projektet.
- Test af det udviklede modul på produktionsmiljøer hos kunder.
 - Da det ikke er muligt at tjekke hvorledes modulet ville agere i større miljøer, vil en succesfuld test hos en middelstor kunde anses som en succes.
- Selv om der er forskel på SCCM 2007 og SCCM 2012, ønskes der en ens brugeroplevelse.
 - For at undgå unødvendig udvikling af interfaces, ønskes der at det samme interface med få modifikationer kan bruges til SCCM 2007 og SCCM 2012 versionen.

Disse målsætninger var dét som der fra start kunne anses som et succesfuldt projekt, hvis de blev opfyldt. Dog ville det ikke være kritisk hvis disse ikke blev opfyldt fuldtud. Målene er opsat ud fra den viden der er opnået igennem faget 42430 Projektledelse omhandlende SMART mål, hvor smart står for Specifikke - Målbare - Ambitiøse - Realistiske – Tidsbestemte mål.

3.2 Succeskriterier:

Følgende succeskriterier er blevet opstillet for, om projektet anses som en succes. I forhold til målsætningerne skal alle disse overholdes.

- Dokumentation for det færdige projekt afleveres til den aftalte dato 10. juni 2013.
 - Dette succeskriterium opfyldes ved færdiggørelsen af en samlet rapport, som opfylder de krav som er stillet til denne.
- Modulet skal være implementeret til at kunne køre på både SCCM 2007 og SCCM 2012, og ligeledes sende opdateringer ud til en testmaskine.
 - Dette succeskriterium opfyldes ved at have opsat et testmiljø og se en række opdateringer blive installeret på en testcomputer. Herefter skal der kunne ses at der bliver sendt oplysninger tilbage så der igennem modulet kan ses om installationen gik som ønsket.

3.3 Kravspecifikation:

Efter analysen i prestage afsnitte, er der kommet frem til en række af komponenter, som var nødvendige for at en automatiseret opdateringsservice kan køre. Hver af disse komponenter har krav som de skal opfylde før de kan godkendes. Derudover er der også generelle krav til selve systemet, som skal opfyldes før det kan kaldes en succes.

Generelle krav:

- Systemet skal køre både på SCCM 2007 og SCCM 2012.
 - Hovedkravet til modulet er at sende opdateringer ud automatisk, da der i selve SCCM 2007 ikke findes noget indbygget system som gør dette. I SCCM 2012 er dette lavet, dog ønskes der en mere simpel måde at opsætte dette på.
- De to versioner skal så vidt mulig ligne hinanden.
 - Dette skyldes at hvis der skulle være en kunde som opgraderet fra SCCM 2007 til SCCM 2012 vil de ikke opleve nogen ændringer i oplevelsen af modulet.
- Systemet skal køre automatisk med mindst muligt brugerinteraktion.
 - Det andet hovedkrav som skal opfyldes, så en bruger efter at have konfigureret systemet kun skal godkende eventuelle opdateringer.
- Detaljeret informationer hvordan en opdatering er kørt skal være mulig.
 - Dette skal være muligt, så der nemt kan fejlfindes hvis eventuelle opdateringer fejler på maskinerne uden at skulle ind på SCCM serveren og tjekke.

Komponentspecifikke krav:

- **Synkronisering af WSUS**
 - Det skal være muligt at synkronisere specifikationer med WSUS fra interfacet.
 - Det skal være muligt at synkronisere produkter med WSUS fra interfacet.
- **Search Folder**
 - Kun nødvendige kriterier skal bruges, og kun kriterier det er muligt at søge på.
 - En search folder i SCCM 2007 og SCCM 2012 indeholder mange forskellige kriterier som kan benyttes til at finde konkrete opdateringer. Mange af disse er ligegyldige, og skal derfor udelukkes. Ydermere har SCCM 2007 listet en række ting som der ikke kan søges på i listen over kriterier.
 - Det skal være muligt at oprette/slette og opdatere en search folder.
 - Da alle data til komponenterne tilgås ved hjælp af WMI eller SQL, skal de alle overholde principperne i CRUD (Create, Read, Update, Delete)

- En beskrivelse en i læsbar tekst skal kunne generes.
 - Eftersom søgekriterier er lagt i en XML fil, skal det være muligt at give brugeren et nemt overblik over hvad en søgefolder indeholder i læsbar tekst.
- **Patch Management template**
 - Templatet skal være med som hjælp, og skal indeholde alle de indstillinger der kan sættes på en updateservice.
 - Da der ikke kan bruges SCCM egne templates, skal dette laves fra bunden og håndteres af SoftwareCentral 100 %.
 - Skal overholde reglerne fra CRUD, denne gang kun via SQL.
 - Datamængden skal minimeres i databasen.
 - Da der er mange indstillinger, skal der findes en måde hvorpå det undgås at have en kolonne til hver indstilling.
- **Patch Management package**
 - Skal overholde principperne i CRUD igennem SQL og WMI
 - Skal kunne tildeles distributionspunkter.
 - Hvis ikke dette gøres vil serveren blive overbelastet.

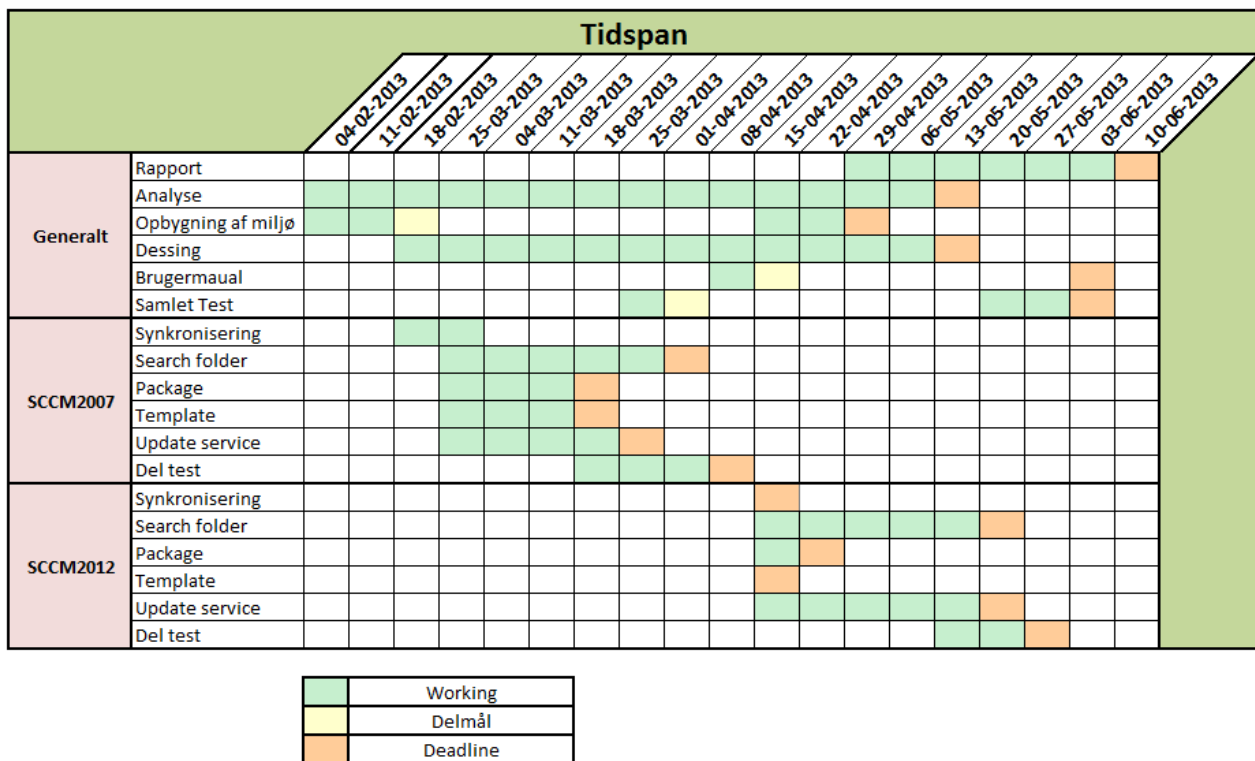
3.4 Tidsplan:

Projekter af denne størrelse kan nemt gå galt og blive forsinket, så det var derfor hensigtsmæssigt at have en tidsplan under hele projektet. I en sådan tidsplan kan der skabes et overblik over hvornår der skal indlægges flere ressourcer f.eks. er i form af overarbejde samt weekendarbejde. Tidsplanen blev tilføjet en række milepæle samt større deadlines for hvornår dele af projektet skulle være færdigt. Dette er gjort for at optimere ressourceforbruget.

Projektet strakte sig over 126 kalenderdage, hvad svarer til 18 uger. Under denne tid blev der sideløbende taget 10 ECTS point på DTU, som krævede deltagelse mandag og tirsdag formiddag. Derudover var der en stor aflevering i begge af disse fag, som også trak kalenderdage ud. Alt dette er blevet estimeret til ca. 22 kalenderdage. Fagene der blev fulgt var som følger.

- 42430 – Projektledelse
- 02180 - Introduktion til kunstig intelligens

Ydermere var der en række helligdage i projektperioden, som samlet strakte sig over 10 kalenderdage. Dette giver 94 tilbagestående dage hvor projektet blev lavet. På figur 2 ses den færdige tidsplan for hele projektperioden.



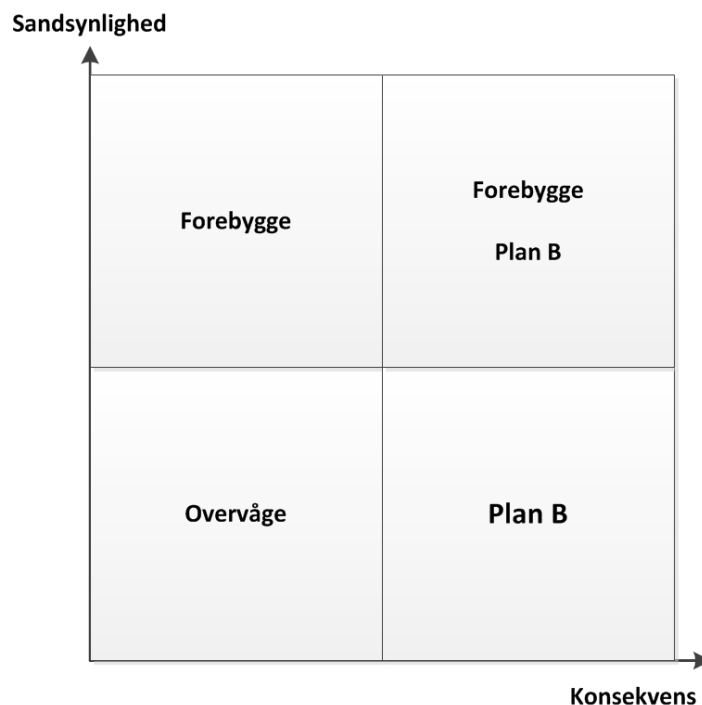
Figur 2: Tidsplan over projektet

3.5 Risikoanalyse:

For at finde de risici som kunne opstå under projektperioden, blev en risikoanalyse foretaget til at beregne hvilken skade samt sandsynlighed forskellige risici kunne have på det endelige resultat. Analysen blev foretaget ud fra den teori der er blevet lært i faget 42430 Projektledelse.

En risiko beregnes ud fra to variabler: Hvilken konsekvens dette ville have, som sættes til én værdi ud fra en skala fra 1 til 5. Den anden variabel er sandsynligheden som også sættes til en værdi ud fra en skala fra 1 til 5. Disse to variabler benyttes til at beregne et risikotal, som giver den endelige risiko. Dette gjordes ved at multiplicere de to variabler.

Alt efter hvor stort risikotallet var, kan det på grafen på figur 3, ses hvilke handlinger som der skulle foretages.



Figur 3: Riskotal tabel

3.5.1 Tidsmangel:

Eftersom projektet ikke var det eneste projekt som skulle færdiggøres i projektperioden, ville der være en stor sandsynlighed for, at der skulle bruges ekstra tid på disse projekter. Dette ville i værste tilfælde give et stort tidspres i slutningen af perioden, og kunne resultere i at dele af projektet ikke ville kunne færdiggøres i den kvalitet der var ønsket.

Konsekvens: 4
Sandsynlighed: 4
Risikotal: 16

Forebygge:

Der laves en udførlig tidsplan, som der tager højde for at der kan forekomme tidspunkter i slutningen af perioden, hvor der vil blive brugt mere tid på DTU til andre projekter.

Plan B:

Hvis projektet får akut mangel på tid, vil delen til SCCM 2007 færdiggøres som det første, og versionen til SCCM 2012 vil færdiggøres til størst mulig færdighedsgrad. Ydermere laves der en revideret tidsplan.

3.5.2 Udviklingsmiljø:

Til dele af projektet skulle der bruges virtuelle udviklingsmiljøer. Disse skulle først opbygges så de virkede som ønsket med en kørende WSUS server. Der ville her være store risici indblandet, hvis det skulle hende at disse ikke kunne laves eller brød sammen. Denne problematik ville stoppe udviklingen helt.

Konsekvens:	4
<u>Sandsynlighed:</u>	<u>5</u>
Risikotal:	20

Forebygge:

Der skulle bruges udviklingsmiljøer som var testet og brugt af andre. Hertil skulle der forsøges at installere WSUS på dem. Ydermere skulle der tages backup når et kørende miljø var til stede, for at der kunne vendes tilbage og genbruge dette, hvis der skulle ske nedbrud.

Plan B:

Plan B var at bruge allerede eksisterende testserver i firmaet, og udvikle op imod disse.

3.5.3 Datamangel:

Da dét system som skal udvikles skal stå for at opdatere store mængder af maskiner, og da det kræver store ressourcer at opbygge et sådant miljø, ville datamangel med sikkerhed opstå.

Konsekvens:	2
<u>Sandsynlighed:</u>	<u>5</u>
Risikotal:	10

Forebygge:

Der var ikke noget som kunne forebygge datamangel, da der herved skulle opsættes hundredvis er virtuelle computere, for at kunne få den datamængde der skulle til, for at lave de test som ville kunne bestemme, om dele af systemet fungerer efter hensigten.

3.5.4 Datatab:

Datatabs-risikoen var minimal, da der blev kørt med revisionsstyrning. Der kunne til stadighed forekomme strømafbrydelser, eller systemnedbrud og derved give et mindre datatab. Dog kunne det ske at alt på serveren blev slettet, men der forefindes her en lokal backup.

Konsekvens:	2
<u>Sandsynlighed:</u>	<u>1</u>
Risikotal:	2

Overvåge:

Der skulle sørges for, at når vigtige ændringer blev foretaget i koden, at der herefter blev lagt en ny revision op, så den nyeste kode altid er på serveren.

3.5.5 Uopfyldte succeskriterier:

Denne risiko havde den største konsekvens på projektet. Det er på baggrund af de opstillet succeskriterier i afsnit 3.2, da hvis én eller flere af disse ikke blev overholdt, ville projektet ikke kunne anse som nogen succes. Sandsynligheden for dette var forholdsvis lav, da der blev arbejdet efter en udførlig tidsplan, og dermed blev risikotallet også lavt.

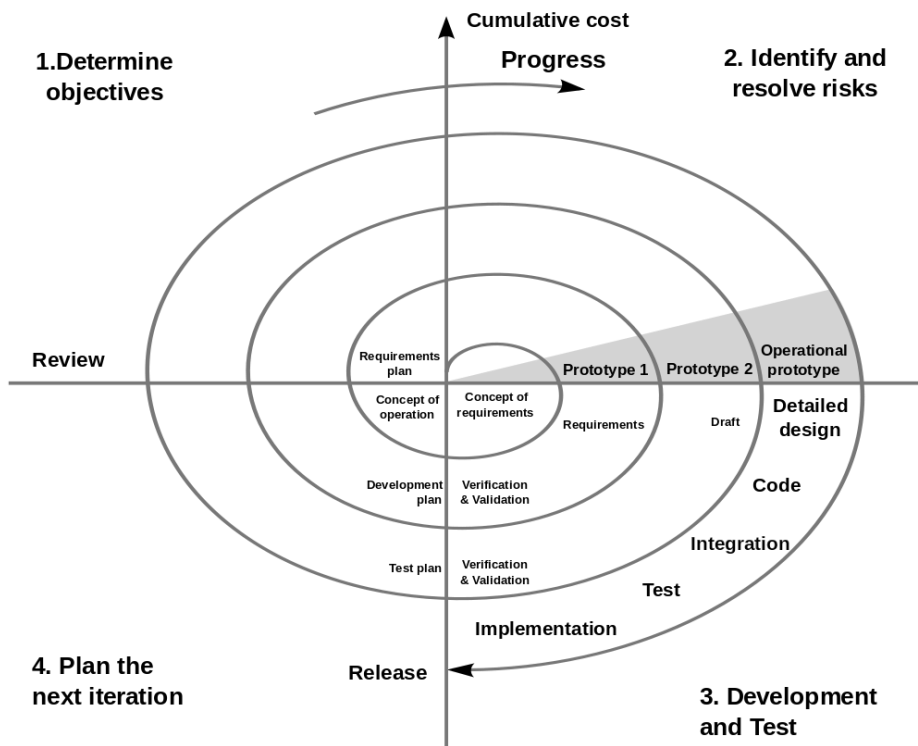
Konsekvens:	5
Sandsynlighed:	<u>1</u>
Risikotal:	5

Plan B:

Hvis dette skulle ske, var en plan B, at aflevere det som var blevet udviklet, og derefter færdigudvikle modulet bagefter.

3.6 Udviklingsmetoder:

Der blev under hele projektføreløbet brugt iterative udviklingsmetoder. Da det modul som der skulle udvikles, var helt ny kode, og da der løbende kom ændringer i hvorledes det skulle laves, passede spiral modellen, som er afbilledet på figur 4, godt på den arbejdsgangen.



Figur 4: Spiralmodellen

Der har været udviklet såkaldte prototyper af produktet, hvor den første var det færdige SCCM 2007 model, og den næste var den færdige SCCM 2012 prototype. Under disse forløb kom der undervejs nye problematikker som der skulle tages højde for, og nye funktioner som skulle bruges før systemet kørte efter planen.

4. Afgrænsning:

For at sikre at projektet kunne færdiggøres til den planlagte tid, var det nødvendigt at afgrænse projektet. Dette blev gjort ved at implementere den funktionalitet som gjorde, at det var muligt at kunne udrulle opdateringer som er tilgængelige via Windows Update.

Selv om det ville være optimalt at systemet kunne udsende opdateringer fra tredjepartprogrammer, så som Java opdateringer og lignende programmer som bruges på Windows maskiner, blev dette ikke gjort, da dette krævede en længere research.

De bruger definerede opdateringer blev ligeledes fravalgt, altså opdatering som kun bruges internt i et firma. Dette er på baggrund af mange forskellige parametre som der var svære at håndtere.

Ydermere blev selve installationen samt opsætningen af WSUS fravalgt. Hvis det var muligt at lave en guide igennem SoftwareCentral som nemt og enkelt kunne installere WSUS på serveren, og dermed sikre at alle korrekte indstillinger blev sat, så systemet kørte bedst muligt ville dette være optimalt. Dog blev der ikke gjort nogen research i, om hvorledes dette var en mulighed, men sat som et krav til kunden at WSUS skulle være installeret.

5. Implementering:

6. Test

7. Videreudvikling:

I dette afsnit ville der blive gennemgået en række forbedringer som der ville kunne laves til dette modul. Mange af disse funktioner er nogen som der er blevet skåret fra i første omgang, men er nogen som ligger på tegnebrættet, og skal laves når der tid til det.

7.1 Tredje parts produkter

Den vigtigste af disse forbedringer, er support af tredjeparts produkter, samt brugerdefinerede opdateringer. Dette anses som den vigtigste, da mange kunder gerne ønsker at kunne opdatere f.eks. Java, da der fortiden er mange vigtige sikkerhedsopdateringer til dette stykke software. Derfor er der blevet undersøgt mulighederne for denne implementation, og hertil er der blevet fundet et Microsoft produkt der hedder System Center Updates Publisher kort kaldet SCUP. Dette er mindre program som installeres oven på SCCM og WSUS og der kan her vælges imellem en række produkter fra andre firmaer end Microsoft og ikke mindst selv vælge nogen. Disse kan så komme ind og ligge på SCCMen, så de kan ses som almindelige opdateringer.

Hvis dette er tilfældet ville det udviklede modul allerede kunne håndtere dette uden nogen form for programmering. Dog ville det være hensigtsmæssig at lave et interface hvor det ville være muligt at kunne håndtere sine tredjeparts produkter.

7.2 EULA manuel acceptering

En anden ting som der skal udvikles, er en måde at kunne se EULA (End User Licens Agreement) på, så dem der ikke ønsker at have automatisk godkendelse på, manuelt kan gå ind og godkende dem inden opdateringerne bliver downloadet. Dette vil kræve en større omstrukturering i interfacet hvor det skal være muligt at læse omkring en EULA før den godkendes.

7.3 Håndtering af fejlende opdateringer i SoftwareCentral

En ting der er på tegnebrættet, er en intern håndtering af opdateringer som der fejler i SoftwareCentral, det vil sige at opdateringer som af årsager ikke kan installeres korrekt, vil kunne vælges fra og skrive en kommentar til hvorfor de fejler. Dette er for at der på et senere tidspunkt kan ses hvorfor en given opdatering ikke er ude på computerne i firmaet. Det skal være muligt at se hvem som har sagt at disse opdateringer ikke virker og hvornår dette er opdaget. Derudover skal det være muligt at kunne sende disse opdateringer ud igen i en speciel opdateringspakke, som manuelt bliver lavet igennem SoftwareCentral.

7.4 Support

Ydermere vil der være en del support indblandet med dette modul, da der kan forekomme ønsker fra kunder om ny funktionalitet som der ikke er blevet tænkt over, eller hvis de finder fejl som der skal rettes.

8. Diskussion

I dette afsnit ville der blive gennemgået de problemer som opstod under projektet, og hvad der kunne have været gjort for at de var gået bedre.

Den første ting som der har været brugt lang tid på i starten af projektet, var opbygningen af et fungerende udviklingsmiljø, hvor der var mindst en virtuel server med installeret SCCM og WSUS, og så en test computer hvor der kunne sendes opdateringer ud til. Da der ikke var et sådan miljø til rådighed, skulle der forsøges at installeres WSUS på et allerede eksisterende testmiljø. Dette var et større problem da det eneste hjælp der var i starten af projektet, var youtube videoer hvor der blev installeret en WSUS rolle. Dette virkede ikke efter hensigten, og det var først da jeg fik hjælp fra en af firmaet servereksperter, at der kom styr på det, da der var flere ekstra ting som skulle gøres før dette virkede.

Grunden til at jeg selv forsøgte først var, at jeg ønskede den fulde viden for hvorledes WSUS fungerede. På sin vis blev installationen jeg selv lavede foretaget korrekt, men der var specielle hensyn som jeg ikke havde taget højde for da det som sagt var et virtuelt miljø der blev udviklet på. Derfor burde der ved eftertanke have været taget kontakt til ekspertiseviden lidt før.

Da SCCM 2012 versionen skulle udvikles blev der anvendt en kørende server hos firmaet. Dette gav problemer, da der var flere servere som delte det som kaldes sitecode, og de virtuelle test computere som der blev sat op, havde svært ved at modtage opdateringer rigtigt da de ikke vidste hvilken server som de skulle melde ind til. Dette kunne have været undgået ved at benytte et fuldt virtuelt miljø, men da jeg efter erfaring med opbygningen af det første kunne se dette ville tage lang tid, blev denne løsning brugt, da det fungerede hvis jeg geninstallerede klienten hver gang på test computeren når den mistede sin forbindelse. Dog skulle computeren så køre et stykke tid før alt data var korrekt meldt tilbage til SCCMen.

9. Konklusion:

De opsatte succeskriterier ved starten af dette projekt var, at der skulle laves en rapport der redegjorde for det udviklede modul, sat et modul der kunne sende opdateringer ud til test maskine, se dem blive installeret og til sidst modtage data om dette. Det første kriterium er opfyldt, ellers ville dette ikke læses, og det næste er blevet opfyldt ud fra afsnit 6.3 af denne rapport, hvor en test computer korrekt installerer de opdateringer den har fået at vide, samt automatisk henter nye opdateringer og sender disse ud hvis der er brug for det. Denne test er blevet udført i både SCCM 2007 og SCCM 2012 med enslydende resultater.

Hvis der ses på de målsætninger som der blev opsat, er de ikke alle sammen overholdt, da det udviklede modul ikke kom ud til betatest hos eksisterende kunder, hvor der er kommet feedback. Det er dog kommet ud til én kunde, men der er her ikke kommet noget feedback endnu om det er afprøvet eller ej. Dog blev de andre målsætninger opfyldt ved at tidsplanen næsten blev overholdt, dog med lidt pres på til sidst grundet andre eksamener. Målet med at de to interfaces skal ligne hinanden, er opfyldt, kun med en lille ændring i den måde man vælger collections på. Dette skyldes at der var en ændring i SCCM 2012, der gjorde at det skulle vises på denne måde.

Der er blevet brugt meget tid igennem dette projekt på at lede efter data, da der fra Microsoft side ikke har været nogen dokumentation på, hvor jeg fandt det data der var nødvendigt for at få en viden om hvordan en opdatering har kørt. Derfor er projektet blevet begrænset til at det kun tager opdateringer fra Microsofts Windows Update, og ikke tredje parts produkter. Der er tænkt over løsninger til denne implementering, men er ikke fuldført ved afleveringen.

Hvis der ses på de krav som blev stillet til de forskellige komponenter der skulle udvikles er alle disse overholdt, og de forskellige komponenter spiller sammen som de skal.

Så alt i alt kan det konkluderes at projektet har været en succes, og der nu er et udviklet patch management modul til SoftwareCentral der kører både på SCCM 2007 og SCCM 2012.

10. Kildehenvisning:

Følgende kilder er blevet brugt igennem projektet til at få den viden der har været nødvendig.

SCCM 2007 SDK

<http://msdn.microsoft.com/en-us/library/cc145334.aspx>

Software Updates in SCCM 2007

<http://technet.microsoft.com/en-us/library/bb680701.aspx>

SCCM 2012 SDK

<http://msdn.microsoft.com/en-us/library/hh948960.aspx>

Software Updates in SCCM 2012

<http://technet.microsoft.com/en-us/library/gg682068.aspx>

SCCM 07 - Deploying Windows Updates Tutorial

<http://www.youtube.com/watch?v=jXcPUyanY8o>

Deploying Software Updates with Configuration Manager 2012

<http://www.youtube.com/watch?v=4XmN189nbMc>

Tredje parts understøttelse i WSUS

<http://www.youtube.com/watch?v=fyEGWSFWyy0>

11. Bilag:
