



Scriptie | Onderzoek Deployit

Een onderzoek naar en de invoering van de tool Deployit bij de afdelingen ADM en B&E van Rabobank Nederland, Groep ICT

R. van Helten

Scriptie

Onderzoek Deployit

Een onderzoek naar en de invoering van de tool Deployit bij de afdelingen ADM en B&E van Rabobank Nederland, Groep ICT

Auteur	R. van Helten
Versie	1.0
Datum	22 mei 2013
Student	R. van Helten
Studentnummer	1545097
Opleiding	Business IT & Management
Onderwijsinstelling	Hogeschool Utrecht
Docentbegeleider:	K. van Wensen
Stage	Afstudeerstage
Periode	4 februari tot en met 28 mei 2013
Afstudeerorganisatie	Rabobank Nederland, Groep ICT
Locatie	Utrecht, Croeselaan 18
Afdeling	ADM, PDS
Bedrijfsbegeleider:	W. Leeuwis
Opdrachtgever:	W. Leeuwis

Managementsamenvatting

In 2012 is de tool Deployit, van de leverancier Xebialabs, aangeschaft en geïmplementeerd door de afdeling Infra Service, van Rabobank Nederland Groep ICT, voor het programma SEPA. Het programma SEPA heeft geen uitvoerig onderzoek gedaan naar verdere mogelijkheden van de tool. De kracht van Deployit is het automatisch verplaatsen, oftewel deployen, van een Java-applicatie met bijhorende configuratiebestanden en databasescripts naar een geselecteerde omgeving. Hoe Deployit generiek ingezet kan worden bij de Java-applicaties van de Rabobank was na de implementatie voor SEPA onduidelijk. Er is daarom onderzoek gedaan naar het gebruik van de tool Deployit in de ontwikkel- en beheerprocessen van Java-applicaties zodat de doorlooptijd van het deploymentproces gereduceerd wordt. Dit impliceerde een inventarisatie en analyse van de ontwikkel- en beheerprocessen en het opstellen van voorstellen voor de inpassing van Deployit in die processen.

Uit het onderzoek is gebleken dat de deployments die de meeste tijd innemen degene zijn naar de UAT-, PAT- en P-omgeving. Deze benodigde tijd heeft voornamelijk te maken met het ontvangen en moeten begrijpen van (onvolledige) installatiehandleidingen, niet aanwezige koppelingen en het moeten schrijven van een draaiboek. De problemen rondom de manuals, de aanwezige koppelingen en het draaiboek worden opgelost door de tool. Om de deploymenttijd te reduceren moet de tool vanaf de ontwikkeling tot en met het in productie nemen van een applicatie worden gebruikt. Daardoor worden fouten in de configuratie van de tool in een vroeg stadium (lees: ontwikkeling) gevonden in plaats van tijdens het in productie nemen van de applicatie. Dit betekent dat de ontwikkelaars, testers, functioneel applicatie beheerders en technisch applicatie beheerders gebruik maken van de tool. De werkzaamheden voor deze rollen krijgen in het kader van de deployment van een applicatie een monitorend karakter. Daarnaast biedt een koppeling met het aanwezige continuous integration systeem de mogelijkheid om de doorlooptijd van het deploymentproces verder te reduceren.

Voor de invoering van Deployit zijn drie situaties (A, B en C) beschreven waarbij situatie B een uitbreiding is op situatie A en situatie C een uitbreiding is op situatie B. Situatie A is de eenvoudigste implementatie waarbij de deployment van de applicatie geautomatiseerd is. Het aanpassen van o.a. de database moet handmatig uitgevoerd worden. Situatie B beschrijft de mogelijkheid tot een koppeling met het aanwezige continuous integration systeem waardoor de Java-code na de opslag in het versiebeheersysteem (commit) automatisch gecontroleerd, getest, gebuild en vervolgens gedeployed wordt. Situatie C beschrijft de mogelijkheden van een geautomatiseerde deployment pipeline en de toepassing van de Kanban methode.

De aanbeveling is om de tool te implementeren indien het ook technisch haalbaar is. Of het technisch haalbaar is wordt onderzocht binnen de pilot Deployit BBS van project ARA. Het project ARA heeft, op het moment van schrijven, nog geen aanbeveling gedaan. Indien het technisch haalbaar is, en op dit moment is er geen reden daaraan te twifelen, is er de aanbeveling om de situaties stapsgewijs te implementeren beginnend bij situatie A, daarna situatie B en vervolgens situatie C. De implementatie van de situaties vraagt om een projectmatige aanpak. Omdat het project ARA al opgezet is, is het voor de hand liggend dat dit project de implementatie van alle situaties uitvoert. De verwachting is dat situatie C in het jaar 2016 geïmplementeerd is.

Voorwoord

In een tijdsbestek van vier maanden heb ik een onderzoek uitgevoerd bij Rabobank Nederland, Groep ICT. Het onderzoek betrof de inzet van de tool Deployit in de ontwikkel- en beheerprocessen van Java-applicaties. De Rabobank is altijd een toonaangevende organisatie geweest voor mij en dit beeld is tijdens het afstudeeronderzoek ongewijzigd gebleven. Ik ben trots op hetgeen wat ik heb bereikt in deze maanden en ik heb veel geleerd van de medewerkers van de Rabobank. Ik wil daarom iedereen bedanken die heeft bijgedragen aan mijn onderzoek.

In het bijzonder wil ik Wil Leeuwis bedanken voor zijn professionele en intensieve begeleiding tijdens het onderzoek. Dankzij zijn inzichten en feedback ben ik van mening dat ik veel heb bereikt. Naast de bedrijfsbegeleider wil ik ook mijn docentbegeleider Koos van Wensen bedanken. De docentbegeleider heeft mij prima begeleid en gaf altijd spoedig en waardevolle feedback.

Als laatste wil ik de medewerkers van het project ARA en de leden van de kerngroep Deployit ADM bedanken omdat ze me opgenomen hebben als volwaardig lid en vanwege de hulp die ze mij geboden hebben.

Ik wens u veel plezier met het lezen van mijn scriptie.

Roan van Helten
Utrecht, 22 mei 2013

Inhoud

Managementsamenvatting	3
Voorwoord	4
1 Inleiding	7
2 Rabobank Groep	8
2.1 Groep ICT	8
3 Opdrachtschrijving	11
3.1 Aanleiding	11
3.2 Probleemstelling	11
3.3 Doelstellingen	13
4 Onderzoekontwerp	14
4.1 De onderzoeker en adviseur	14
4.2 Populatie	14
4.3 Betrouwbaarheid, validiteit en bruikbaarheid	15
4.4 Dataverzameling	17
4.5 Ontwerp- en analysemethoden	18
5 Huidige situatie	20
5.1 De applicatie BBS	20
5.2 Applicatie en koppelingen	20
5.3 Systemen en hulpmiddelen	21
5.4 Omgevingen	21
5.5 Rollen	22
5.6 Processen	23
5.7 Doorlooptijden	30
5.8 Beveiliging en functiescheiding	30
5.9 BBS specifiek	31
6 SWOT analyse	32
6.1 Toelichting model	33
7 Gewenste situaties	34
7.1 Situatie A	34
7.2 Situatie B	37

7.3	Situatie C.....	40
7.4	Weerstand	41
7.5	Configuratiemanagement	42
7.6	Application Lifecycle Management	42
7.7	Kosten/baten analyse.....	43
7.8	Vervolgplanning.....	43
8	Conclusie.....	44
9	Aanbeveling	45
10	Discussie	46
11	Glossarium	47
12	Literatuurlijst	48
	Bijlagen.....	51
	Bijlage A: Plan van Aanpak	51
	Bijlage B: Evaluatie van eigen functioneren	84
	Bijlage C: Organogram ADM.....	87
	Bijlage D: Theoretisch kader.....	88
	Bijlage E: Deployit beschrijving	96
	Bijlage F: Stakeholderanalyse.....	100
	Bijlage G: Interviewopzet	102
	Bijlage H: BPMN 2.0.....	106
	Bijlage I: Contextplaat BBS	109
	Bijlage J: Rolbeschrijvingen	110
	Bijlage K: Benodigde resources	113
	Bijlage L: Doorlooptijden	115
	Bijlage M: Wensen, verwachtingen en aandachtspunten.....	117
	Bijlage N: Gewenste situatie A	119
	Bijlage O: Gewenste situatie B	122

1 Inleiding

In 2012 is de tool Deployit, van de leverancier Xebialabs, aangeschaft en geïmplementeerd door de afdeling Infra Service, van Rabobank Nederland Groep ICT, voor het programma SEPA. Het programma SEPA heeft geen uitvoerig onderzoek gedaan naar verdere mogelijkheden van de tool. De kracht van Deployit is het automatisch deployen van een Java-applicatie met bijhorende configuratiebestanden en databasescripts naar een geselecteerde omgeving. Hoe Deployit generiek ingezet kan worden bij de Java-applicaties van de Rabobank was na de implementatie voor SEPA onduidelijk. Deze scriptie beschrijft het onderzoek dat gedaan is naar het gebruik van de tool Deployit in de ontwikkel- en beheerprocessen van Java-applicaties zodat de doorlooptijd van het deploymentproces gereduceerd wordt.

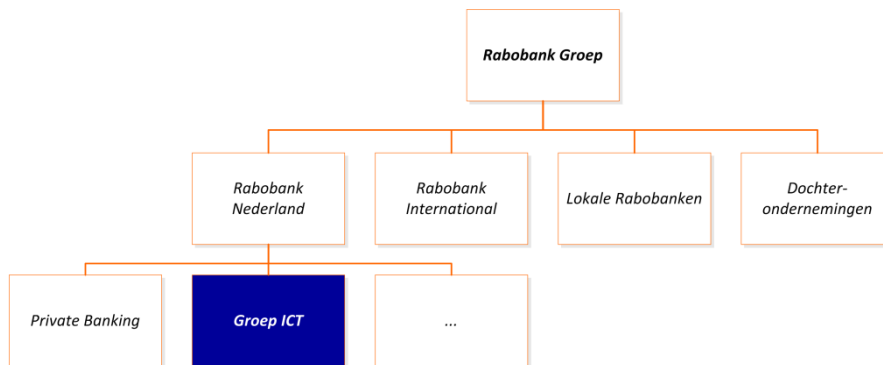
De scriptie is opgedeeld in twaalf hoofdstukken waarvan de inleiding het eerste hoofdstuk is. Het tweede hoofdstuk beschrijft de organisatie waarin het onderzoek werd uitgevoerd, namelijk de Rabobank Groep. Hoofdstuk 2 beschrijft de opdrachtschrijving bestaande uit de aanleiding van het onderzoek, de probleemstelling en de doelstellingen. Hoofdstuk 3 gaat in op het onderzoekontwerp. Hoofdstuk 4 beschrijft de huidige situatie van de ontwikkel- en beheerprocessen van Java-applicaties en enkele daarmee samenhangende onderwerpen. Hoofdstuk 5 beschrijft de SWOT-analyse van de huidige situatie. Hoofdstuk 6 beschrijft de gewenste situaties, namelijk hoe Deployit ingezet kan worden. Hoofdstuk 8 en 9 bevatten de conclusie en de aanbevelingen. Hoofdstuk 10 beschrijft de discussie van de onderzoeksresultaten. Hoofdstuk 11 is het glossarium, een nadere toelichting op enkele begrippen die gebruikt zijn in de scriptie. Tot slot bevat hoofdstuk 12 de literatuurlijst.

Voor de opmaak van het document is de literatuur Leren Communiceren geraadpleegd¹.

¹ Michaël Steenhouders, 2006

2 Rabobank Groep

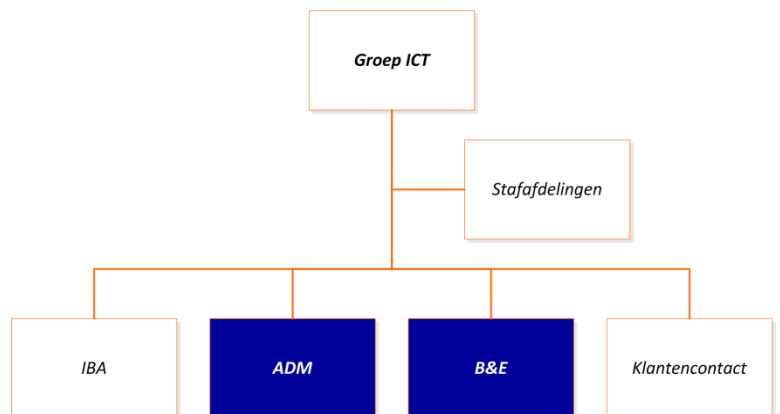
De Rabobank Groep is actief in 47 landen en bestaat uit de lokale Rabobanken, Rabobank Nederland, Rabobank International en de dochterondernemingen². De lokale Rabobanken vormen het coöperatieve kernbedrijf van de Rabobank Groep en zijn lid en aandeelhouder van de overkoepelende coöperatie Rabobank Nederland. Rabobank Nederland ondersteunt de lokale Rabobanken, waaronder op het gebied van ICT (Groep ICT). Zie *Figuur 1 - Organogram Rabobank Groep* voor een visuele weergave.



Figuur 1 - Organogram Rabobank Groep

2.1 Groep ICT

Groep ICT (GICT) bestaat in totaal uit ongeveer 3.000 medewerkers en ondersteunt de Rabobank Groep met ICT-diensten waarbij continuïteit en effectiviteit centraal staan³. GICT kent in totaal vier afdelingen, namelijk ICT Beleid & Architectuur (IBA), Application Development & Maintenance (ADM), Beheer & Exploitatie (B&E) en Klantencontact. GICT heeft drie stafafdelingen, namelijk HR, Control en Communicatie. Zie voor het organogram, *Figuur 2 - Organogram Groep ICT*. De twee afdelingen van GICT die voor deze afstudeeropdracht met name van belang zijn, zijn ADM en B&E.



Figuur 2 - Organogram Groep ICT

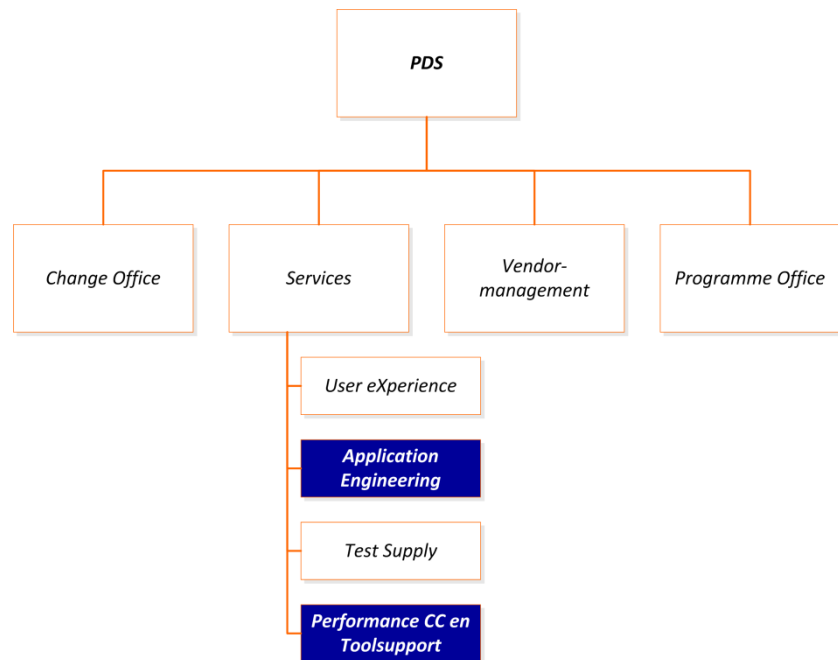
² (Rabobank, Rabobank Groep)

³ (Rabobank, Groep ICT)

2.1.1 Application Development & Maintenance

Application Development & Maintenance ontwikkelt en onderhoudt alle business applicaties voor de bancaire dienstverlening van Rabobank Nederland. ADM heeft een op de businessafdelingen van de Rabobank gebaseerde, indeling wat portfolio's worden genoemd. Zo is er een portfolio Betalen & Sparen, een portfolio Financieringen, enz. ADM kent ook een centrale ondersteunende afdeling, Professional Development Support (PDS). Alle ontwikkel- en onderhoudswerkzaamheden vinden in de portfolio's plaats, PDS ondersteunt de portfolio's. Zie *Bijlage C: Organogram ADM* voor het organogram van ADM.

PDS kent vier afdelingen. Vendor Management organiseert de inkoop voor ADM, waarbij de nadruk ligt op de inhuur van externe medewerkers; voor een belangrijk deel via omvangrijke outsourcingcontracten. De afdeling Programme Office verzorgt de portfolio overstijgende administratie en rapportages. De afdeling Change Office begeleidt veranderprocessen binnen ADM. De afdeling Services ondersteunt het ontwikkelproces. De afdeling Services draagt onder meer zorg voor de ontwikkelomgevingen zoals deze binnen ADM worden gebruikt. Het zijn met name de afdelingen Application Engineering en Tool Support die hierin een rol vervullen, en relevant zijn voor het afstudeeronderzoek. Zie *Figuur 3 - Organogram PDS* voor het organogram van ADM.

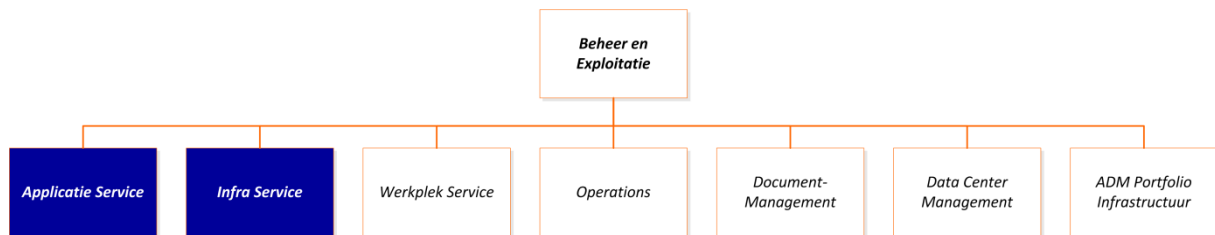


Figuur 3 - Organogram PDS

Application engineers (AE's) zijn binnen ADM verantwoordelijke voor de technische applicatiearchitectuur, en deskundig op het gebied van ontwikkelomgevingen. AE's zijn dan ook degenen die de ontwikkelomgevingen definiëren en, wanneer deze eenmaal operationeel zijn, actueel houden. De afdeling Tool Support verzorgt het technisch beheer van de ontwikkelomgevingen. Om een indruk te geven: er wordt onderscheid gemaakt in circa tien ontwikkelomgevingen (onder meer: Java, .Net, Cobol z/OS, Cobol NonStop, Siebel, AppBuilder, SAP), die in totaal bestaan uit honderden afzonderlijke tools.

2.1.2 Beheer en Exploitatie

Beheer en Exploitatie maakt de business applicaties voor gebruik beschikbaar aan de centrale afdelingen van Rabobank Nederland en aan de door heel Nederland verspreide lokale Rabobanken, en ondersteunt dat gebruik. B&E bestaat uit zeven afdelingen, namelijk Applicatie Service, Infra Service, Werkplek Service, Operations, Document Management, Data Center Management en ADM Portfolio Infrastructuur. De afdelingen Applicatie Service en Infra Service zijn relevant voor dit afstudeeronderzoek. Zie *Figuur 4 - Organogram Beheer en Exploitatie* voor het organogram van B&E.



Figuur 4 - Organogram Beheer en Exploitatie

Applicatie Service (AS) beheert de portfolio applicatie. AS draagt bij aan de ambitie van Groep ICT en B&E door het leveren van Applicatief Beheer conform de afspraken met opdrachtgevers en de verwachtingen van afnemers, tegen beter dan marktconforme inspanning⁴. Infra Service (IS) verzorgt de infrastructuur bij een applicatie en levert en beheert de infrastructuur omgeving. Infra Service levert onder andere Windows, z/Os en Linux omgevingen inclusief de benodigde netwerk-, storage- en middleware infrastructuur⁵.

2.1.3 Strategie – The best IT enabled bank

Groep ICT heeft een strategie ontwikkeld om daarmee 'the best IT enabled bank' te worden. Deze strategie is afkomstig uit het strategieplan van 2010⁶. In dit strategieplan staat beschreven wat GICT in drie jaar tijd gaat doen om 'the best IT enabled bank' te worden, oftewel:

"We willen de bank worden die ICT het slimst toepast voor haar klanten: 'The best IT enabled bank'"

Gericht op het strategieplan komen er drie doelen naar voren die belangrijk zijn voor de afstudeeropdracht, namelijk:

- Systemen dienen stabiel te zijn;
- Kosten van ontwikkeling en beheer verlagen;
- Time to market verkorten.

⁴ (Rabobank, Applicatie Service)

⁵ (Rabobank, Infra Service)

⁶ (Rabobank Groep ICT, 2009)

3 Opdrachtomschrijving

Dit hoofdstuk beschrijft de aanleiding, probleemstelling en doelstellingen van het afstudeeronderzoek. Deze onderwerpen zijn ten aanzien van het Plan van Aanpak deels aangescherpt.

3.1 Aanleiding

De tool Deployit van de leverancier Xebialabs, zie *Bijlage E: Deployit beschrijving*, realiseert een geautomatiseerde deployment van een applicatie naar verschillende OTAP omgevingen. OTAP staat voor Ontwikkeling, Test, Acceptatie en Productie. Infra Service heeft Deployit aangeschaft voor het programma SEPA. Het doel van SEPA, wat staat voor Single Euro Payments Area, is om in Europa één 'betaalruimte' te creëren waarin betalingen op vergelijkbare wijze functioneren⁷. Het doel van het programma is het implementeren van de benodigde SEPA betaalproducten met ondersteuning van de benodigde infrastructuur en systemen. De aangeschafte licentie van Deployit is voor alle afdelingen binnen Rabobank Nederland geldig. Eind 2012 is een kerngroep binnen ADM opgezet die Deployit heeft opgepakt ter onderzoek. Deze kerngroep stelt Deployit ter discussie met daarbij het onderzoek naar de inpassing van Deployit voor alle Java-applicaties van de Rabobank, met als uiteindelijk doel een generieke implementatie van Deployit binnen ADM.

3.2 Probleemstelling

Deployit ondersteunt het proces dat de brug vormt tussen development (ADM) en operations (B&E). Het is van groot belang dat het gebruik van Deployit in beide afdelingen goed op elkaar is afgestemd, en dat het life cycle management van de tool rekening houdt met de eisen en wensen van beide afdelingen. De kern van de opdracht is de inpassing van Deployit in de processen van ADM, op een wijze die leidt tot een soepele aansluiting op de processen van B&E. Dit impliceert een inventarisatie en analyse van processen (m.n. development en het ondersteunende proces change & configuration management) en het opstellen van voorstellen voor de inpassing van Deployit in die processen.

3.2.1 Centrale onderzoeksvraag

De centrale onderzoeksvraag van het afstudeeronderzoek luidt als volgt:

Hoe kan de tool Deployit ingezet worden in de ontwikkel- en beheerprocessen van Java-applicaties binnen de afdelingen ADM en B&E zodat de doorlooptijd van het deploymentproces gereduceerd wordt?

Deployit	Een tool om een deployment van een applicatie te automatiseren naar zowel de ontwikkelomgeving, testomgeving, acceptatieomgeving en productieomgeving.
Ontwikkel- en beheerprocessen	De processen die bestaan uit de ontwikkeling, het testen, de acceptatie en deployment van Java-applicaties.
Java-applicaties	Applicaties die zijn geprogrammeerd in de programmeertaal Java.
ADM	Application Development & Maintenance (ADM) ontwikkelt en onderhoudt alle business applicaties voor de bancaire dienstverlening van Rabobank Nederland.

⁷ (Wikipedia, SEPA)

B&E	Beheer en Exploitatie (B&E) maakt de business applicaties voor gebruik beschikbaar voor de centrale afdelingen van Rabobank Nederland en aan de door heel Nederland verspreide lokale Rabobanken, en ondersteunt dat gebruik.
Deploymentproces	Het proces dat bestaat uit alle activiteiten die het mogelijk maken om een applicatie klaar voor gebruik te maken ⁸ . De deployment betekent het verplaatsen van een applicatie naar een gewenste omgeving.

3.2.2 Deelvragen

De deelvragen zijn onderverdeeld in drie categorieën namelijk huidige situatie, gewenste situatie en vervolgprojecten. Bij iedere deelvraag is aangegeven welke paragraaf antwoord geeft.

Huidige situatie

- Hoe zijn de organisatorische aspecten vormgegeven voor de ontwikkeling en beheer van een Java-applicatie in de huidige situatie? *(Zie paragraaf 5 Huidige situatie)*
- Hoe zijn de processen vormgegeven voor de ontwikkeling en beheer van een Java-applicatie in de huidige situatie? *(Zie paragraaf 5.6 Changemanagement Processen)*
- Hoe is de infrastructuur vormgegeven voor de ontwikkeling en beheer van een Java-applicatie in de huidige situatie? *(Zie paragraaf 5.3 Systemen en hulpmiddelen)*
- Hoe wordt Application Lifecycle Management (ALM) toegepast bij de Java-applicaties in de huidige situatie? *(Zie paragraaf 5.6.5 Ondersteuning)*
- Hoe wordt configuratiemanagement toegepast bij de Java-applicaties? *(Zie paragraaf 5.6.5 Ondersteuning)*

Gewenste situatie

- Welke gewenste situaties zijn mogelijk gezien de functionaliteiten van Deployit? *(Zie paragraaf 7 Gewenste situaties)*
 - o Wat zijn/is de veranderingen/impact van de gewenste situaties ten opzichte van de huidige situatie?
 - o Wat zijn de risico's van de gewenste situaties?
 - o Wat zijn de kosten en baten van de gewenste situaties?
- Hoe kan Application Lifecycle Management (ALM) voor Deployit ingericht worden rekening houdend met de wensen en eisen van de afdelingen B&E en ADM? *(Zie paragraaf 7.6 Application Lifecycle Management)*

Vervolgprojecten

- Welke projecten en/of werkzaamheden moeten uitgevoerd te worden als vervolg van dit onderzoek? *(Zie paragraaf 7.8 Vervolgplanning)*

⁸ (Wikipedia, Software deployment)

3.3 Doelstellingen

De resultaten van het onderzoek creëren duidelijkheid en inzicht in de huidige situatie van de ontwikkeling en beheer van Java-applicaties. Vanuit deze huidige situatie worden meerdere scenario's beschreven die Deployit plaatsen. De scenario's geven de aandachtspunten weer voor de implementatie van de tool. De verschillende scenario's geven meerdere opties weer voor de implementatie met daarbij per scenario een impactanalyse en een risicoanalyse. Vanuit de huidige situatie en potentiële gewenste situaties wordt er een onderbouwd advies opgesteld dat aangeeft welke gewenste situatie de beste keuze is en of de implementatie van de tool wel of niet aan te raden is. Het advies wordt gebruikt door de afdelingen B&E en ADM. Deze afdelingen verzorgen de implementatie van Deployit, rekening houdend met de conclusies van dit onderzoek.

Concreet zijn dit de volgende doelstellingen:

- Voor 28 mei 2013 heeft de student de huidige situatie van de ontwikkel- en beheeromgeving voor Java-applicaties op organisatie- en procesniveau in kaart gebracht.
- Voor 28 mei 2013 heeft de student minimaal drie gewenste situaties beschreven met daarbij voor iedere situatie de aangegeven veranderingen, impact en een risicoanalyse.
- Voor 28 mei 2013 heeft de student een advies opgesteld voor de keuze van een gewenste situatie.

4 Onderzoekontwerp

Voor het onderzoekontwerp is er gebruik gemaakt van de literatuur 'Wat is onderzoek?'⁹, 'Praktijkonderzoek voor bachelors'¹⁰ en 'Methoden & Technieken van Onderzoek'¹¹. Het onderzoek is op hoofdlijnen een interpretatief onderzoek, wat betekent dat het onderzoek kwalitatief is van karakter. Doordat er ook gebruik gemaakt wordt van bronnenstudie (literatuuronderzoek) is er een triangulaire onderzoeksmethode gehanteerd.

4.1 De onderzoeker en adviseur

Doordat het onderzoek kwalitatief is van karakter, is er een open en flexibele houding vereist van de onderzoeker¹². De onderzoeker geeft een advies aan de Rabobank waardoor de onderzoeker ook een adviseur¹³ is. De rolopvatting van de adviseur is een marktgerichte en productgerichte oriëntatie. Dit betekent dat de adviseur voornamelijk kijkt naar de behoeften van opdrachtgevers en klanten over het in kaart gebrachte huidige en gewenste procesmodel.

4.2 Populatie

De populatie van het onderzoek zijn alle Java-applicaties op het Linux platform. Omdat er ongeveer 200 Java-applicaties aanwezig zijn binnen de Rabobank is het onmogelijk alle applicaties te onderzoeken binnen de gestelde afstudeerperiode van vier maanden. Met die gedachte is er een steekproef geselecteerd. De geselecteerde steekproef is de ontwikkeling en beheer van de Java-applicatie BBS en bijhorende satelliet-applicaties van portfolio Financiering Bedrijven. BBS staat voor Bedrijven Bediening Systeem en ondersteunt het proces om bedrijven te voorzien van kredieten.

Het uitgangspunt van het onderzoek is om te streven naar een zo generiek mogelijk eindproduct. Dit betekent dat het eindproduct van toepassing is bij iedere Java-applicatie. In principe moet een steekproef representatief zijn, voldoende groot zijn en dient, voor de gelijkwaardigheid, random-select gekozen te worden¹⁴. De Java-applicatie BBS voldoet niet aan alle eisen. De applicatie is niet representatief voor de andere Java-applicaties. Er zijn meerdere mogelijkheden voor de ontwikkeling en beheer van een Java-applicatie binnen de Rabobank waardoor geen Java-applicatie voldoet aan deze eis. BBS is niet random-select gekozen. Het project ARA, Application Release Automation, heeft deze applicatie gekozen als scope. Het project ARA heeft gekozen voor deze scope omdat de medewerkers rondom de ontwikkeling en beheer van BBS ambitieus zijn om te gaan werken met Deployit en de benodigde resources beschikbaar stellen. Het afstudeeronderzoek is aangehaakt bij het project ARA wat heeft geleid tot een win-win situatie. Deze win-win situatie resulteert in procesmodellen die benodigd zijn voor het project ARA en anderzijds het beschikbaar stellen van resources voor het afstudeeronderzoek. BBS voldoet wel aan een voldoende omvang, er spelen tientallen medewerkers een rol voor de ontwikkeling en beheer van de applicatie. Naast het aantal betrokken medewerkers wordt BBS als een van de uitgebreidere Java-applicaties gezien van de Rabobank.

⁹ (Verhoeven, 2010)

¹⁰ (Mertens, 2010)

¹¹ (Schreuder, 2005)

¹² (Verhoeven, 2010)

¹³ (Nathans, 1999)

¹⁴ (Mertens, 2010)

4.3 Betrouwbaarheid, validiteit en bruikbaarheid

Een onderzoek wordt beoordeeld op de betrouwbaarheid en validiteit van de resultaten¹⁵. Tijdens de uitvoering van het afstudeeronderzoek zijn deze twee onderwerpen een belangrijk onderdeel geweest. Naast de betrouwbaarheid en validiteit speelt ook de bruikbaarheid een rol.

4.3.1 Betrouwbaarheid

De betrouwbaarheid geeft de mate aan waarin het onderzoek vrij is van toevallige fouten. Om de betrouwbaarheid te verhogen zijn de volgende manieren toegepast¹⁶:

Geraadpleegde bronnen

Er zijn verschillende bedrijfsrollen geïnterviewd die ieder een andere kijk op het geheel hadden. Een bedrijfsrol kan namelijk gericht zijn op uitvoerend niveau en managementniveau. De uitvoerende rol is degene die de taak daadwerkelijk uitvoert, de managementrol geeft aan hoe de uitvoering zou moeten verlopen. Naast de uitvoerende- en managementrol is ook de aanwezige documentatie op het intranet onderzocht waarin staat hoe men behoort te werken.

Interbeoordelaarsbetrouwbaarheid¹⁷.

Er zijn meerdere medewerkers geïnterviewd met dezelfde bedrijfsrol, waardoor de overeenstemming tussen de verkregen resultaten bij deze medewerkers resulteerde in betrouwbaardere resultaten. Dit wordt *interbeoordelaarsbetrouwbaarheid* genoemd.

Triangulatie

Met triangulatie wordt het gebruik van meer dan één onderzoeksmethode verstaan. Voor het beantwoorden van de centrale vraag is er gebruik gemaakt van kwalitatief onderzoek en bronnenstudie. De bronnenstudie gaf details aan en richtlijnen voor de verkregen resultaten uit de interviews, een duidelijk voorbeeld hiervan is de toepassing van continuous integration.

Proefinterview

De opgestelde interview is uitvoerig besproken met de bedrijfsbegeleider.

Peer examination

Een stagiair die een onderzoek deed naar de toepassing van Agile binnen dezelfde steekproef bekritiseerde de onderzoeksresultaten en vice versa.

¹⁵ (Verhoeven, 2010)

¹⁶ (Verhoeven, 2010)

¹⁷ (Verhoeven, 2010)

4.3.2 Validiteit

De validiteit geeft de geldigheid aan van het onderzoek, oftewel dat er gemeten is wat we willen meten en dat er geen systematische fouten gemaakt zijn. Om de validiteit van de onderzoeksresultaten te behouden zijn er verschillende manieren toegepast¹⁸.

Bespreking voortgang met opdrachtgever

De resultaten zijn wekelijks met de opdrachtgever en bedrijfsbegeleider besproken, dit om de voortgang weer te geven en daarnaast om, indien nodig, de richting van het onderzoek aan te passen.

Wekelijkse standup project ARA

Vanaf het moment dat project ARA gestart is wordt er wekelijks een standup gehouden. Tijdens deze standup bespreken de projectleden de voortgang van de werkzaamheden. De onderzoeker was tijdens de afstudeerperiode een projectlid van project ARA.

Bespreking voortgang met projectleider ARA

In *Bijlage F: Stakeholderanalyse* zijn de stakeholders van het afstudeeronderzoek beschreven. De opdrachtgever heeft aangegeven dat de resultaten afgestemd moeten zijn met de projectleider van het project ARA. Daarom zijn er gesprekken gevoerd, naast de wekelijkse standups, met de projectleider rondom de voortgang en zijn belangen van/bij het afstudeeronderzoek.

Theoretische basis

De onderzoeker heeft een duidelijke theoretische basis van de problematiek voordat gestart wordt met de interviews waardoor dit de validiteit verhoogt. In *Bijlage D: Theoretisch kader* is de benodigde theoretische kennis beschreven.

Terugkoppeling interviewverslagen

Het vergaren van de benodigde onderzoeksresultaten is o.a. gebeurd aan de hand van interviews. Na ieder interview zijn interviewverslagen geschreven. De gemaakte interviewverslagen zijn teruggekoppeld bij de desbetreffende respondent ter controle. Dit vergroot de validiteit en betrouwbaarheid van de onderzoeksresultaten. De interviewverslagen zijn niet opgenomen in de scriptie. Mocht hier vraag naar zijn dan kunnen de verslagen opgevraagd worden bij de student.

4.3.3 Bruikbaarheid

Het vaststellen van de betrouwbaarheid en validiteit van de onderzoeksresultaten is niet altijd haalbaar¹⁹. Het is daarom van groot belang dat het onderzoek bruikbaar is voor de Rabobank. Het onderzoek is bruikbaar voor het project ARA en de kerngroep Deployit ADM. De projectleider van het project ARA heeft het eindproduct geaccepteerd. Daarvoor is het eindproduct mondeling besproken met verschillende projectleden waaruit is gebleken dat ze tevreden zijn. De kerngroep Deployit ADM gebruikt de resultaten van het onderzoek voor het maken van beslissingen over de generieke implementatie van Deployit. De kerngroep en het project zijn wekelijks op de hoogte gesteld van de resultaten en voortgang van het onderzoek.

¹⁸ (Verhoeven, 2010)

¹⁹ (Verhoeven, 2010)

4.4 Dataverzameling

De gegevens zijn vergaard via literatuuronderzoek en interviews.

4.4.1 Literatuuronderzoek

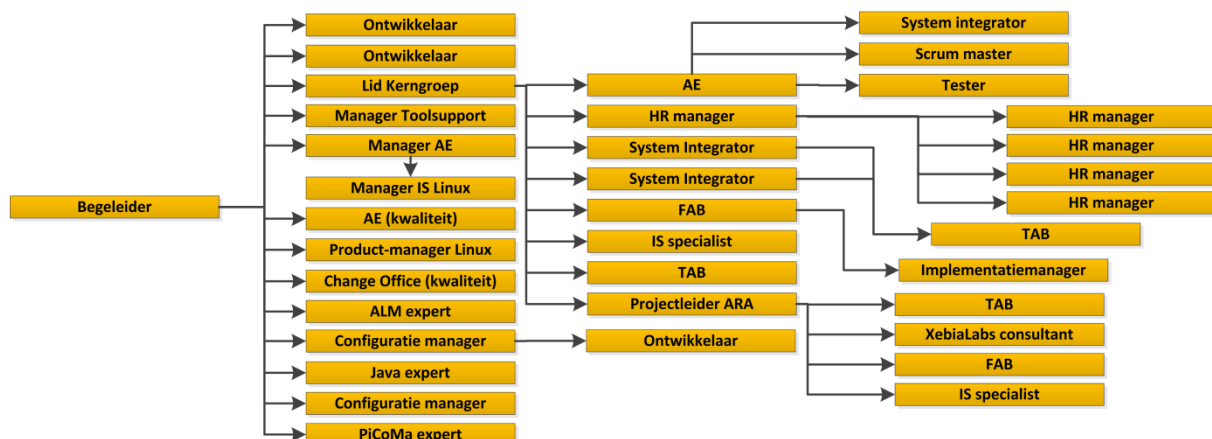
Tijdens het literatuuronderzoek is er literatuur verzameld om kennis te verkrijgen van de ontwikkeling en het in beheer nemen van Java-applicaties. Om gericht te werk te gaan zijn specialisten van de Rabobank benaderd om zoektermen te verkrijgen die gebruikt zijn tijdens het literatuuronderzoek. Onder andere de zoektermen Java, Application Lifecycle Management, DevOps en continuous delivery zijn gebruikt. De literatuur is vergaard bij de specialisten en de vrijgegeven databanken van Hogeschool Utrecht. Voornamelijk de databanken Safari Books Online²⁰ en Gartner²¹ zijn intensief benaderd voor de benodigde theoretische achtergrond. Voor het zoeken naar informatie is er gebruik gemaakt van de Big6™ methode²². Zie *Bijlage D: Theoretisch kader* voor de resultaten van het literatuuronderzoek.

4.4.2 Interviews

Het verzamelen van de benodigde informatie is gebeurd aan de hand van half gestructureerde interviews. Voor het opstellen van het interview zijn topics geselecteerd, bij iedere topic zijn vragen geformuleerd om het interview structuur te geven. Ieder topic en/of vraag biedt de vrijheid om dieper in te gaan op een bepaald onderwerp. Het interviewopzet is gebruikt bij ieder interview. Omdat niet elke respondent kennis had over alle topics zijn tijdens de interviews enkel de relevante topics behandeld. Het interviewopzet is weergegeven in *Bijlage G: Interviewopzet*.

Sneeuwbalmethode

Het vergaren van de benodigde respondenten is gebeurd aan de hand van de sneeuwbalmethode. De sneeuwbalmethode bestaat uit het vragen naar potentiële respondenten binnen de aanwezige kleine kring van respondenten. De reden daarvan is dat de onderzoeker, in de eerste weken van het onderzoek, weinig contactpersonen binnen het onderzoekdomein binnen handbereik had. Zie voor een visuele weergave van de geraadpleegde medewerkers *Figuur 5 - Gespreksnetwerk*. Het figuur visualiseert een duidelijk sneeuwbaaleffect.



Figuur 5 - Gespreksnetwerk

²⁰ www.safaribooksonline.com

²¹ www.gartner.com

²² (Verhoeven, 2010)

4.5 Ontwerp- en analysemethoden

Er zijn verschillende ontwerp- en analysemethoden gebruikt. De resultaten uit het literatuuronderzoek zijn beschreven in het theoretisch kader, zie *Bijlage D: Theoretisch kader*. De resultaten uit het interview zijn gecodeerd aan de hand van categorieën²³. Het coderen van de resultaten is een intensieve bezigheid. Het resultaat van het coderen bestaat uit alle vergaarde resultaten per categorie. De paragrafen van de huidige situatie, verwachtingen, eisen en wensen zijn als categorieën gebruikt. Vanuit de verkregen resultaten per categorie zijn er verschillende ontwerp- en analysemethoden toegepast, deze zijn verwerkt in de volgende paragrafen met uitzondering van de stakeholderanalyse.

4.5.1 Stakeholderanalyse

De belangrijkste stakeholders moeten op de hoogte gesteld zijn van de resultaten en tevreden gehouden te worden. Daarom is er een stakeholderanalyse toegepast om de stakeholders in kaart te brengen. Voor iedere stakeholder zijn de bijhorende belangen, mate van macht, ondersteuning en beïnvloeding in kaart gebracht. Zie *Bijlage F: Stakeholderanalyse*.

4.5.2 Procesanalyse

De huidige- en gewenste situatie(s) zijn beschreven met behulp van procesmodellen. Voor het modelleren van de processen is BPMN2.0 toegepast, zie *Bijlage H: BPMN 2.0* voor een korte uitleg. Het uitgangspunt van de modellen is dat ze leesbaar en duidelijk zijn voor de medewerkers van de Rabobank en niet of er volledig is voldaan aan alle eisen van BPMN2.0. Een aanpassing met betrekking tot de BPMN2.0 is dat er naast de rollen ook systemen weergegeven zijn met activiteiten. Daardoor ontstaat er een duidelijk beeld hoe Deployit ingezet kan worden in combinatie met de aanwezige systemen en rollen van de processen. Voor de procesanalyse zijn ook de huidige doorlooptijden geanalyseerd. De doorlooptijd is de tijd die nodig is om een klantorde of opdracht een proces(keten) te laten doorlopen²⁴.

4.5.3 SWOT analyse

Nadat de huidige situatie in kaart gebracht is dient er een richting gegeven te worden naar gewenste situaties. Daarvoor is een SWOT analyse²⁵ toegepast. De SWOT analyse is uitgewerkt en toegelicht in *paragraaf 6 SWOT analyse*.

4.5.4 Verandermanagement

Een implementatie van een tool zoals Deployit vraagt om verandering van de huidige werkwijze van de medewerkers. Deze verandering kan weerstand opwekken bij verschillende medewerkers. Om de impact van de gewenste situaties te beschrijven is er gebruik gemaakt van verandermanagement²⁶. De nadruk ligt daarbij op het onderwerp weerstand. De komst van Deployit heeft invloed op de huidige activiteiten, bevoegdheden en verantwoordelijkheden van verschillende medewerkers. De gewenste situaties beschrijven de mate van verwachte impact en bijhorende weerstand, zie *paragraaf 7 Gewenste situaties*.

²³ (Mertens, 2010)

²⁴ (Obers & Achterberg, 2012)

²⁵ (Kleijn & Rorink, 2007)

²⁶ (Caluwé, 2006)

4.5.1 Risicoanalyse

Iedere gewenste situatie beschrijft potentiële risico's die zich kunnen voordoen indien de betreffende situatie geïmplementeerd wordt. Zoals Prince²⁷ aangeeft kan een risico het beste worden gedefinieerd als het product van de kans dat een dreiging zich voordoet en de schade ($R = K \times S$). In *paragraaf 7 Gewenste situaties* is per gewenste situatie een risicoanalyse beschreven.

4.5.2 Kosten/baten analyse

De kosten/baten analyse beschrijft per gewenste situatie de bijkomende kosten en baten. De analyse is niet weergegeven in kwantitatieve waarden wat ook niet concreet aangegeven is in het plan van aanpak. Echter is een dergelijke analyse meestal wel kwantitatief van karakter. De reden waarom de kosten en baten in de vorm van onderwerpen zijn beschreven is omdat de getallen moeilijker te vergaren zijn dan verwacht. Dit was niet realistisch met de beschikbare tijd die is aangegeven in de planning van het plan van aanpak. In *paragraaf 7.7 Kosten/baten analyse* is de kosten/baten analyse beschreven.

²⁷ (Janssen, 2007)

5 Huidige situatie

Dit hoofdstuk beschrijft de huidige situatie van de ontwikkeling en het in beheer nemen van BBS.

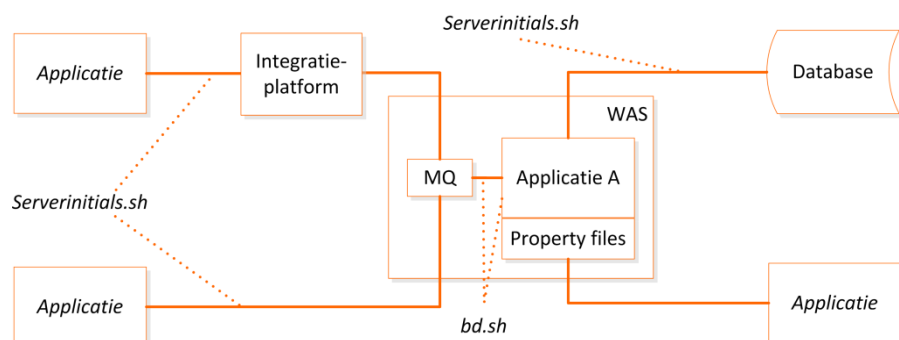
5.1 De applicatie BBS

Het bedrijf en bediening systeem (BBS) is het verkoopsysteem voor het verstrekken van kredieten aan bedrijven. Een bedrijf vraagt een krediet aan waarop een accountmanager informatie vergaart van het bedrijf. Vervolgens stelt de accountmanager een offerte op. Uiteindelijk kan dit leiden tot het verstrekken van een krediet aan het bedrijf. Cognizant en Ordina leveren de ontwikkelaars voor de ontwikkeling van BBS. Cognizant is grotendeels gevestigd in India. BBS is een complexe applicatie daarom is de planning om BBS te ontmantelen in verschillende satellietapplicaties/componenten waarna uiteindelijk BBS wordt vervangen. Om inzicht te geven in de aanwezige applicaties rondom het systeem BBS is in de bijlage een contextplaat weergegeven, zie *paragraaf 14.1 Bijlage A – Contextplaat BBS*.

5.2 Applicatie en koppelingen

Een deployment van een applicatie kan opgesplitst worden in twee onderdelen, namelijk de deployment van de applicatie en de deployment van de benodigde koppelingen, configuratie- en databasebestanden. De focus van dit rapport is de deployment van de applicatie. Omdat een applicatie afhankelijk is van de aanwezige structuur met bijhorende koppelingen, configuratie- en databasebestanden is dit kort toegelicht in deze paragraaf.

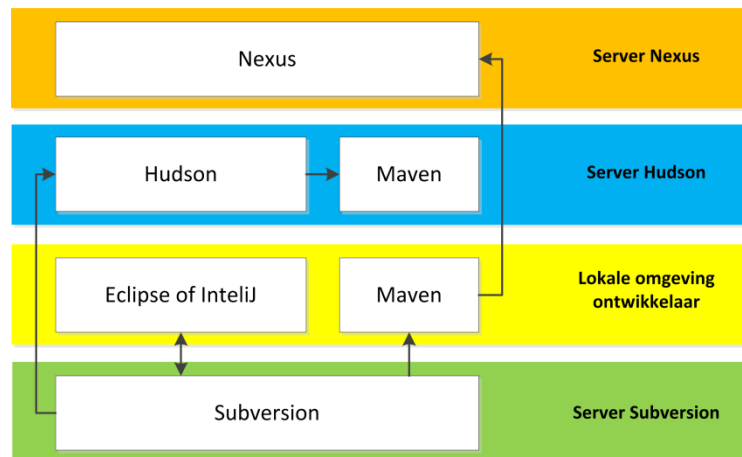
Een Java-applicatie draait onafhankelijk op een Websphere Applicatie Server (WAS). Een applicatie heeft o.a. database-, Message Queue- (MQ) en http(s) koppelingen met andere applicaties en databases. De koppelingen worden in een WAS geconfigureerd. Een MQ-koppeling wordt gelegd vanuit de applicatie naar MQ. MQ heeft een koppeling met de andere applicatie wat in sommige gevallen via een integratieplatform wordt verzorgd. Het integratieplatform beheert voor meerdere applicaties koppelingen. De databasekoppeling legt een koppeling met een interne of externe database. De interne koppelingen zijn beschreven in property files. De property files zijn http(s) koppelingen die verwijzen naar andere applicaties. Om de koppelingen te verzorgen zijn er zelfgemaakte scripts ontwikkeld. Voor het systeem BBS is dit een `serverinitials.sh` en een `bd.sh` bestand (bd staat voor batch deployer). Zie *Figuur 6 - Technische aspecten* voor een visuele weergave van de aanwezige technische aspecten.



Figuur 6 - Technische aspecten

5.3 Systemen en hulpmiddelen

Deze paragraaf beschrijft de systemen en hulpmiddelen die gebruikt worden tijdens de ontwikkeling van een Java-applicatie. In *Figuur 7 - Systemen en hulpmiddelen* zijn de systemen en hulpmiddelen visueel weergegeven per omgeving. Ieder onderdeel van het figuur is kort toegelicht in *Tabel 1 - Systemen en hulpmiddelen*. Voor de toelichting is er gebruik gemaakt van het document 'Java Development Infrastructure, versie 1.7'²⁸



Figuur 7 - Systemen en hulpmiddelen

Naam	Beschrijving
Subversion (SVN)	SVN is een versiebeheersysteem waar de ontwikkelaar zijn source code opslaat.
Hudson	Hudson is een continuous integration systeem. Hudson voert, met een ingestelde tijdsfrequentie, een poll uit op het systeem SVN om aangepaste source code te laten controleren. Indien de source code is aangepast laat het systeem de code testen door Maven, die is gevestigd op de Hudson server. Naast Maven roept Hudson de systemen/groepen Fortify (wekelijks), Sonar (dagelijks) en SIG (Software Improvement Group, wekelijks) aan. Fortify controleert op beveiligingslekken, Sonar voert kwaliteitscontroles uit en SIG voert een uitgebreidere kwaliteitscontrole uit t.o.v. Sonar. De server Hudson heeft geen schrijf-autorisatie op de server Subversion.
Nexus	Nexus is een versiebeheersysteem waar de ontwikkelaar de releases opslaat. Op het moment dat de release in Nexus is geplaatst kan deze niet meer aangepast worden.
Eclipse/Intelij	Eclipse en IntelliJ zijn beiden IDE's waar de ontwikkelaar een keuze uit heeft. IDE staat voor Integrated Development Environment, oftewel de persoonlijke integrale ontwikkelomgeving van de ontwikkelaar. Beide IDE's vragen de source code op uit SVN (uitchecken) en slaan de source code op in SVN (inchecken).
Maven	Maven is een programma dat aanwezig is op de lokale omgeving van de ontwikkelaar en op de server van Hudson. Maven compileert source code, voert (unit) tests uit en maakt builds. Via Maven (op de omgeving van de ontwikkelaar) wordt een release gemaakt en geplaatst op de server Nexus, daarvoor geeft de ontwikkelaar aan welke source code gebruikt moet worden uit SVN. Tijdens het maken van een release wijzigt Maven de versienummer in SVN.

Tabel 1 - Systemen en hulpmiddelen

5.4 Omgevingen

Voor de ontwikkeling en het in beheer nemen van een Java-applicatie is er een ontwikkel- (O), test- (T), acceptatie- (a/A) en productieomgeving (P) nodig om controles en testen uit te kunnen voeren voordat de applicatie in productie gaat. Een omgeving is een fysieke machine met daarbij de benodigde virtuele/digitale inrichting en bijhorende koppelingen met andere applicaties. Naast de testomgeving kan er ook een IRT-omgeving (Integratie Regressie Test) aanwezig zijn. Bij IRT wordt er in de keten getest met de laatste versies van de systemen. Een acceptatie-omgeving is gesplitst in twee omgevingen, namelijk de UAT-omgeving (User Acceptance Test, a) en de PAT-omgeving (Production Acceptance Test, A).

²⁸ (Bosman, 2011)

5.5 Rollen

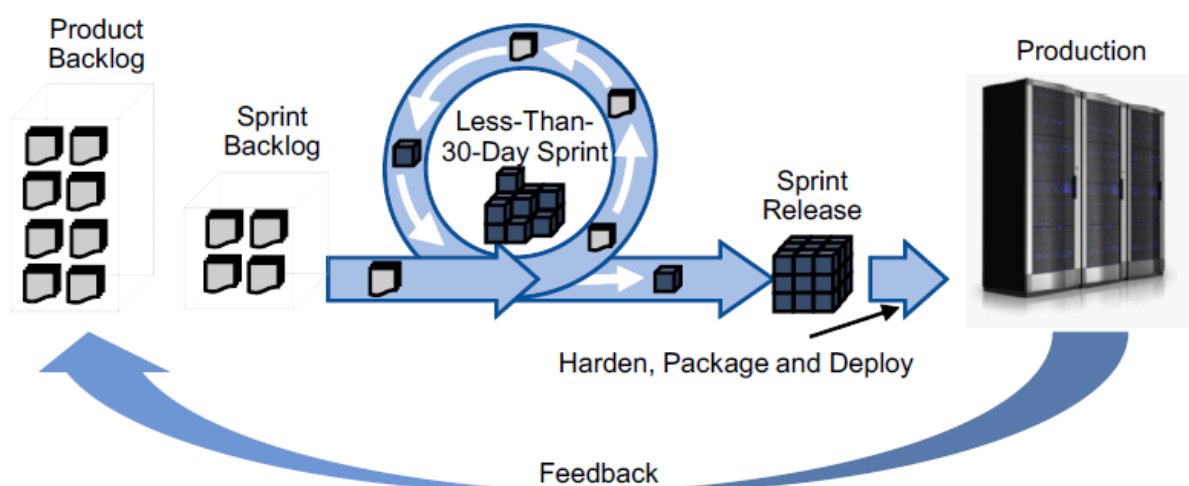
Tijdens het deploymentproces zijn verschillende rollen benodigd met ieder hun eigen taken en verantwoordelijkheden. In tabel *Tabel 2 - Rollen* zijn de benodigde rollen aangegeven en kort toegelicht. Een volledige uitwerking van de aanwezige rollen, met bijhorende verantwoordelijken, is weergegeven in *Bijlage J: Rolbeschrijvingen*.

Naam rol	Toelichting
Application Engineer	Vertalen van de business requirements conform de eisen van beleid en architectuur.
Configuratiemanager	Managen van informatie rondom de ontwikkelversies en releaseversies.
Database administrator	Beheert de databases in de (grote) acceptatieomgeving en de productieomgeving.
Functioneel applicatie beheerder	Het functioneel applicatief inrichten en beheren van een of meerdere complexe geautomatiseerde omgevingen.
Implementatie-manager	Organiseren en vergaren van de juiste resources voor de implementatie van een applicatie.
Infra Service medewerker	Het landen van een applicatie in de aanwezige infrastructuur en de fysieke machine.
Ontwikkelaar	Het ontwikkelen van Java -applicaties.
Product owner	Bepaalt welke werkzaamheden nodig zijn vanuit de business kant.
System Integrator	Verzorgt de benodigde omgevingen.
Technisch applicatie beheerder	Het technisch applicatief inrichten en beheren van een of meerdere complexe geautomatiseerde omgevingen.
Tester	Het testen van Java-applicaties.

Tabel 2 - Rollen

5.5.1 Ontwikkelpject

Het ontwikkelproject werkt volgens de Agile Scrum methode. Een Scrumteam bestaat uit een scrum master, product owner, business analyst, ontwikkelaars en testers. Een Scrumteam voert de, van te voren geselecteerde, taken in een tijdsbestek van twee of drie weken uit wat sprints worden genoemd. Zie *Figuur 8 - Agile-Scrum methode*.²⁹

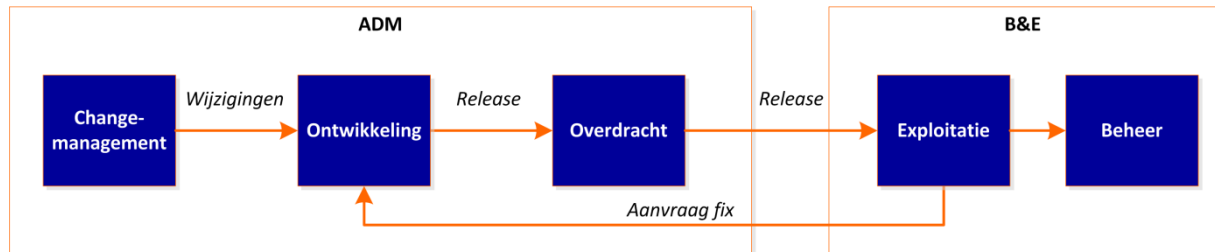


Figuur 8 - Agile-Scrum methode

²⁹ (Gartner, Achieving Continuous Delivery, 2013)

5.6 Processen

In *Figuur 9 - Processen hoog niveau* is op hoog niveau weergegeven welke processen binnen de afdelingen ADM en B&E benodigd zijn voor de ontwikkeling en het in beheer nemen van een Java-applicatie. Per paragraaf is ieder proces toegelicht.



Figuur 9 - Processen hoog niveau

5.6.1 Changemanagement

Het proces changemanagement ontvangt een verzoek om een applicatie te wijzigen (Request for Change (RFC). Dit verzoek kan afkomstig zijn van verschillende partijen. De RFC wordt beoordeeld en de goedgekeurde RFC's worden op de releaselijst geplaatst. Releases worden gevormd door een selectie van verschillende RFC's van de releaselijst. Een release wordt overgedragen aan het proces ontwikkeling. Het proces ontwikkeling bepaalt de benodigde taken om de release te bouwen en verdeelt deze over de scrum teams. De release wordt door één of meerdere scrum teams ontwikkeld. Na de ontwikkeling van de taken vindt er een integratie plaats van de uitgevoerde taken tot een volledig ontwikkelde release.

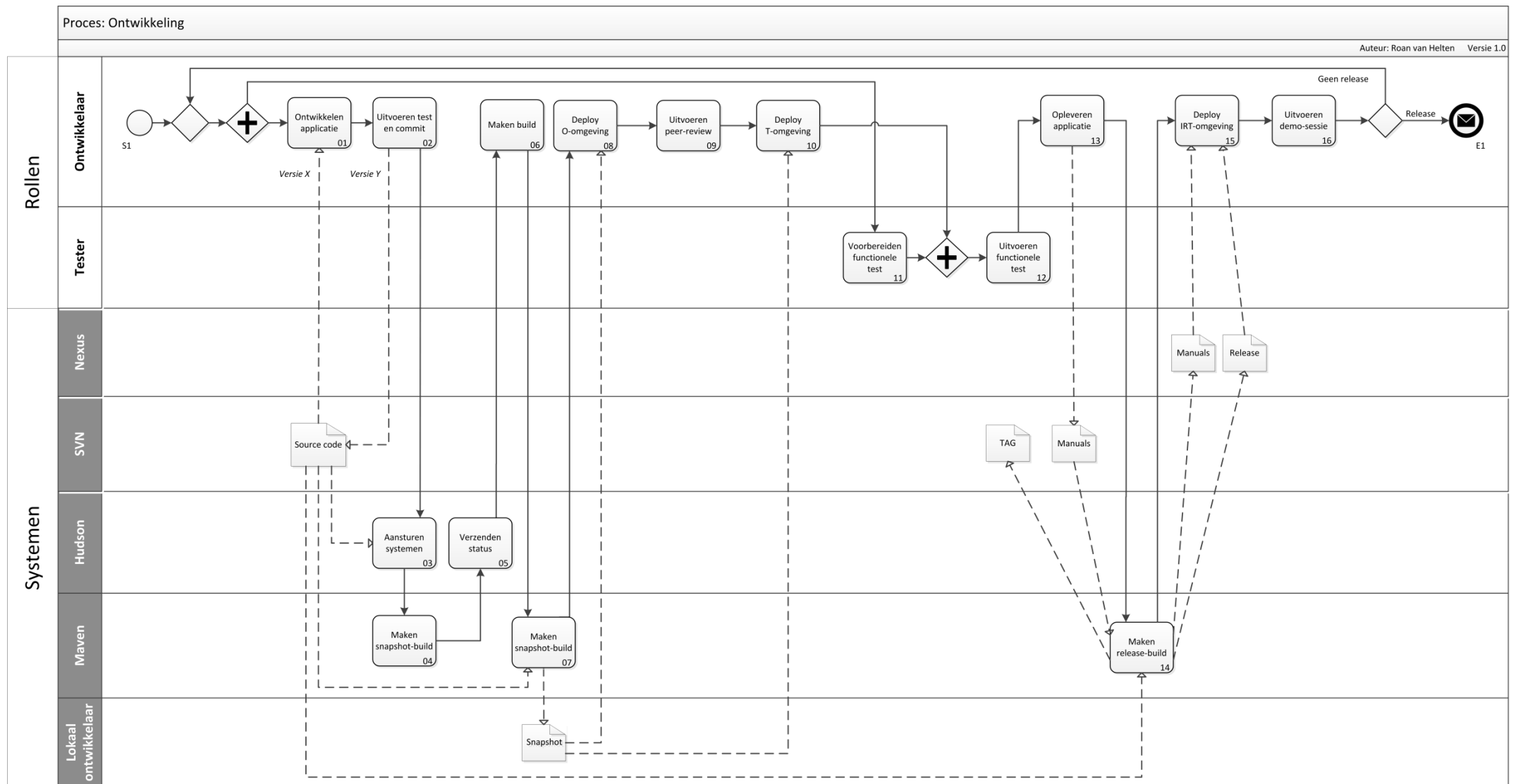
5.6.2 Ontwikkeling

Het proces ontwikkeling behandelt de ontwikkeling en het testen van een (nieuwe wijziging van een) Java-applicatie. In *Figuur 10 - Proces Ontwikkeling* is het proces gevisualiseerd. De nadruk van het model ligt op de deployment van de ontwikkelde applicatie op de omgevingen. De onderdelen van het ontwikkel- en testproces die geen raakvlak hebben met het deploymentproces zijn niet gemodelleerd. Dit betekent onder andere dat enkel de ontwikkelaar en tester van het scrumteam in het proces aanwezig zijn en dat gedetailleerde activiteiten niet benoemd worden. De scenariobeschrijving voor het proces ontwikkeling is beschreven in *Tabel 3 - Ontwikkeling*.

Nr.	Beschrijving
S	Het proces start bij het verkrijgen van een taak die wordt uitgedeeld vanuit het scrumteam. De taak betreft daarbij het programmeren en (vervolgens) functioneel testen van Java-code. Dit kan zowel een nieuwe taak zijn als het realiseren van een fix die is aangevraagd tijdens het proces Exploitatie en Beheer.
01	De ontwikkelaar vernieuwt zijn werkomgeving door de laatste versie ('versie x') van de source code uit het systeem Subversion (SVN) te halen, dit heet uitchecken bij SVN. Vervolgens ontwikkelt de ontwikkelaar de applicatie.
02	De ontwikkelaar test de ontwikkelde code. Daarvoor maakt de ontwikkelaar een build via Maven waarbij, naast het compileren en integreren van de componenten, de verschillende unit tests worden uitgevoerd. De ontwikkelaar gebruikt Maven die op zijn lokale omgeving staat. Als de build succesvol is commit de ontwikkelaar de code in het systeem Subversion (SVN), dit heet inchecken bij SVN. Het systeem Subversion maakt automatisch een nieuwe versie ('versie y') aan van de laatste source code, dit wordt revisie genoemd.
03	Het systeem Hudson voert een poll uit op SVN. Indien er source code is aangepast stuurt Hudson

	Maven aan. Hudson gebruikt Maven die op de server van Hudson staat. Maven maakt een build en stuurt SIG (wekelijks), Fortify (wekelijks) en Sonar (dagelijks) aan.
04	Maven, die tijdens deze activiteit op de server van Hudson staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een snapshot, wordt op de server van Hudson geplaatst.
05	Het resultaat van Maven en, indien aanwezig, van SIG, Fortify en Sonar stuurt Hudson door aan de ontwikkelaar.
06	De ontwikkelaar laat een build maken. Daarvoor stuurt de ontwikkelaar Maven aan die op zijn lokale omgeving staat. De build is hetzelfde resultaat als de build die op de Hudson server staat.
07	Maven, die tijdens deze activiteit op de lokale omgeving van de ontwikkelaar staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een snapshot, wordt op de lokale omgeving van de ontwikkelaar geplaatst.
08	De ontwikkelaar selecteert de snapshot, die gemaakt is door Maven, van zijn lokale omgeving en deployt deze naar de ontwikkelomgeving. Een andere optie is om de build van de Hudson server te gebruiken, daarvoor worden stap 06 en 07 overgeslagen.
09	De ontwikkelaar en/of andere leden van het scrumteam testen/controleren/bespreken de ontwikkelde Java-code.
10	De ontwikkelaar deployt de snapshot op de testomgeving. Daarvoor wordt dezelfde snapshot gebruikt als degene die gebruikt is tijdens de deployment op de ontwikkelomgeving.
11	De tester bereidt zich, vanaf het begin van de start van het proces, voor op het uitvoeren van een functionele test. De voorbereiding is een globale uitwerking van o.a. de testscenario's.
12	Nadat de snapshot gedeployd is op de testomgeving voert de tester functionele testen uit.
13	Nadat de functionele testen succesvol zijn uitgevoerd levert de ontwikkelaar de nieuwe applicatie op. Daarbij plaatst de ontwikkelaar de benodigde manuals in SVN en roept de ontwikkelaar Maven, die op zijn lokale omgeving staat, aan om een build te maken. De manuals bestaan uit een installatievolgorde van de applicatie, install notes en release notes.
14	Maven, die tijdens deze activiteit op de lokale omgeving van de ontwikkelaar staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een release, wordt in Nexus geplaatst samen met de manuals. Maven maakt een tag aan in SVN. Een tag is een etiket met een versienummer op elk bestand die onderdeel is van de release. De tag wordt in een aparte map opgeslagen in SVN zodat de versie waarvan een release gemaakt wordt te herleiden is.
15	De ontwikkelaar deployt de release op een IRT-omgeving. IRT staat voor Integratie Regressie Test. Op de IRT-omgeving worden alle applicaties, die in een keten samen werken, getest. Applicatie X kan bijvoorbeeld een printmogelijkheid bieden die benodigd is voor Applicatie Y.
16	De demo-sessie en een vastgestelde deadline bepalen of de release wel/niet wordt afgeleverd aan de configuratiemanager.
E	Indien is besloten om de release op te leveren aan de configuratiemanager, wordt dit doorgegeven aan de configuratiemanager. De release dient goedgekeurd te zijn door de product owner.

Tabel 3 - Ontwikkeling



Figuur 10 - Proces Ontwikkeling

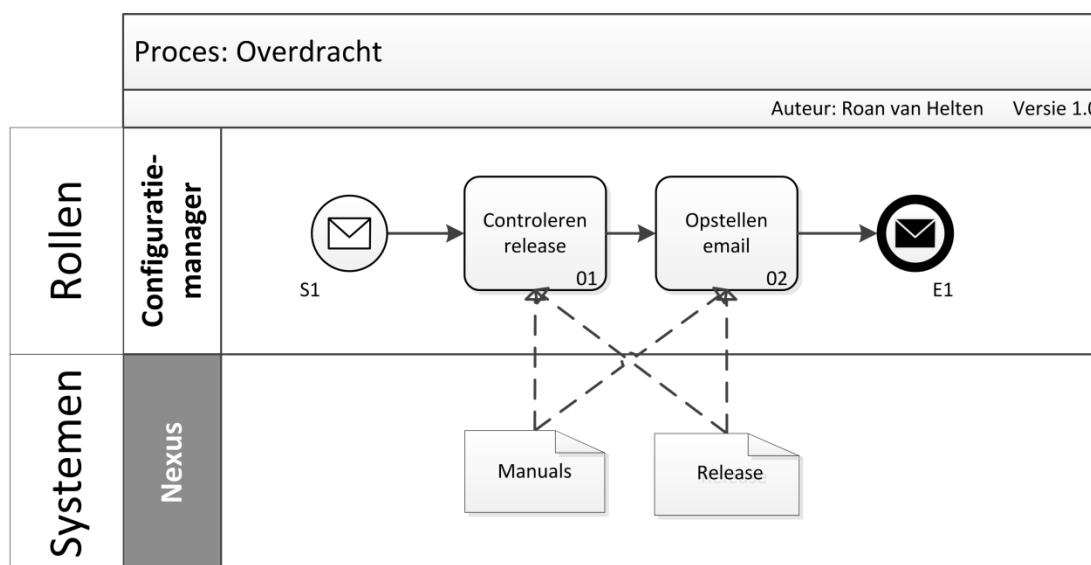
5.6.3 Overdracht

Indien, tijdens het ontwikkel- en testproces, is besloten om de release op te leveren wordt dit doorgegeven aan de configuratiemanager. Dit kan ook worden verstaan als een overdracht gezien het feit dat niet binnen elk project of programma de configuratiemanager deze taak op zich neemt. In *Figuur 11 - Proces Overdracht* is het proces gevisualiseerd.

De beschrijving van het procesmodel is gebeurd aan de hand van een scenariobeschrijving, zie *Tabel 4 - Overdracht*.

Nr.	Beschrijving
S1	Indien tijdens het ontwikkel- en testproces is besloten om de release, met goedkeuring van de product owner, op te leveren aan de configuratiemanager wordt dit doorgegeven aan de betreffende configuratiemanager. Daardoor start het proces van configuratiemanagement.
A1	De configuratiemanager ontvangt een melding vanuit het scrumteam dat een release ontwikkeld is. Vervolgens controleert de configuratiemanager de verkregen releases en manuals op volledigheid en juistheid. De releases en manuals zijn opgeslagen in het systeem Nexus.
A2	Indien de controle succesvol is verlopen stelt de configuratiemanager een email op met daarin o.a. de wijzigingen en de locaties van de manuals en releas(es). De configuratiemanager houdt handmatig, in een Excel-sheet, de versie bij voor integratie indien dit van toepassing is. Het Excel-sheet wordt op de Sharepointpagina van het Scrumteam geplaatst.
E1	De configuratiemanager verzendt de email naar het scrumteam, system integrator(s), functioneel applicatie beheerder(s) en technisch applicatie beheerder(s).

Tabel 4 - Overdracht



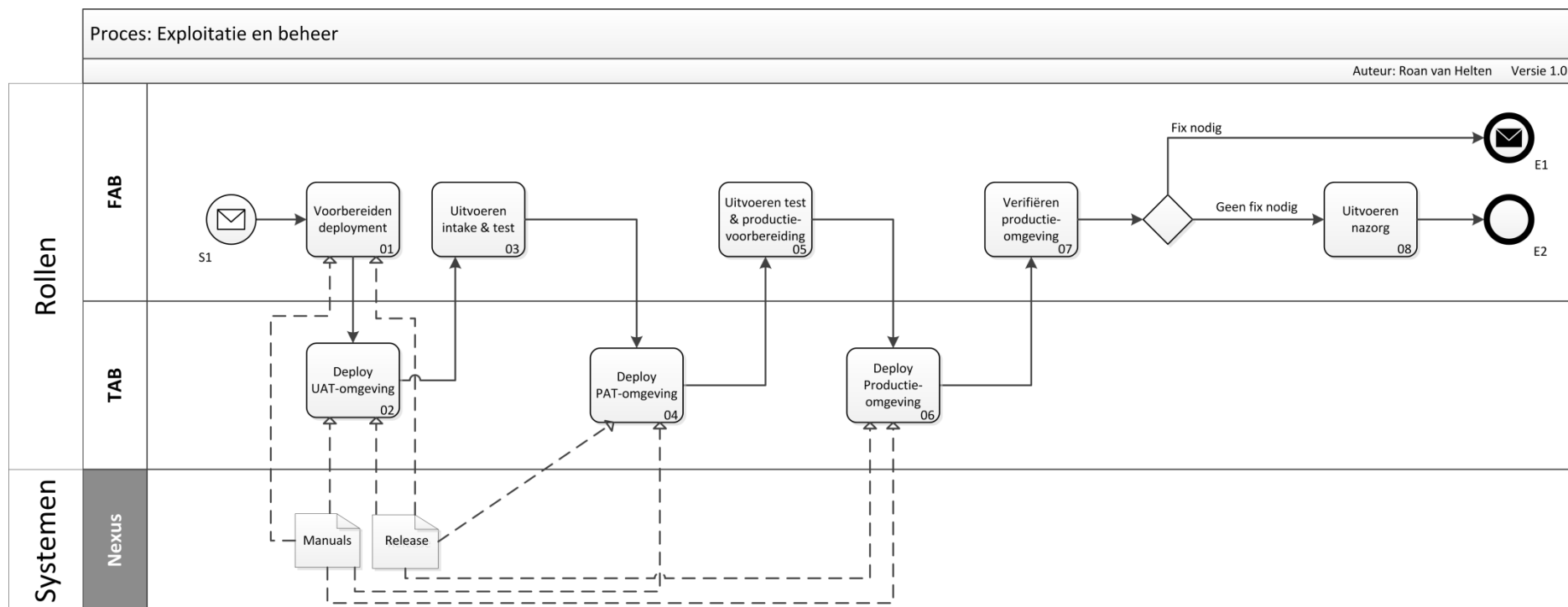
Figuur 11 - Proces Overdracht

5.6.4 Exploitatie en beheer

Het proces exploitatie en beheer behandelt de exploitatie en het beheer van een (nieuwe) Java-applicatie. In *Figuur 12 - Proces Exploitatie en Beheer* is het proces gevisualiseerd. De nadruk van het model ligt op de deployment van de ontwikkelde applicatie op de aanwezige omgevingen. De beschrijving van het procesmodel is gebeurd aan de hand van een scenariobeschrijving, zie *Tabel 5 - Exploitatie en beheer*.

Nr.	Beschrijving
S1	De functioneel applicatie beheerder (FAB) ontvangt van de configuratiemanager een email met de verwijzing vanuit het systeem Nexus naar de betreffende release(s) en bijhorende manual(s).
01	De FAB bereidt de deployment voor door middel van de controle van de verkregen release(s) en manual(s). De controle focust zich voornamelijk op de volledigheid en juistheid van de verkregen bestanden. Na de controle brengt de FAB de technisch applicatie beheerder (TAB) op de hoogte van de benodigde deployment op de UAT-omgeving (kleine acceptatieomgeving). Tijdens deze activiteit worden de benodigde resources al vergaard en ingepland, zie <i>paragraaf 7.4.2 Benodigde resources</i> .
02	De TAB ontvangt van de FAB de melding om te starten met de deployment op de UAT-omgeving. Voor de deployment wordt er gebruik gemaakt van de aanwezige manual(s) en release(s) uit het systeem Nexus. Omdat de ontwikkelaar de manual(s) heeft geschreven en geen kennis heeft van de omgevingen binnen het B&E domein, schrijft de TAB een draaiboek. Het draaiboek geeft aan hoe de betreffende applicatie geïmplementeerd moet worden op technisch niveau.
03	De FAB voert een intake uit op de UAT-omgeving. De intake bestaat uit de controle of de applicatie(s) functioneert naar behoren. Vervolgens wordt er een zogeheten smoke-test en regressietest uitgevoerd. Op de UAT-omgeving wordt ook al getest welke benodigde koppelingen aanwezig zijn en welke aangevraagd dienen te worden.
04	Nadat de FAB succesvol de testen uitgevoerd heeft, start de TAB met de deployment op de PAT-omgeving. Voor de deployment op de PAT-omgeving worden de manual(s) en release(s) gebruikt uit het systeem Nexus.
05	De FAB test de applicatie op functioneel niveau. Daarbij komt kijken dat op de PAT-omgeving een productieomgeving wordt nagespeeld. Na het uitvoeren van de testen wordt de productievoorbereiding uitgevoerd. Tijdens de productievoorbereiding worden de benodigde resources op de hoogte gesteld. In de PAT wordt de fall back bij iedere stap getest. Fall back is de mogelijkheid om de installatie/deployment van de applicatie terug te kunnen draaien.
06	De TAB ontvangt van de FAB de melding om te starten met de deployment op de productieomgeving. Voor de deployment wordt er gebruik gemaakt van de aanwezige manual(s) en release(s) uit het systeem Nexus.
07	De FAB verifieert de productieomgeving. Dit betekent dat er testen worden uitgevoerd op de productieomgeving.
08	De activiteit nazorg wordt gestart indien er geen fix nodig is. Een fix kan echter al op de UAT-omgeving en PAT-omgeving nodig zijn. (<i>Dit is niet gevisualiseerd in het procesmodel omdat dit leidt tot een onleesbaar document bestaande uit vele beslissingsmomenten</i>). De nazorg wordt verleend aan de eindgebruikers en bestaat uit het verlenen van ondersteuning bij de eindgebruiker.
E1	Een fix wordt aangevraagd bij het proces ontwikkeling.
E2	De applicatie is succesvol in productie genomen.

Tabel 5 - Exploitatie en beheer



Figuur 12 - Proces Exploitatie en Beheer

Benodigde resources

Voor het in productie nemen van een Java-applicatie, nadat de release is opgeleverd vanuit het proces ontwikkeling, dienen er meerdere partijen geraadpleegd te worden. In tegenstelling tot een ontwikkelaar, die de database en koppelingen zelf beheert, zijn er voor verschillende koppelingen en de wijziging van databases verschillende partijen nodig. Opgesomd zijn de volgende partijen nodig:

- Infra Service Oracle, Database administrator - *Het aanpassen van de databasestructuur;*
- Infra Service Oracle, IS-koppelingen specialist – *Het plaats van de benodigde koppelingen;*
- Infra Service Linux, Specialist – *Inrichting van de applicatieserver;*
- Rabobank International (RI) – *Plaatsen van externe koppelingen met RI;*
- CRMi – *Plaatsen van interne koppelingen binnen de Rabobank.*

Zie *Bijlage J: Rolbeschrijvingen* voor een gedetailleerde beschrijving van de benodigde resources.

5.6.5 Ondersteuning

Voor de ontwikkeling en beheer van een Java-applicatie zijn er ondersteunende processen aanwezig, namelijk de inrichting van een omgeving, de inrichting van technische koppelingen en de registratie van de ontwikkel- en releaseversies.

Inrichting van een omgeving

Voor de inrichting van een omgeving is een system integrator en een application engineer benodigd. De system integrator vraagt de benodigde onderdelen aan voor het opzetten van de omgeving. De application engineer bepaalt de ontwikkelomgeving en de productieomgeving.

Inrichting van technische aspecten

De inrichting van onder andere de hardware, WAS en bijhorende technische koppelingen wordt verzorgd door een IS-specialist. De IS-specialist is ook degene die de handmatige deployment-scripts schrijft zodat een ontwikkelaar of TAB de applicatie en koppelingen kan deployen op de gewenste omgeving.

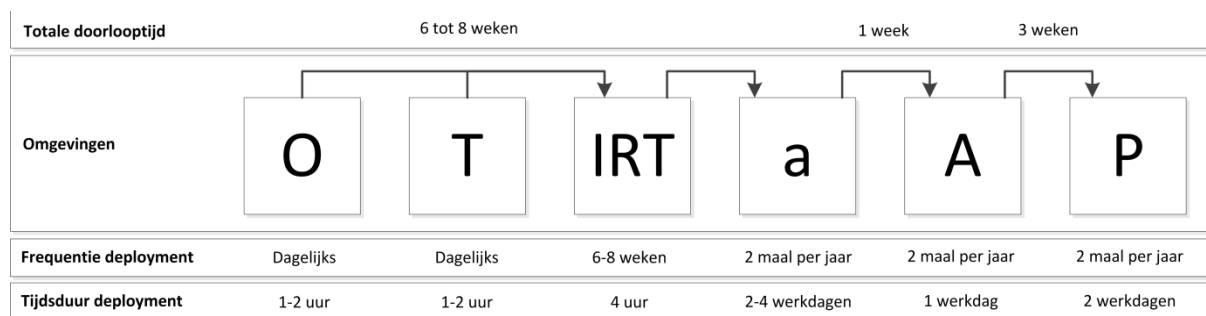
Registratie van de ontwikkel- en releaseversies

Configuratiemanagement is een discipline om software (source code en releases) te beheersen door o.a. de registratie van versienummers. Binnen de Rabobank wordt configuratiemanagement verstaan als Software Configuratiemanagement. Een configuratiemanager heeft inzicht in de ontwikkelversie (die opgeslagen zijn in SVN) en de releaseversies (die opgeslagen zijn in Nexus). Daardoor is het bekend welke versie in productie is en welke versies in ontwikkeling zijn. Daarnaast heeft de configuratiemanager inzicht in welke aangevraagde wijziging behoort bij welke source code, tag en release. Dit heeft meerdere voordelen waaronder het kunnen herleiden van een gewijzigde stuk software. Ook is het van belang dat externe toezichthouders moeten vaststellen welke versie in productie is. Naast de registrerende rol heeft een configuratiemanager ook een controlerende rol waarbij de betreffende versies gecontroleerd worden op volledigheid. Dit geldt voor alle applicaties binnen de keten(s). Software Configuratiemanagement wordt binnen de Rabobank verstaan als Application Lifecycle Management³⁰.

³⁰ (Jacobs, 2011)

5.7 Doorlooptijden

In *Figuur 13 - Doorlooptijden* zijn de doorlooptijden van het deploymentproces weergegeven. Er is onderscheid gemaakt in de totale doorlooptijd, frequentie van de deployment en de tijdsduur van een deployment per omgeving. Een verklaring van de aangegeven tijden is beschreven in de *Bijlage L: Doorlooptijden*.



Figuur 13 - Doorlooptijden

5.8 Beveiliging en functiescheiding

Dit hoofdstuk beschrijft de beveiliging, autorisatie en functiescheiding die wordt gehanteerd.

5.8.1 OTAP

Er is een strikte scheiding aanwezig tussen de OTa en de AP. Leden van het scrumteam hebben geen toegang tot en inzicht in de acceptatie(A)- en productieomgeving. Dit betekent ook geen toegang tot de gegevens op deze omgevingen, zoals databases en log files. Omgekeerd kan iemand die actief is binnen de acceptatieomgeving (A) geen sourcecode aanpassen.

5.8.2 Ontwikkeling

Een ontwikkelaar en/of tester kan via DDL³¹ statements (om de database structuur aan te passen) een database aanpassen. Een ontwikkelaar en/of tester kan niet via DML statements (om de gegevens in de tabellen te beheren) een database aanpassen.

5.8.3 Beheer en exploitatie

Voor het in beheer nemen van een applicatie (waaronder zowel de A-omgeving en de P-omgeving valt) is er een functiescheiding aanwezig. Daardoor zijn er specifieke medewerkers die een server/serverruimte beheren, linux beheren, netwerkvoorzieningen beheren, de applicatie beheren (scheiding tussen TAB en FAB), Websphere (WAS) en andere middleware applicaties beheren, database beheren of koppelingen beheren. De scheiding heeft te maken met zowel interne als externe wet- en regelgeving. Een voorbeeld van externe wet- en regelgeving is SOx. SOx geeft daarnaast ook aan welke voorwaarden een applicatie moeten voldoen.

³¹ (Orafaq.com)

5.9 BBS specifiek

In de vorige paragrafen van de huidige situatie zijn de organisatorische aspecten van de ontwikkeling en het in beheer nemen van de Java-applicatie BBS toegelicht. Waar mogelijk is deze informatie generiek beschreven, oftewel dat het geldig is voor iedere Java-applicatie binnen de Rabobank. Deze paragraaf behandelt de BBS specifieke onderdelen die niet generiek zijn beschreven in de vorige paragrafen.

Opschalen

Binnen de acceptatieomgeving (A) en productieomgeving wordt de applicatie aan de hand van opschalen geïmplementeerd. Het opschalen betekent dat eerst een en/of twee banken getest worden met de nieuwe versie van de applicatie en dat zich dit vervolgens opschaaft totdat alle banken gebruik maken van deze applicatie.

IRT omgeving

Voor de ontwikkeling en beheer van de applicatie BBS wordt er gebruik gemaakt van een IRT omgeving.

Agile Scrum

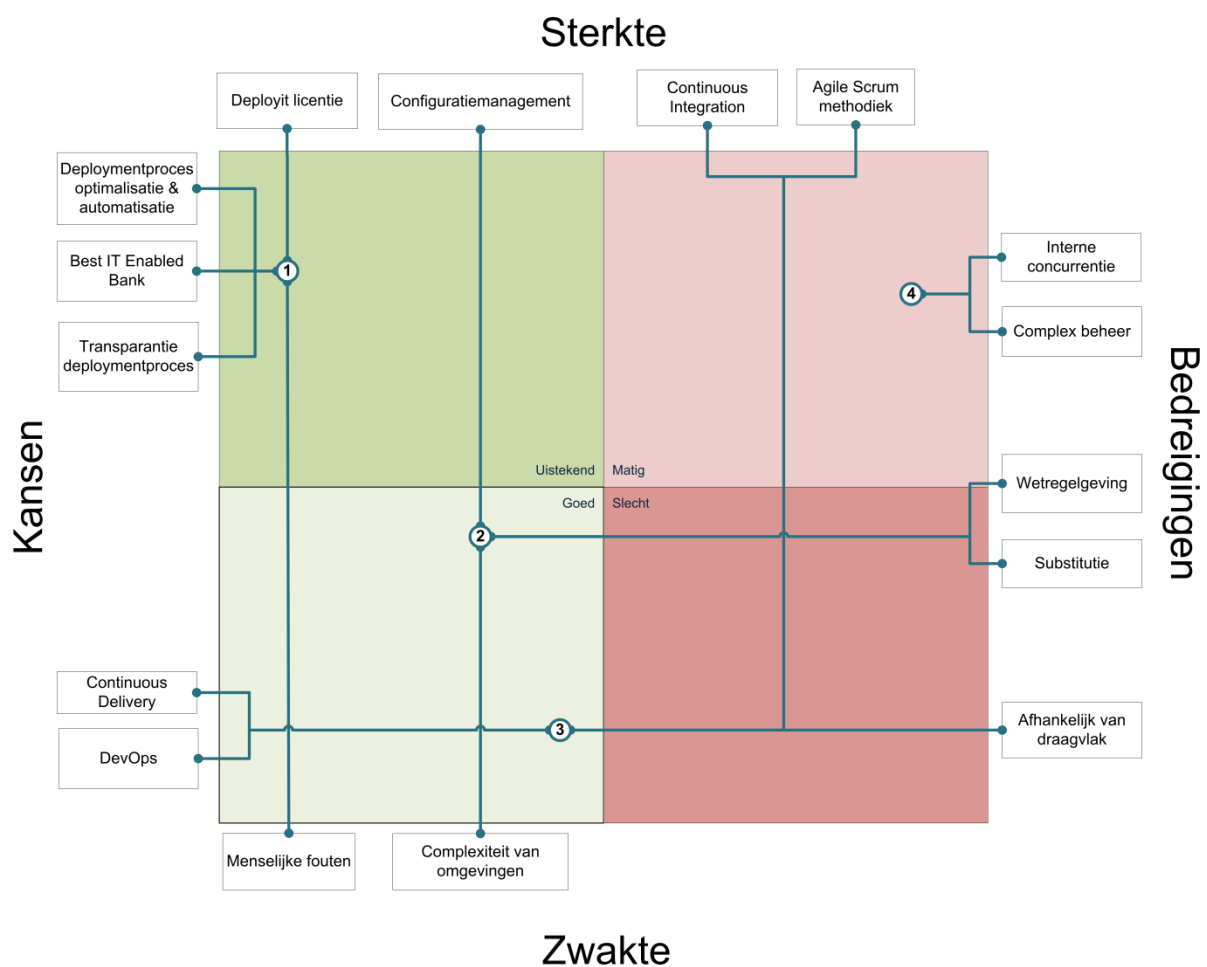
Voor de ontwikkeling van de applicatie BBS wordt er gebruik gemaakt van de Agile Scrum methode.

Testdocumentatie

De functionele documentatie van BBS (ontwerp) wordt goed bijgewerkt, maar de testgevallen worden slechts globaal beschreven. Dit heeft als nadeel dat de testen lastiger overdraagbaar zijn maar als groter voordeel dat er meer tijd voor het testen zelf over blijft. Documentatie wordt dan geen doel op zich meer.

6 SWOT analyse

In een SWOT analyse^{32,33}, zie *Figuur 14 - SWOT analyse*, zijn de sterke punten (**Strenghts**), zwakke punten (**Weaknesses**), kansen (**Oppertunities**) en bedreigingen (**Threats**) in kaart gebracht. De scope van de analyse is de in kaart gebrachte huidige situatie. De analyse is verricht met behulp van de aangegeven wensen, verwachtingen en aandachtspunten die tijdens het onderzoek zijn vergaard, zie *Bijlage M: Wensen, verwachtingen en aandachtspunten*. Voor het in kaart brengen van de kansen en bedreigingen is er gebruik gemaakt van het vijfkrachtenmodel van Porter³⁴. De bedreigingen en kansen doelen op potentiële gewenste situaties. De sterktes, zwaktes, kansen en bedreigingen zijn verbonden met een nummer (1-4) via lijnen. Ieder nummer is uitgewerkt in *paragraaf 6.1 Toelichting model*.



Figuur 14 - SWOT analyse

³² (Roij, Miltenburg, Steenis, & Helten, 2011)

³³ (Kleijn & Rorink, 2007)

³⁴ (Kleijn & Rorink, 2007)

6.1 Toelichting model

De verbindingen, aangegeven met nummers, in het SWOT model zijn in deze paragraaf toegelicht.

Punt 1, *Uitstekend scenario*

Deze verbinding focust zich op het scenario om het deploymentproces te optimaliseren en te automatiseren. Deze kans is sterk aanwezig omdat de licentie van een tool om de deployment te automatiseren al is aangeschaft binnen de Rabobank, namelijk Deployit. Door deze optimalisatie en automatisatie worden menselijke fouten gereduceerd. De menselijke fouten bestaan o.a. uit de manuals die in sommige gevallen onvolledig en/of incorrect. De 'Best IT Enabled Bank' is de strategie van de Rabobank, voor dit onderdeel betekent dit het verkorten van de Time to Market, stabiel houden van de systemen en het verlagen van de kosten van ontwikkeling en beheer. De transparantie van het deploymentproces betekent het inzicht hebben van aanwezige applicatieversies op de verschillende omgevingen en daarnaast het genereren van rapportages en dashboards.

Punt 2, *Goed scenario*

Een goed scenario is om preventief in te spelen op de aanwezige bedreigingen, namelijk substitutie en wet- en regelgeving. Substitutie omvat de vervanging van een van de systemen binnen de Java-ontwikkelstraat. Daarnaast dient er ook rekening gehouden te worden met de kans dat een tool, zoals Deployit, down-time ondervindt op een cruciaal moment. Wet- en regelgeving kan bestaan op het wijzigingen van bijvoorbeeld de aanwezige functiescheiding wat invloed heeft op het deploymentproces. Om hier op in te spelen dient configuratiemanagement aanwezig te zijn. Configuratiemanagement is een sterk punt van de Rabobank, waarbij het o.a. inzicht geeft in de aanwezige complexiteit van de verschillende ontwikkelomgevingen. Om preventief te werk te gaan dient een tool zoals Deployit opgenomen te worden binnen configuratiemanagement.

Punt 3, *Goed scenario*

De derde verbinding is het goede scenario om DevOps en continuous delivery te implementeren. In *Bijlage D: Theoretisch kader* is DevOps en continuous delivery toegelicht. Voor een optimale invoering van DevOps en continuous delivery is de Agile methodiek en continuous integration benodigd. Twee van de sterke punten van de Rabobank is dat de ontwikkeling van een Java-applicatie wordt ondersteund met de Agile Scrum methodiek en een basale continuous integration. De bedreiging van de afhankelijkheid van draagvlak heeft te maken met de ontwikkel- en beheerorganisatie. De ontwikkelorganisatie is voornamelijk innovatief ingesteld en de beheerorganisatie heeft een groot belang bij continuïteit. De gedachte van de DevOps beweging is om dit draagvlak te optimaliseren.

Punt 4, *Matig scenario*

De combinatie van interne concurrentie en de complexiteit van het beheer is een matig scenario. Binnen de Rabobank is er een soortgelijke tool zoals Deployit actief, namelijk PiCoMa. Het complexe beheer bestaat uit de afhankelijkheid van procedures, verschillende partijen en een complex systemenlandschap. Om een tool zoals Deployit organisatie-breed te implementeren dient de tool flexibel te zijn met de keuzes voor de inrichting zodat een tool zoals PiCoMa ook over zou kunnen gaan naar Deployit.

7 Gewenste situaties

In het kader van de SWOT-analyse zijn er drie gewenste situaties beschreven, namelijk situatie A, B en C. Punt één van de analyse komt in iedere situatie terug, namelijk de automatisering en optimalisatie van het deploymentproces via o.a. Deployit. Punt twee bespreekt configuratiemanagement wat als losse paragraaf is opgenomen, zie *paragraaf 7.5 Configuratiemanagement*. Punt drie wordt behandeld bij situatie B en C. Punt vier, wat een matig scenario is, wordt voorkomen aan de hand van een ingericht Application Lifecycle Management, zie *paragraaf 7.6 Application Lifecycle Management*.

7.1 Situatie A

De gewenste situatie A geeft de eenvoudigste implementatie van Deployit weer.

7.1.1 Veranderingen

De veranderingen ten opzichte van de huidige situatie zijn per onderwerp toegelicht.

Deployit

De tool wordt in deze situatie bij elke omgeving (OTAP) toegepast. Dit betekent dat de ontwikkelaar, tester en technisch applicatiebeheerder gebruik maken van de tool.

Maven plugin³⁵

Deployit heeft een DAR-bestand nodig als input om een deployment uit te voeren. De creatie van het DAR-bestand kan, middels een plugin, uitgevoerd worden door Maven tijdens het maken van een build. Het DAR-bestand wordt op de lokale computer van de ontwikkelaar geplaatst die het vervolgens selecteert in Deployit.

Manuals

De manuals bestaan enkel uit release notes. De benodigde install notes, die beschrijven hoe de applicatie gedeployd moet worden, wordt vervangen door Deployit.

7.1.2 Procesmodel

De veranderingen ten op zichte van de huidige situatie zijn toegelicht per proces. De activiteiten die ongewijzigd blijven zijn niet beschreven. Zie *Bijlage N: Gewenste situatie A* voor de procesmodellen.

7.1.2.1 Ontwikkeling

Zie *Tabel 6 - Ontwikkeling situatie A* voor de wijzigingen van het proces ontwikkeling.

Nr.	Beschrijving
07	Maven, die tijdens deze activiteit op de lokale omgeving van de ontwikkelaar staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een snapshot-DAR, wordt op de lokale omgeving van de ontwikkelaar geplaatst.
08	De ontwikkelaar selecteert in Deployit de snapshot-DAR van zijn lokale omgeving en geeft Deployit de opdracht om te deployen.
A1	Deployit pakt het DAR-bestand uit, slaat het op de server, en deployt de applicatie op de omgeving.
10	De tester geeft Deployit de opdracht om de applicatie naar de T-omgeving te deployen.

³⁵ (XebiaLabs, Deployit Plugin, 2012)

A2	Deployit deployt de applicatie op de T-omgeving.
14	Maven, die tijdens deze activiteit op de lokale omgeving van de ontwikkelaar staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een release-DAR, wordt in Nexus geplaatst samen met de release notes. Maven maakt een tag aan in SVN.
15	De ontwikkelaar geeft Deployit de opdracht om de applicatie naar de IRT-omgeving te deployen.
A3	Deployit deployt de applicatie op de IRT-omgeving.

Tabel 6 - Ontwikkeling situatie A

7.1.2.2 Overdracht

De configuratiemanager dient voor de controle van de bestanden en manuals in situatie A de release notes en DAR-bestanden te raadplegen uit Nexus.

7.1.2.3 Exploitatie en beheer

Zie *Tabel 7 - Exploitatie en beheer situatie A* voor de wijzigingen van het proces exploitatie en beheer.

Nr.	Beschrijving
02	De TAB ontvangt van de FAB de melding om te starten met de deployment op de UAT-omgeving. TAB leest de release notes en geeft Deployit de opdracht om te deployen naar de UAT-omgeving.
A1	Deployit deployt de applicatie op de UAT-omgeving.
04	TAB geeft Deployit de opdracht om te deployen naar de PAT-omgeving.
A2	Deployit deployt de applicatie op de UAT-omgeving.
06	TAB geeft Deployit de opdracht om te deployen naar de P-omgeving.
A3	Deployit deployt de applicatie op de UAT-omgeving.

Tabel 7 - Exploitatie en beheer situatie A

7.1.3 Impactanalyse

De impactanalyse beschrijft de effecten van de veranderingen ten opzichte van de huidige situatie. De verwachte impact is per onderwerp toegelicht.

Deployit kennis

Tester, ontwikkelaar en technisch applicatiebeheerder moeten kennis hebben van Deployit.

Frequenter deployen

Het efficiënter maken van de deployment resulteert in de mogelijkheid om vaker te releasen. Naar verwachting worden de releases daardoor qua omvang kleiner. De impact is dat de configuratiemanager, functioneel- en technisch applicatiebeheerders frequenter moeten controleren, testen en/of deployen.

Monitorende rol

De TAB en FAB worden door de implementatie van situatie A meer betrokken bij de ontwikkeling van (een nieuwe versie van) een applicatie. De TAB en FAB nemen namelijk nadat Deployit is geconfigureerd een monitorende rol op zich.

Handmatige acties

De database administrator (DBA) geeft in deze situatie geen rechten aan Deployit om direct aanpassingen te maken in de databases van de PAT- en productieomgeving. Dit resulteert in een handmatige actie om de database voor deze omgevingen handmatig aan te passen, dit verlengt de benodigde deploymenttijd.

7.1.4 Randvoorwaarden

Voor een succesvolle implementatie van situatie A dienen de omgeving specifieke onderdelen geconfigureerd te zijn in Deployit. Daarnaast moeten de omgevingen op dezelfde manier ingericht zijn. Dit betekent o.a. dezelfde WAS-versie, databasestructuur en dezelfde beschikbare koppelingen. Een andere randvoorwaarde is dat Deployit geïnstalleerd is op een server en beschikbaar is voor de ontwikkelaar, tester, TAB en FAB. In het kader van een generieke implementatie moet het organisatieonderdeel dat Deployit wil implementeren dezelfde opzet hebben als is beschreven in de huidige situatie.

7.1.5 Risicoanalyse

Deze paragraaf beschrijft de risico's die zich kunnen voordoen tijdens de uitvoering van situatie A. De risico's zijn beschreven in *Tabel 8 - Risicoanalyse situatie A*. Ieder risico beschrijft de bijhorende kans, impact, prioriteit, maatregel en uitvalscenario. De kans en impact is een cijfer van 1 tot en met 5, daarbij is 1 laag en 5 hoog. De prioriteit is de kans maal de impact. De maatregel is hetgeen wat wordt gedaan om de kans en impact van het risico te minimaliseren. Het uitvalscenario beschrijft het scenario op het moment dat het risico zich voordoet.

Risico	Kans	Impact	Prioriteit	Maatregel	Uitvalscenario
TAB/DBA heeft scripts benodigd die niet worden ondersteund door Deployit.	3	5	15	TAB/DBA in vroeg stadium van de ontwikkeling betrekken bij de configuratie van Deployit.	De scripts handmatig uitvoeren. Zoeken naar een oplossing.
Deployit en Nexus slaan dezelfde DAR-bestanden op. Dit resulteert in inefficiënte opslag.	5	3	15	Deployit opslag maandelijks handmatig opschonen.	Grotere opslag aanvragen.
Database administrator (DBA) ondervindt problemen met databasetoegang.	2	5	10	DBA in vroeg stadium van de ontwikkeling betrekken bij de configuratie van Deployit.	De databaseaanpassingen handmatig verrichten. Zoeken naar een oplossing.
Medewerker ondervindt problemen met deployen	2	2	4	Medewerker een korte cursus geven voor het deployen via Deployit.	Een andere medewerker inschakelen om te deployen en/of het probleem op te lossen.

Tabel 8 - Risicoanalyse situatie A

7.1.6 Benodigde activiteiten

Voor de implementatie van Situatie A dienen de volgende activiteiten uitgevoerd te worden:

1. Installatie Deployit op server
2. Configuratie Deployit (waaronder het instellen van een autorisatiemodel)
3. Maatwerk Deployit
 - a. Maven plugin installeren
 - b. Ondersteuning van scripts van TAB/DBA.
4. Ontwikkelaars, testers, TAB, FAB en DBA een cursus geven over de mogelijkheden van Deployit.

7.2 Situatie B

De gewenste situatie B is een uitbreiding op de gewenste situatie A. Situatie B geeft ten opzichte van situatie A de mogelijke connectie weer tussen Deployit en de huidige mogelijkheden van continuous integration.

7.2.1 Veranderingen

De veranderingen ten opzichte van situatie A zijn per onderwerp toegelicht.

Hudson/Deployit

De Rabobank heeft een continuous integration systeem ingericht voor de ontwikkeling van Java-applicaties. Dit betekent het automatisch uitvoeren van code-inspectie, (unit-)testen en het laten maken van een build. Om CI te optimaliseren is er de mogelijkheid om automatisch te deployen via het CI systeem³⁶. Daarvoor is er een koppeling nodig met, in dit geval, Hudson. In situatie B roept Hudson Deployit aan om te deployen, zonder tussenkomst van een ontwikkelaar. Om dit te realiseren dient een script/plugin gemaakt te worden om Hudson de tool aan te kunnen laten sturen. Hudson laat na iedere commit Maven een DAR-bestand (snapshot) aanmaken, vervolgens plaatst Maven het bestand in Nexus. Hudson selecteert het bestand en laat het door Deployit deployen naar de O-omgeving.

Hudson schrijfrechten Subversion (SVN)

Hudson heeft in situatie B een uitgebreidere taak gekregen. Namelijk het automatisch aanmaken en plaatsen van builds (snapshots/releases) in Nexus om deze vervolgens automatisch te deployen, via Deployit, naar de gewenste omgeving. In de huidige situatie heeft Hudson geen schrijfrechten in SVN. De schrijfrechten zijn noodzakelijk om Maven een volledige release op te leveren waarbij de versie van source code in SVN verhoogd wordt. Hudson zal daarom schrijfrechten moeten krijgen in SVN.

Databaserechten Deployit UAT en P

In situatie B heeft Deployit databaserechten waardoor Deployit aanpassingen kan maken in de databases van de PAT- en P-omgeving.

Geen gebruik van lokale omgeving na commit

In situatie B wordt na een commit geen gebruik meer gemaakt van de lokale ontwikkelomgeving voor de opslag van snapshots. In situatie B worden zowel de snapshots en releases opgeslagen in Nexus. Het voordeel dat alle snapshot-DAR bestanden in Nexus staan is dat het toegankelijk is voor ieder ander project. Een project kan daardoor de andere benodigde applicaties synchroniseren om in een vroeger stadium een kleinschalige integrale regressie test (IRT) uit te voeren. Daardoor zal de volledige IRT sneller slagen.

³⁶ (Duvall, Matyas, & Glover, 2007)

Opslag Deployit

Deployit heeft geen eigen opslag meer om de uitgepakte DAR-bestanden op te slaan. Dit gebeurt in situatie B in Nexus. Nexus is immers het versiebeheersysteem van alle releases en die verantwoordelijkheid moet niet deels uit handen gegeven worden aan Deployit.

7.2.2 Procesmodel

Enkel het proces ontwikkeling wijzigt ten opzichte van situatie A. De activiteiten die ongewijzigd blijven zijn niet beschreven. Zie *Bijlage O: Gewenste situatie B* voor het procesmodel.

7.2.2.1 Ontwikkeling

Zie *Tabel 9 - Ontwikkeling situatie B* voor de wijzigingen van het proces ontwikkeling.

Nr.	Beschrijving
04	Maven, die tijdens deze activiteit op de server van Hudson staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een DAR-snapshot, wordt in Nexus opgeslagen.
B1	Hudson selecteert het DAR-bestand uit Nexus en stuurt Deployit aan om het bestand te deployen op de O-omgeving.
B2	Hudson geeft Maven de opdracht om een DAR-bestand (in dit geval spreken we van een release) aan te maken.
14	Maven, die tijdens deze activiteit op de server van Hudson staat, compileert source code, voert (unit) test uit en maakt een build. Het resultaat, een DAR-release, wordt in Nexus opgeslagen.
B3	Hudson selecteert het DAR-bestand uit Nexus en stuurt Deployit aan om het bestand te deployen op de IRT-omgeving.

Tabel 9 - Ontwikkeling situatie B

7.2.3 Impactanalyse

De impactanalyse beschrijft de effecten van de veranderingen ten opzichte van de situatie A. De verwachte impact is per onderwerp toegelicht.

Monitorende rol

De DBA heeft een monitorende rol in situatie B. Deployit heeft de rechten om een database aan te passen in de PAT- en P-omgeving. Dit betekent dat de DBA meer betrokken zal moeten worden bij de ontwikkeling van (een nieuwe versie van) een applicatie om zekerheid te hebben dat de database juist wordt aangepast.

Ontwikkelaar

De ontwikkelaar is in situatie B genoodzaakt zijn om voor een deployment naar de O- of IRT-omgeving gebruik te maken van Hudson. Het handmatig maken en deployen van een build is niet mogelijk.

7.2.4 Randvoorwaarden

De randvoorwaarde voor situatie B is dat eerst situatie A geïmplementeerd is. Situatie A kan zowel afzonderlijk of gezamenlijk met situatie B geïmplementeerd worden.

7.2.5 Risicoanalyse

Deze paragraaf beschrijft de risico's die zich kunnen voordoen tijdens de uitvoering van situatie B. De risico's zijn beschreven in *Tabel 10 - Risicoanalyse situatie B*. Ieder risico beschrijft de bijhorende kans, impact, prioriteit, maatregel en uitvalscenario. De kans en impact is een cijfer van 1 tot en met 5, daarbij is 1 laag en 5 hoog. De prioriteit is de kans maal de impact. De maatregel is hetgeen wat wordt gedaan om de kans en impact van het risico te minimaliseren. Het uitvalscenario beschrijft het scenario op het moment dat het risico zich voordoet.

Risico	Kans	Impact	Prioriteit	Maatregel	Uitvalscenario
Hudson schrijfrechten geven in SVN onhaalbaar.	3	5	15	Onderzoek doen naar reden waarom er geen schrijfrechten gegeven kunnen worden.	Onderzoek doen naar alternatief, bijvoorbeeld het CI-systeem Jenkins.
DBA geeft geen rechten aan Deployit om aanpassingen te doen in de database van de UAT- en P-omgeving.	2	5	10	DBA meer betrekken bij de ontwikkeling van (een nieuwe versie van) een applicatie.	Database handmatig aanpassen.
Medewerker ondervindt problemen met deployen	2	2	4	Medewerker een korte cursus geven voor het deployen via Deployit.	Een andere medewerker inschakelen om te deployen en/of het probleem op te lossen.
De ontwikkelaar maakt en deployt handmatig de build.	3	5	15	Regel invoeren dat de ontwikkelaar dit niet mag doen.	Deployit de enige mogelijkheid laten zijn om toegang te hebben tot de o-omgeving.

Tabel 10 - Risicoanalyse situatie B

7.2.6 Benodigde activiteiten

1. Situatie A implementeren.
2. Nexus directory aanpassen zodat DAR-bestanden opgeslagen kunnen worden.
3. Hudson schrijfrechten geven in Subversion (eventueel in combinatie met onderzoek naar het probleem).
4. Script/plugin schrijven om Hudson Deployit aan te laten sturen.

7.3 Situatie C

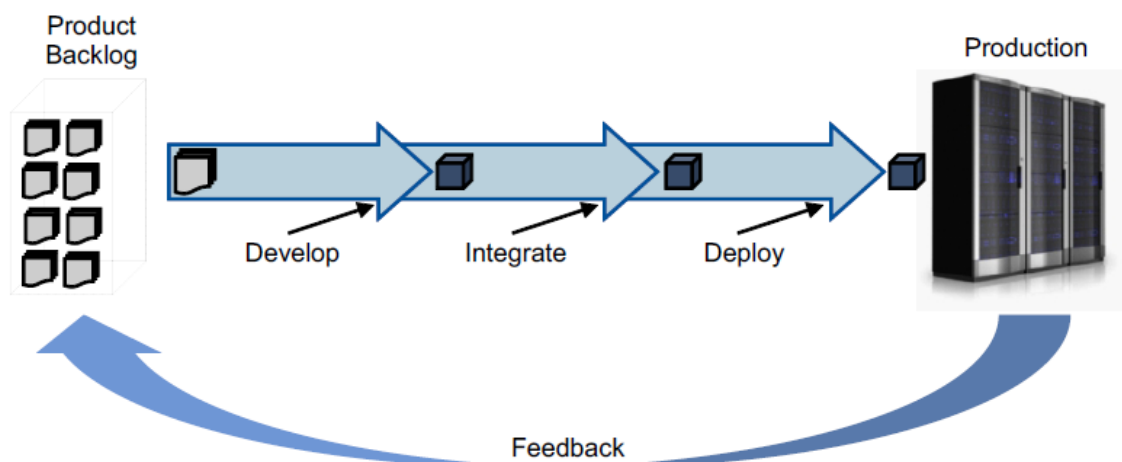
Situatie A en B geven een duidelijke weergave voor de invoering van Deployit om de doorlooptijd van het deploymentproces te reduceren. Situatie C beschrijft meerdere opties om situatie B te optimaliseren zodat de benodigde doorlooptijd verder gereduceerd kan worden. Voor het uitvoeren van de beschreven opties is het vereist om situatie B geïmplementeerd te hebben. Voor een implementatie van situatie C dienen de betreffende opties nader onderzocht te worden.

7.3.1 Volledig geautomatiseerde deployment pipeline

In *Bijlage D: Theoretisch kader* is de deployment pipeline toegelicht. Een volledig geautomatiseerde deployment pipeline is een end-to-end automatisering van de build, deploy, test en release processen. Voor een volledig geautomatiseerde deployment pipeline dient, ten opzichte van situatie B, de functionele testen en acceptatie testen (UAT/PAT) geautomatiseerd te worden. Voor het automatiseren van deze testen dienen er tools geïmplementeerd te worden en moet de huidige werkwijze van de testers en functioneel applicatiebeheerders wijzigen naar een monitorende rol.

7.3.2 Kanban methode

Een groot voordeel van een geautomatiseerde deployment pipeline is de mogelijkheid om het deploymentproces te versnellen. Daardoor is het mogelijk om (twee)wekelijks een release in productie te nemen. Het (twee)wekelijks releasen vereist na iedere sprint een release die geschikt is om in productie te nemen. Het is moeilijk om een planning op te stellen, het ontwikkelen van verschillende taken en het uitvoeren van een demo in een (twee)wekelijkse sprint volgens de Agile-Scrum methode. Een opkomende trend is om de Kanban methode³⁷ te gebruiken in plaats van de Scrum methode. Het verschil tussen Kanban en Scrum is dat Kanban geen gebruik maakt van tijdsgebonden sprints maar dat een ontwikkelteam zich focust op het uitvoeren van één taak. Zie figuur *Figuur 15 - Kanban methode*³⁸ voor een weergave van de Kanban methode.



Figuur 15 - Kanban methode

³⁷ (Gartner, Achieving Continuous Delivery, 2013)

³⁸ (Gartner, Achieving Continuous Delivery, 2013)

7.4 Weerstand

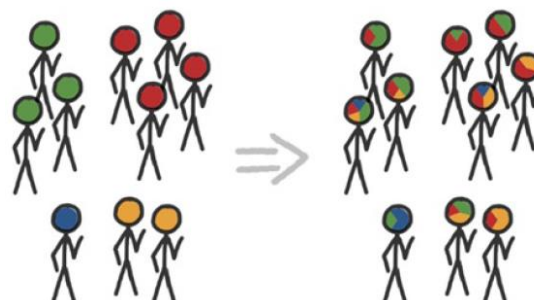
De verandering in de huidige werkwijze van medewerkers kan weerstand opwekken, waarbij weerstand niet altijd kwaadaardig is³⁹. Er zijn verschillende oorzaken waardoor weerstand kan ontstaan. Medewerkers kunnen slechte ervaringen hebben met, in dit geval, de implementatie van een nieuwe tool. Een andere oorzaak van weerstand is omdat de belangen van de medewerkers geschaad worden vanwege de verandering. De meest relevante weerstand is de conflicterende zienswijzen over mens en organisatie. De bijkomende partijen voor het implementeren van Deployit hebben ieder andere ideeën en meningen over wat goed is voor de organisatie. Daarbij is een terugkomende discussie: innovatie versus continuïteit.

Voor een optimale implementatie dient er een draagvlak gecreëerd te worden. Dat het creëren van draagvlak tussen de benodigde niet makkelijk is geeft de organisatie National Instruments aan⁴⁰. De organisatie geeft namelijk in de evaluatie voor de implementatie van DevOps aan dat er nog gewerkt wordt aan een optimale samenwerking tussen de benodigde partijen. Het probleem dat deze organisatie ondervindt is dat de benodigde partijen ieder andere disciplines hebben en kijken vanuit andere perspectieven.

Om draagvlak te creëren tussen de aanwezige partijen zijn er twee mogelijkheden beschreven.

Beheer is betrokken bij ontwikkeling en vice versa

In de huidige situatie is er een duidelijke scheiding tussen de beheerafdeling (B&E) en de ontwikkelafdeling (ADM). Met de komst van Deployit worden werkzaamheden van de beheerafdeling al uitgevoerd tijdens de werkzaamheden van de ontwikkelafdeling. Een voorbeeld is de configuratie van de productieomgeving specifieke onderdelen in Deployit. Om als beheerafdeling de controle te behouden over deze configuratie dient de afdeling



Figuur 16 - Gedeelde kennis

betrokken te worden bij de werkzaamheden van de ontwikkelafdeling. Daarnaast resulteert het delen van elkaars vaardigheden, kennis en ervaringen zich in een team wat elkaar beter begrijpt. Een medewerker is daarbij gespecialiseerd in zijn of haar eigen onderdeel maar heeft daarnaast ook de basiskennis over de werkzaamheden van zijn of haar collega's. Zie *Figuur 16 - Gedeelde kennis*⁴¹.

Ontwikkeling en beheer in dezelfde ruimte⁴²

De organisatie Rally is een leverancier van Agile-specifieke Application Lifecycle Management tools, Agile coaching en Agile training. Rally past de Kanban methode toe voor het optimaliseren van continuous delivery. Een belangrijk leerpunt is dat Rally de ontwikkelafdeling en de beheerafdeling gevestigd heeft in dezelfde ruimte om de communicatie tussen deze afdelingen te bevorderen.

³⁹ (Caluwé, 2006)

⁴⁰ (Gartner, Case Study: DevOps Used at National Instruments, 2011)

⁴¹ (Hüttermann, 2012)

⁴² (Gartner, Achieving Continuous Delivery, 2013)

7.5 Configuratiemanagement

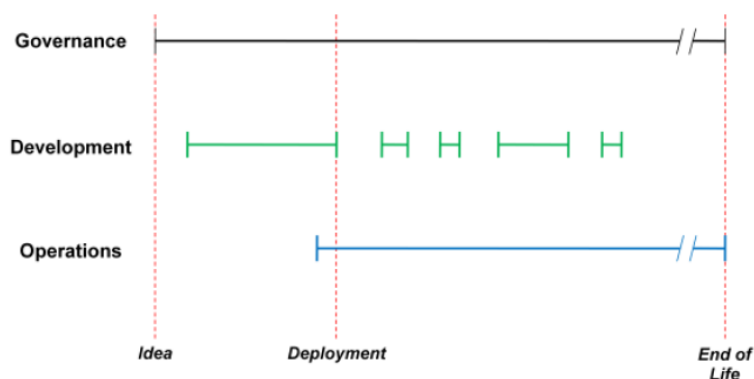
Voor iedere toekomstige situatie is configuratiemanagement een belangrijk onderdeel, zie ook punt twee van de SWOT-analyse in *hoofdstuk 6 SWOT analyse*. Het is van belang om inzicht te houden in welke release behoort bij welke label (tag), source code en wijziging (change). De implementatie van Deployit biedt de mogelijkheid om sneller en vaker te releasen. Het frequenter releasen hoeft geen nadelige gevolgen te hebben voor de huidige nauwkeurigheid van configuratiemanagement. Daarvoor dient configuratiemanagement wel intensief opgenomen te worden tijdens de implementatie van Deployit.

7.6 Application Lifecycle Management

Application Lifecycle Management (ALM), zie ook *Bijlage D: Theoretisch kader*, beschrijft de manier hoe een applicatie wordt beheerd vanaf concept, bouwen en tot en met het uit dienst nemen⁴³. Binnen ALM kan er onderscheid gemaakt worden tussen bestuur, ontwikkeling en beheer, zie *Figuur 17 - Application Lifecycle Management*⁴⁴. Dit hoofdstuk beschrijft de inrichting van ALM voor Deployit.

Governance

Het bestuur (governance) maakt de beslissingen rondom de tool en voert het projectmanagement uit. Voor Deployit is dit zowel de dienst ARA (Application Release Automation) en de leverancier XebiaLabs. De dienst ARA wordt binnen de Rabobank opgezet door het project ARA. Dienst ARA valt onder de verantwoordelijkheid van Infa Service Linux (IS-Linux) van B&E. Het bepalen of er nieuwe versies uitkomen voor de tool wordt beslist door XebiaLabs.



Figuur 17 - Application Lifecycle Management

Development

De ontwikkeling van de tool en het uitbrengen van nieuwe versies wordt verzorgd door de leverancier, namelijk XebiaLabs.

Operations

Het leveren van support aan de eindgebruiker valt onder de verantwoordelijkheid van de dienst ARA. Ook het in productie nemen van de laatste versie van de tool, die door de leverancier wordt aangeboden, valt onder de verantwoordelijkheid van de dienst ARA.

⁴³ (Gousset, Keller, & Woodward, 2012)

⁴⁴ (Chappell, 2008)

7.7 Kosten/baten analyse

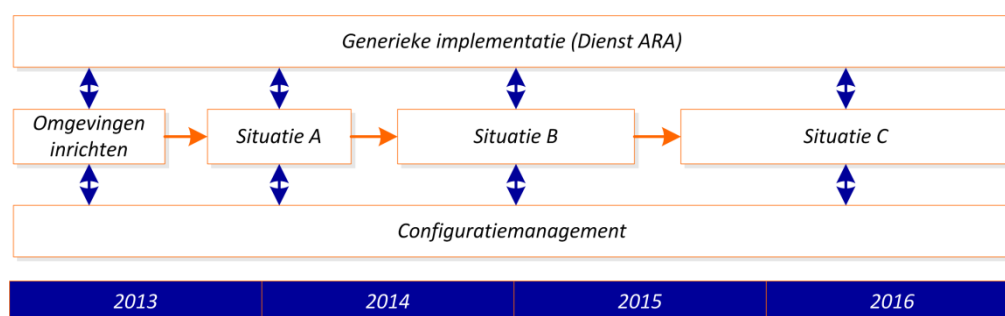
In deze paragraaf zijn de kosten en baten kort toegelicht in *Tabel 11 - Kosten/baten analyse*. De kosten beschrijven de onderwerpen, zoals benodigde onderzoeken, waarbij kosten gemaakt worden. De baten beschrijven de onderwerpen, zoals het automatiseren van de deployment pipeline, waar de meeste baat bij te behalen is. De kosten en baten zijn niet in kwantitatieve waarden uitgedrukt, zie ook *paragraaf 4.5 Ontwerp- en analysemethoden*.

Kosten	Tijdsfrequentie	Situatie		
		A	B	C
Server Deployit	<i>Jaarlijks</i>	X	X	X
Implementatie Maven-plugin	<i>Eenmalig</i>	X	X	X
Cursus Deployit voor TAB, FAB, Tester en ontwikkelaars	<i>Op aanvraag</i>	X	X	X
Cursus Deployit voor DBA	<i>Op aanvraag</i>		X	X
Maatwerk koppeling Hudson en Deployit	<i>Eenmalig</i>		X	X
Onderzoek koppeling Hudson en SVN	<i>Eenmalig</i>		X	X
Onderzoek naar geautomatiseerde deployment pipeline	<i>Eenmalig</i>			X
Onderzoek naar mogelijkheden Kanban-methode	<i>Eenmalig</i>			X
Baten		A	B	C
Reduceren time to market	<i>Permanent</i>	X	X	X
Reduceren menselijke fouten	<i>Permanent</i>	X	X	X
Frequenter deployen	<i>Permanent</i>	X	X	X
Geoptimaliseerd continuous integration system	<i>Permanent</i>		X	X
Geautomatiseerde deployment pipeline	<i>Permanent</i>			X
Geïmplementeerde Kanban-methode	<i>Permanent</i>			X

Tabel 11 - Kosten/baten analyse

7.8 Vervolgplanning

Voor de implementatie van de gewenste situaties is er een globale planning gemaakt, zie *Figuur 18 - Vervolgplanning*. In de vorige paragrafen is per situatie beschreven welke activiteiten benodigd zijn, deze zijn daarom niet benoemd in de planning. De implementatie van de situaties zal projectmatig uitgevoerd worden, daarom dient er een projectgroep gespecificeerd te worden. Omdat het project ARA al opgezet is, is het voor de hand liggend dat dit project de implementatie van alle situaties uitvoert. De generieke implementatie (dienst ARA) is opgenomen in de planning omdat er vanaf het begin van de ontwikkelingen rondom een ARA functionaliteit de onderzoeken generiek uitgevoerd moeten worden zodat er uiteindelijk een organisatie-brede ARA dienst is. Configuratiemanagement is opgenomen in de planning omdat bij iedere implementatie richting een volledige ARA functionaliteit dit onderwerp van belang is. Volgens de planning is situatie C in 2016 generiek geïmplementeerd.



Figuur 18 - Vervolgplanning

8 Conclusie

De conclusie van het onderzoek is het antwoord op de centrale onderzoeksvraag. Om antwoord te geven op de centrale onderzoeksvraag zijn er deelvragen geformuleerd en beantwoord, zie *paragraaf 3.2.2 Deelvragen*. De centrale onderzoeksvraag luidt als volgt:

Hoe kan de tool Deployit ingezet worden in de ontwikkel- en beheerprocessen van Java-applicaties binnen de afdelingen ADM en B&E zodat de doorlooptijd van het deploymentproces gereduceerd wordt?

De kracht van Deployit is het automatisch deployen van een Java-applicatie met bijhorende configuratiebestanden en databasescripts naar een geselecteerde omgeving. Het reduceren van de doorlooptijd van het deploymentproces focust zich daarom op de werkelijke deployments die tijdens dit proces plaatsvinden. De deployments die de meeste tijd innemen zijn degenen naar de UAT-, PAT- en P-omgeving. Deze benodigde tijd heeft voornamelijk te maken met het ontvangen en moeten begrijpen van (onvolledige) manuals, niet aanwezige koppelingen en het moeten schrijven van een draaiboek. De problemen rondom de manuals, de aanwezige koppelingen en het draaiboek worden opgelost door de tool. Om de deploymenttijd te reduceren moet de tool vanaf de ontwikkeling tot en met het in productie nemen van een applicatie worden gebruikt. Daardoor worden fouten in de configuratie van de tool in een vroeg stadium (lees: ontwikkeling) gevonden in plaats van tijdens het in productie nemen van de applicatie. Dit betekent dat de ontwikkelaars, testers, functioneel applicatie beheerders en technisch applicatie beheerders gebruik maken van de tool. De werkzaamheden voor deze rollen krijgen in het kader van de deployment van een applicatie een monitorend karakter.

De invoering van Deployit betekent, in het kader van de aanwezige infrastructuur, omgevingen die zoveel mogelijk op de productieomgeving lijken. Ook een koppeling met het aanwezige continuous integration systeem biedt voordelen voor het reduceren van de doorlooptijd van het deploymentproces.

De gebruikte manuals en het draaiboek in de huidige situatie worden vervangen door Deployit. Daarvoor dient de tool ingesteld/geconfigureerd te worden. Deze configuratie dient bijgehouden te worden door configuratiemanagement. Daardoor wordt de discipline behouden dat er te herleiden is welke release bij welke wijziging, source code en label hoort. En wat daarnaast belangrijk is, is dat een gedeployde release herhaalbaar is.

Een belangrijk aandachtspunt tijdens de implementatie van de tool zijn de verschillende partijen. Tijdens het deploymentproces spelen namelijk vele partijen een belangrijke rol met bijhorende functiescheidingen en autorisatiebeveiligingen. Iedere partij heeft zijn eigen disciplines en kijkt vanuit een eigen perspectief.

Een generieke invoering van Deployit betekent een helder ingerichte Application Lifecycle Management waarbij duidelijk is wie beslist, ontwikkelt en beheert. Voor Deployit betekent dit een combinatie van de (externe) leverancier XebiaLabs en de (interne) dienst ARA die opgezet wordt door het project ARA.

9 Aanbeveling

In *hoofdstuk 8 Conclusie* is de conclusie van het onderzoek beschreven. Vanuit de conclusie kunnen de volgende vijf aanbevelingen gedaan worden.

Deployit

Dit rapport beschrijft de invoering van Deployit binnen de huidige organisatorische aspecten en processen. Vanuit de conclusie wordt aanbevolen om de tool te implementeren indien het ook technisch haalbaar is. Of het technisch haalbaar is wordt onderzocht binnen de pilot Deployit BBS van project ARA. Het project ARA heeft, op het moment van schrijven, nog geen aanbeveling gedaan.

Vervolgplanning

Voor de implementatie van Deployit wordt aanbevolen dit stapsgewijs uit te voeren om zodanig naar een hoger volwassenheidsniveau te komen binnen continuous delivery. Deze stappen komen nadrukkelijk naar voren van situatie A naar B en C in de vervolgplanning, zie *paragraaf 7.8 Vervolgplanning*. Waar nodig worden er uitzonderingen gemaakt indien een afnemer niet voldoet aan de randvoorwaarden van situatie A.

Betrokken partijen

Met de komst van Deployit kan weerstand worden verwacht, zie *paragraaf 7.4 Weerstand*. Deze weerstand komt voort uit de betrokken partijen die ieder eigen disciplines hebben en kijken vanuit verschillende perspectieven. Om de tool een succes te laten worden, dienen de partijen samen te werken. De aanbeveling is om de benodigde partijen intensief op te nemen tijdens de implementatie van Deployit en iedere partij op de hoogte te brengen van de mogelijkheden en verwachte impact van de tool. Het project ARA dient deze verantwoording op zich te nemen.

Generieke implementatie

Een succesvolle generieke implementatie van Deployit betekent dat iedere organisatieonderdeel volgens dezelfde wijze de tool toepast. Het wordt aanbevolen de ondersteuning van de tool centraal te beleggen, in dit geval bij de dienst ARA. De dienst neemt de verantwoordelijkheid op zich om de afnemers te voorzien van de laatste updates, support en onderhoudt het contact met de leverancier voor onder andere nieuwe versies van de tool. De dienst zal worden ingericht door project ARA.

En nu?.. DevOps!

Deployit is een stap richting continuous delivery wat een onderdeel is van DevOps, zie *Bijlage D: Theoretisch kader*. DevOps behandelt onderwerpen rondom 'CAMS' (Culture, Automation, Measurement, Sharing)⁴⁵. De tool valt binnen het onderwerp automatisering. De andere drie aspecten zijn onvoldoende besproken in dit rapport en de tool is enkel een onderdeel van het automatiseringsaspect om DevOps succesvol te maken binnen de Rabobank. De aanbeveling is om onderzoek te doen naar de mogelijkheden met betrekking tot DevOps.

⁴⁵ (Willis, 2010)

10 Discussie

De onderzoeksresultaten geven een duidelijke weergave van de huidige situatie en de gewenste situaties van het deploymentproces met betrekking tot de invoering van Deployit. Er zijn verschillende maatregelen genomen om de resultaten valide, betrouwbaar en bruikbaar te houden, zie daarvoor *paragraaf 4.3 Betrouwbaarheid, validiteit en bruikbaarheid*.

Aan het begin van het onderzoek heeft de onderzoeker aan verschillende medewerkers de vraag gesteld of er wel of geen toolanalyse uitgevoerd is voor een tool die deployments automatisch uitvoert. Het antwoord was dat er geen toolanalyse is uitgevoerd en meteen gekozen is voor Deployit. De onderzoeker had drie opties, het plan van aanpak wijzigen en eerst een toolanalyse uitvoeren, de waarneming negeren of Deployit beschouwen als een dergelijke tool wat automatisch deployments kan uitvoeren. Er is gekozen voor de laatste optie. De eerste stap voor het realiseren van de deployment pipeline is namelijk het in kaart brengen van de huidige processen⁴⁶. Dit is dan ook de kern van het onderzoek. De gewenste situaties geven een invoering van een deployment-tool aan waarbij de keuze van de tool achteraf gewijzigd zou kunnen worden. Enkel de aangegeven Maven-plugin is specifiek voor Deployit en daardoor mogelijk niet bij iedere soortgelijke tool aanwezig. Het uitvoeren van een toolanalyse na dit onderzoek heeft daarom weinig tot geen impact op de beschreven resultaten.

Het uitgangspunt van het onderzoek is om de resultaten zo generiek mogelijk te beschrijven. Om een oordeel te kunnen geven of de resultaten daadwerkelijk generiek zijn is dit aan het eind van het onderzoek overlegd met twee deskundigen binnen de Rabobank. Beide deskundigen hebben veel organisatie-brede ervaring en kennis in het ontwikkelen van Java-applicaties en de aanwezige Java-ontwikkelstraten binnen de Rabobank. Uit het overleg is gebleken dat de resultaten geldig zijn bij de meeste (lees: 85-95%) Java-applicaties. Een deel van de onderwerpen verschillen namelijk bij iedere portfolio of programma. Dit is het gebruik van de verschillende omgevingen (OTAP), het handmatig dan wel automatisch functioneel testen en de benodigde resources tijdens het in productie nemen van de applicatie. Wanneer een ander programma of project gebruik wil maken van Deployit moeten deze aangegeven onderwerpen specifiek voor deze groep in kaart gebracht worden.

⁴⁶ (Humble & Farley, 2011)

11 Glossarium

ADM	Application Development & Maintenance. Afdeling van Groep ICT.
AE	Application Engineer. Bedrijfsrol binnen de Rabobank.
ALM	Application Lifecycle Management, omvat de volledige tijd vanaf het idee tot en met het einde van de applicatie. ⁴⁷
ARA	Application Release Automation. Vormt de brug tussen ontwikkel- en beheerorganisatie voor het automatisch deployen van software ⁴⁸ .
Artifact	Een artifact is een (combinatie van) EAR-, WAR- en/of JAR-bestand(en).
AS	Application Service. Een afdeling van B&E.
B&E	Beheer en Exploitatie. Afdeling van Groep ICT.
BBS	Bedrijf en Bediening Systeem wat kredieten verstrekt aan bedrijven.
Build	Resultaat van Maven. Bestaat uit artifacts, configuratiebestanden en databasescripts.
Commit	Opslaan van de ontwikkelde Java-code in een versiebeheersysteem.
Continuous delivery	Het continu kunnen leveren van de ontwikkelde software aan de eindgebruiker.
Continuous integration	Maakt het mogelijk om 'builds' regelmatig te integreren met als doel dat het geen specifieke en tijdrovende taak wordt van het project. ⁴⁹
Deployit	Een tool die een geautomatiseerde deployment realiseert.
Deployment	Het verplaatsen van een applicatie en bijorende bestanden naar een omgeving.
DevOps	DevOps is een filosofie wat een technologische en procesmatige integratie realiseert tussen ontwikkeling en beheer ⁵⁰ .
EAR	Enterprise ARchive. Bestandsformaat van Java-applicaties.
Eclipse	Een persoonlijke integrale ontwikkelomgeving van de ontwikkelaar (IDE).
FAB	Functioneel Applicatie Beheerder. Bedrijfsrol binnen de Rabobank.
FIN Bedrijven	Financieren Bedrijven. Een portfolio van ADM.
GICT	Groep ICT. Organisatieonderdeel van Rabobank Nederland.
IDE	Integrated Development Environment. Een persoonlijke integrale ontwikkelomgeving van de ontwikkelaar.
Intelij	Een persoonlijke integrale ontwikkelomgeving van de ontwikkelaar (IDE).
IRT	Integratie Regressie Test.
IS	Infra Service. Een afdeling van B&E.
JAR	Java ARchive. Bestandsformaat van Java-applicaties.
Java	Een object georiënteerd programmeertaal om software te schrijven ⁵¹ .
Maven	Een programma dat source code compileert, testen uitvoert uit en builds maakt.
Nexus	Een versiebeheersysteem voor de opslag van releases van Java-applicaties.
OTAP	Ontwikkeling, Test, Acceptatie, Productie.
PAT	Production Acceptance Test. De 'A'-omgeving.
PDS	Professional Development Support. Een afdeling van ADM.
Script	Uitvoerbaar bestand.
SVN	Subversion. Een versiebeheersysteem voor Java-code.
TAB	Technisch Applicatie Beheerder. Bedrijfsrol binnen de Rabobank.
UAT	User Acceptance Test. De 'a'-omgeving.
WAR	Web application archive. Bestandsformaat van Java-applicaties.
WAS	Websphere Applicatie server.

⁴⁷ (Chappell, 2008)

⁴⁸ (XebiaLabs, Whitepaper Continous Delivery)

⁴⁹ (Duvall, Matyas, & Glover, 2007)

⁵⁰ (Computable)

⁵¹ (Horstmann & Cornell, 2012)

12 Literatuurlijst

Bosman, R. (2011, juni 7). Development Java Infrastructure Rabobank Groep v1.7.

Caluwé, L. d. (2006). *Leren Veranderen: een handboek voor de veranderkundige*. Deventer: Kluwer.

Chappell, D. (2008). *What is application lifecycle management?* Opgeroepen op april 29, 2013, van [www.davidchappell.com](http://www.davidchappell.com/WhatIsALM--Chappell.pdf): <http://www.davidchappell.com/WhatIsALM--Chappell.pdf>

Computable. (sd). Opgeroepen op februari 18, 2013, van DevOps: <http://www.computable.nl/artikel/opinie/development/4629565/1277180/is-devops-het-nieuwe-agile.html>

Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous Integration*. Pearson Education, Inc.

Gartner. (2011, augustus 23). *Application Release Automation is a key to DevOps*. Opgeroepen op april 30, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1773923&ref=QuickSearch&stkw=Application+Release+Automation+is+a+key+to+DevOps): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1773923&ref=QuickSearch&stkw=Application+Release+Automation+is+a+key+to+DevOps>

Gartner. (2011, november 21). *Case Study: DevOps Used at National Instruments*. Opgeroepen op mei 5, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1854517&ref=QuickSearch&stkw=Case+study+DevOps): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1854517&ref=QuickSearch&stkw=Case+study+DevOps>

Gartner. (2012, mei 15). *How to build a DevOps Release team*. Opgeroepen op april 30, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2015315&ref=QuickSearch&stkw=How+to+Build+a+DevOps+Release+Team): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2015315&ref=QuickSearch&stkw=How+to+Build+a+DevOps+Release+Team>

Gartner. (2012, juli 30). *Hype Cycle Application Development*. Opgeroepen op april 30, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2098316&ref=QuickSearch&stkw=Continuous+Deployment+Java): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2098316&ref=QuickSearch&stkw=Continuous+Deployment+Java>

Gartner. (2013, januari 21). *Achieving Continuous Delivery*. Opgeroepen op april 30, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2310016&ref=QuickSearch&stkw=Achieving+continuous+delivery): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2310016&ref=QuickSearch&stkw=Achieving+continuous+delivery>

Gartner. (2013, april 12). *Best Practices to Sync and Streamline DevOps Cycles for Faster IT Service Delivery*. Opgeroepen op april 30, 2013, van [www.gartner.com](http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2426115&ref=QuickSearch&stkw=DevOps): <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=2426115&ref=QuickSearch&stkw=DevOps>

Gousset, M., Keller, B., & Woodward, M. (2012). *Professional Application Lifecycle Management with Visual Studio 2012*. Wrox.

Horstmann, C. S., & Cornell, G. (2012). *Core Java: Volume I - Fundamentals, Ninth Edition*. Prentice Hall.

- Humble, J., & Farley, D. (2011). *Continuous Delivery*. Pearson Education, Inc.
- Hüttermann, M. (2012). *DevOps for Developers*. Apress.
- Jacobs, J. (2011, maart 24). IT management Ontwikkelbedrijf- SOLL situatie.
- Janssen, P. (2007). *Prince2 Compact*. Amsterdam: Pearson Education Benelux.
- Kleijn, H., & Rorink, F. (2007). *Verandermanagement: Een plan van aanpak voor integrale organisatieverandering en innovatie*. Amsterdam: Pearson Education Benelux.
- Mertens, J. (2010). *Praktijkonderzoek voor bachelors*. Bussum: Uitgeverij coutinho.
- Nathans, H. (1999). *Adviseren als tweede beroep*. Deventer: Kluwer.
- Obers, G.-J., & Achterberg, K. (2012). *Grip op processen in organisaties*. Zaltbommel: Van Haren Publishing.
- Orafaq.com. (sd). Opgeroepen op maart 15, 2013, van What are the difference between DLL, DML and DCL commands?:
http://www.orafaq.com/faq/what_are_the_difference_between_ddl_dml_and_dcl_commands
- O'Regan, O. (2012, augustus). Enterprise-class Application Release Automation. A Best Practice.
- Rabobank. (2010, juli). Functieprofiel Application Engineer.
- Rabobank. (2011, mei). Functieprofiel Beheerder versie 1.0.
- Rabobank. (2011, mei). Functieprofiel Ontwikkelaar versie 1.0.
- Rabobank. (2013, januari). Functieprofiel System Integrator Test Supply.
- Rabobank. (sd). *Applicatie Service*. Opgeroepen op februari 11, 2013, van www.afdelingen.rabobank.nl:
<http://afdelingen.rabobank.nl/SITES/AS/OVERAPPLICATIESERVICE/Pages/default.aspx>
- Rabobank. (sd). *Groep ICT*. Opgeroepen op februari 11, 2013, van www.rabobank.com:
https://www.rabobank.com/nl/group/About_Rabobank_group/Corporate_governance/Rabobank_Nederland/index.html
- Rabobank Groep ICT. (2009). *Strategieplan 2010-2012*. Utrecht: Rabobank.
- Rabobank. (sd). *Infra Service*. Opgeroepen op februari 11, 2013, van www.afdelingen.rabobank.nl:
<http://afdelingen.rabobank.nl/sites/InfraService/over%20IS/Pages/default.aspx>
- Rabobank. (sd). *Rabobank Groep*. Opgeroepen op februari 11, 2013, van www.afdelingen.rabobank.nl:
<http://afdelingen.rabobank.nl/sites/GroepICT/over%20GICT/Pages/default.aspx>
- Rooij, E. d., Miltenburg, K., Steenis, R. v., & Helten, R. v. (2011). *Informatieplan project 8: De Vooruitgang*. Utrecht: Hogeschool Utrecht.

- Schreuder, P. R. (2005). *Methoden & Technieken van Onderzoek*. Den Haag: Sdu Uitgevers bv.
- Schudel, M. (2013, maart 13). Lunchsessie DevOps - 2. Aan de slag, Rabobank. Utrecht.
- Steehouder, M. (2006). *Leren communiceren: handboek voor de mondelinge en schriftelijke communicatie*. Groningen: Wolters Noordhoff.
- Verhoeven, N. (2010). *Wat is onderzoek?* Den Haag: Boom Lemma uitgevers.
- Wikipedia. (sd). Opgeroepen op februari 27, 2013, van SEPA: <http://nl.wikipedia.org/wiki/SEPA>
- Wikipedia. (sd). Opgeroepen op februari 12, 2013, van Software deployment: http://en.wikipedia.org/wiki/Software_deployment
- Willis, J. (2010, juli 16). Opgeroepen op april 25, 2013, van What Devops Means to Me: <http://www.opscode.com/blog/2010/07/16/what-devops-means-to-me/>
- Xebialabs. (sd). Opgeroepen op april 18, 2013, van Releasedashboard Deployit: <http://docs.xebialabs.com/releases/3.7/deployit/releasedashboardmanual.html>
- Xebialabs. (sd). Opgeroepen op februari 13, 2013, van Deployit product information: <http://docs.xebialabs.com>
- Xebialabs. (sd). Opgeroepen op februari 13, 2013, van Deployit 3.8.x: <http://docs.xebialabs.com/releases/3.8/deployit/startthere.html>
- Xebialabs. (2012, december 7). *Deployit Plugin*. Opgeroepen op mei 5, 2013, van tech.xebialabs.com: <http://tech.xebialabs.com/deployit-maven-plugin/3.8.4/>
- Xebialabs. (sd). *Whitepaper Continuous Delivery*. Opgeroepen op maart 1, 2013, van www.xebialabs.com: http://go.xebialabs.com/EBCDWhitepaper_Continuous_Delivery.html

Bijlagen

Bijlage A: Plan van Aanpak

Z.o.z.

Plan van Aanpak

Onderzoek Deployit

*Een onderzoek naar en de inrichting van het tool Deployit bij ADM
en B&E van Rabobank Nederland, Groep ICT*

R. van Helten



Plan van Aanpak

Onderzoek Deployit

*Een onderzoek naar en de inrichting van het tool Deployit bij
ADM en B&E van Rabobank Nederland, Groep ICT*

Auteur	R. van Helten
Versie	1.1
Datum opmaak	11 maart 2013
Student	R. van Helten
Studentnummer	1545097
Opleiding	Business IT & Management
Onderwijsinstelling	Hogeschool Utrecht
Docentbegeleider:	K. van Wensen
Stage	Afstudeerstage
Periode	4 februari tot en met 28 mei 2013
Afstudeerorganisatie	Rabobank Nederland, Groep ICT
Locatie	Utrecht, Croeselaan 18
Afdeling	ADM, PDS
Bedrijfsbegeleider:	W. Leeuwis
Opdrachtgever:	W. Leeuwis

Inhoud

Versiebeheer.....	56
Inleiding	57
1 Rabobank Groep.....	58
1.1 Groep ICT	8
1.1.1 Application Development & Maintenance	9
1.1.2 Beheer en Exploitatie.....	10
1.1.3 Strategie – The best IT enabled bank.....	10
1.1.4 DevOps	61
2 Het afstudeeronderzoek.....	62
2.1 Aanleiding	11
2.2 Probleemstelling	11
2.2.1 Centrale onderzoeksvraag	11
2.2.2 Deelvragen	12
2.3 Doelstellingen	12
2.4 Producten.....	65
2.4.1 Deelproducten adviesrapport.....	65
2.5 Projectafbakening	67
3 Context van het onderzoek	68
3.1 Projectorganisatiestructuur	68
3.2 Betrokken partijen	69
3.3 Benodigde kennis.....	70
3.4 Verwachtingen	70
4 Methodes, technieken en middelen.....	71
4.1 Methodes en technieken	71
4.1.1 Literatuuronderzoek	71
4.1.2 Kwalitatief onderzoek	71
4.1.3 Kwantitatief onderzoek.....	72
4.1.4 Analyses	72
4.2 Middelen	72
5 Globale planning.....	73

5.1	De projectactiviteiten	73
5.2	Planning.....	76
6	Risico's.....	77
7	Bedrijfs- en persoonsgegevens.....	78
8	Glossarium.....	79
9	Bronvermelding	80
10	Bijlagen.....	81
10.1	PBS	81
10.2	Organogram ADM	82

Versiebeheer

Versiedatum	Uitgave	Wijzigingen
1 maart 2013	Concept uitgave (versie 0.9)	
8 maart 2013	Eerste uitgave (versie 1.0)	<ul style="list-style-type: none">- paragraaf 1.1.3, eerste alinea- paragraaf 1.1.4, eerste alinea- paragraaf 2.2.1, centrale onderzoeksvraag- paragraaf 2.2.2, deelvragen- paragraaf 2.4.1, eerste alinea- paragraaf 2.5, tabel- paragraaf 3.1, projectorganisatiestructuur
11 maart 2013	Tweede uitgave (versie 1.1)	<ul style="list-style-type: none">- paragraaf 1.1.1, eerste alinea

Inleiding

In de periode van 4 februari tot en met 28 mei 2013 studeert de student, Roan van Helten, af aan de Hogeschool Utrecht voor de opleiding Business IT & Management bij Rabobank Nederland, Groep ICT, afdeling ADM, PDS. Dit document is het Plan van Aanpak (PvA) voor het afstudeerproject van de student. Het PvA beschrijft de benodigde aanpak om het afstudeerproject uit te voeren, waaronder de probleem- en doelstelling, projectafbakening en projectorganisatiestructuur.

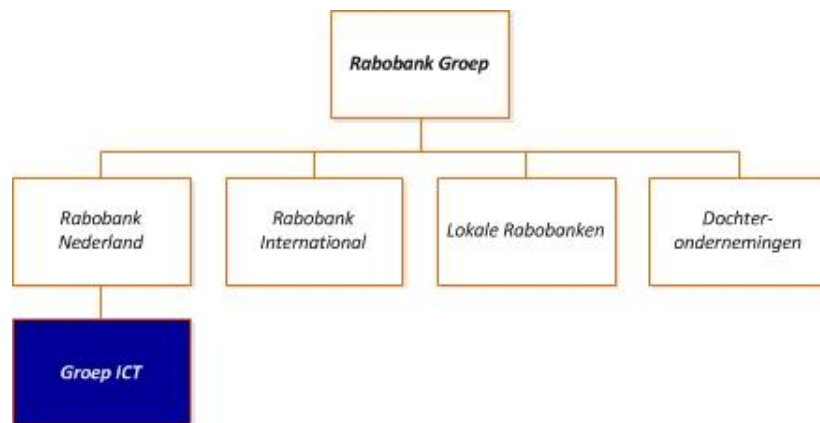
Het PvA is opgedeeld in tien hoofdstukken. Het eerste hoofdstuk beschrijft de afstudeerorganisatie, namelijk de Rabobank Groep. Het tweede hoofdstuk behandelt het afstudeeronderzoek met daarop volgend de context van het onderzoek, hoofdstuk 3, en de gebruikte methodes, technieken en middelen, hoofdstuk 4. Hoofdstuk 5 geeft de globale planning weer van het afstudeerproject. De risico's van het afstudeerproject zijn opgenomen in hoofdstuk 6. Hoofdstuk 7 bestaat uit de bedrijfs- en persoonsgegevens van de bedrijfsbegeleider, docentbegeleider en student. Het glossarium is weergegeven in hoofdstuk 8. Hoofdstuk 9 omvat de bronvermelding en tot slot geeft hoofdstuk 10 de bijlagen weer.

Voor de opmaak van het document is de literatuur Leren Communiceren geraadpleegd⁵².

⁵² Michaël Steenhouders, 2006

1 Rabobank Groep

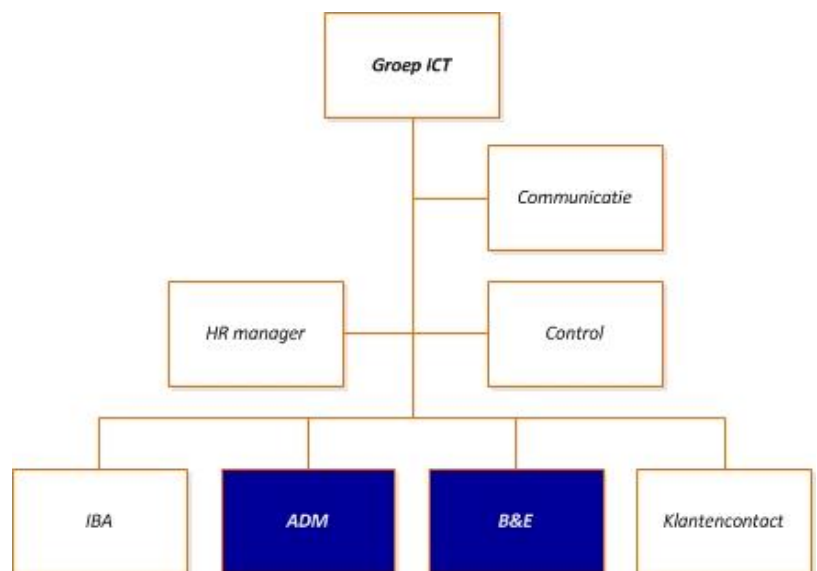
De Rabobank Groep is actief in 47 landen en bestaat uit de lokale Rabobanken, Rabobank Nederland, Rabobank International en de dochterondernemingen⁵³. De lokale Rabobanken vormen het coöperatieve kernbedrijf van de Rabobank Groep en zijn lid en aandeelhouder van de overkoepelende coöperatie Rabobank Nederland. Rabobank Nederland ondersteunt de lokale Rabobanken, waaronder op het gebied van ICT (Groep ICT). Zie *Figuur 19 | Organogram Rabobank Groep* voor een visuele weergave.



Figuur 19 | Organogram Rabobank Groep

1.1 Groep ICT

Groep ICT bestaat in totaal uit ongeveer 3.000 medewerkers en ondersteunt de Rabobank Groep met ICT-diensten waarbij continuïteit en effectiviteit centraal staan⁵⁴. GICT kent in totaal vier afdelingen, namelijk ICT Beleid & Architectuur (IBA), Application Development & Maintenance (ADM), Beheer & Exploitatie (B&E) en Klantencontact. GICT heeft drie stafafdelingen, namelijk HR, Control en Communicatie. Zie voor het organogram, *Figuur 20 | Organogram Groep ICT*. De twee afdelingen van GICT die voor deze afstudeeropdracht met name van belang zijn, zijn Application Development & Maintenance (ADM) en Beheer & Exploitatie (B&E).



Figuur 20 | Organogram Groep ICT

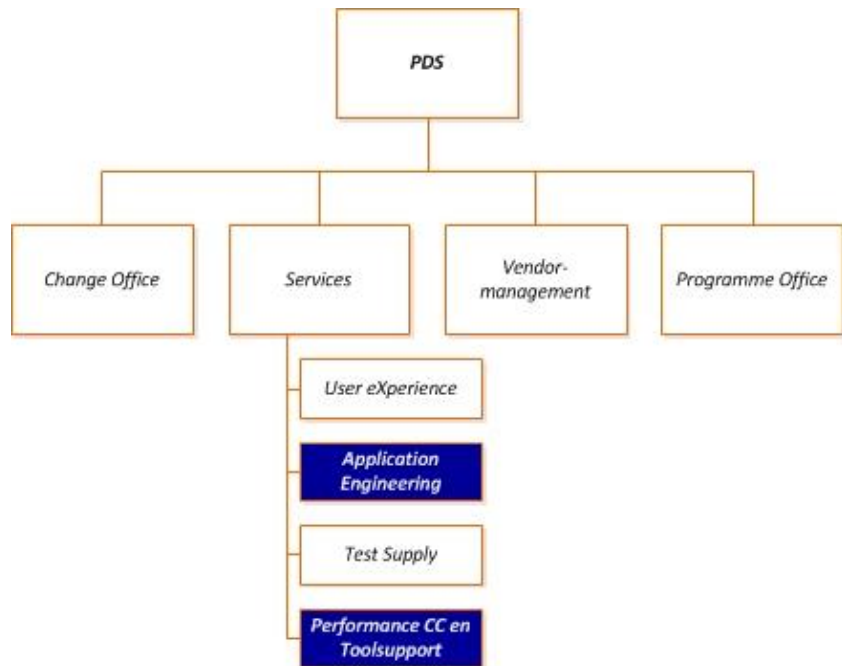
⁵³ Rabobank, 2013 (A)

⁵⁴ Rabobank, 2013 (B)

1.1.1 Application Development & Maintenance

Application Development & Maintenance (ADM) ontwikkelt en onderhoudt alle business applicaties voor de bancaire dienstverlening van Rabobank Nederland. ADM heeft een, op de business afdelingen van de Rabobank gebaseerde, indeling wat portfolio's worden genoemd. Zo is er een portfolio Betalen & Sparen, een portfolio Financiering, enz. ADM kent ook een centrale ondersteunende afdeling, PDS (Professional Development Support). Alle ontwikkel- en onderhoudswerkzaamheden vinden in de portfolio's plaats, PDS ondersteunt de portfolio's. Zie bijlagen, *paragraaf 10.2 Organogram ADM* voor het organogram van ADM.

PDS kent vier afdelingen. De afdeling Vendor Management organiseert de inkoop voor ADM, waarbij de nadruk ligt op de inhuur van externe medewerkers; voor een belangrijk deel via omvangrijke outsourcingcontracten. De afdeling Programme Office verzorgt de portfolio overstijgende administratie en rapportages. De afdeling Change Office begeleidt veranderprocessen binnen ADM. De afdeling Services ondersteunt het ontwikkelproces. De afdeling Services draagt onder meer zorg voor de ontwikkelomgevingen zoals deze binnen ADM worden gebruikt. Het zijn met name de afdelingen Application Engineering en Tool Support die hierin een rol vervullen, en relevant zijn voor het afstudeeronderzoek. Zie *Figuur 21 / Organogram PDS* voor het organogram van ADM.

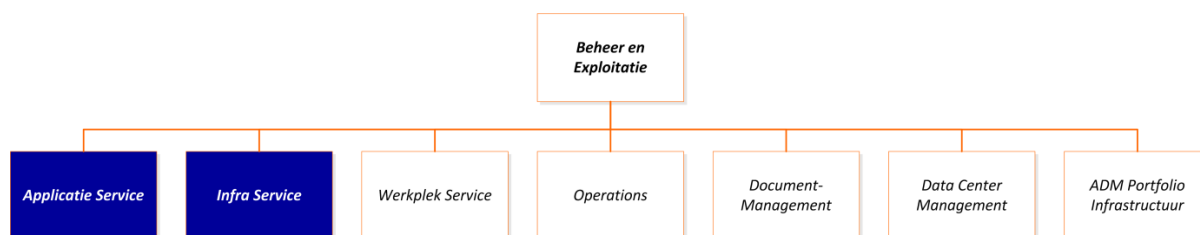


Figuur 21 | Organogram PDS

Application engineers (AE's) zijn binnen ADM verantwoordelijke voor de technische applicatiearchitectuur, en deskundig op het gebied van ontwikkelomgevingen. AE's zijn dan ook degenen die de ontwikkelomgevingen definiëren en, wanneer deze eenmaal operationeel zijn, actueel houden. De afdeling Tool Support verzorgt het technisch beheer van de ontwikkelomgevingen. Om een indruk te geven: er wordt onderscheid gemaakt in circa tien ontwikkelomgevingen (onder meer: Java, .Net, Cobol z/OS, Cobol NonStop, Siebel, AppBuilder, SAP), die in totaal bestaan uit honderden afzonderlijke tools.

1.1.2 Beheer en Exploitatie

Beheer en Exploitatie (B&E) maakt de business applicaties voor gebruik beschikbaar aan de centrale afdelingen van Rabobank Nederland en aan de door heel Nederland verspreide lokale Rabobanken, en ondersteunt dat gebruik. B&E bestaat uit zeven afdelingen, namelijk Applicatie Service, Infra Service, Werkplek Service, Operations, Document Management, Data Center Management en ADM Portfolio Infrastructuur. De afdelingen Applicatie Service en Infra Service zijn relevant tijdens dit afstudeeronderzoek. Zie *Figuur 22 | Organogram Beheer en Exploitatie* voor het organigram van B&E.



Figuur 22 | Organogram Beheer en Exploitatie

Applicatie Service beheert de applicatie portfolio. AS draagt bij aan de ambitie van Groep ICT en B&E door het leveren van Applicatief Beheer conform de afspraken met opdrachtgevers en de verwachtingen van afnemers, tegen beter dan marktconforme inspanning⁵⁵. Infra Service verzorgt de infrastructuur bij een applicatie en levert en beheert de infrastructuur omgeving. Infra Service levert onder andere Windows, z/Os en Linux omgevingen inclusief de benodigde netwerk-, storage- en middleware infrastructuur⁵⁶.

⁵⁵ Rabobank, 2013 (C)

⁵⁶ Rabobank, 2013 (D)

1.1.3 Strategie – The best IT enabled bank

Groep ICT heeft een strategie ontwikkeld om daarmee ‘the best IT enabled bank’ te worden. Deze strategie is afkomstig uit het strategieplan van 2010⁵⁷. In dit strategieplan staat beschreven wat GICT in drie jaar tijd gaat doen om ‘the best IT enabled bank’ te bereiken, oftewel:

“We willen de bank worden die ICT het slimst toepast voor haar klanten: ‘The best IT enabled bank’”

Gericht op het strategieplan komen er drie doelen naar voren die belangrijk zijn voor de afstudeeropdracht, namelijk:

- Systemen dienen stabiel te zijn;
- Kosten van ontwikkeling en beheer verlagen;
- Time to market verkorten.

Alle afdelingen, waaronder dus ook ADM en B&E, van GICT richten zich op deze strategie.

1.1.4 DevOps

DevOps is een beweging wat binnen de Rabobank nog in zijn kinderschoenen staat. De DevOps beweging is een opkomende methode binnen de ICT-wereld. Bij DevOps gaat het om de samenwerking tussen ontwikkeling en beheer. Daarbij staat ‘Dev’ voor Development (ontwikkeling) en ‘Ops’ voor Operations (beheer). DevOps realiseert technologische en procesmatige integratie tussen ontwikkeling en beheer⁵⁸.

Een andere belangrijke term met betrekking tot DevOps is continuous integration. Continuous integration (CI) maakt het mogelijk om 'builds' regelmatig te integreren met als doel dat het geen specifieke en tijdrovende taak wordt van het project. Een build is het proces dat de broncode samenvoegt en controleert of de software functioneert als een samenhangende eenheid⁵⁹.

⁵⁷ Rabobank Groep ICT, 2009

⁵⁸ Comutable, 2013

⁵⁹ Paul M. Duvall, 2007

2 Het afstudeeronderzoek

2.1 Aanleiding

Het tool Deployit, van de leverancier XebiaLabs, realiseert een geautomatiseerde deployment van een applicatie vanuit ontwikkeling, naar test, acceptatie en vervolgens productie. Dit proces wordt OTAP genoemd, wat staat voor Ontwikkeling, Test, Acceptatie en Productie. Binnen Infra Service (IS) van de afdeling B&E is het tool Deployit aangeschaft voor het programma SEPA. Het doel van SEPA, wat staat voor Single Euro Payments Area, is om in Europa één 'betaalruimte' te creëren waarin betalingen op vergelijkbare wijze functioneren⁶⁰. Het doel van het programma is het implementeren van de benodigde SEPA betaalproducten met ondersteuning van de benodigde infrastructuur en systemen. Het tool Deployit is aangeschaft om de benodigde infrastructuur efficiënter en productiever te configureren en op te zetten binnen de productieomgeving. Eind 2012 is een stuurgroep binnen ADM opgezet die Deployit heeft opgepakt ter onderzoek. Deze stuurgroep stelt Deployit ter discussie met daarbij het onderzoek naar de inpassing van Deployit voor alle Java-applicaties van de Rabobank, met als uiteindelijk doel een generieke implementatie van het tool binnen Groep ICT.

2.2 Probleemstelling

Deployit ondersteunt het proces dat de brug vormt tussen development (ADM) en operations (B&E). Het is van groot belang dat het gebruik van het tool in beide afdelingen goed op elkaar is afgestemd, en dat het life cycle management van het tool rekening houdt met de eisen en wensen van beide afdelingen. De kern van de opdracht is de inpassing van Deployit in de processen van ADM, op een wijze die leidt tot een soepele aansluiting op de processen van B&E. Dit impliceert een inventarisatie en analyse van processen (m.n. development en het ondersteunende proces change & configuration management) en het opstellen van voorstellen voor de inpassing van Deployit in die processen.

2.2.1 Centrale onderzoeksvraag

De centrale onderzoeksvraag van het afstudeeronderzoek luidt als volgt:

Hoe kan het tool Deployit ingericht worden in de huidige situatie van de ontwikkel- en beheeromgeving van Java-applicaties binnen de afdelingen ADM en B&E zodat de benodigde doorlooptijd van het deploymentproces gereduceerd is?

Deployit	Een tool om een deployment van een applicatie te automatiseren binnen zowel de ontwikkelomgeving, testomgeving, acceptatieomgeving en productieomgeving.
Huidige situatie	De huidige situatie bestaat uit de huidige processen, actoren, bevoegdheden, verantwoordelijkheden, (autorisatie) beveiliging, (wet)regelgeving, (technische) functiescheiding, infrastructuur en configuratiemanagement van een vastgesteld kader.
Ontwikkel- en beheeromgeving	De ontwikkel- en beheeromgeving bestaat uit de ontwikkeling, het testen, de acceptatie en deployment van java-applicaties.

⁶⁰ Wikipedia, 2013 (A)

Java-applicaties	Applicaties die zijn geprogrammeerd volgens de programmeertaal Java.
ADM	Application Development & Maintenance (ADM) ontwikkelt en onderhoudt alle business applicaties voor de bancaire dienstverlening van Rabobank Nederland.
B&E	Beheer en Exploitatie (B&E) maakt de business applicaties voor gebruik beschikbaar aan de centrale afdelingen van Rabobank Nederland en aan de door heel Nederland verspreide lokale Rabobanken, en ondersteunt dat gebruik.
Deploymentproces	Het proces wat bestaat uit alle activiteiten die het mogelijk maken om een applicatie klaar voor gebruik te maken ⁶¹ .

2.2.2 Deelvragen

Huidige situatie

- Hoe zijn de organisatorische aspecten vormgegeven voor de ontwikkeling en beheer van een Java-applicatie binnen de huidige situatie?
- Hoe zijn de processen vormgegeven voor de ontwikkeling en beheer van een Java-applicatie binnen de huidige situatie?
- Hoe is de infrastructuur vormgegeven voor de ontwikkeling en beheer van een Java-applicatie binnen de huidige situatie?
- Hoe wordt Application Lifecycle Management (ALM) toegepast bij de Java-applicaties binnen de huidige situatie?
- Hoe wordt configuratiemanagement toegepast bij de Java-applicaties binnen de huidige situatie?

Gewenste situatie

- Welke gewenste situaties zijn mogelijk gezien de functionaliteiten van Deployit?
 - o Wat zijn/is de veranderingen/impact van de gewenste situaties ten opzichte van de huidige situatie?
 - o Wat zijn de risico's van de gewenste situaties?
 - o Wat zijn de kosten en baten van de gewenste situaties?
- Hoe kan Application Lifecycle Management (ALM) voor Deployit ingericht worden met kijk naar de wensen en eisen van de afdelingen B&E en ADM?

Vervolgprojecten

- Welke projecten en/of werkzaamheden moeten uitgevoerd te worden als vervolg van dit onderzoek?

⁶¹ Wikipedia, 2013 (C)

DevOps (optioneel)

- Wat is DevOps?
- Wat is de aanleiding voor de populariteit van DevOps?
- Is DevOps toepasbaar binnen een bank?
- Hoe verhoudt Deployit zich binnen DevOps en in welke mate kan Deployit hieraan een bijdrage leveren?

2.3 Doelstellingen

De resultaten van het onderzoek creëren duidelijkheid en inzicht in de huidige situatie van de ontwikkeling en beheer van Java-applicaties. Vanuit deze huidige situatie moeten meerdere scenario's beschreven zijn die het tool Deployit plaatsen binnen de huidige situatie. De scenario's geven de benodigde aandachtspunten weer voor de implementatie van het tool. De verschillende scenario's geven meerdere opties weer voor de implementatie met daarbij per scenario een kosten/baten analyse en een risicoanalyse. Vanuit de huidige situatie en potentiële gewenste situaties dient er een onderbouwd advies opgesteld te zijn dat aangeeft welke gewenste situatie de beste keuze is en of de implementatie van het tool wel of niet aan te raden is. Het advies zal gebruikt worden door de afdelingen B&E en ADM. Deze afdelingen zullen de implementatie van het tool Deployit verzorgen, indien het advies dit aanraadt.

De doelstellingen zijn SMART gedefinieerd, wat staat voor Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden. Concreet zijn dit de volgende doelstellingen:

- Voor 28 mei 2013 heeft de student de huidige situatie van de ontwikkel- en beheeromgeving voor Java-applicaties op organisatie- en procesniveau in kaart gebracht.
- Voor 28 mei 2013 heeft de student minimaal drie gewenste situaties beschreven met daarbij voor iedere situatie de aangegeven veranderingen, impact, kosten/baten analyse en een risicoanalyse.
- Voor 28 mei 2013 heeft de student een advies opgesteld voor de keuze van een gewenste situatie.

2.4 Producten

Het eindresultaat van het afstudeeronderzoek is een adviesrapport aan de afdelingen ADM en B&E van Groep ICT van Rabobank Nederland. De deelproducten van het adviesrapport zijn beschreven in *paragraaf 2.4.1 Deelproducten adviesrapport*.

Naast het adviesrapport levert de student voor de Hogeschool Utrecht verschillende producten op. In een vorige fase is er al een voorstel van de afstudeeropdracht beschreven. Dit document behandelt het Plan van Aanpak. *Tabel 12 | Producten* geeft de benodigde producten weer met daarbij de deadline van de oplevering.

Naam product	Beschrijving	Deadline
Scriptie	Scriptie van het afstudeeronderzoek	28 mei 2013
Beoordeling	Beoordeling van de student door de bedrijfsbegeleider	31 mei 2013
Presentatie	Presentatie tijdens de afstudeerzitting.	Specifieke datum onbekend, periode: 17 juni t/m 28 juni 2013

Tabel 12 | Producten

2.4.1 Deelproducten adviesrapport

Deze paragraaf behandelt de deelproducten van het adviesrapport. De producten zijn per kopje beschreven. Iedere beschrijving behandelt een korte toelichting en eventuele grenzen en randvoorwaarden. In de bijlagen, *paragraaf 10.1 PBS*, is de Product Breakdown Structure (PBS) weergegeven van het adviesrapport volgens de methodiek van Prince2⁶².

Organisatiebeschrijving

De organisatiebeschrijving behandelt de benodigde kennis van de organisatie zodat de afstudeeropdracht uitgevoerd kan worden. De organisatieomschrijving omvat een introductie van de organisatie met daarbij specifiek ingezoomd op de afdelingen waar de student actief is en zijn onderzoek uitvoert.

Achtergrondinformatie

Dit deelproduct omvat de benodigde achtergrondinformatie voor de uitvoering van het afstudeeronderzoek. De achtergrondinformatie bestaat uit een beschrijving van de onderwerpen Application Lifecycle Management (ALM), (Software) configuratiemanagement, Deployit en Java-applicaties. De vergaring van de informatie zal enkel gebruikt worden als basis van de benodigde kennis. Diepere analyses over het onderwerp zullen uitgevoerd worden indien dit is aangegeven bij de projectactiviteiten.

Huidige situatie

Dit deelproduct beschrijft de huidige situatie van ontwikkel- en beheeromgeving die geraakt zullen worden door het tool Deployit. Dit deelproduct heeft als doel om de behandelde deelvragen in *paragraaf 2.2.2 Deelvragen* onder het kopje *huidige situatie* te behandelen en te zoeken naar een passend antwoord. De huidige situatie behandelt organisatorische aspecten, processen, infrastructuur, Application Lifecycle Management en (software) configuratiemanagement. De

⁶² Peter Janssen, 2007

organisatorische aspecten omvatten daarbij de aanwezige actoren, bevoegdheden, verantwoordelijkheden, (autorisatie) beveiliging, (wet)regelgeving en (technische) functiescheiding. Tijdens het in kaart brengen van dit deelproduct houdt de student rekening met de eisen en wensen van de twee afdelingen B&E en ADM. De eisen en wensen zullen meegenomen worden als input voor de gewenste situaties. Het onderzoek naar de huidige situatie wordt uitgevoerd bij het systeem BBS. Vanuit de vergaarde resultaten uit dit onderzoek ontwerpt de student een generieke huidige situatie.

Gewenste situatie

Dit deelproduct behandelt minstens drie gewenste situaties. Iedere gewenste situatie geeft een mogelijkheid weer voor de inrichting van Deployit binnen de eerder beschreven generieke huidige situatie. Een situatie behandelt daarbij de verandering, impact, risicoanalyse en een kosten/baten analyse van de beschreven inrichting.

DevOps

Dit deelproduct bestaat uit een beschrijving van de DevOps beweging. Het product geeft een korte introductie weer van het onderwerp, wat het te bieden heeft voor een bank zoals Rabobank en hoe het tool Deployit in relatie staat tot DevOps.

Application lifecycle management Deployit

Dit deelproduct bestaat uit een inrichting van de Application Lifecycle Management (ALM) van Deployit. Deployit is een (ontwikkel)tool waardoor het vereist is zijn levenscyclus in kaart te brengen. Daarbij kan onder andere gedacht worden aan wie het tool onderhoudt, beheert en wie de versiebeheer uitvoert. De inrichting van ALM zal op hoofdlijnen uitgewerkt worden. Daarbij behandelt de student in ieder geval de onderwerpen beheer, ontwikkeling en governance.

Vervolgprojecten

Na voltooiing van dit afstudeeronderzoek worden er waarschijnlijk vervolgprojecten en/of werkzaamheden uitgevoerd. Dit deelproduct behandelt de benodigde vervolgprojecten en/of werkzaamheden die uitgevoerd moeten worden na voltooiing van het onderzoek. Een vervolgproject bestaat uit een korte introductie van het project, onderbouwing van het project en de benodigde vakmensen. De vervolgprojecten worden opgenomen in een globale planning.

Conclusie

De conclusie geeft antwoord op de centrale vraag en deelvragen. Het antwoord is verkregen uit de ondervonden resultaten.

Advies en discussie

Dit deelproduct bestaat uit een advies en een discussie. Het advies behandelt de verkregen resultaten en geeft een advies betreffende deze resultaten aan de Rabobank. De discussie stelt de verkregen resultaten ter discussie.

Evaluatie

Dit deelproduct bestaat uit een evaluatie van het afstudeeronderzoek. De evaluatie omvat de ondervonden problemen, wat goed ging en wat beter kon tijdens het project.

2.5 Projectafbakening

Deze paragraaf beschrijft de projectafbakening (scope) van de afstudeeropdracht. De scope is in *Tabel 13 | Projectafbakening* beschreven. Per onderwerp is er aangegeven wat er binnen en buiten de scope valt.

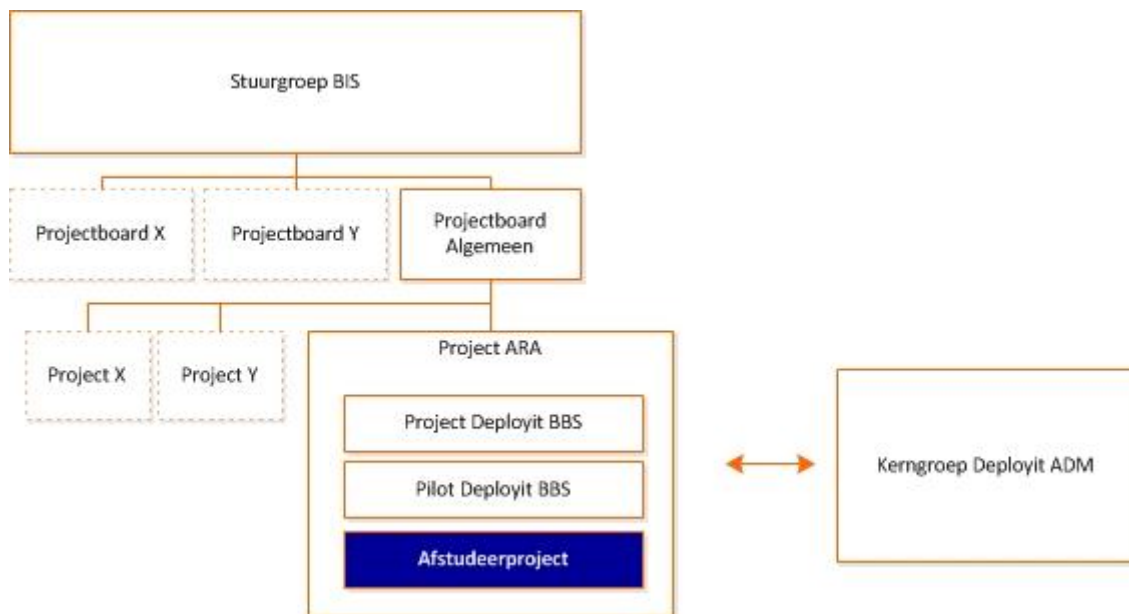
Onderwerp	Binnen de scope	Buiten de scope
Applicaties	Java-applicaties op het Linux platform	Overige applicaties zoals .Net en Cobol.
Organisatiebeschrijving	Organisatieonderdelen die raakvlak hebben met de afstudeeropdracht	Organisatieonderdelen die geen raakvlak hebben met de afstudeeropdracht
Achtergrondinformatie	Java-applicaties, Application Lifecycle Management, (Software) configuratiemanagement, Deployit (Functionaliteiten)	Deployit (Technische aspecten)
Huidige situatie	Onderdelen Organisatorische aspecten, Aanwezige actoren en rollen, Bevoegdheden en verantwoordelijkheden, (Autorisatie)beveiliging, (Wet)regelgeving, (Technische) functiescheiding, Processen, (Software) Configuratiemanagement en Java-infrastructuur Overige Gericht op ervaring uit BBS Generieke huidige situatie (gebaseerd op huidige situatie BBS)	
Gewenste situatie	Onderdelen Veranderingen en impact, Risicoanalyse, Kosten/baten analyse Overige Minstens drie gewenste situaties	Algemeen Functioneel Ontwerp Technisch Ontwerp Informatieplan Implementatieplan Deployit Configureren van (virtuele) hardware en infrastructuur.
Application Lifecycle Management	Deployit	Bedrijfsapplicaties
DevOps beweging	Optioneel	Buiten de scope indien geval van tijdsnood project.
Overige	Conclusie, Advies, Discussie, Evaluatie	Vooronderzoek Deployit, een onderzoek naar soortgelijke tools

Tabel 13 | Projectafbakening

3 Context van het onderzoek

3.1 Projectorganisatiestructuur

Deze paragraaf behandelt de aanwezige projectorganisatiestructuur. De afstudeeropdracht heeft (in)directe relaties met Stuurgroep BIS, Projectboard Algemeen, Project ARA, Project Deployit BBS, Pilot Deployit BBS en de kerngroep Deployit ADM. Zie *Figuur 23 | Projectorganisatiestructuur* voor een visuele weergave van de relaties. Vanwege de recente opzet van de groepen kan het voorkomen dat de groepen wijzigen tijdens het afstudeertraject.



Figuur 23 | Projectorganisatiestructuur

Stuurgroep BIS

BIS, wat staat voor Bedrijfs Informatie Systeem, is de stuurgroep dat de infrastructuurwijzigingen bestuurt. Het programma bestaat uit een aantal projectboards waaronder de projectboard Algemeen.

Projectboard Algemeen

De projectboard Algemeen heeft contact met stuurgroep BIS en Kerngroep Deployit ADM. De functie van de projectboard is de sturing en ondersteuning van project ARA.

Project ARA

Het project ARA, wat staat voor Application Release Automation, richt zich op de invoering van Deployit binnen de Rabobank. Binnen dit project vallen kleinere projecten, waaronder Project Deployit BBS, het afstudeerproject en een pilot van Deployit bij BBS.

Kerngroep Deployit ADM

Deze kerngroep heeft als belangrijkste doel om het gebruik van het tool Deployit binnen ADM in goede banen te leiden. Andere aspecten van de kerngroep zijn het generiek invoeren van Deployit en bijsturen waar nodig.

Afstudeerproject

Het afstudeerproject wordt uitgevoerd door de student Roan van Helten. De student volgt de opleiding Business IT & Management aan de Hogeschool Utrecht. De opdrachtgever en de bedrijfsbegeleider zijn dezelfde persoon, namelijk Wil Leeuwis. De bedrijfsbegeleider adviseert de student tijdens het afstudeertraject indien de student dit aangeeft. Het afstudeerproject is voor een groot deel een onderdeel van project ARA. Tijdens dit afstudeerproject zal de student contact hebben met de projectleider van dat project.

Het onderzoek

Zoals de literatuur Wat is onderzoek?⁶³ beschrijft dient de student, de adviseur, tijdens het onderzoek een open en onafhankelijke houding aan te nemen. De adviseur dient te beschikken over de juiste kennis, zie *paragraaf 3.3 Benodigde kennis*, en vaardigheden, zie *hoofdstuk 4 Methodes, technieken en middelen*. Met kijk naar de literatuur Adviseren als tweede beroep⁶⁴ heeft de betreffende advies de competenties zelfstandigheid, proactief, oog voor detail en resultaatgericht. De opdracht geeft de competenties aan waar de adviseur aan moet voldoen, namelijk samenwerken, analytisch, communicatief vaardig, zelfstandigheid en resultaatgericht. In vergelijking tot de kenmerken van de adviseur en deze die gesteld zijn in de opdrachtbeschrijving, is de betreffende adviseur de geschikte kandidaat.

3.2 Betrokken partijen

In deze paragraaf zijn de betrokken partijen beschreven die naar verwachting benodigd zijn tijdens het afstudeeronderzoek. De betrokken partijen zijn onderverdeeld in projectgroepen, afdelingen en benodigde bedrijfsfuncties.

Projectgroep

Project Deployit
Projectboard
Stuurgroep Deployit ADM

Afdelingen

B&E – Application Service
B&E – Infra Service
ADM – Configuratiemanagement
ADM – Performance CC en Toolsupport
ADM – Tool Supply
ADM – Application Engineering

Bedrijfsfuncties

Applicatiebeheerders
Application Engineers
Configuratie managers
Managers
Ontwikkelaars
Testers

⁶³ Nel Verhoeven, 2010

⁶⁴ Hannah Nathans, 2004

3.3 Benodigde kennis

Voor de uitvoering van het afstudeeronderzoek dient de student op de hoogte te zijn van verschillende onderwerpen. In *Tabel 14 | Onderwerpen* zijn de benodigde onderwerpen weergegeven.

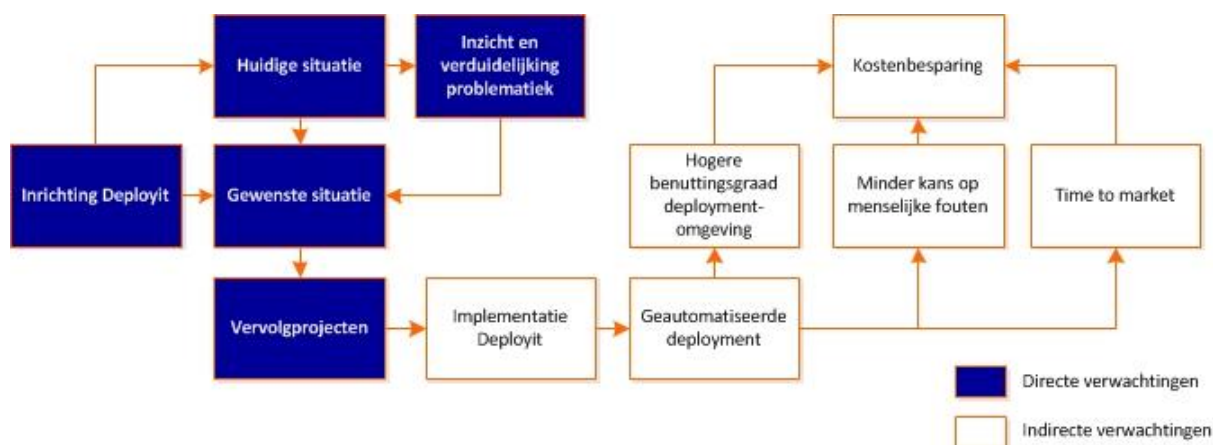
(Software) configuratiemanagement	Deployit
Application development	Deployment
Application Lifecycle Management	DevOps
Application management	Geld Nodig Bedrijven (GNB)
Bedrijven Bediening Systeem (BBS)	ITIL
Change management	Java (applicaties)
Continuous deployment	OTAP
Continuous integration	Procesoptimalisatie

Tabel 14 | Onderwerpen

3.4 Verwachtingen

De kern van dit onderzoek is de inrichting van Deployit binnen de organisatie van de Rabobank. De inrichting van Deployit levert een huidige situatie en minstens drie gewenste situaties op. De huidige situatie en het daaruit verkregen inzicht en verduidelijking van de problematiek dragen bij aan de creatie van de gewenste situaties. Uit de verkregen gewenste situaties kunnen er vervolgprojecten opgezet worden waarvan het uiteindelijke doel de brede implementatie van Deployit binnen Groep ICT is. Op het moment dat Deployit geïmplementeerd is kunnen er automatische deployments van Java-applicaties uitgevoerd worden. Een geautomatiseerde deployment brengt, naar verwachting, een hogere benuttingsgraad van de deploymentomgeving, kortere time to market en minder kans op menselijke fouten met zich mee. Deze drie verwachtingen leveren een kostenbesparing op ten opzichte van het huidige deployment proces.

In *Figuur 24 | Verwachtingsmodel* zijn de verwachtingen weergegeven in een verwachtingsmodel. Het verwachtingsmodel maakt onderscheid in directe en indirecte verwachtingen. De directe verwachtingen zijn de verwachtingen die direct af te leiden zijn na voltooiing van het onderzoek. De indirecte verwachtingen zijn de verwachtingen die na voltooiing van de beschreven vervolgprojecten gerealiseerd kunnen worden.



Figuur 24 | Verwachtingsmodel

4 Methodes, technieken en middelen

Voor de uitvoering van het afstudeerproject zal er gebruik gemaakt worden van verschillende methodes, technieken en instrumenten. Deze zijn in dit hoofdstuk beschreven.

4.1 Methodes en technieken

Het onderzoek is opgezet met behulp van de literatuur Wat is onderzoek⁶⁵. Het onderzoek is op hoofdlijnen een interpretatief onderzoek, wat betekent dat het onderzoek kwalitatief is van karakter. Omdat er ook gebruik gemaakt zal worden van bronnenstudie (literatuuronderzoek) en, indien gewenst, kwantitatief onderzoek zal er een triangulaire onderzoeksmethode gehanteerd worden.

4.1.1 Literatuuronderzoek

Voor de vergaring van informatie zal er een literatuuronderzoek uitgevoerd worden. Het literatuuronderzoek behandelt begrippen en methodieken die nog niet of onvoldoende beheerst worden door de student. Voor het literatuuronderzoek zal er gebruik worden gemaakt van The Big6-methode en zullen twee zoekmachines gebruikt worden die meerdere databanken doorzoeken, namelijk LUCAS en Google Scholar⁶⁶. LUCAS is een functionaliteit die door de Hogeschool Utrecht wordt aangeboden. Door een zoekopdracht te doen met LUCAS (The Library's Uniform and Central Access to Sources) worden tegelijkertijd meerdere bronnen doorzocht⁶⁷.

4.1.2 Kwalitatief onderzoek

Tijdens het kwalitatief onderzoek zullen er open interviews uitgevoerd worden. Een open interview kent drie soorten, namelijk:

- Ongestructureerd interview (diepte interview)
Vrijheid om dieper in te gaan op een bepaald onderwerp tijdens het gesprek.
- Half gestructureerd interview
Vrijheid in combinatie met structuur.
- Gestructureerd interview
Persoonlijke enquête.

Het soort interview dat toegepast wordt kan variëren per soort gesprek dat uitgevoerd wordt. Het meest voor de hand liggende soort is een half gestructureerd interview waarbij er een lijst opgezet is met punten die besproken moeten worden. De geïnterviewde bespreekt per punt zijn eigen kennis over dat onderwerp. Het detailniveau van de kennis over een bepaald onderwerp kan variëren waardoor tijdens het gesprek bepaald wordt of er wel of niet meer vragen gesteld worden.

⁶⁵ Nel Verhoeven, 2010

⁶⁶ Google Scholar, 2013

⁶⁷ Hogeschool Utrecht, 2013

4.1.3 Kwantitatief onderzoek

Naast het literatuuronderzoek en kwalitatief onderzoek is er de mogelijkheid tot een kwantitatief onderzoek. Op dit moment is er geen aanleiding tot gebruik van dit type onderzoek. Voor het kwantitatief onderzoek kan er gebruik gemaakt worden van enquêtes. Enquêtes hebben een groter bereik binnen de onderzoeksgroep. Indien nodig, kan er tijdens het afstudeeronderzoek voor gekozen worden om kwantitatief onderzoek toe te passen.

4.1.4 Analyses

Vanuit het literatuuronderzoek, kwalitatief- en kwantitatief onderzoek zullen er resultaten worden vergaard. Deze resultaten zullen geanalyseerd worden. Deze paragraaf behandelt toepasbare analyse technieken die nuttig kunnen zijn tijdens het afstudeeronderzoek.

Impactanalyse

De impactanalyse geeft de mate van impact weer van een verandering. In de context van de afstudeeropdracht zal de impactanalyse zich richten op de mate van verandering binnen de gewenste situatie ten opzichte van de huidige situatie. Daarnaast zal de analyse de aandachtspunten aanwijzen.

Kosten/baten analyse

Een kosten/baten analyse onderzoekt de aanwezige kosten en baten van de invoering van een nieuw idee. In de context van de afstudeeropdracht zal de kosten/baten analyse toegepast worden bij de verschillende gewenste situaties. Een kosten/baten analyse geeft duidelijk aan wat de kosten en baten zijn in een bepaalde tijdsbestek. Vanuit een kosten/baten analyse kan er een beslissing gemaakt worden rondom de keuze van een gewenste situatie.

Verandermanagement

Verandermanagement onderzoekt in welke mate de verandering van een nieuw idee, in dit geval de invoering van het tool Deployit, uitgevoerd dient te worden. Verandermanagement behandelt daarbij de organisatie, de verandering en de uitvoerder(s). Tijdens deze methode zal er gebruik worden gemaakt van de literatuur Leren Veranderen⁶⁸. Verandermanagement zal toegepast worden tijdens de opzet van de gewenste situaties.

4.2 Middelen

Voor de uitvoering van het afstudeeronderzoek zijn middelen beschikbaar gesteld vanuit de organisatie. In *Tabel 15 | Middelen* zijn deze middelen aangegeven.

Laptop (Randapparatuur, Office en Visio 2010)	Toegang tot documentatie (o.a. intranet en wiki)
Printer	Werkplek
Schrijfgerei (pen, kladblok, etc.)	

Tabel 15 | Middelen

⁶⁸ Léon de Caluwé, 2006

5 Globale planning

Dit hoofdstuk behandelt de benodigde projectactiviteiten en geeft daarbij deze activiteiten weer in een planning.

5.1 De projectactiviteiten

De benodigde projectactiviteiten zijn per fase ingedeeld en zijn er overige activiteiten die indirect baat hebben voor het afstudeeronderzoek.

Vorbereiden

Geschatte benodigde uren: 70 uur
Activiteiten: Plan van Aanpak schrijven
Stagecontract opstellen

De voorbereiding bestaat uit twee activiteiten, namelijk het schrijven van een Plan van Aanpak en het opstellen van een stagecontract. Na voltooiing van de voorbereiding kan er gestart worden met fase 1.

Fase 1: Vooronderzoek uitvoeren

Geschatte benodigde uren: 50 uur
Activiteiten: Organisatiebeschrijving schrijven.
Benodigde achtergrondinformatie beschrijven

Fase 1 bestaat uit twee activiteiten, namelijk het schrijven van een organisatiebeschrijving en de benodigde achtergrondinformatie. De benodigde achtergrondinformatie bestaat uit het beschreven deelproduct *Achtergrondinformatie* van *paragraaf 2.4.1 Deelproducten adviesrapport*. Voor de volledige beschrijving van de achtergrondinformatie dient er een literatuuronderzoek uitgevoerd te worden. Het literatuuronderzoek onderzoekt welke informatie al aanwezig is over de termen Application Lifecycle Management (ALM), (Software) configuratiemanagement en Java-applicaties.

Fase 2: Huidige situatie ontwerpen

Geschatte benodigde uren: 130 uur
Activiteiten: Organisatorische aspecten beschrijven
Processen beschrijven
Java-infrastructuur beschrijven
ALM beschrijven
Configuratiemanagement beschrijven
Generieke huidige situatie ontwerpen

Fase 2 bestaat uit het ontwerp van de huidige situatie. Het ontwerp van de huidige situatie bestaat uit een onderzoek naar de organisatorische aspecten, processen, aanwezige Java-infrastructuur, ALM van de business applicaties en configuratiemanagement. Voor het in kaart brengen van deze aspecten dienen interviews gevoerd te worden met zowel ADM personeel als B&E personeel. ADM richt zich daarbij op de ontwikkeling van de Java-applicaties en B&E op het beheer van deze applicaties. Het personeel waarbij interviews gevoerd worden zullen daardoor bestaan uit onder andere ontwikkelaars, testers, configuratiemanagers en beheerders van Java-applicaties. Omdat er vele java-applicaties aanwezig zijn binnen Rabobank zal de huidige situatie gebaseerd worden op het systeem BBS. Vanuit de verkregen resultaten van BBS stelt de student een generieke huidige situatie op.

Fase 3: Gewenste situatie ontwerpen

Geschatte benodigde uren: 90 uur
Activiteiten: Veranderingen en impact beschrijven
Risicoanalyse beschrijven
Kosten/baten analyse uitvoeren beschrijven

Fase 3 bestaat uit de inrichting van minimaal drie gewenste situaties. De gewenste situaties beschrijven de inrichting van het tool Deployit binnen de huidige situatie. De veranderingen en impact van de bepaalde inrichting zal beschreven worden met daarbij een risicoanalyse en kosten/baten analyse.

Fase 4: ALM Deployit ontwerpen

Geschatte benodigde uren: 30 uur
Activiteiten: Governance beschrijven
Beheer beschrijven
Ontwikkeling beschrijven

Fase 4 bestaat uit de inrichting van het Application Lifecycle Management (ALM) voor Deployit. Omdat het tool Deployit zowel binnen het domein van de afdeling ADM als de afdeling van B&E valt dient het ALM afgestemd te worden met beide afdelingen. Daarvoor dienen er interviews gehouden te worden met onder andere application engineers en medewerkers van de afdeling Application Service (AS).

Fase 5: DevOps beweging (optioneel)

Geschatte benodigde uren: 50 uur
Activiteiten: DevOps beweging beschrijven
DevOps in verhouding tot Deployit beschrijven

Fase 5 bestaat uit een onderzoek naar de DevOps beweging. Het tool Deployit draagt bij aan deze beweging. Er zal daarom onderzocht worden in welke mate DevOps aanwezig is binnen de Rabobank en hoe de Rabobank profijt kan hebben van deze beweging. Daarnaast zal er onderzocht worden in welke mate een tool zoals Deployit een bijdrage kan leveren aan deze beweging. Deze fase is optioneel, wat betekent dat deze fase niet wordt uitgevoerd indien het afstudeerproject in tijdsnood komt. Fase 5 kan uitgevoerd worden op het moment dat de gewenste situaties (fase 3) in kaart gebracht zijn.

Fase 6: Afronding

Geschatte benodigde uren: 30 uur
Activiteiten: Advies en discussie schrijven
Evaluatie schrijven

Fase 6 bestaat uit het schrijven van een advies, discussie en evaluatie. Dit is de laatste fase van het onderzoek.

Scriptie

Geschatte benodigde uren: 160 uur
Activiteiten: Scriptie schrijven

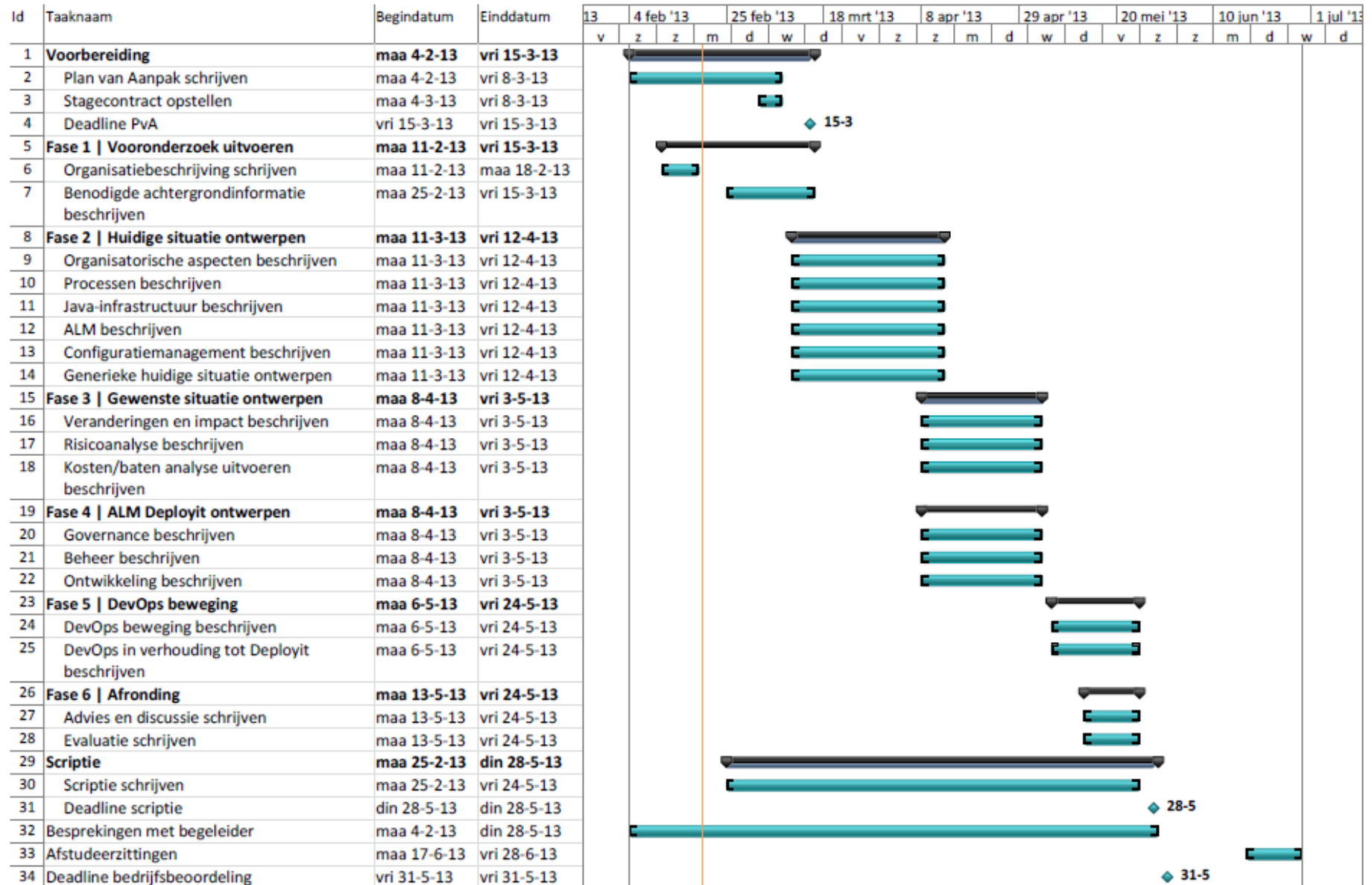
Het schrijven van de scriptie is een activiteit naast de aangegeven fases.

Overige activiteiten

Geschatte benodigde uren: 30 uur
Activiteiten: Gevarieerd

Naast de fases van het afstudeeronderzoek zullen er parallel overige activiteiten uitgevoerd worden. Deze overige activiteiten zullen een indirecte baat hebben voor het afstudeeronderzoek. Niet alle overige activiteiten zijn op dit moment duidelijk. Zoals het nu bekend is zal er meegeholpen worden met een pilot (proefimplementatie) van het tool Deployit bij het systeem BBS. Daarnaast zal de student aanwezig zijn bij stuurgroep bijeenkomsten, projectgroep bijeenkomsten en overige bijeenkomsten die relevant zijn voor de implementatie van het tool Deployit. Tijdens deze bijeenkomsten zal de student de rol van notulist op zich nemen.

5.2 Planning



6 Risico's

Dit hoofdstuk beschrijft de risico's die kunnen ontstaan tijdens de uitvoering van het onderzoek. De risico's zijn beschreven in *Tabel 16 | Risicoanalyse*. Ieder risico's beschrijft de bijhorende kans, impact, prioriteit, maatregel en uitvalscenario. De kans en impact is een cijfer van 1 tot en met 5, daarbij is 1 laag en 5 hoog. De prioriteit is de kans maal de impact. De maatregel is hetgeen wat wordt gedaan om de kans en impact van het risico te minimaliseren. Het uitvalscenario beschrijft het scenario op het moment dat het risico zich voordoet.

Risico	Kans	Impact	Prioriteit	Maatregel	Uitvalscenario
De belangen van een of meerdere (project)groepen die in relatie staan tot de afstudeeropdracht wijzigen.	4	3	12	Wijzigingen met betrekking tot de belangen nauwkeuring in de gaten houden.	De veranderde belangen dienen in kaart gebracht te worden met daarbij de mogelijke impact op het afstudeeronderzoek.
Het project Deployit, waarvan de afstudeeropdracht een onderdeel is, staakt het project.	2	5	10	De probleemstelling van het onderzoek onafhankelijk van een ander project formuleren.	Onderzoek plaatsen onder een ander project of het project individueel en onafhankelijk voortzetten.
Benodigde medewerkers nemen niet deel aan het afstudeeronderzoek	4	2	8	Duidelijk communiceren wat het doel en belang is van het onderzoek.	Zoeken naar een, gericht op zijn of haar functie, soortgelijke medewerker.
Het project raakt in tijdsnood	2	4	8	Wekelijkse planning opstellen zodat een vertraging direct kan worden aangepakt.	Fase 5 DevOps wordt niet uitgevoerd.
Projectboard, ADM of B&E staakt het project	1	5	5	Communicatie behouden met (in)direct betrokken partijen.	Afstudeeropdracht individueel voortzetten en/of probleemstelling wijzigen.
Student raakt ziek	1	5	5	Marges opnemen in de planning.	Project raakt in tijdsnood, planning wijzigen.

Tabel 16 | Risicoanalyse

7 Bedrijfs- en persoonsgegevens

Stagedocent:

Naam: Koos van Wensen

Telefoon: *[wegens privacy redenen verwijderd]*

Email: koos.vanwensen@hu.nl

Bedrijfsbegeleider / Opdrachtgever:

Naam: Wil Leeuwis

Telefoon: *[wegens privacy redenen verwijderd]*

Email: w.leeuwis@rn.rabobank.nl

Student:

Naam: Roan van Helten

Telefoon: *[wegens privacy redenen verwijderd]*

Email: roan.vanhelten@student.hu.nl

8 Glossarium

Dit hoofdstuk beschrijft de begrippen en onderwerpen die terugkomen in de uitvoering van het onderzoek. De begrippen en onderwerpen zijn beschreven zodat er geen verwarring kan zijn over de betekenis.

.NET	Een raamwerk voor ontwikkelaars om software te schrijven ⁶⁹ .
Application Lifecycle Management (ALM)	ALM omvat de volledige tijd vanaf het idee tot en met het einde van de applicatie. Anders geformuleerd: de totale tijd dat de organisatie geld spendeert aan de applicatie ⁷⁰ .
Application Release Automation (ARA)	Vormt de brug tussen ontwikkel- en beheerorganisatie voor het automatisch deployen van software ⁷¹ .
Bedrijven Bediening Systemen (BBS)	BBS is het kernsysteem voor het vertrekken van bedrijfsleningen (kredieten).
Change management	Het beheerst doorvoeren van wijzigingen waarbij de betrokkenen inzicht hebben in de status ervan ⁷² .
Cobol	Een zakelijk georiënteerde programmeertaal om software te schrijven ⁷³ .
Continuous Deployment	Het frequent releasen van gebouwde software onderdelen ⁷⁴
Continuous Integration (CI)	Maakt het mogelijk om 'builds' regelmatig te integreren met als doel dat het geen specifieke en tijdrovende taak wordt van het project. ⁷⁵
Deployit	Een tool die een geautomatiseerde deployment realiseert.
Deployment	Alle activiteiten die het mogelijk maken om een applicatie klaar voor gebruik te maken ⁷⁶
DevOps	DevOps is een methode wat een technologische en procesmatige integratie realiseert tussen ontwikkeling en beheer ⁷⁷ .
Geld Nodig Bedrijven (GNB)	Een term die gebruikt wordt binnen het programma Financiering Bedrijven voor het verlenen van kredieten aan bedrijven.
Java	Een object georiënteerd programmeertaal om software te schrijven ⁷⁸ .
Java-applicatie	Applicatie die geprogrammeerd is volgens de programmeertaal Java.
OTAP	Ontwikkeling, Test, Acceptatie, Productie
SEPA	Single Euro Payments Area
Software configuratiemanagement	De taak om wijzigingen in applicaties te beheersen en te monitoren ⁷⁹ .

⁶⁹ Microsoft, 2013

⁷⁰ David Chappell, 2008

⁷¹ XebiaLabs, 2013

⁷² Rabobank, 2013 (E)

⁷³ Wikipedia, 2013 (B)

⁷⁴ Jez Humble; David Farley, 2011

⁷⁵ Paul M. Duvall, 2007

⁷⁶ Wikipedia, 2013 (C)

⁷⁷ Computable, 2013

⁷⁸ Wikipedia, 2013 (D)

⁷⁹ Wikipedia, 2013 (E)

9 Bronvermelding

Websites

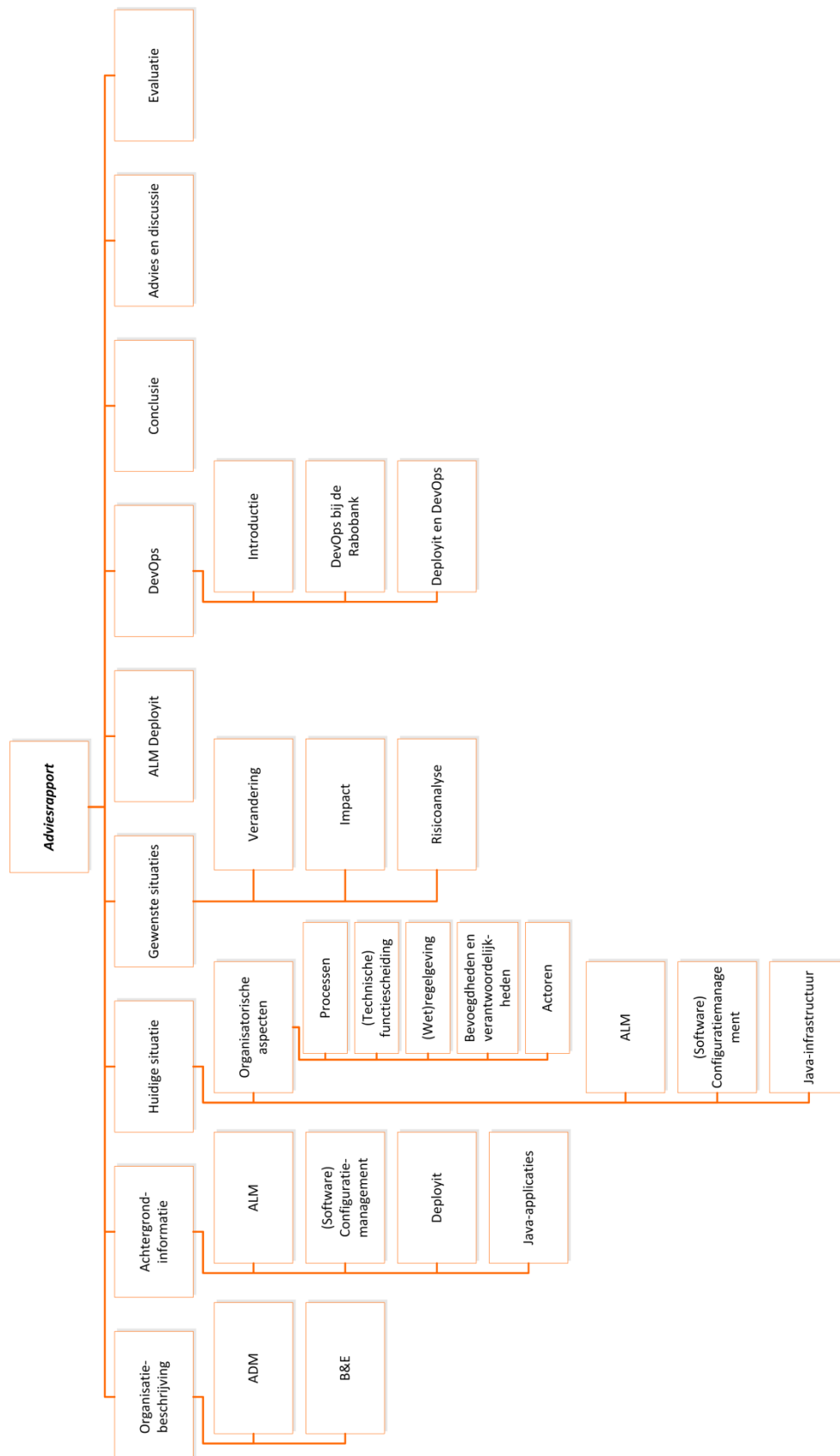
- Computable, DevOps. Geraadpleegd op 18 februari 2013, van <http://www.computable.nl/artikel/opinie/development/4629565/1277180/is-devops-het-nieuwe-agile.html>
- Google Scholar. Geraadpleegd op 28 februari 2013, <http://scholar.google.nl/>
- Hogeschool Utrecht, LUCAS. Geraadpleegd op 28 februari 2013, <http://www.mediatheek.hu.nl/~link.aspx?id=EFA05914A6E244F7BA4A319216379797>
- Microsoft, .NET. Geraadpleegd op 11 februari 2013, van <http://www.microsoft.com/net>
- Rabobank, Rabobank Groep. Geraadpleegd op 11 februari 2013 (A), van https://www.rabobank.com/nl/group/About_Rabobank_group/Corporate_governance/Rabobank_Nederland/index.html
- Rabobank, Groep ICT. Geraadpleegd op 11 februari 2013 (B), van <http://afdelingen.rabobank.nl/sites/GroepICT/over%20GICT/Pages/default.aspx>
- Rabobank, Applicatie Service. Geraadpleegd op 11 februari 2013 (C), van <http://afdelingen.rabobank.nl/SITES/AS/OVERAPPLICATIESERVICE/Pages/default.aspx>
- Rabobank, Infra Service. Geraadpleegd op 11 februari 2013 (D), van <http://afdelingen.rabobank.nl/sites/InfraService/over%20IS/Pages/default.aspx>
- Rabobank, Changemanagement. Geraadpleegd op 1 maart 2013 (E), van <http://afdelingen.rabobank.nl/sites/ADM/themas/changemanagement/Pages/default.aspx>
- Wikipedia, SEPA. Geraadpleegd op 27 februari 2013 (A), van <http://nl.wikipedia.org/wiki/SEPA>
- Wikipedia, COBOL. Geraadpleegd op 20 februari 2013 (B), van <http://nl.wikipedia.org/wiki/COBOL>
- Wikipedia, Software Deployment. Geraadpleegd op 12 februari 2013 (C), van http://en.wikipedia.org/wiki/Software_deployment
- Wikipedia, Java (programmeertaal). Geraadpleegd op 12 februari 2013 (D), van [http://nl.wikipedia.org/wiki/Java_\(programmeertaal\)](http://nl.wikipedia.org/wiki/Java_(programmeertaal))
- Wikipedia, Software Configuration Management. Geraadpleegd op 1 maart 2013 (E), van http://en.wikipedia.org/wiki/Software_configuration_management
- Xebialabs; Continuous Integration. Geraadpleegd op 1 maart 2013, van http://go.xebialabs.com/EBCDWhitepaper_Continuous_Delivery.html

Literatuur

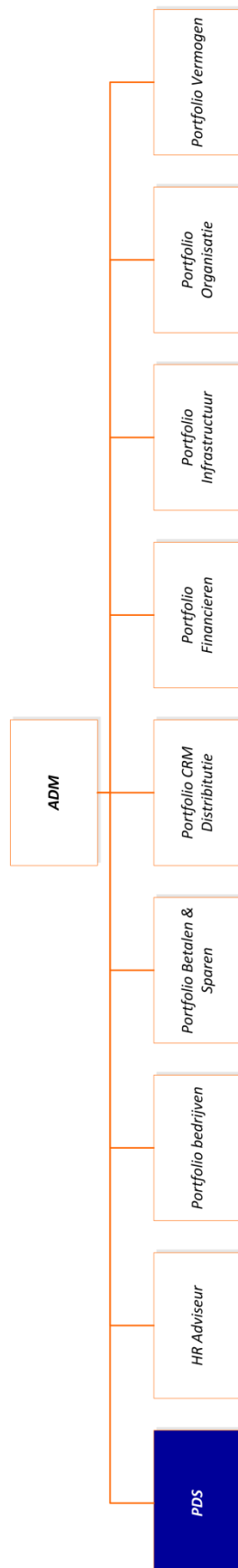
- Léon de Caluwé; Leren Veranderen; 2006; 2e druk; Kluwer, Deventer
- David Chappell; What is ALM; december 2008; Chapell & Associates. Digitale versie: <http://www.davidchappell.com/WhatIsALM--Chappell.pdf>
- Paul M. Duvall; Continuous Integration; 2007; Pearson Education, Inc.
- Jez Humble, David Farley; Continuous Delivery; 2011; Pearson Education, Inc
- Peter Janssen; PRINCE2 Compact; 2007; Pearson Education Benelux, Amsterdam
- Hannah Nathans; Adviseren als tweede beroep; 2004; 3e druk; Kluwer, Deventer
- Rabobank Groep ICT; Strategieplan 2010 – 2012; September 2009
- Michaël Steenhouders; Leren Communiceren; 2006; 5e druk; Wolters-Noordhoff, Groningen/Houten
- Nel Verhoeven; Wat is onderzoek?; 2010; 3e druk; Boom Onderwijs, Den Haag

10 Bijlagen

10.1PBS



10.2 Organogram ADM



Contract afstudeeropdracht
Instituut voor ICT
Nijenoord 1, 3552 AS, UTRECHT

Datum:

Naam student:	R. (Roan) van Helten
Opleiding:	Business IT & Management
Variant: voltijd / deeltijd / duaal	Voltijd
Adres student:	[wegens privacy redenen verwijderd]
Postcode / Woonplaats student:	[wegens privacy redenen verwijderd]
Studentnummer:	1545097
Telefoonnummer privé:	[wegens privacy redenen verwijderd]
E-mailadres:	roan.vanhelten@student.hu.nl r.helten@rn.rabobank.nl

Naam bedrijf (afstuderen):	Rabobank
Adres bedrijf:	Croeselaan 18
Postcode / Woonplaats bedrijf:	3521CB Utrecht
Naam bedrijfsbegeleider:	W. (Wil) Leeuwis
Telefoonnummer bedrijfsbegeleider:	[wegens privacy redenen verwijderd]
E-mailadres bedrijfsbegeleider:	w.leeuwis@rn.rabobank.nl

Beoogde datum van afstuderen:	juni 2013
Geheimhouding geaccordeerd door HU op:	<i>Niet van toepassing</i>

Ondergetekenden verklaren hiermee akkoord te gaan met de inhoud van bijgesloten PvA.

Handtekeningen

Student :

Docentbegeleider :

Bedrijfsbegeleider⁸⁰ :

⁸⁰ Door ondertekening van dit formulier verklaart de bedrijfsbegeleider minimaal een hbo- of vergelijkbare opleiding te hebben.

Bijlage B: Evaluatie van eigen functioneren

In deze bijlage is het afstudeeronderzoek persoonlijk geëvalueerd door de afstudeerder. De evaluatie is opgedeeld in de onderdelen organisatie, afstudeeropdracht, aanpak, begeleiding, resultaten en persoonlijke ontwikkeling.

De organisatie

Tijdens het zoeken naar een passende afstudeeropdracht wilde ik afstuderen binnen een toonaangevende en grote organisatie. De Rabobank voldoet aan deze eisen en is daarnaast ook goed bereikbaar via het openbaar vervoer. Binnen mijn netwerk wordt er positief gesproken over deze organisatie en dit, samen met de andere factoren, hebben er toe geleid om een open sollicitatie te sturen. Al vrij snel werd hier op gereageerd en werd ik uitgenodigd voor een gesprek. Uiteindelijk heeft mij dit een interessante en uitdagende opdracht opgeleverd.

Tijdens het afstuderen is mijn mening over de organisatie ongewijzigd gebleven. De medewerkers binnen Groep ICT hebben veel kennis en ervaring met ICT-gerelateerde onderwerpen en er wordt professioneel en hard gewerkt. De medewerkers zijn bereid je te helpen indien je hier om vraagt. Het contact met de medewerkers verliep telefonisch, per email of je zocht degene op.

De afstudeeropdracht

De opdracht bestond uit de invoering van Deployit binnen de afdelingen ADM en B&E op organisatorisch- en procesniveau. Een van de uitdagingen was om, zonder te kunnen programmeren in Java, de activiteiten van een Java-ontwikkelaar te beschrijven. Een Java-applicatie heeft daarbij ook vele koppelingen nodig en deze termen waren (bijna) allemaal nieuw voor mij. Daardoor moest ik aan het begin van het onderzoek verschillende literatuur raadplegen en medewerkers vragen om een uitgebreide uitleg.

Aanpak

In de eerste maand van het onderzoek is het plan van aanpak geschreven waarin al vrij duidelijk stond wat er van mij werd verwacht. Dit is deels omdat er tijdens deze eerste maand een aantal gesprekken zijn gevoerd met medewerkers die een duidelijke inschatting konden geven van de omvang en complexiteit van de opdracht.

Werkelijke voortgang

In het kader van het plan van aanpak en de werkelijke voortgang van het onderzoek zijn er enige verschillen waargenomen. De planning is niet volledig uitgevoerd zoals deze is beschreven in het plan van aanpak. De benodigde uren per activiteit kwamen redelijk overeen maar de waterval-aanpak verliep anders. Het beschrijven van de huidige situatie ging namelijk gepaard met het schrijven van de gewenste situaties. Dit had voornamelijk te maken met het moeilijk (van te voren) kunnen inschatten van het benodigde detailniveau.

Kosten/baten analyse

De kosten/baten analyse is niet weergegeven in kwantitatieve waarden wat ook niet concreet aangegeven is in het plan van aanpak. Echter is een dergelijke analyse meestal wel kwantitatief van karakter. De reden waarom de kosten en baten in de vorm van onderwerpen zijn beschreven is omdat de getallen moeilijker te vergaren zijn dan verwacht. Dit was niet realistisch met de beschikbare tijd die is aangegeven in de planning van het plan van aanpak.

Steekproef

De keuze van de steekproef van de applicatie BBS en de bijhorende satelliet-applicaties was naar mijn mening de beste keuze. In de eerste weken van het onderzoek ben ik uitgenodigd om deel te nemen aan de kerngroep. Dit accepteerde ik omdat ik draagvlak een van de belangrijkste onderdelen vind om een onderzoek tot een succes te maken. De kerngroep heeft als belangrijkste doel om het gebruik van Deployit binnen ADM in goede banen te leiden. Andere aspecten van de kerngroep zijn het generiek invoeren van Deployit en bijsturen waar nodig.

De kerngroep bracht mij op de hoogte van een project (project ARA) dat werd opgezet om een pilot uit te voeren bij BBS. Op het moment dat het project ARA werd opgestart is er meteen contact gelegd met de projectleider en de projectleden. Ik ben opgenomen als project-lid van het project en heb daarbij de verantwoordelijkheid gekregen om de huidige en gewenste deploymentprocessen in kaart te brengen. Dankzij deze verantwoordelijkheid heb ik gemerkt dat benodigde medewerkers toegankelijker werden om deel te nemen aan mijn onderzoek. Het project had wekelijks een Stand-Up waarbij iedereen zijn voortgang besprak van zijn taken. De projectleider besprak de planning waarop stond wie waarvoor verantwoordelijk was. Tijdens deze Stand-Ups brachten de projectleden mij ook op de hoogte van bruikbare informatie en gaven ze mij tips.

Begeleiding vanuit de organisatie

De begeleiding binnen de Rabobank is uitstekend geregeld. De bedrijfsbegeleider nam wekelijks minimaal een uur de tijd om de voortgang te spreken en mij bij te sturen waar dat nodig was. De bedrijfsbegeleider heeft een brede kennis over o.a. de organisatie, de ontwikkeling van Java, schrijfvaardigheid en de balans tussen de verwachtingen vanuit de onderwijsinstelling en de organisatie. Tijdens het onderzoek dacht de begeleider met mij mee indien ik hier om vroeg. De begeleider heeft een groot contactennetwerk en hielp mij om een eigen netwerk op te bouwen. Ik was vooral enthousiast over het feit dat mijn begeleider niet snel tevreden is, dit leidde er toe dat ik tot het uiterste ging en ik daarin ook werd begeleid.

Begeleiding vanuit de onderwijsinstelling

De begeleiding vanuit school verliep vlekkeloos en voorspoedig. De docentbegeleider gaf duidelijke en concrete feedback op de verslagen en gaf helder antwoord op de vragen die ik stelde. Er waren geen momenten waarbij ik nog onduidelijkheden had na een gesprek. De begeleider was goed bereikbaar en reageerde snel.

Resultaten

In het kader van het onderzoek ben ik tevreden over de resultaten. De resultaten zijn op een valide en betrouwbare manier vergaard. Ik ben het meest enthousiast over de bruikbaarheid van de resultaten. Deze zijn namelijk al deels tijdens het onderzoek gebruikt door het project ARA. Het eindproduct is een scriptie wat, naar mijn mening, een heldere rode lijn heeft bestaande uit voortgangaspecten, onderzoeksresultaten en het gebruik van literatuur.

Persoonlijke ontwikkeling

Tijdens de afstudeerperiode heb ik mezelf ontwikkeld. Ik heb o.a. kennis en ervaring opgedaan over de ontwikkeling van Java, DevOps en Continuous Delivery. In het kader van het uitvoeren van een dergelijk onderzoek heb ik enkele ontwikkelpunten opgesomd.

Kritisch zijn

Tijdens het afstudeeronderzoek heeft mijn bedrijfsbegeleider er op toegewezen kritisch te zijn. Dit heb ik tijdens ieder interview meegenomen. Daarbij wilde ik ook niet afhankelijk zijn van één medewerker en betrok ik meerdere medewerkers met dezelfde rol. Het kritisch leren te zijn heb ik tijdens het onderzoek verder ontwikkeld.

Projectmatig werken

Doordat ik een project-lid was in het project ARA heb ik mijn projectmatig werken verder ontwikkeld. Dit is mijn eerste keer dat ik in een dergelijk groot project heb meegewerkt. Ik ben verrast hoe een project in drie tot vier maanden een groot succes kan worden.

Kerngroep

Tijdens de kerngroep bijeenkomsten werden visies besproken m.b.t. een generieke implementatie en de verwachte impact die het kan hebben. Daarbij moet gedacht worden aan de benodigde rollen die nodig zijn en de weerstand die kan ontstaan. Ik heb ervaring opgedaan in het begrijpen van o.a. het belang van deze gespreksonderwerpen.

Stakeholders

Ik heb geleerd wat het belang is om intensief contact te behouden met de stakeholders. Tijdens het project is dit in een enkel geval in mindere mate gebeurd. De impact is dat de stakeholder geen akkoord geeft op je resultaten en je de resultaten moet verbeteren.

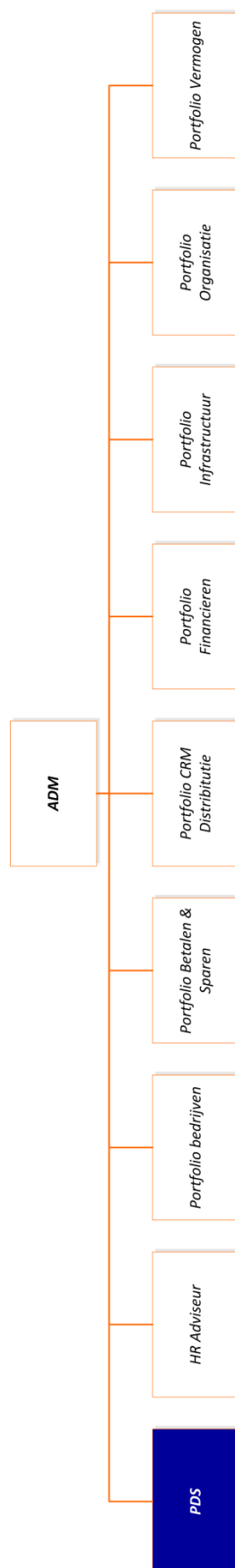
Interviews

Ik heb geleerd om geen interviews uit te voeren met meer dan één respondent. Omdat het tijdens het onderzoek in een enkel geval niet anders kon is dit wel gedaan. Er ontstaat een soort discussie tussen de medewerkers. Dit is informatief maar niet productief. Daarnaast beïnvloeden de medewerkers elkaar ook. Dit kan zowel negatief als positief zijn. De interviewer moet de mogelijkheid hebben om het interview te sturen, dit is lastig met twee (of meer) respondenten.

Taal- en schrijfvaardigheid

Tijdens het onderzoek heb ik verschillende verslagen en conceptversies opgestuurd naar mijn bedrijfsbegeleider ter controle. De bedrijfsbegeleider gaf onder andere feedback over mijn taal- en schrijfvaardigheid. Dankzij deze feedback heb ik mijn taal- en schrijfvaardigheid verder ontwikkeld.

Bijlage C: Organogram ADM



Bijlage D: Theoretisch kader

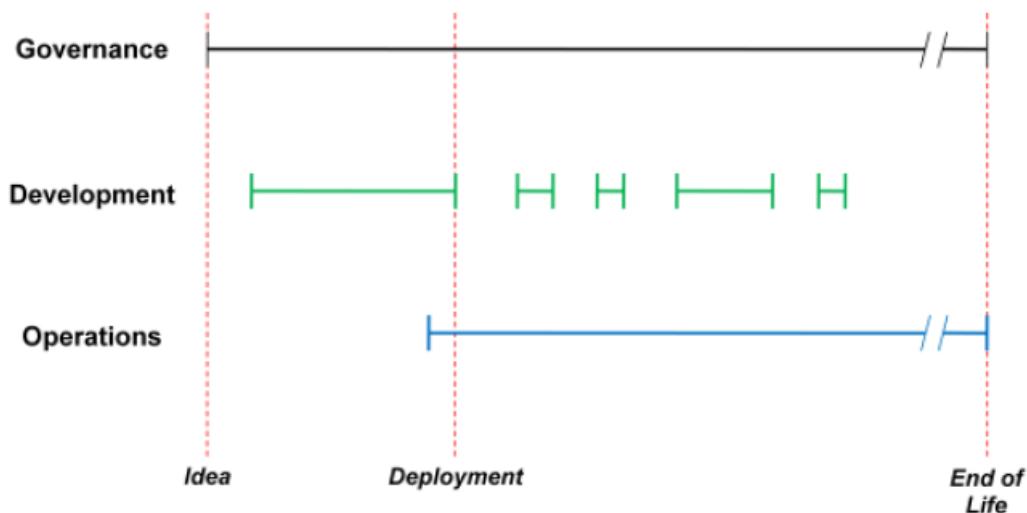
De literatuurstudie betreft de onderwerpen Java, Application Lifecycle Management, DevOps, Continuous Delivery, Continuous Integration en Continuous Deployment. Voor ieder onderdeel is de Big6⁸¹ methode gehanteerd. In de *paragraaf Geraadpleegde literatuur* zijn per onderwerp de geraadpleegde bronnen weergegeven.

Java

Java is niet enkel een programmeertaal maar een volledig platform bestaande uit bibliotheken (libraries)⁸². Voor het programmeren van Java-code kan er gebruikt gemaakt worden van verschillende Integrated Development Environments (IDE). Twee voorbeeld van IDE's zijn IntelliJ en Eclipse. Om Java-code op een server te deployen wordt er een build gemaakt, bijvoorbeeld een Websphere Applicatie Server (WAS). Een build bestaat uit (een combinatie van) EAR-, WAR- en JAR-bestand(en), configuratiebestanden en database scripts.

Application Lifecycle Management

Application Lifecycle Management (ALM) beschrijft hoe een applicatie wordt beheerd vanaf concept, bouwen en tot en met het uit dienst nemen⁸³. Binnen ALM wordt er onderscheid gemaakt tussen bestuur, ontwikkeling en beheer, zie *Figuur 25 - Application Lifecycle Management*⁸⁴.



Figuur 25 - Application Lifecycle Management

Bestuur maakt de beslissing en voert het projectmanagement uit rondom de applicatie. Ontwikkeling start bij het schrijven van de applicatie en tijdens de levenscyclus kunnen er verschillende aanpassingen gemaakt worden zoals onderhoud en updates. Beheer focust zich op het operationeel houden van de applicatie voor de gebruikers.

⁸¹ (Verhoeven, 2010)

⁸² (Horstmann & Cornell, 2012)

⁸³ (Gousset, Keller, & Woodward, 2012)

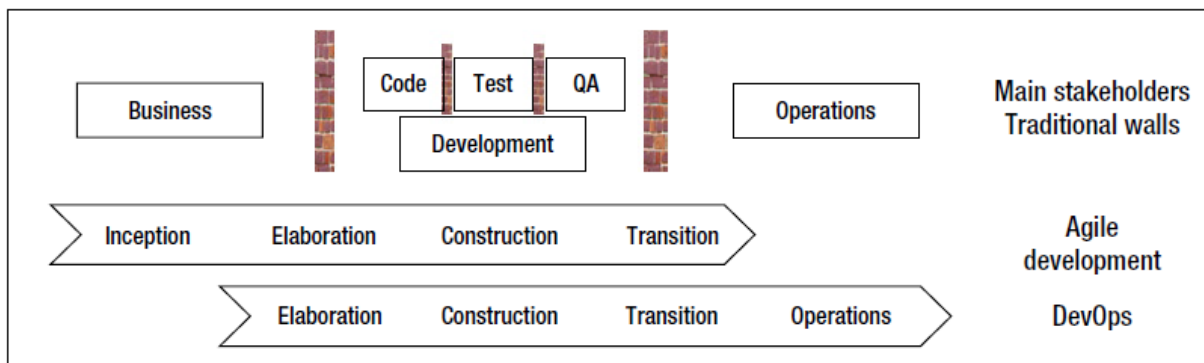
⁸⁴ (Chappell, 2008)

DevOps

DevOps is een adolescente⁸⁵ filosofie⁸⁶ binnen de ICT-wereld die binnen de Rabobank nog in haar kinderschoenen staat. Bij DevOps gaat het om de samenwerking tussen *Dev*, Development (ontwikkeling) en *Ops*, Operations (beheer). DevOps realiseert technologische en procesmatige integratie tussen ontwikkeling en beheer⁸⁷.

DevOps bestaat uit werkwijzen om het proces, vanaf de ontwikkeling tot en met het in productie nemen, sneller te laten verlopen. Daarnaast helpt DevOps om software te leveren van hogere kwaliteit⁸⁸. DevOps behandelt onderwerpen met betrekking tot cultuur, automatisering, meetbaarheid en sharing. Dit wordt 'CAMS' genoemd (Culture, Automation, Measurement, Sharing)⁸⁹. Het afstudeeronderzoek heeft raakvlak met het onderwerp automatisering. Een tool dat deployments automatiseert, zoals Deployit, ondersteunt een van de doelen van DevOps, namelijk het verkorten van de cycle time.

Een duidelijke opvatting vanuit DevOps is dat de ontwikkelorganisatie gericht is op het maken van wijzigingen en daartoe innovatief ingesteld is. De beheerorganisatie is gericht op continuïteit van de verwerking. DevOps heeft raakvlak met Agile. Agile focust zich op het omschrijven, bouwen en construeren van software. In *Figuur 26 - DevOps en Agile* is weergegeven wat het raakvlak is⁹⁰.



Figuur 26 - DevOps en Agile

Zoals het figuur aangeeft brengt Agile de business en development samen. DevOps ondersteunt het leveren van de benodigde software. Een combinatie van Agile en DevOps betekent daartoe een volledige samenhang tussen business, development en operations. In de traditionele organisatie wordt de minder optimale samenhang gezien als een 'muur' tussen business, development en operations. Het onderwerp continuous integration, *paragraaf Continuous Integration*, is een onderdeel wat gepositioneerd kan worden binnen de grenzen van Agile. Continuous delivery, *paragraaf Continuous Delivery*, gaat een stap verder en wordt gepositioneerd binnen de grenzen van DevOps.

⁸⁵ (Gartner, Hype Cycle Application Development, 2012)

⁸⁶ (Gartner, How to build a DevOps Release team, 2012)

⁸⁷ (Computable)

⁸⁸ (Hüttermann, 2012)

⁸⁹ (Willis, 2010)

⁹⁰ (Hüttermann, 2012)

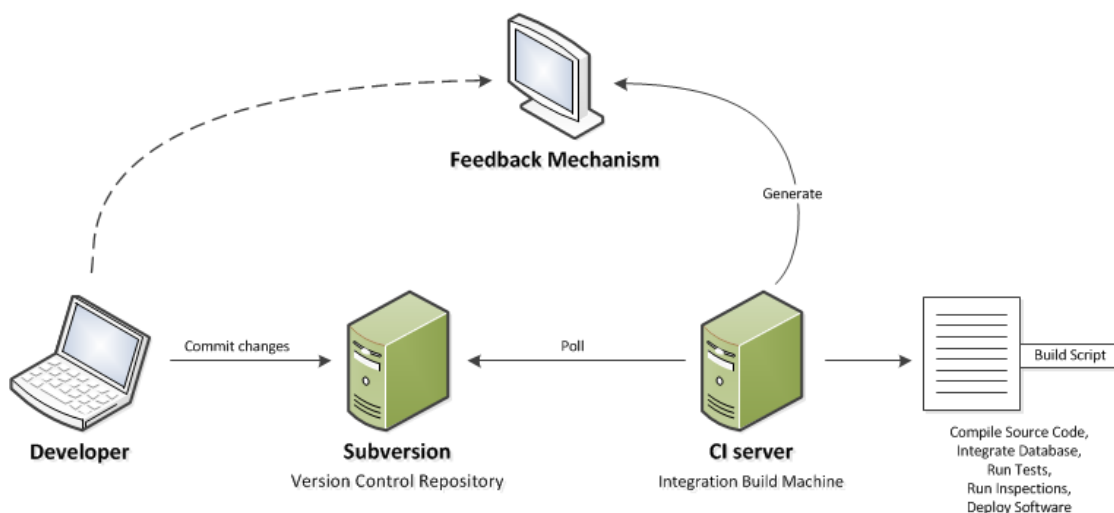
Continuous Integration

Continuous integration (CI) maakt het mogelijk om 'builds' regelmatig te integreren met als doel dat het geen specifieke en tijdrovende taak wordt van het project. Een build is het proces dat de broncode samenvoegt en controleert of de software functioneert als een samenhangende eenheid⁹¹.

Voor continuous integration betekent dit:⁹²

- Ontwikkelaars voeren builds uit op hun eigen ontwikkelomgeving.
- Ontwikkelaars commiten minimaal dagelijks de ontwikkelde source code naar een versiebeheer systeem.
- Een CI server voert dagelijks geïntegreerde builds uit.
- 100% van de testen moeten geslaagd zijn voor iedere build.
- De geïntegreerde build (het product) kan functioneel getest worden.
- Repareren van een kapotte build is de hoogste prioriteit.
- Ontwikkelaars bekijken de verkregen rapportages die gemaakt zijn door de build (server) om mogelijke verbeteringen te vinden.

De bovenstaande punten zijn weergegeven in *Figuur 27 - Componenten van een CI systeem*⁹³.



Figuur 27 - Componenten van een CI systeem

⁹¹ (Duvall, Matyas, & Glover, 2007)

⁹² (Duvall, Matyas, & Glover, 2007)

⁹³ (Duvall, Matyas, & Glover, 2007)

Continuous Delivery

De term continuous delivery betekent het continu kunnen leveren van de ontwikkelde software aan de eindgebruiker. Continuous delivery gaat een stap verder dan continuous integration waardoor het aangeraden wordt om continuous integration ingericht te hebben⁹⁴. In het kader van het afstudeeronderzoek zijn de onderwerpen deployment pipeline en continuous deployment relevant.

Deployment Pipeline

Continuous integration focust zich op het ontwikkelteam. De output van het CI proces is de input voor het uitvoeren van o.a. testen aan de beheerkant⁹⁵. De volgende problemen komen dan vaak voor⁹⁶:

- Beheerorganisatie wacht op documentatie en/of (spoed)fixes.
- Testers wachten op geslaagde builds van de ontwikkelde software.
- Ontwikkelteams ontvangen 'bug reports' weken na het releasen van de betreffende functionaliteit.

De oplossing is de invoering van een end-to-end automatisering van de build, deploy, test en release processen. Dit betekent een geautomatiseerde deployment pipeline. Deze aanpak resulteert zich in een 'pull' systeem. Dit systeem betekent voor de tester het zelf simpel, met één druk op de knop, kunnen deployen van de ontwikkelde software op de testomgeving. Voor de beheerder betekent dit het simpel kunnen deployen van de release naar de productieomgeving. De ontwikkelaars hebben inzicht op welke omgeving welke ontwikkelde code zich bevind en welke problemen ondervonden zijn. Managers hebben meer inzicht in de totale tijd, deployment-tijd en codekwaliteit.

In *Figuur 28 - Deployment pipeline* zijn de mogelijke wijzigingen binnen de deployment pipeline weergegeven⁹⁷. De kleur groen betekent dat de betreffende controle en/of test geslaagd is. De kleur rood betekent dat de betreffende controle en/of test mislukt is. Het doel van de deployment pipeline is het zo spoedig mogelijk ontvangen van feedback⁹⁸. Het spoedig ontvangen van feedback betekent het eerder ontdekken van een mogelijke fout, waardoor eerder en sneller ingespeeld kan worden op deze fout zonder dat dit grote gevolgen heeft voor de voortgang van het proces. Om een voorbeeld te geven is een fout in de source code van de applicatie makkelijker te verhelpen op het moment dat de applicatie nog in de handen is van de ontwikkelaar zelf.

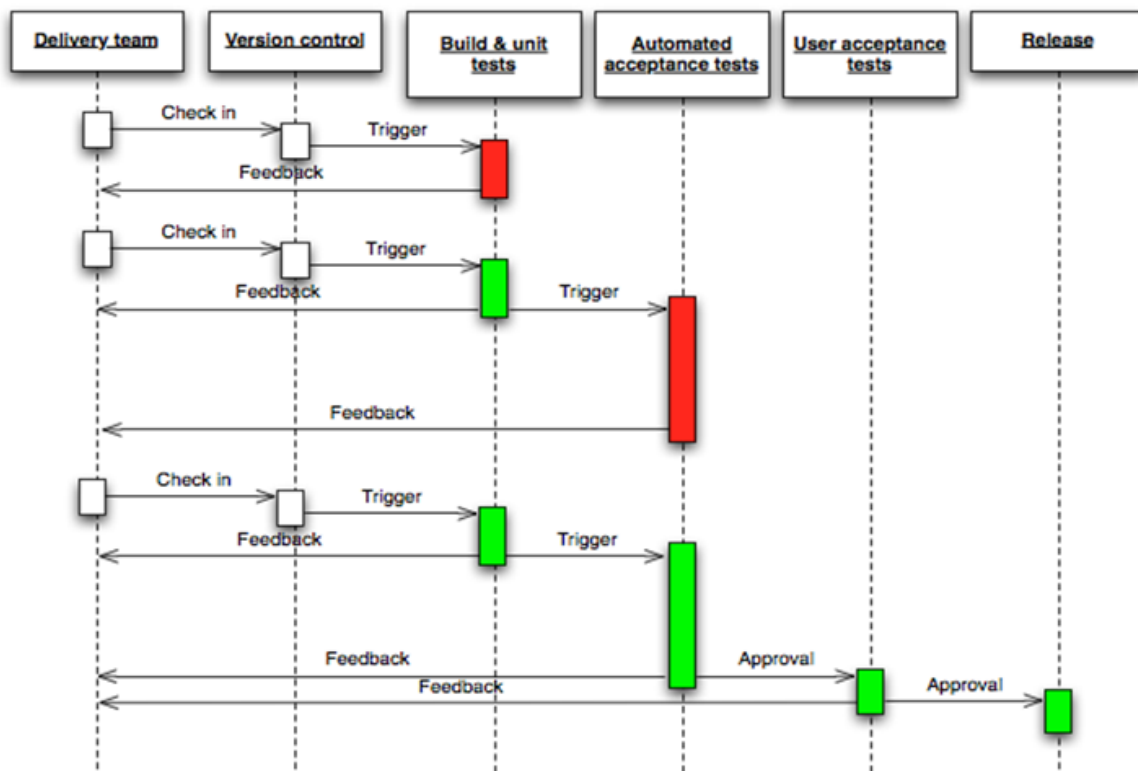
⁹⁴ (Gartner, Achieving Continuous Delivery, 2013)

⁹⁵ (Humble & Farley, 2011)

⁹⁶ (Humble & Farley, 2011)

⁹⁷ (Schudel, 2013)

⁹⁸ (Humble & Farley, 2011)



Figuur 28 - Deployment pipeline

Er zijn verschillende fases in de deployment pipeline, namelijk de commit stage, acceptance stage, UAT, capacity stage en production⁹⁹.

Commit stage

De ontwikkelaars commiten hun ontwikkelde source code in een versiebeheersysteem. Via een continuous integration systeem wordt de code gebuild. De geslaagde builds worden opgeslagen in de artifact repository. Dit is het Continuous Integration onderdeel.

Acceptance stage

Nadat de commit stage succesvol is uitgevoerd wordt de acceptance stage automatisch gestart. Deze fase voert automatische acceptatie testen uit.

UAT, Capacity stage en Production

Deze fases worden niet automatisch gestart nadat de acceptance stage is geslaagd. De testers binnen deze verschillende fases moeten zelf de ontwikkelde software deployen naar hun omgeving. Daarvoor is een automatische deployment vereist. De betreffende tester dient inzicht te hebben welke wijzigingen van de software geslaagd zijn binnen welke stage. Daarnaast komt het voor dat bijvoorbeeld enkel de medewerker binnen de stage Production de software mag deployen op de productieomgeving.

⁹⁹ (Humble & Farley, 2011)

Voor de realisatie van een deployment pipeline zijn de volgende stappen nodig¹⁰⁰:

1. Modelleren van de waardeketen
2. Build- en deploymentproces automatiseren
3. Unit tests en code analyses automatiseren
4. Acceptatietest automatiseren
5. Verder ontwikkelen van de pipeline.

Voor het inrichten van continuous delivery zijn er de volgende best practices¹⁰¹:

- Gebruik de source code niet voor iedere stage, maar maak eenmalig een build aan en plaats deze niet in het versiebeheersysteem van de source code.
- Deploy de applicatie op dezelfde manier bij elke omgeving. Indien de productieomgeving beheerd wordt door een ander team dan de ontwikkel- en testomgeving, dan dienen deze teams samen te werken om het automatisch deployen effectief te houden.
- Nadat de deployment uitgevoerd is dient er een automatische script te zijn die een smoke test uitvoert. Een smoke test bekijkt of de applicatie volledig draait en daarbij ook de benodigde externe services (bijvoorbeeld een database van een andere applicatie).
- Voer continuous integration en het testen uit op een omgeving die zoveel mogelijk gelijk is aan de productieomgeving.
- Een nieuwe versie van de source code die wordt gecommited naar het versiebeheersysteem dient meteen opgepakt te worden door het continuous integration systeem. Omdat het voor kan komen dat de ontwikkelaars in hoge frequentie nieuwe code commiten moet het CI systeem de intelligentie hebben om de laatste versies te builden, anders lopen de builds achter de feiten aan.

¹⁰⁰ (Humble & Farley, 2011)

¹⁰¹ (Humble & Farley, 2011)

Continuous Deployment

Het continu kunnen deployen betekent het direct en automatisch kunnen deployen naar de productieomgeving. Dit betekent dat, nadat de ontwikkelaar zijn code gecommitt heeft in een versiebeheer systeem, de code automatisch wordt getest en vervolgens wordt gedeployd naar de productieomgeving. De volgende voorwaarden zijn hier van toepassing¹⁰²:

- Geautomatiseerde unit test;
- Geautomatiseerde component test;
- Geautomatiseerde acceptatie test (zowel functioneel als non-functioneel).

Continuous deployment kan risico's met zich meebrengen gezien het feit dat er meteen naar de productieomgeving wordt gedeployd, oftewel meteen 'live' gaan zonder de tussenkomst van een handmatige test.

Application Release Automation (ARA) tools ondersteunen continuous (release) deployment. Een ARA tool maakt het mogelijk om een applicatie (build) met bijhorende configuratiebestanden en databasescripts automatisch te verplaatsen naar verschillende omgevingen. Een voorbeeld van een ARA tool is Deployit. Een ARA tool elimineert het tijdrovend proces om zelfgebouwde scripts te schrijven tijdens applicatie updates¹⁰³. De hype cycle Application Development van Gartner geeft aan dat ARA qua volwassenheid opkomend is.¹⁰⁴

De volgende tips en tricks zijn van toepassing op continuous deployment¹⁰⁵:

- De medewerkers die de deployment uitvoeren moeten betrokken worden bij het inrichten van het deploymentproces.
- Registreer deployment activiteiten.
- Verwijder geen oude bestanden, verplaats deze.
- De deployment is de verantwoordelijkheid van het hele team.
- Heb een 'warm-up' periode voor een nieuwe deployment.
- Breng geen wijzigingen aan de applicatie direct op de productieomgeving.

¹⁰² (Humble & Farley, 2011)

¹⁰³ (Gartner, Application Release Automation is a key to DevOps, 2011)

¹⁰⁴ (Gartner, Hype Cycle Application Development, 2012)

¹⁰⁵ (Humble & Farley, 2011)

Geraadpleegde literatuur

In *Tabel 17 - Geraadpleegde literatuur* is de geraadpleegde literatuur beschreven. De tabel beschrijft per gebruikte term de gebruikte databank, het aantal hits dat deze zoekuitvoering heeft opgeleverd, welke hits gebruikt zijn en daarvan de bronverwijzing.

Term	Databank	Hits	Gebruikt	Bronverwijzing
Java Fundamentels	Safari Books Online	516	1	(Horstmann & Cornell, 2012)
Application Lifecycle Management	Safari Books Online	705	1	(Gousset, Keller, & Woodward, 2012)
	DavidChappell	1	1	(Chappell, 2008)
Continuous deployment Java	Lucas HU	6	0	
	Safari Books Online	341	0	
	Gartner	523	1	(Gartner, Hype Cycle Application Development, 2012)
Continuous delivery Java	Lucas HU	0	0	
	Safari Books Online	138	1	(Hüttermann, 2012)
	Gartner	504	1	(Gartner, Achieving Continuous Delivery, 2013)
	Medewerker organisatie	2	2	(Duvall, Matyas, & Glover, 2007) (Humble & Farley, 2011)
Application release automation Java	Lucas HU	2	0	
	Safari Books Online	189	0	
	Gartner	901	2	(Gartner, Hype Cycle Application Development, 2012) (Gartner, Application Release Automation is a key to DevOps, 2011)
DevOps	Gartner	142	2	(Gartner, How to build a DevOps Release team, 2012) (Gartner, Best Practices to Sync and Streamline DevOps Cycles for Faster IT Service Delivery, 2013)

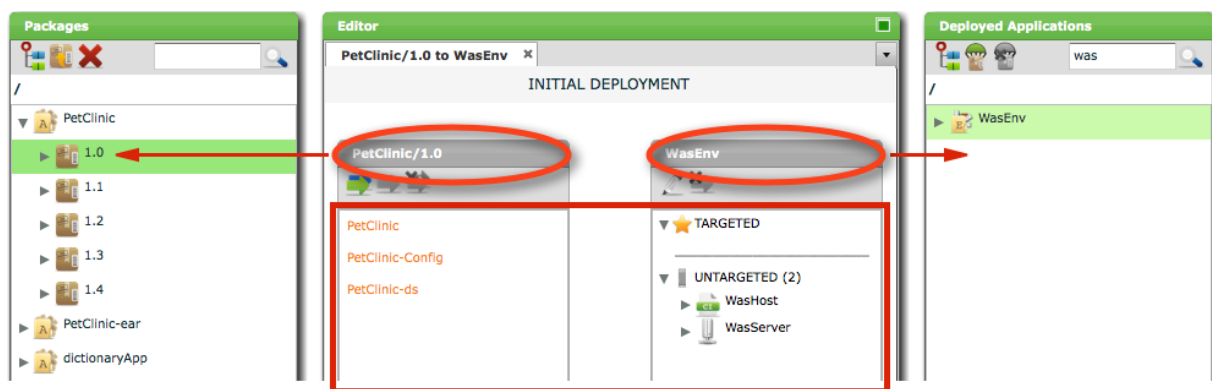
Tabel 17 - Geraadpleegde literatuur

Bijlage E: Deployit beschrijving

Dit hoofdstuk beschrijft de tool Deployit (versie 3.8¹⁰⁶). Voor de beschrijving van de tool is er gebruik gemaakt van de aanwezige documentatie en het beeldmateriaal afkomstig van de website van de leverancier, XebiaLabs¹⁰⁷.

Introductie

Deployit is een oplossing voor de realisatie van een geautomatiseerde deployment. Met een deployment worden alle activiteiten bedoeld om de applicatie te installeren en configureren op een geselecteerde omgeving. *Figuur 29 - Deployit GUI* geeft de graphical user interface (GUI) van de tool weer. Naast GUI kent de tool ook een command line interface (CLI). De CLI kan gebruikt worden om taken binnen de tool te automatiseren.



Figuur 29 - Deployit GUI

De GUI van Deployit bestaat uit drie vensters, namelijk 'Packages', 'Editor' en 'Deployed Applications'. Het venster 'Packages' geeft een lijst van versies weer van een bepaalde applicatie. Daarbij kent een versie (package) zowel 'artifacts' als 'resource specifications'. Artifacts zijn fysieke bestanden zoals EAR, WAR of statische HTML bestanden. Resource specifications zijn o.a. de gegevensbronnen om een applicatie te draaien. Het venster 'Deployed Applications' geeft de gedeployde applicatie weer op de geselecteerde omgeving.

Om een versie van een applicatie te deployen wordt de gewenste package en de gewenste omgeving geselecteerd. Vervolgens dient er via de editor, dit is het middelste venster, enige aanpassingen gemaakt te worden, daarbij kan gedacht worden aan de acceptatie van nieuwe bestanden of bestanden die in de nieuwe versie niet meer aanwezig zijn, en kan de package via een knop automatisch gedeployd worden. Tijdens de deployment geeft de tool de voortgang en eventuele problemen (errors) aan. De deployment wordt sequentieel uitgevoerd en in geval van een error stopt het proces en dient er ingegrepen te worden.

¹⁰⁶ (XebiaLabs, Deployit 3.8.x)

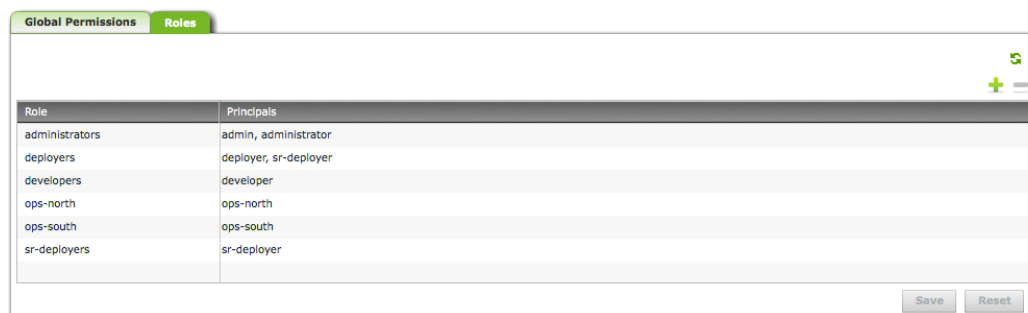
¹⁰⁷ (XebiaLabs, Deployit product information)

Functionaliteiten en plugins

Deployit kent verschillende functionaliteiten en plugins. Deze paragraaf behandelt de belangrijkste functionaliteiten die niet beschreven zijn in de vorige paragraaf en alle aanwezige plugins.

Rollen en beveiliging

In de tool kunnen rollen gespecificeerd worden. Rollen hebben concrete toegankelijkheden op onder andere bestanden, mappen en functionaliteiten binnen de tool. In *Figuur 30 - Rollen* is de GUI weergegeven van het beheer van de rollen. Door de mogelijkheid van rollen kan er een degelijke beveiliging gewaarborgd worden. Daarnaast is het ook mogelijk bestanden en/of mappen te beveiligen met een login autorisatie.



Role	Principals
administrators	admin, administrator
deployers	deployer, sr-deployer
developers	developer
ops-north	ops-north
ops-south	ops-south
sr-deployers	sr-deployer

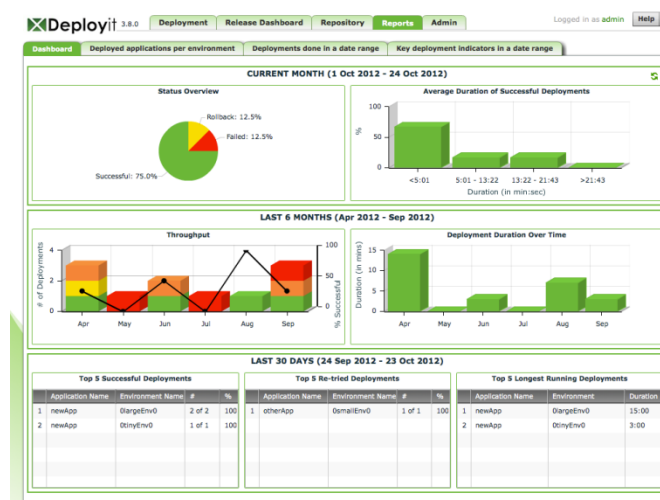
Figuur 30 - Rollen

Rollback

Deployit ondersteunt handmatige 'rollbacks' van deployments. Een rollback maakt het mogelijk om de versie van de applicatie terug te zetten voordat de deployment werd uitgevoerd.

Dashboard

Deployit heeft een dashboard waar rapportages weergegeven worden. Daarbij kan gedacht worden aan het aantal deployments in een bepaalde tijd, langdurige deployments, succesvolle deployments, mislukte deployments en een geschiedenis van geïmplementeerde versies binnen een bepaalde applicatie. Naast de mogelijkheid om deze informatie op te vragen binnen het dashboard kan het ook geëxporteerd worden als CSV formaat wat een brede standaard is voor verschillende bekende programma's waaronder Microsoft Excel. Zie *Figuur 31 - Dashboard* voor een weergave van een dashboard.



Figuur 31 - Dashboard

Plugins

Een plugin is een toevoeging van functionaliteiten aan een applicatie, in dit geval Deployit. Plugins kunnen onder andere benodigd zijn voor connecties met andere applicaties. Deployit kent verschillende plugins, in *Tabel 18 - Plugins* zijn de aanwezige plugins beschreven.

Naam plugin	Functionaliteiten
Command plugin	Het creëren van packages voor en het uitvoeren van willekeurige commando's
Database plugin	Maakt SQL implementaties naar databases mogelijk.
File Plugin	Verantwoordelijk voor de implementatie van bestanden en mappen.
Generic Model Plugin	Een 'bouwdoos' om nieuwe plugins te ontwikkelen
JEE Plugin	Bevat standaard JEE configuratie items
Remoting Plugin	Maakt het mogelijk om te communiceren met remote systems (systemen die afstand geraadpleegd kunnen worden).
Webserver Plugin	Maakt het mogelijk om statische content te implementeren op webserverns.
Trigger Plugin	Maakt het mogelijk om een actie toe te voegen aan een gebeurtenis. Bijvoorbeeld een email versturen op het moment dat er een error is.
Maven Plugin	Maakt het mogelijk om via Maven een bestand (.DAR) aan te maken dat de benodigde input is voor Deployit.

Tabel 18 - Plugins

Voor- en nadelen

Deze paragraaf behandelt de mogelijke voor- en nadelen van Deployit. De voor- en nadelen zijn gericht op de tool in het algemeen, de organisatie en best practices¹⁰⁸. De voor- en nadelen zijn als volgt opgesomd:

Voordelen

- Handmatige deployments zijn geautomatiseerd, wat het volgende als voordeel heeft:
 - o Minder kans op (menselijke) fouten tijdens de deployment van een applicatie.
 - o Time to market, de totale tijd van de ontwikkeling, test, acceptatie en productie (OTAP) van een applicatie is gereduceerd vanwege de geautomatiseerde deployment.
 - o Kostenbesparing, de benodigde handwerk is vervangen door de tool wat leidt tot minder benodigd personeel.
- De mogelijkheid om deployments in samenhang te brengen met prestatieproblemen.
- Hogere benuttingsgraad van de aanwezige implementatieomgeving.

Nadelen

- Complex om te implementeren.
 - o Binnen de Rabobank worden er verschillende werkwijzen gehanteerd rondom de ontwikkel- en beheerprocessen van Java-applicaties. Een generieke implementatie van Deployit kan daardoor complex zijn.
 - o De aanwezige wet- en regelgeving met betrekking tot functiescheiding binnen OTAP kan leiden tot een complexe en onoverzichtelijke rolverdeling binnen Deployit.

¹⁰⁸ (O'Regan, 2012)

Keuze van de tool

Voor de keuze van Deployit is Gartner in 2011 geraadpleegd. Gartner kent weinig soortgelijke continuous delivery tools. Naast het feit dat er weinig aanbod is, heeft de Rabobank een goede relatie met de leverancier XebiaLabs. Deployit is onderzocht om de internetketen automatisch te deployen. Daaruit kwam de conclusie dat de tool nog teveel tekortkomingen had. Gezien deze conclusie hebben de applicaties van het Persoonlijk Internet Framework (PIF) een eigen tool ontwikkeld, namelijk PiCoMa. PiCoMa is specifiek geschikt voor PIF.

Rond de maand mei/juni van 2012 was er de uitdaging om SEPA te deployen. Dit zou veel tijd in beslag nemen gezien de benodigde handmatige deployments. Daarom is er voor SEPA onderzocht of Deployit geschikt is. Daaruit werd geconcludeerd dat Deployit geschikt is. Er is echter niet gekeken naar de mogelijkheden voor de configuratie van o.a. de aanwezige functiescheiding, beveiliging en wet- en regelgeving binnen de Rabobank. In de business case van het project SEPA is de ROI van de tool drie maanden. De licentie van Deployit is van toepassing voor de gehele Rabobank.

PiCoMa

PiCoMa, wat staat voor **PIF CO**nfig **MA**nagement, is een web-based CRUD applicatie wat Serverinitials en batch deployments uitvoert. Ook verplaatst PiCoMa de applicatie zelf. Het is een tool dat intern ontwikkeld is. PiCoMa is enkel ontwikkeld voor alle PIF applicaties. PiCoMa heeft veel overeenkomst met een tool zoals Deployit, waarbij Deployit uitgebreider is en toegankelijker voor iedere Java-applicatie.

Bijlage F: Stakeholderanalyse

In deze paragraaf zijn de belanghebbenden (stakeholders) van het onderzoek Deployit beschreven. Voor het in kaart brengen van de stakeholders is het voorbeeld gebruikt van het rapport 'Informatieplan De Vooruitgang'¹⁰⁹. De stakeholders zijn beschreven in *Tabel 19 - Stakeholders*. Elke belanghebbende heeft een beschrijving bestaande uit een doel, macht, macht versus interesses, ondersteuning en beïnvloeding.

Doel

Elke belanghebbende heeft een doel met betrekking tot de mogelijke veranderingen binnen de Rabobank. Daarbij is in grote lijnen vermeld waar de betreffende belanghebbende baat bij heeft en waar in mate opgelet moet worden. Deze mate is beschreven in macht en beïnvloeding.

Macht

De macht van de belanghebbende hangt af van zijn of haar status. De macht is onderverdeeld in laag, gemiddeld en hoog. Deze niveaus zijn bepaald door middel van degene met het laagste niveau en degene met het hoogste niveau te vergelijken. Macht heeft weer betrekking op het doel dat de belanghebbende voor zich heeft.

Macht versus Interesses:

Om de belanghebbende verder te kunnen specificeren wordt op het gebied van macht versus interesses gekeken. Uit dit resultaat kan opgemaakt worden in welke mate een belanghebbende wordt opgenomen in een project.

Er wordt onderscheid gemaakt tussen de volgende mogelijkheden:

- Laag niveau tevreden houden (weinig interesse, weinig macht)
- Tevreden houden (weinig interesse, veel macht)
- Informeren (veel interesse, weinig macht)
- Intensief opnemen binnen project (veel interesse, veel macht)

Ondersteuning:

De ondersteuning heeft betrekking op dit project. Welke ondersteuning wordt verwacht van de belanghebbende tijdens het uitvoeren van het project.

Beïnvloeding:

De beïnvloeding is de mate waarin de belanghebbende invloed kan hebben op dit project. Deze mate is onderverdeeld in laag, gemiddeld en hoog.

¹⁰⁹ (Rooij, Miltenburg, Steenis, & Helten, 2011)

	Doel	Macht	Macht versus Interesses	Ondersteuning	Beïnvloeding
Opdrachtgever	<ul style="list-style-type: none"> Een overzicht van de aanwezige processen en rollen, inclusief het raakvlak met changemanagement, configuratiemanagement en application lifecycle management. Een tevreden projectleider van project ARA. 	H	Intensief opnemen binnen project	Positief	H
Projectleider project ARA	<ul style="list-style-type: none"> Beschrijving van de huidige en gewenste deploymentflow, bestaande uit de activiteiten en rollen. Daarbij, bij iedere rol, de bijhorende taken en verantwoordelijkheden. Gewenste situatie is een uitgewerkte fase 1. 	H	Intensief opnemen binnen project	Positief	H
Configuratie-manager	<ul style="list-style-type: none"> Overzicht van de configuratiemanagement aspecten en het raakvlak met andere processen en infrastructuur. 	G	Informeren	Positief	H
Scrum master, Application Engineer, Functioneel applicatie beheerder	<ul style="list-style-type: none"> Overzicht van aanwezige processen voor de ontwikkeling en beheer van Java-applicaties. Dit voor zowel de huidige en gewenste situatie(s). 	G	Informeren	Positief	H
Kerngroep Deployit ADM	<ul style="list-style-type: none"> Een generieke implementatie van Deployit. 	G	Informeren	Positief	G
Technisch applicatie beheerder, System Integrator, Implementatie-manager, Infra Service medewerker, Ontwikkelaar, Tester, Changemanager	<ul style="list-style-type: none"> Een naar waarheid beschreven eindproduct. 	L	Laag niveau tevreden houden	Positief	L

Tabel 19 - Stakeholders

Bijlage G: Interviewopzet

Ontwikkelproces huidige situatie, systeem BBS	
Interviewer	
Respondent	
Datum opmaak	
Versie	
Datum interview	
Tijd interview	
Locatie interview	

Het ontwikkel- en testproces

Specifiek versus generiek

De antwoorden op de vragen uit deze interviewopzet zijn vergaard bij de ontwikkelaars, testers, beheerders, etc. van het systeem BBS. De verkregen specifieke BBS-resultaten moeten generiek gemaakt worden. De generieke resultaten zijn de resultaten die ook geldig zijn bij de andere Java-applicaties op het Linux platform.

Topics

1. Achtergrondinformatie
2. Projectopzet
3. Procesbeschrijving
4. Actoren en rollen
5. Autorisatie, beveiliging en functiescheiding
6. Doorlooptijd
7. Changemanagement en configuratiemanagement
8. Wet- en regelgeving
9. BBS specifiek
10. Ontwikkeltool
11. Documentatie
12. Contactpersonen

Vragenlijst

Achtergrondinformatie

Introductie van het onderzoek

Wat is uw functie/rol binnen de organisatie?

Procesbeschrijving

Wat is de input van het proces ?

Wat is de output van het proces?

Wordt er een project opgezet voor de ontwikkeling van een applicatie?

Zo ja:

Hoe is de projectorganisatiestructuur ingedeeld?

Welke methodes worden gebruikt tijdens het project?

Beschrijf het proces door antwoord te geven op de volgende vragen:

- Welke activiteiten worden er uitgevoerd?
- Wat is de chronologische volgorde van deze activiteiten?
- Wie (rol) voert welke activiteiten uit?
- Welke beslissingen worden tussen de activiteiten gemaakt?
- Welke systemen en applicaties worden tijdens welke activiteiten gebruikt?
- Welke databronnen worden tijdens welke activiteiten gebruikt?

Actoren en rollen

Wat zijn de verantwoordelijkheden van de rollen tijdens het proces?

Autorisatie, beveiliging en functiescheiding

Wie (actor/rol) hebben autorisatie voor de uitvoering van de activiteiten, systemen en applicaties? *Geef daarbij per actor/rol aan voor welke activiteiten, systemen of applicaties dit geldt.*

Welke beveiligingsmaatregelen zijn van toepassing tijdens het ontwikkelproces?

Hoe wordt er gebruik gemaakt van functiescheiding?

Doorlooptijd

Wat is de gemiddelde doorlooptijd van het totale proces?

Wat is de gemiddelde doorlooptijd van het deploymentproces, oftewel de verplaatsing van de ontwikkelde applicatie naar een OTAP-omgeving?

Voor beide vragen:

Wat kan de reden zijn op het moment dat de doorlooptijd onder het gemiddelde ligt?

Wat kan de reden zijn op het moment dat de doorlooptijd boven het gemiddelde ligt?

Changemanagement en configuratiemanagement

Wat is de koppeling/communicatie tussen het ontwikkel- en testproces en changemanagement?

Wat is de koppeling/communicatie tussen het ontwikkel- en testproces en configuratiemanagement?

Wet- en regelgeving

Is er specifieke wet- en regelgeving van toepassing op het beschreven ontwikkel- en testproces ?

BBS specifiek

Wat zijn BBS-specifieke eigenschappen/kenmerken voor het proces? En wat is daar de achterliggende gedachte van?

Ontwikkeltool

Welke aandachtspunten zijn er voor de implementatie en inrichting van een ontwikkeltool?
In dit geval voornamelijk een ontwikkeltool die gericht is op het automatisch deployen van een applicatie.

Zijn er eisen en/of wensen m.b.t. het automatisch deployen van applicaties?

- Is er behoefte naar het automatisch deployen van applicaties? En waarom?
- Dient er bijvoorbeeld nog aan andere onderwerpen gedacht te worden?
- Zijn er onderwerpen en/of functionaliteiten die een ontwikkelaar graag voor ogen ziet met kijk naar het automatisch deployen van applicaties?
- Zijn er onderwerpen en/of functionaliteiten die behouden moeten worden met kijk naar het 'traditionele' deployen van applicaties?

Documentatie

Is er documentatie aanwezig van de besproken onderwerpen van het proces?

Zo ja, waar kan ik deze vinden?

Contactpersonen

Met welke andere rollen heb jij intensief contact? En waarom is dit contact benodigd?

Welke collega's kan ik benaderen voor de bespreking van de ondervonden resultaten?

Wie kan ik benaderen voor de bespreking van de generieke huidige situatie?


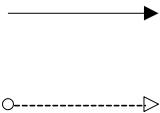
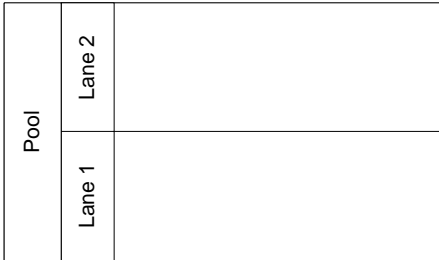
Wie kan ik benaderen voor de beschrijving van de processen die koppelingen hebben met het ontwikkelproces (bijvoorbeeld: testproces, changemanagementproces, configuratiemanagementproces)?

Bijlage H: BPMN 2.0

Deze paragraaf geeft een uitleg van de gebruikte onderdelen afkomstig van de modelleertaal BPMN 2.0 (Business Proces Modelling Notation, versie 2.0) zoals deze is gebruikt tijdens het onderzoek. In deze uitleg wordt enkel de onderdelen behandeld die van toepassing zijn voor het inrichten van de huidige- en gewenste situaties. Tijdens het documenteren van de uitleg is er gebruikt gemaakt van de BPMN 2.0 poster¹¹⁰.

Uitleg

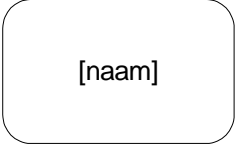
BPMN is opgebouwd vanuit vier basisobjecten namelijk, flow-objecten, connectie-objecten, en swimlanes (zwembanen).

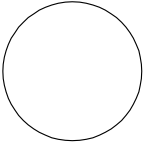
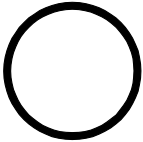

	Flow-objecten worden gebruikt om het normale procesverloop weer te geven. Deze objecten bevatten de bekende symbolen zoals events, activiteiten en gateways.
	Connectie-objecten worden gebruikt om de events, activiteiten, gateways en berichten aan elkaar te koppelen.
	Zwembanen bestaan uit 'Pools' en 'Lanes'. Pools en lanes kunnen gebruikt worden om verschillende rollen en verantwoordelijkheden te definiëren: bijvoorbeeld klant, leverancier en producent.

¹¹⁰ Berliner BPM-Offensive; BPMN 2.0 Poster; Geraadpleegd op 28 april 2013, van http://www.bpmb.de/images/BPMN2_0_Poster_NL.pdf

Objecten

In deze paragraaf zijn de objecten weergegeven die gebruikt zijn in de procesmodellen van dit afstudeeronderzoek.



Activiteiten	
	<p>Activiteit</p> <p>Een activiteit bestaat een handeling/werkzaamheid. In dit geval een handeling van de documentstroming.</p> <p><i>De [naam] dient compact verwoord te zijn (1-4 woorden).</i></p>

Gebeurtenissen (events)	
	<p>Start-event</p> <p>Een object om de start van het proces aan te geven. In één model is één start-event aanwezig.</p>
	<p>End-event</p> <p>Een object om het einde van het proces aan te geven. In één model zijn één of meer end-events en/of end-messages aanwezig.</p>
	<p>End-message</p> <p>Een object om het einde van het proces aan te geven inclusief een bericht wat verzonden wordt. In één model zijn één of meer end-events en/of end-messages aanwezig.</p>

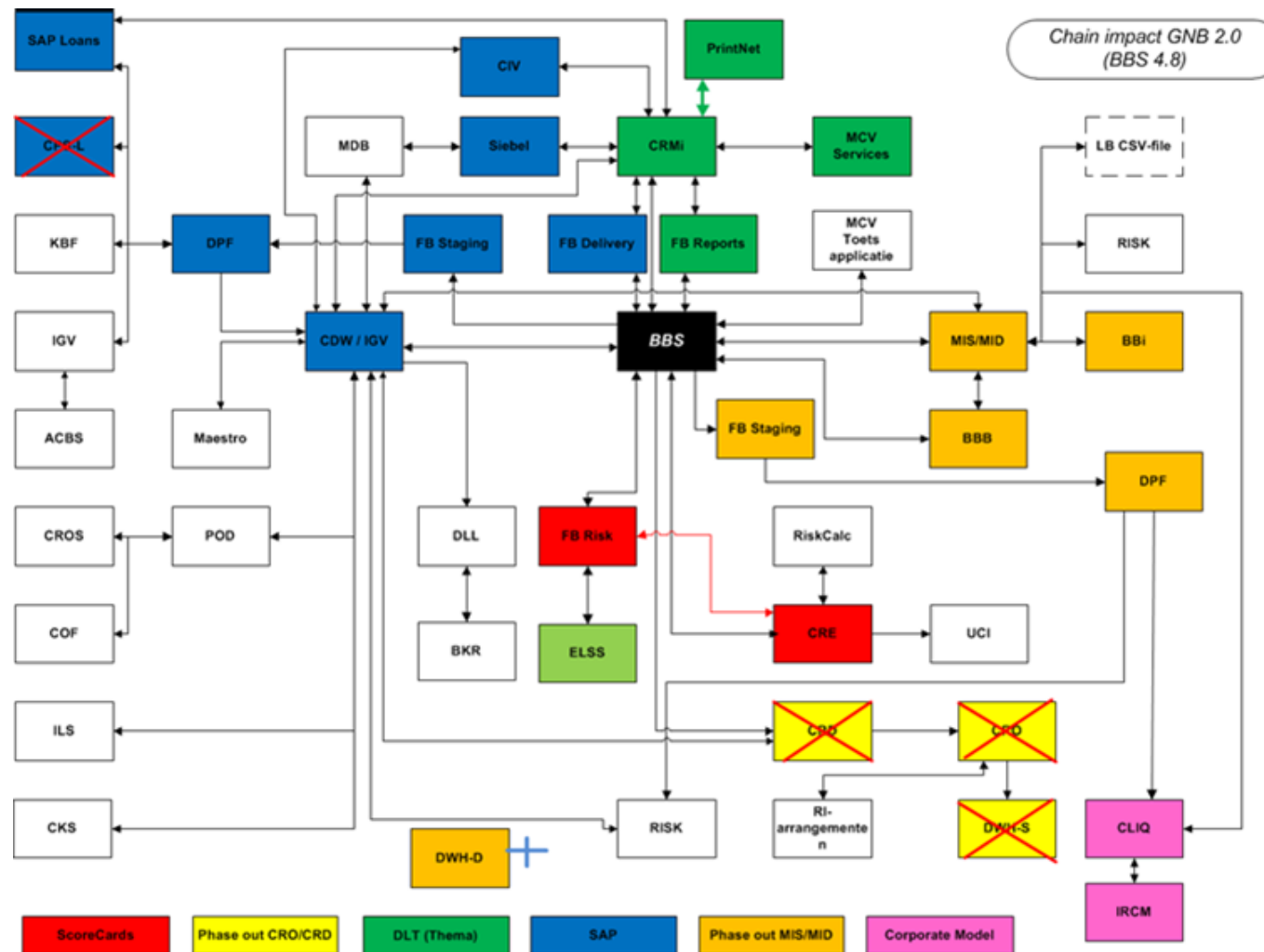
Beslissingsmomenten

	<p>Standaard gateway</p> <p>Bij het splitsen van een stroom wordt de flow naar exact één uitgaande stroom gerouteerd. Als twee stromen bij elkaar komen, wacht de uitgaande stroom totdat er één input binnengekomen is.</p>
	<p>Parallel gateway</p> <p>Als de inkomende stroom gesplitst wordt, worden de uitgaande stromen simultaan geactiveerd. Als er twee stromen bij elkaar komen, wordt de uitgaande stroom pas in gang gezet als alle inkomende stromen binnen zijn.</p>

Connectie-objecten

	<p>Sequence Flow</p> <p>Definieert de volgorde waarin activiteiten uitgevoerd worden.</p>
	<p>Message Flow</p> <p>Dit object symboliseert informatiestromen tussen organisaties. Een message flow kan worden verbonden aan pools, activiteiten of gebeurtenissen.</p>

Bijlage I: Contextplaat BBS



Bijlage J: Rolbeschrijvingen

Deze paragraaf beschrijft de aanwezige rollen tijdens het proces rondom de deployment van een Java-applicatie. De rollen zijn weergegeven in onderstaande tabel, bestaande uit de naam van de rol, een beschrijving en de verantwoordelijkheden. Er is gebruik gemaakt van de aanwezige functieprofielen die beschikbaar zijn gesteld door HR managers.

Rol	Beschrijving	Verantwoordelijkheden (m.b.t. het deploymentproces)
Application Engineer ¹¹¹	De Application Engineer vertaalt de business requirements conform de eisen van beleid en architectuur naar passende technische oplossingen. Hierbij ligt de primaire focus op zorgvuldige integratie van de applicatie en applicatie-onderdelen in het bestaande systemenlandschap. Deze integratie vindt plaats binnen de randvoorwaarden voor onder andere performance, betrouwbaarheid, beschikbaarheid, schaalbaarheid, beveiliging en reductie van complexiteit (standaardisatie). Het uitgangspunt is complexiteitsreductie c.q. optimale onderhoudbaarheid/maintainability op de langere termijn. De Application Engineer is in staat specifieke productkennis, applicatielandschap van een domein en de specifieke kennis van/over het te volgen proces binnen het VoortBrengingsProces (VBP) op een effectieve en efficiënte wijze te combineren.	<ul style="list-style-type: none">- Assisteren van (systeem)ontwikkeling en integratie bij de uitvoering van projecten en de invoering van Pakketten
Configuratiemanager	Managen van informatie rondom de ontwikkelversies en releaseversies.	<ul style="list-style-type: none">- Inzicht hebben in de aanwezige ontwikkel- en releaseversies op de aanwezige versiebeheersystemen.- Samenstellen en controleren van een geïntegreerde oplevering.
Database administrator	Beheert de databases in de (grote) acceptatieomgeving en de productieomgeving.	<ul style="list-style-type: none">- Het verzorgen van de database van de Java-applicatie binnen de

¹¹¹ (Rabobank, Functieprofiel Application Engineer, 2010)

		acceptatieomgeving (A) en productieomgeving.
Functioneel applicatie beheer (FAB) ¹¹²	Het functioneel applicatief inrichten en beheren van een of meerdere complexe geautomatiseerde omgevingen, volgens afgesproken service levels, zodanig dat de continuïteit van de business processen optimaal gewaarborgd wordt.	<ul style="list-style-type: none"> - Het op juiste wijze (doen) implementeren van nieuwe en gewijzigde onderdelen in de te beheren omgeving - Het participeren in projecten met als taken het bewaken dat de op te leveren producten voldoen aan de geldende beheereisen en het opdoen van kennis zodanig dat de beheerdienstverlening na oplevering zelfstandig uitgevoerd kan worden. - Het opstellen en onderhouden van beheerhandboeken en –handleidingen.
Implementatiemanager	Organiseren en vergaren van de juiste resources voor de implementatie van een applicatie.	<ul style="list-style-type: none"> - Het plannen van de juiste resources
Infra service medewerker	Het landen van een applicatie in de aanwezige infrastructuur en de fysieke machine.	<ul style="list-style-type: none"> - Het verzorgen van de technische koppelingen rondom een Java-applicatie.
Ontwikkelaar ¹¹³	Het (in projectverband) ontwikkelen en onderhouden van kwalitatief hoogwaardige IT-systemen, waarbij zowel de bedrijfsdoelstellingen van de opdrachtgevers optimaal ondersteund worden (functionaliteit) als de doelstellingen van de IT-omgeving m.b.t. het ontwikkelen, beheren en exploiteren van de systemen. In dit geval betreft het de ontwikkeling van Java-applicaties.	<ul style="list-style-type: none"> - Ontwikkelen van software. - Opstellen en onderhouden van programma- en systeemdokumentatie conform de standaards en richtlijnen.
Product Owner	Bepaalt welke werkzaamheden nodig zijn vanuit de business kant.	<ul style="list-style-type: none"> - Geeft een approval op de ontwikkelde release om gereleased te worden naar de omgevingen binnen het B&E domein.
System integrator ¹¹⁴	Is binnen de projecten verantwoordelijk voor de levering en werking van ontwikkel-, test- en acceptatieomgevingen (OTa), alsmede ketentestomgevingen en productie acceptatie omgevingen (PA).	<ul style="list-style-type: none"> - Uitvoeren intake ten behoeve van realisatie en exploitatie van de testomgeving(en) n.a.v. een vraag vanuit een project of programma. - Borgen dat randvoorwaardelijke inputdocumentatie aanwezig is en kwalitatieve toets op documentatie uitvoeren (teststrategie, testplan, PSA, PAC, SOPO, ISP) - Op basis van de inputdocumentatie een ontwerp maken voor de benodigde

¹¹² (Rabobank, Functieprofiel Beheerder versie 1.0, 2011)

¹¹³ (Rabobank, Functieprofiel Ontwikkelaar versie 1.0, 2011)

¹¹⁴ (Rabobank, Functieprofiel System Integrator Test Supply, 2013)

		<p>testomgevingen en dit afstemmen met de betrokken stakeholders (Projectmanager, Testmanager, Application Engineer, Functioneel Applicatie Beheerders, Technisch Applicatie Beheerders)</p> <ul style="list-style-type: none"> - Inzichtelijk maken uit welke onderdelen de benodigde testomgeving(en) bestaat, inclusief afhankelijkheden naar testomgevingen die in gebruik zijn bij andere projecten of beheerafdelingen - Bewaken van de technische samenhang van de testomgeving(en)
Technisch applicatie beheerder (TAB) ¹¹⁵	Het technisch applicatief inrichten en beheren van een of meerdere complexe geautomatiseerde omgevingen, volgens afgesproken service levels, zodanig dat de continuïteit van de business processen optimaal gewaarborgd wordt.	<ul style="list-style-type: none"> - Het op juiste wijze (doen) implementeren van nieuwe en gewijzigde onderdelen in de te beheren omgeving - Het participeren in projecten met als taken het bewaken dat de op te leveren producten voldoen aan de geldende beheereisen en het opdoen van kennis zodanig dat de beheerdienstverlening na oplevering zelfstandig uitgevoerd kan worden. - Het opstellen en onderhouden van beheerhandboeken en –handleidingen.
Tester	Test Java-applicaties.	<ul style="list-style-type: none"> - Het functioneel testen van de Java-applicatie(s).

¹¹⁵ (Rabobank, Functieprofiel Beheerder versie 1.0, 2011)

Bijlage K: Benodigde resources

Voor het in productie nemen van een Java-applicatie, nadat de release is opgeleverd vanuit het proces ontwikkeling, dienen er meerdere partijen geraadpleegd te worden. Deze paragraaf beschrijft de benodigde partijen met daarbij de benodigde input en output.

Infra Service Oracle: Database administrator

Voor aanpassingen aan de structuur van de database dient een database administrator (DBA) ingeschakeld te worden. De aanvraag gebeurt via het systeem Servicemanager aan de hand van een change. Voor de realisatie van de databasewijziging wordt een doorlooptijd van maximaal 5 dagen toegekend. In de praktijk is dit in de meeste gevallen 1 dag, dit is echter wel afhankelijk van de omvang van de aanvraag.

Input:

1. Deploymentscript (verkregen van de ontwikkelaars).
2. Productieconfiguratie (PAT en P specifieke omgevingsconfiguraties en omgevingsinrichting).

Output:

1. Log file script (waarin de voortgang en voltooiing van de aanpassing geregistreerd is).
2. Bijgewerkte database.

Infra Service Oracle: IS-koppelingen specialist

Voor het plaatsen van IS-koppelingen, waaronder MQ-koppelingen en http-koppelingen, dient een specialist ingeschakeld te worden. De aanvraag voor een koppeling gebeurt op dezelfde wijze als voor een aanpassing aan de database.

Input:

1. Aanvraagformulier (wordt ingevuld door TAB BBS, FAB BBS en IS-Linux).

Output:

1. Werkende koppeling.

Infra Service Linux: Specialist

Voor de inrichting van Websphere dient een specialist van IS Linux ingeschakeld te worden. De specialist communiceert met de ontwikkelaar van de applicatie voor de specifieke details.

Input:

1. Release.
2. Manuals.
3. Productieconfiguratie (PAT en P specifieke omgevingsconfiguraties en omgevingsinrichting).

Output:

1. Ingerichte Websphere omgeving.

Rabobank International

Een applicatie communiceert met externe systemen. Voornamelijk is dit met de systemen van Rabobank International (RI). Voor de communicatie dient een koppeling gelegd te worden, dit betreft in bijna alle gevallen een http(s)-koppeling. Om de koppeling te realiseren dient de RI omgeving hierop ingericht te worden. De communicatie met RI vindt telefonisch en per email plaats, voor de deployment van een applicatie dient er rekening gehouden te worden met dat RI medewerkers niet in de avonden en weekenden raadpleegbaar zijn.

Input:

1. Informatie-uitwisseling over de structuur van de RI en RN omgeving.

Output:

1. Werkende koppeling(en).

CRMi

Voor de plaatsing van een interne koppeling wordt dit in de meeste gevallen gedaan via het systeem CRMi. CRMi is een breed draagvlak voor koppelingen met andere applicaties. Indien er een koppeling nodig is dient er rekening gehouden te worden met de zogeheten distributierelease. De distributierelease is de release voor meerdere applicaties, waaronder CRMi en Siebel. Iedere twee maanden vindt deze release plaats. Voor CRMi betekent dit dat na iedere release de aangevraagde koppelingen beschikbaar zijn. Een release betekent voor CRMi de aanpassing of toevoeging van ongeveer 150 tot 200 koppelingen.

Input:

1. Scope document en oplevering vanuit CRMi-ADM.
2. Formulier met verzoek om de koppeling te 'unlocken'.

Output:

1. Werkende koppeling.

Bijlage L: Doorlooptijden

In paragraaf 5.7 *Doorlooptijden* zijn de doorlooptijden weergegeven van het deploymentproces. Deze paragraaf geeft een toelichting aan deze tijden.

Ontwikkelomgeving (O) en testomgeving (T)

De ontwikkelomgeving en testomgeving hebben dezelfde deployment frequentie, namelijk dagelijks. De tijd die benodigd is om de ontwikkelde en/of geteste applicatie te deployen op de omgeving duurt één tot twee uur. Op het moment dat er meerdere bestanden aan de orde zijn, bijvoorbeeld een EAR bestand met configuratiebestanden, is de kans op fouten groter. In zeker 30% van de gevallen kan de deployment daardoor twee uur in beslag nemen. Na zes tot acht weken wordt de ontwikkelde en geteste applicatie gedeployd op de IRT-omgeving.

De reden van totale doorlooptijd van zes tot acht weken heeft binnen BBS een specifieke reden. In de eerste sprints wordt er namelijk gebruik gemaakt van de productieve versie van het 'moedersysteem' BBS. Naarmate er sprints uitgevoerd worden, wordt de nieuwste versie van BBS gebruikt. Er wordt gebruik gemaakt van de laatste versie zodat de kans groter is dat de integratie regressie test (IRT) succesvol wordt uitgevoerd.

IRT-omgeving (IRT)

Iedere zes tot acht weken wordt de ontwikkelde en geteste applicatie op de IRT-omgeving gedeployd. Een deployment van de ontwikkelde en geteste applicatie neemt ongeveer vier uur in beslag. Dit heeft te maken met de keten van applicaties die gedeployd worden op de omgeving, dit is een complexere actie dan de deployment van een enkele applicatie.

Acceptatieomgeving (a)

Een grote (keten)release, voor portfolio Financieren Bedrijven is de GNB-release (Geld Nodig Bedrijven) een voorbeeld, wordt twee maal per jaar gedeployd naar de acceptatieomgeving. Deze kleine 'a'-omgeving wordt ook de UAT-omgeving genoemd. De deployment naar een UAT omgeving neemt twee tot vier dagen in beslag. De totale doorlooptijd op de UAT omgeving duurt één week. De benodigde deployment-tijd heeft voornamelijk te maken met het ontvangen en moeten begrijpen van (onvolledige) manuals, niet aanwezige koppelingen en het moeten schrijven van een draaiboek.

Acceptatieomgeving (A)

Deze grote 'A'-omgeving wordt ook de PAT-omgeving genoemd. De deployment naar een PAT omgeving neemt één dag in beslag. De totale doorlooptijd op de UAT omgeving duurt, inclusief wachttijden, drie weken. Omdat de deployment-fouten tijdens de UAT verholpen zijn verloopt de deployment naar de PAT omgeving sneller. De totale doorlooptijd op deze omgeving heeft te maken met de benodigde stappen die uitgevoerd moeten worden. Er wordt namelijk een Pre-PAT gecreëerd waarop een 'proof' wordt gedaan bij één lokale Rabobank database, dit neemt ongeveer twee dagen in beslag. Vervolgens wordt de applicatie opgeschaald naar meerdere Rabobanken, dit neemt drie dagen in beslag. Nadat alle databases van de lokale Rabobanken gekoppeld zijn aan de nieuwe ontwikkelde applicatie vindt er een clean-up plaats. De clean-up bestaat uit het opschonen van de oude database. Na de clean-up start de werkelijke PAT, deze implementatie neemt inclusief de deployment en de databasemigratie drie dagen in beslag.

Productieomgeving (P)

Hetgeen wat in de PAT getest is wordt op de productieomgeving in de werkelijkheid uitgevoerd. Dezelfde onderdelen als op de PAT worden uitgevoerd, namelijk de proof, het opschalen en de clean-up. Dit omvat een deployment naar de Pre-productie en een deployment naar de productieomgeving wat samen gemiddeld twee dagen in beslag neemt.

Bijlage M: Wensen, verwachtingen en aandachtspunten

De wensen en verwachtingen zijn gescheiden in de afdeling ADM en de afdeling B&E. De laatste paragraaf beschrijft de aandachtspunten met betrekking tot de implementatie van een tool die automatische deployments mogelijk maakt.

ADM domein

Wensen

- Verhogen van de consistentie van het deploymentproces
- Verhogen van de herhaalbaarheid van het deploymentproces.
- Verhogen van de voorspelbaarheid van het deploymentproces.
- Frequenter deployen.
- Reductie van fouten tijdens het deploymentproces.
- Ontwikkelaar zo min mogelijk (extra) overhead.
- Inzicht in de versies van de applicaties in de OTAP omgeving.
- Zoveel mogelijk automatiseren vanaf de ontwikkeling van de applicatie tot en met de applicatie in productie nemen.
- Het overzichtelijk houden van de releaselijst.

Verwachtingen

- Meeste baat bij de deployment naar de A en P omgevingen (daar wordt de meeste reductie van het deploymentproces behaald).
- Afhankelijkheid van de mensen wordt minder met de komst van Deployit.
- Repeterende werk wordt geautomatiseerd.
- Continu kunnen testen van een wijziging in de hele keten.
- Eerder ontvangen van feedback over het deploymentproces.
- Alle informatie dient opgeslagen te worden zodat een deployment reproduceerbaar is.

B&E domein

Wensen

- In een vroegere fase inzicht hebben in de ontwikkeling van een release.
- Ontwikkelaar configureert (richt in) Deployit zodat eventuele problemen al in een vroeg stadium worden opgelost.
- Meer inzicht krijgen in de ontwikkeling van een applicatie, dit (ook) vanuit het DevOps principe.
- Deployit wordt gedragen door Oracle, dat wil zeggen dat Oracle de databasewijzigingen via Deployit laat verlopen.
- TAB wil de volgende onderdelen kunnen aanpassen in Deployit:
 - o Eigenschappen rondom de log files (zoals: het aantal logs dat wordt opgeslagen).
 - o Property files, dit i.v.m. de omgeving specifieke configuratie.

Verwachtingen

- De aanwezige functiescheiding dient geconfigureerd te kunnen worden in Deployit.
- Een Infra Service medewerker die de benodigde koppelingen configureert behoudt hetzelfde werk. Enkel de tool verandert.
- Een medewerker zonder ervaring en kennis van Websphere zal kunnen deployen.
- Een deployment zal bestaan uit een deployment van de applicatie en een deployment van de benodigde koppelingen.
- Reductie van fouten tijdens het deploymentproces (zowel van de applicatie en de koppelingen).
- Overzichtelijke rapportages over o.a. het aantal deployments met daarbij de deployment-tijden.

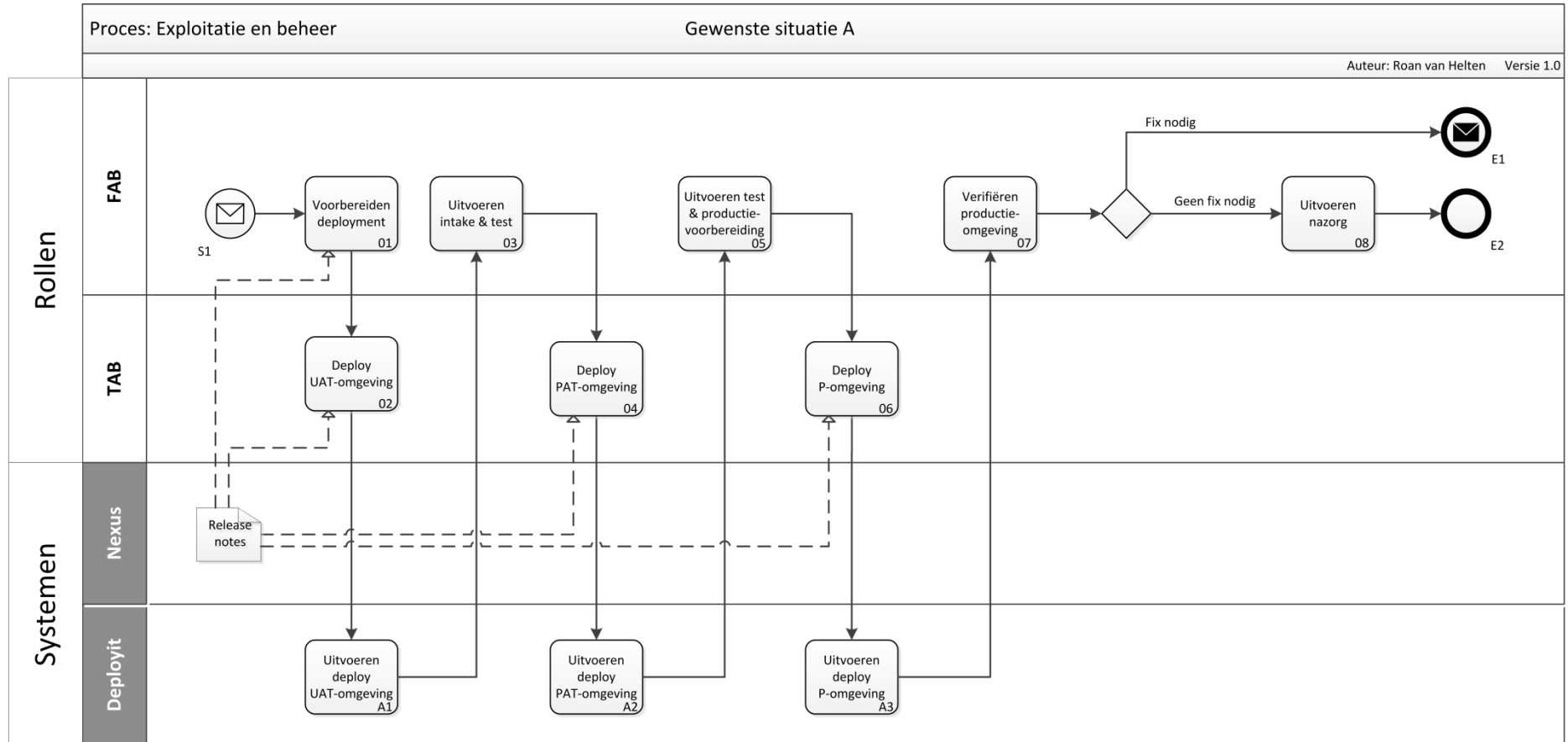
Aandachtspunten

Met kijk naar de implementatie van Deployit zijn de volgende aandachtspunten van belang.

- **Ontwikkeling**
 - o Applicatie niet stabiel genoeg na 1 sprint.
 - o De tester vertrouwt de ontwikkelaar dat hij/zij geen aanpassingen maakt aan de ontwikkelde source code voordat een build gemaakt wordt na de positief doorlopen test.
 - o De server Hudson heeft geen schrijfautorisatie op de server Subversion.
- **Potentiele kans op menselijke fouten**
 - o Aanwezige manuals bij de releas(es) zijn onvolledig en/of incorrect.
 - o Tijdens de deployment worden handmatige fouten gemaakt tijdens het overnemen van de koppelingen.
 - o De aanvraag van benodigde koppelingen wordt soms over het hoofd gezien.
 - o Firewalls zijn niet altijd correct geconfigureerd.
 - o De deployment van de applicatie en koppelingen gebeurt aan de hand van zelfgemaakte scripts die niet altijd volledig functioneren.
 - o Tijdens de eerste deployment binnen het B&E-domein moeten veel koppelingen handmatig ingesteld worden.
- **Complexiteit**
 - o Een omgeving heeft ontwikkel specifieke configuratiebestanden nodig om de deployment succesvol te laten verlopen.
 - o Aanwezige structuur van de databases binnen de OTAP omgevingen kunnen verschillen.
 - o Afhankelijkheid van de benodigde resources binnen het B&E domein
- **Overige**
 - o Tijdens kleine releases kan het voorkomen dat de UAT overgeslagen wordt.

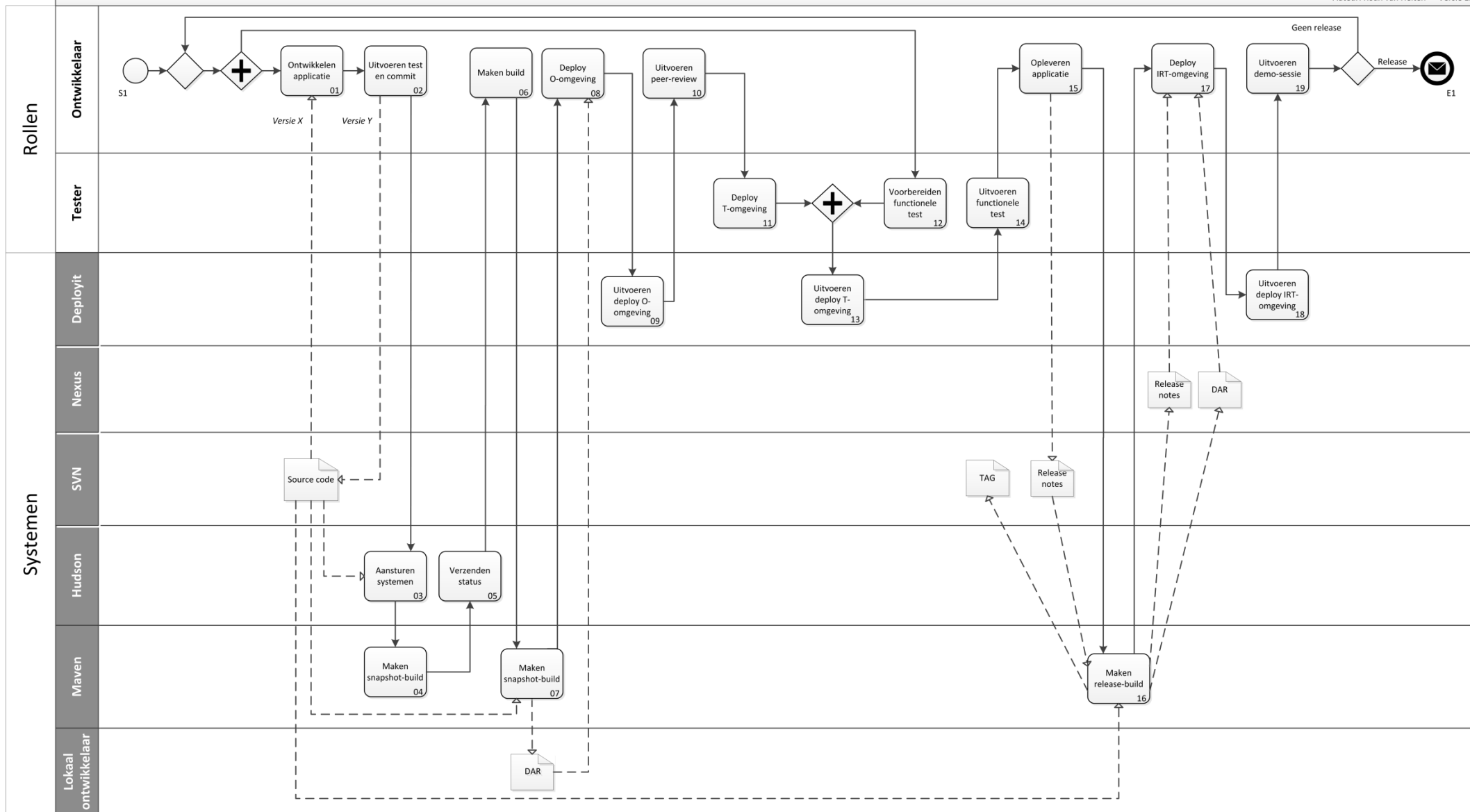
Bijlage N: Gewenste situatie A

Exploitatie en beheer

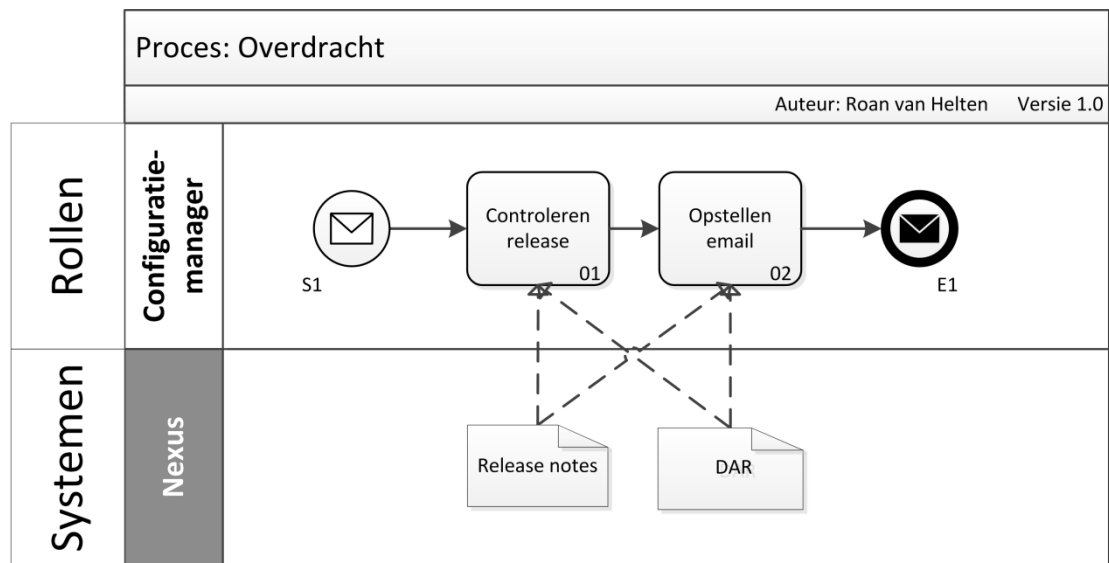


Ontwikkeling

Z.O.Z.



Overdracht



Bijlage O: Gewenste situatie B

Ontwikkeling

Z.O.Z.

