



Faculteit Bedrijf en Organisatie

Geautomatiseerd gebruiker acceptatie testen in Microsoft Dynamics 365 Finance & Operations gebruik
makende van de Regression Suite Automation Tool

Jules Decock

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Martine Van Audenrode
Co-promotor:
Wout Van den Broucke

Instelling: Avanade

Academiejaar: 2020-2021

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Geautomatiseerd gebruiker acceptatie testen in Microsoft Dynamics 365 Finance & Operations gebruik
makende van de Regression Suite Automation Tool

Jules Decock

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Martine Van Audenrode
Co-promotor:
Wout Van den Broucke

Instelling: Avanade

Academiejahr: 2020-2021

Tweede examenperiode

Woord vooraf

Deze bachelorproef werd geschreven tot het bekomen van de graad van professionele bachelor in de toegepaste informatica met als afstudeerrichting e-business. Ik heb ervoor gekozen de Regression Suite Automation Tool (RSAT) te onderzoeken, dit vanwege de vernieuwende technologie alsook het uitbreiden van mijn persoonlijke kennis binnen ERP. Na reeds intensief gebruik te maken van SAP, dit zowel tijdens de lessen als tijdens mijn stage leek het me interessant en leerrijk om ook een diepere kennis te verwerven van Microsoft ERP software. Naast de zeer technische opleiding leek het me ook verrijkend een minder technisch onderwerp te onderzoeken en bij te dragen aan de latere uitrol zodat niet technisch geschoolden deze tool kunnen gebruiken.

Deze scriptie zou niet tot stand zijn gekomen zonder de hulp van verschillende personen. Hierbij wil ik deze personen oprecht bedanken voor de hulp tijdens het onderzoek.

Om aan te vangen wil ik mijn co-promotor, Wout Van den Broucke, bedanken voor de hulp gedurende het hele onderzoek. Dankzij de regelmatige feedback en bijsturing kon ik deze studie tot een goed einde brengen. Daarnaast kon ik ook steeds bij hem terecht indien ik met onduidelijkheden zat of niet wist hoe verder te gaan. Daarnaast wil ik ook de firma Avanade¹ bedanken voor de kans die ze me gaven dit onderzoek bij hen uit te voeren.

Ten slotte wil ik ook mijn promotor, Martine Van Audenrode, bedanken voor de feedback betreffende de inhoud en structuur van de scriptie. Bij onduidelijkheden of vragen kon ik ook steeds bij haar terecht voor verdere uitleg en begeleiding.

Ten slotte wil ik ook mijn ouders bedanken voor de steun tijdens de gehele opleiding.

¹ www.avanade.com

Samenvatting

Automatiseren van processen is een trend die zich reeds vele jaren doorzet. Maar kan deze trend ook worden toegepast bij het testen van applicaties? De Regression Suite Automation Tool van Microsoft laat je toe geautomatiseerde gebruiker acceptatie testen te schrijven voor Microsoft Dynamics 365 Finance & Operations applicaties. In deze studie onderzoeken we hoe de Regression Suite Automation Tool geconfigureerd moet worden. Daarnaast onderzoeken we ook hoe een correcte test opgesteld moet worden. Het is immers zeer belangrijk dat een test correct geconfigureerd is, enkel een correcte test kan je uitsluitsel geven over de correctheid van je applicatie. Naast het configureren van RSAT en correcte testgevallen onderzoeken we ook hoe deze testen nu juist worden uitgevoerd en welke resultaten er worden opgeleverd. Het opstellen van een correcte geautomatiseerde gebruiker acceptatie testen is afhankelijk van verschillende zaken. De grootste hoeveelheid aan mogelijke fouten kan zich voordoen tijdens het opnemen van een test scenario. Hier is het essentieel rekening te houden met verschillende factoren. Zo is het belangrijk steeds compact, nauwkeurig en specifieke te zijn. Eens deze factoren in acht genomen zijn zal het opstellen van een correcte test steeds vlot verlopen.

De Regression Suite Automation Tool in de praktijk is momenteel nog een heel ander verhaal. De workload nodig om deze geautomatiseerde gebruiker acceptatie testen op te stellen en te onderhouden is hoog, wat vervolgens resulteert in een hoge kost. Om deze kost terug te verdienen moet de test zeer frequent worden uitgevoerd, en dit is vandaag de dag nog niet het geval. Naar de toekomst toe, zal de move naar de Cloud een grote invloed hebben op het gebruik van geautomatiseerde gebruiker acceptatie testen en de Regression Suite Automation Tool op zich.

Inhoudsopgave

1	Inleiding	13
1.1	Probleemstelling	14
1.2	Onderzoeksvraag	14
1.3	Onderzoeksdoelstelling	15
1.4	Opzet van deze bachelorproef	15
2	Stand van zaken	17
2.1	ERP	17
2.2	Microsoft Dynamics 365 Finance & Operations	19
2.3	Het V-model	21
2.4	Gebruiker Acceptatie testen	22
2.5	Geautomatiseerd acceptatie testen	23
2.6	Azure Devops	24

2.7	Regression Suite Automation Tool	25
3	Methodologie	27
3.1	Configureren van RSAT	27
3.2	Aanmaken van testen	28
3.3	Uitvoeren en resultaten van testen	29
4	Regression Suite Automation Tool	31
4.1	Configuratie	32
4.1.1	Nodige applicaties	32
4.1.2	Configureren van RSAT	33
4.2	Aanmaken van testen	35
4.2.1	Scenario opnemen	36
4.2.2	Test toevoegen aan Azure DevOps	38
4.2.3	Test toevoegen aan RSAT en testparameters wijzigen	40
4.3	Uitvoeren van testen	41
4.3.1	Geslaagde test	41
4.3.2	Gefaalde test	41
5	Kritische noot	47
6	Conclusie	53
6.1	Hoe configureer je RSAT?	53
6.2	Hoe stel je een correcte test op?	54
6.3	Hoe voer je testen uit aan de hand van RSAT?	55
6.4	Hoe worden de resultaten van een test weergegeven?	56

A	Onderzoeksvoorstel	57
A.1	Introductie	57
A.2	Stand van zaken	58
A.3	Methodologie	59
A.4	Verwachte resultaten	59
A.5	Verwachte conclusies	60
B	Kritsche noot: gestelde vragen	61
	Bibliografie	63

Lijst van figuren

2.1	Overzicht V-Model (Yadav, 2012)	21
4.1	Grafische voorstelling relaties tussen verschillende applicaties.	32
4.2	Azure DevOps Configuration.	33
4.3	Finance and Operations Test Environment.	34
4.4	Run Settings.	35
4.5	Weergave uitgevoerde stappen.	37
4.6	Toevoegen test in Azure DevOps.	39
4.7	Overzicht testen in Azure DevOps Test Suite.	39
4.8	Overzicht testen in de Regression Suite Automation Tool.	40
4.9	Succesvolle test weergegeven in de Regression Suite Automation Tool.	41
4.10	Succesvolle test weergegeven in Azure DevOps.	41
4.11	Gefaalde test weergegeven in de Regression Suite Automation Tool.	42
4.12	Gefaalde test weergegeven in Azure DevOps.	42

1. Inleiding

Microsoft Dynamics 365 Finance & Operations is ERP software ontwikkeld door Microsoft voor middelgrote tot grote ondernemingen. Aan de hand van deze software kunnen bedrijven hun centrale processen beter beheren. Zo kan stock gecontroleerd worden, facturen ingebracht en veel meer. Natuurlijk werkt elke onderneming op zijn eigen manier met elk zijn eigen processen en aanpak. Hierdoor moet elke Dynamics 365 Finance & Operations applicatie voldoen aan een specifieke implementatie op maat van de onderneming die gebruik zal maken van deze software. Een groot deel van deze wijzigingen vallen onder de categorie configuratie. Hierbij worden verschillende instellingen gewijzigd zodat de processen die het systeem bevat aangepast zijn naar de processen van de onderneming. Echter is niet elke vereiste functionaliteit mogelijk door het wijzigen van de configuratie. Indien dit het geval is ontwikkelt men specifieke code voor deze vereiste te verwezenlijken.

Natuurlijk moeten Dynamics 365 Finance & Operations applicaties getest worden zowel voor het live gaan als ook na de lancering. Een applicatie zal tijdens het ontwikkelproces verschillende test fasen doorlopen. Zo zal de code getest worden, de integratie van de verschillende componenten, integratie op het systeem, Een van de type testen die steeds zullen worden uitgevoerd zijn gebruiker acceptatie testen. Hierbij wordt het systeem doorlopen door een medewerker van het bedrijf en controleert men of alle stappen steeds voldaan kunnen worden. Deze testen worden uitgevoerd alvorens de applicatie live gaat maar ook na elke nieuwe update die de applicatie ondergaat.

Automatisatie van processen is een trend die zich reeds vele jaren doorzet. Dankzij automatisatie kan men een proces vaker en met minder resources uitvoeren. Vermits gebruiker acceptatie testen vaak eenzelfde testplan volgen is het opportuun om ook deze processen te automatiseren. Op deze manier kan vaker en met minder resources verschillende test-

plannen worden uitgevoerd. De Regression Suite Automation Tool of RSAT zorgt ervoor dat een test scenario kan worden opgenomen aan de hand van een recorder om vervolgens geautomatiseerd te worden. Deze automatisatie kan men telkens opnieuw laten lopen (Hanssen & Haugset, 2009).

1.1 Probleemstelling

Het testen van Dynamics 365 Finance & Operations applicaties neemt een groot deel van de ontwikkeltijd in beslag. Dit is zeker en vast ook het geval bij acceptatie testen. Deze testen worden in samenwerking met de klant opgezet en vervolgens door de klant zelf uitgevoerd. Echter moet elk van deze testen handmatig worden doorlopen. Deze doorlooptijd neemt veel tijd in beslag. Sinds de lancering van Microsoft One version wordt gebruik gemaakt van continue uitbreiding en verbetering van Dynamics 365 platform. Deze continue updates betekenen echter dat er steeds acceptatie testen moeten worden doorlopen om te controleren of de nieuwste update geen gevolgen heeft voor de werking van de software.

Deze continue wijzigingen zorgen ervoor dat na elke versie opnieuw mensen het opgestelde Test Plan moeten doorlopen. Dit zorgt voor extra werk en een verhoogde kost vermits reeds geteste software opnieuw moeten doorlopen worden, om die momenten kan er geen ander werk verricht worden.

Dankzij de Regression Suite Automation Tool van Microsoft kunnen dergelijke testen geautomatiseerd worden. Vermits deze tool nog maar sinds maart 2020 beschikbaar is, is de kennis die men heeft nog zeer beperkt, hierdoor is niet geweten wat deze tool kan en of er voordelen verbonden zijn aan het gebruik ervan.

1.2 Onderzoeksvraag

Om de Regression Suite Automation tool verder te onderzoeken werden een onderzoeksvraag en een aantal deelvragen opgesteld. Aan de hand van de stand van zaken en onderzoek met de Regression Suite Automation Tool zal dit onderzoek onderstaande vragen proberen te beantwoorden.

Hoe implementeer je de Microsoft Regression Suite Automation Tool voor het geautomatiseerd gebruiker acceptatie testen van Microsoft Dynamics 365 Finance & Operations applicaties?

- Hoe configureer je de Regression Suite Automation Tool?
- Hoe stel je een correcte test op?
- Hoe voer je testen uit aan de hand van RSAT?
- Hoe worden de resultaten van een test weergegeven?

1.3 Onderzoeksdoelstelling

Het doel van dit onderzoek is aantonen hoe de Regression Suite Automation Tool werkt. Welke beperkingen zijn er aan verbonden en waar liggen de valkuilen voor gebruikers. Door middel van het opzetten van verschillende testgevallen, onderzoeken we wat de complexiteit is, waar ondersteuning voor niet technisch geschoolde mensen nodig is,

Een tweede doel van het onderzoek is het achterhalen of de Regression Suite Automation Tool gemakkelijk kan geïmplementeerd worden voor reeds bestaande gebruiker acceptatie testen. Hiermee trachten we te achterhalen of de tool eenvoudig bij reeds lopende projecten toegepast kan worden in de toekomst.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, worden de bekomen resultaten uit het onderzoek weergegeven.

In Hoofdstuk 5, laten we een kritische stem betreffende de Regression Suite Automation Tool aan het woord.

In Hoofdstuk 6, wordt een conclusie getrokken uit de bevonden resultaten. Ook wordt een antwoord geformuleerd op de onderzoeksvraag en verschillende deelvragen.

2. Stand van zaken

In onderstaande literatuurstudie nemen we een diepere kijk op waar de Regression Suite Automation tool zich juist bevindt binnen de bedrijfssoftware. RSAT wordt gebruikt voor het geautomatiseerd testen van Microsoft Dynamics applicaties. Om goed te begrijpen wat Dynamics juist is, bespreken we eerst ERP. Het is vervolgens ook belangrijk de grootste spelers op de ERP-software markt met elkaar te vergelijken. We bekijken SAP en welke veranderingen door de jaren heen plaatsvonden. Vervolgens nemen we op een gelijkaardige manier een kijk bij Microsoft. Na het overlopen van beiden gaan we dieper in op Microsoft en laten we SAP ter informatie. Dit vermits RSAT een door Microsoft ontwikkelde tool is. Om een goed beeld te vormen van RSAT is het ook belangrijk te weten waar de testen die RSAT automatiseert zich bevinden binnen het gehele ontwikkelproces van een applicatie. We overlopen eerst de gehele test-cyclus van een software applicatie aan de hand van het V-Model om vervolgens te focussen op gebruiker acceptatie testen en RSAT op zich.

2.1 ERP

ERP of Enterprise Resource Planning is software die een onderneming of bedrijf in staat stelt de verschillende bedrijfsprocessen die de onderneming uitvoert te ondersteunen. Een belangrijk aspect hierbij is dat een ERP-systeem deze verschillende bedrijfsprocessen combineert tot een samenhangend en geïntegreerd systeem. Voordien maakte men vaak gebruik van verschillende software modules of zelfs applicaties per departement. Het grote voordeel om de verschillende processen te centraliseren, is het minimaliseren van achterhaalde master data en het versnellen van deze processen. Enkele voorbeelden van dergelijke processen zijn: procure-to-pay, goods receipt, sales order, Hierbij is master

data, data die de basis voor een onderneming vormt. Dit kan gaan over klantgegevens, gegevens van leveranciers en dergelijke. Het is essentiële data voor de centrale activiteiten van de onderneming.

Het belangrijkste aspect van ERP-software is de centrale databank. Deze centrale databank bevat alle master- en transactie data die de verschillende bedrijfsprocessen aanspreken. Vermits data steeds naar deze centrale databank wordt weggeschreven en vervolgens ook uitgelezen wordt is er geen sprake van verouderde of redundante data. Dit in grote tegenstelling tot hoe bedrijven vroeger te werk gingen. Toen dezelfde data vaak op meerdere plaatsen opgeslagen werd en hierdoor niet steeds accuraat was.

Er zijn verschillende ondernemingen die ERP-software ontwikkelen. De twee grootste spelers op de markt zijn SAP en Microsoft. Onderstaand bespreken we kort deze twee ERP-software leveranciers. SAP is een van de werelds grootste software producenten met een grote focus op ERP. Met de originele SAP R/2- en SAP R/3-software zette SAP een hoge standaard voor Enterprise Resource Planning. Ondertussen hebben ze met hun SAP S/4Hana systeem de stap naar in-memory computing gezet. Bij in-memory computing wordt de data rechtstreek in het RAM-geheugen opgeslagen. Het grote voordeel van in-memory computing is dat data veel sneller aangesproken kan worden door de verschillende processen. Om de zoveel tijd zal deze in-memory opgeslagen data worden weggeschreven naar de achterliggende databank. Dit is in grote lijnen wat het nieuwe S/4HANA systeem zo krachtig maakt. In het verder verloop van het onderzoek zal SAP niet meer aanbod komen. Dit vermits RSAT een Microsoft specifieke tool is.

Zoals eerder vermeld is Microsoft een tweede grote speler op de ERP-software markt. We nemen een diepere kijk in het Microsoft ecosysteem en de verschillende aspecten die belangrijk zijn om een globaal beeld van de situatie te vormen. We beginnen bij de oorsprong van Microsofts ERP-software.

Tussen 2001 en 2002 heeft Microsoft 4 verschillende accounting systemen overgekocht. Deze accounting systemen waren in 1983 reeds tot de publieke markt toegetreden. Na de overname van deze accounting systemen werden deze applicaties door Microsoft hernoemd. Deze naamswijziging betekende tevens ook het begin van Dynamics ERP software. Dynamics AX was het enige ERP systeem dat geschikt was voor grote ondernemingen. Toen men besloot cloud based te werken was de keuze om AX te herontwikkelen een eenvoudige keuze. Met deze herontwikkeling van Dynamics AX kwam ook een nieuwe naam. Sinds dat moment spreken we niet langer over Microsoft Dynamics AX maar over Microsoft Dynamics 365 Finance & Operations, zoals we het vandaag de dag kennen.

De switch naar een cloud based applicatie bracht vele voordelen met zich mee. Zo heb je steeds toegang op elk met internet verbonden apparaat tot het systeem. Daarnaast heb je nog tal van voordelen zo kunnen nieuwe updates snel en eenvoudig worden doorgevoerd, schalen is eenvoudiger en minder kost gebonden en tal van andere voordelen. Dit terwijl de nadelen eerder beperkt bleven.

2.2 Microsoft Dynamics 365 Finance & Operations

Zoals eerder vermeld is Dynamics 365 Finance & Operations ERP deel van Microsofts software lijn. Microsoft Dynamics is een cloud based business platform. Binnen dit platform heb je verschillende apps, deze applicaties bedienen elk een specifiek onderdeel van een bedrijf. In ons geval is de Finance en Operations applicatie verantwoordelijk voor het ERP deel. Daarnaast heb je ook onderdelen die CRM noodzaken bedienen of HR oplossing bieden en nog verschillende andere. Finance and Operations is ontwikkeld voor middelgrote tot grote ondernemingen. Zoals eerder vermeld heeft Microsoft dit onderdeel overgekocht van de Deense onderneming Damgaard A/S. Toen bestond de software nog onder de naam Axapta. De eerste versie van Axapta werd in 1998 vrijgegeven voor zowel de Deense als de Amerikaanse markt. In de daarop volgende jaren werden steeds uitgebreidere versies van de software in steeds meer landen vrijgegeven. Na de overname veranderde Microsoft de naam naar Dynamics AX. Tijdens de move naar de cloud besloot Microsoft ook Dynamics AX van een update te voorzien. Uiteindelijk in juli 2017 werd Dynamics 365 Finance & Operations officieel gelanceerd.

Dynamics 365 Finance & Operations omvat verschillende kern functionaliteiten gelijk in elk ERP systeem. Onderstaand kan je enkele van deze functionaliteiten terugvinden:

- sales en marketing
- supply chain management
- procurement en sourcing
- production control
- material master planning
- warehouse management
- budgeting
- ...

Bovenstaande voorbeelden zijn slechts een kleine selectie uit de verschillende functionaliteiten van Finance & Operations. Naast deze kern functionaliteiten zijn er ook tal van verschillende industrie specifieke toepassingen onderdeel van het softwarepakket. Dit kan gaan van functionaliteiten specifiek voor distributie tot functionaliteiten die voor het verkoop departement van een onderneming ontwikkeld zijn.

Niet alleen bevat Finance en Operations een grote hoeveelheid aan verschillende functionaliteiten. Daarnaast is het ook nog eens eenvoudig te integreren met verschillende andere Microsoft technologieën. Een veel gebruikt voorbeeld hierbij is de BI-integratie. Dankzij deze functionaliteit kan men snel en eenvoudig rapporten opstellen op basis van de aanwezige data.

Deze integratie en de integratie met verschillende andere functionaliteiten is mogelijk dankzij Microsofts Azure cloud technologie. Deze verschillende cloud oplossingen willen echter niet zeggen dat on-premise systemen niet langer worden geconfigureerd (Luszczak, 2019).

Wanneer we spreken over ERP-software spreken we ook over data. In Dynamics 365

wordt gebruik gemaakt van verschillende soorten data.

- Transactie data
- Master data
- Configuratie data

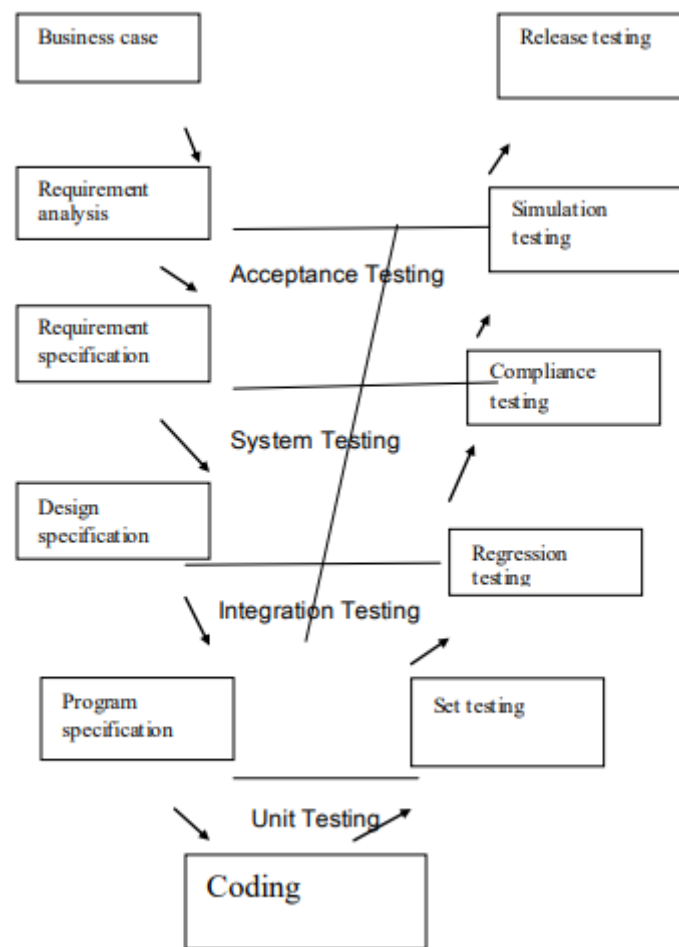
Zo wordt een grote hoeveelheid data bijgehouden die een transactie beschrijft in een Dynamics 365 systeem. Deze data wordt continu bijgewerkt. Dit komende door de activiteiten ondergaan wegens de verschillende bedrijfsprocessen. Bij elke activiteit genereert Dynamics transactie data. Deze data wordt vervolgens weggescheven en later gebruikt voor verdere analyse en rapportering. Een voorbeeld van deze data is het bijhouden van inventaris transacties. Telkens iets wijzigt bij de inventaris zullen verschillende zaken weggeschreven worden naar een daarvoor dienende tabel.

Naast transactie data is er ook data nodig die deze objecten beschrijft. Dit is de master data. De master data is data die over een bepaalde periode weinig tot geen wijzigingen zal ondervinden. Onder master data kunnen we bijvoorbeeld een product of een klant indelen. Master data beschrijft de objecten die van essentieel belang zijn voor het wellopen van de organisatie. Dit kan gaan over klanten, producten, leveranciers. Deze data wijzigt enkel wanneer een wijziging in het object plaatsvindt. Bijvoorbeeld wanneer een bepaalde leverancier andere gegevens heeft of wanneer een klant een adreswijziging doorvoert. Dit zijn slechts enkele voorbeelden van hoe Master Data over de tijd kan wijzigen deze wijzigingen vinden echter niet periodiek plaats. Master data is een zeer belangrijk onderdeel van een organisatie. Hierdoor spendeert men veel tijd en geld aan dit onderdeel. Met een prominente rol voor Master data management.

Ten slotte hebben we ook nog data die de configuratie beschrijft. Hieronder vallen de verschillende configuratie gegevens, of data gerelateerd aan het bijhouden van de huidige versie. Bovenvermelde soorten data zijn de 3 belangrijkste soorten data binnen het Dynamics 365 systeem en met uitbreiding de meeste ERP systemen.

Naast de 3 bovenvermelde soorten data binnen Dynamics zijn er nog andere soorten data die een belangrijke rol spelen. Zo is er ook verschillende setup data aanwezig. Deze data geeft aan hoe een bepaald bedrijfsproces moet worden uitgevoerd. Zo kan je hierin afleiden welke zaken aangesproken worden op welk moment of waarnaar gewezen moet worden. De setup is tevens ook, zoals de naam al aangeeft, de initiële opstart van de applicatie. Hier wordt gezorgd dat de applicatie volledig rond de noden van de klant is opgezet. Later kan in meer detail gekeken worden naar bepaalde zaken die een wijziging moeten ondergaan.

Om een applicatie van dergelijke omvang als Finance en Operations te configureren en te ontwikkelen is het belangrijk gebruik te maken van een goede teststrategie. Dit zodat er geen belangrijke zaken over het hoofd gezien worden tijdens de ontwikkeling ervan. Een veel gebruikte teststrategie is het V-model. In het V-model verdeelt men de aandacht evenredig tussen projectdefinitie en de verificatie en validatie van deze gedefinieerde project waarden en eisen. In dit gebalanceerde model vertrouwd men sterk op de verificatie van voorgaande stap alvorens men overgaat naar de daaropvolgende stap. Op deze manier



Figuur 2.1: Overzicht V-Model (Yadav, 2012)

is de tester sterk betrokken tijdens het ontwikkelproces. Wat een groot voordeel is bij de ontwikkeling van software (Balaji & Murugaiyan, 2012).

2.3 Het V-model

Het v-model geeft een overzicht van de verschillende soorten testen die doorlopen worden tijdens het ontwikkelproces, zoals je kan zien op afbeelding 2.1. Wanneer we beneden het model starten beginnen we met component testen of unit testen. In deze fase controleren we of elke onderdeel geïmplementeerd is en of deze implementatie ook voldoet aan de vooropgestelde vereisten. Tijdens het uitvoeren van deze unit testen ligt de focus op de individuele componenten en minder op het geheel op zich.

Vervolgens gaan we verschillende integratietesten uitvoeren. Integratie testen zijn zeer belangrijke testen voor goedwerkende software. Tijdens deze testen controleert men of de verschillende componenten ook daadwerkelijk als een geheel samenwerken.

Na het uitvoeren van de verschillende integratietesten gaat men over tot systeemtesten. Deze testen worden uitgevoerd wanneer het volledige systeem opgezet is. Men gaat controleren of de gevraagde mogelijkheden ook daadwerkelijk aanwezig zijn. Tijdens deze testfase werkt men nog steeds volgens een principe met grote focus op de ontwikkelaar. Naast het testen of de verschillende vereisten aanwezig zijn kijkt men ook of andere eigenschappen voldaan zijn. Zo zijn er verschillende testen die de performantie testen, of momenten van hoge systeem belasting voldaan zijn, men test of het systeem robuust is,

Wanneer alle systeemtesten voldaan zijn gaat men over tot de acceptatie testen. In deze testen wordt gecontroleerd of de verschillende gebruiker doelstelling aanwezig zijn en functioneren zoals gewenst. In een bepaald opzicht zijn dergelijke testen goed te vergelijken met systeemtesten. Ze onderwerpen ook het volledige systeem aan testen, enkel met dat verschil dat men niet langer kijkt of het geen gespecificeerd werd ook daadwerkelijk werkt maar men kijkt of de verwachtingen van de sponsor aanwezig zijn. In een volgende paragraaf vormen we een meer gedetailleerd beeld van gebruiker acceptatie testen.

Een laatste type testen dat een software project moet doorlopen zijn releasetesten. Wanneer alle vereisten voldaan zijn is er nog steeds een laatste zaak dat beantwoord moet worden, namelijk of de business value al dan niet zal stijgen. Men kijkt of het nieuwe of gewijzigde systeem binnen het huidige landschap zal functioneren. Men gaat na of bepaalde andere systemen hinder ondervinden, of het compatibel is met andere systemen, men kijkt of de performantie nog steeds voldaan is, Wanneer ook al deze testen voldaan zijn kan men succesvolle en goedwerkende software opleveren (Yadav, 2012).

Het v-model geeft een mooi overzicht van de samenhang tussen de verschillende testen onderling maar ook tussen de verschillende projectdefinitie fasen. Een belangrijk onderdeel van elk project is het uitvoeren van gebruiker acceptatie testen. Voor het verder onderzoek is het belangrijk te weten wat gebruiker acceptatie testen juist zijn en wat ermee bereikt wenst te worden.

2.4 Gebruiker Acceptatie testen

User Acceptance Testing of gebruiker acceptatie testen worden gebruikt voor het valideren van een systeem. Men kijkt of het systeem voldoet aan een voldoende hoge kwaliteit dat door de gebruiker en de sponsor of opdrachtgever geaccepteerd zal worden. De kwaliteit kan je vergelijken met de graad waaraan de applicatie aan de vooropgestelde doelstellingen voldoet. Een belangrijke kanttekening bij gebruiker acceptatie testen is dat men niet naar elk stukje code op zich kijkt. Het doel is niet perfecte code, het doel is de applicatie testen of deze voldoet aan de verwachtingen van de gebruiker en de sponsor (Cimperman, 2006).

Een vaak gebruikte methode voor het uitvoeren van gebruiker acceptatie testen is het geven van demos aan de sponsor (Meyer, 2014). Dit is echter geen ideale manier om deze testen uit te voeren wegens verschillende redenen. Een voorbeeld hiervan is agenda synchronisatie moeilijkheden. Dit is zeker een moeilijkheid wanneer het aantal aanwezigen,

deelnemende personen groter wordt. Een tweede pijnpunt is dat een dergelijke test vaak kan overgaan in een training sessie. Waardoor het niet langer een waardige test sessie is (Kupersmith, 2016). Een laatste pijnpunt is dat een sponsor de verantwoordelijkheid naar de ontwikkelaar schuift. Dit vaak door een gebrek aan middelen, hierdoor kan het doel achter de gebruiker acceptatie testen tenietgedaan worden. Wanneer testgevallen echter enkel worden uitgeschreven door het team van ontwikkelaars bestaat er een grote kans dat bepaalde scenarios over het hoofd worden gezien. Dit ook na intensief overleg met de sponsor (Itkonen e.a., 2013). Gebruiker acceptatietesten komen pas tot hun volle recht wanneer een goede samenwerking tussen de ontwikkelaar en de sponsor tot stand gebracht kan worden (Collins & de Lucena, 2012) (Otauy & Diaz, 2017).

2.5 Geautomatiseerd acceptatie testen

Geautomatiseerd acceptatie testen is een veelbelovend onderdeel van software ontwikkeling (Haugset & Hanssen, 2008). Bij geautomatiseerde acceptatie testen zorgt men ervoor dat een bepaald scenario kan worden opgenomen. Deze opname kan gebeuren door de test manueel uit te voeren. Wanneer de opname gemaakt is kan deze steeds opnieuw worden afgespeeld. Het systeem overloopt steeds dezelfde stappen en controleert of deze voldaan worden. Wanneer dit het geval is zal een succes naar boven komen, indien dit niet succesvol verloopt kan men een error tonen. Echter komen met geautomatiseerd testen verschillende andere aspecten naar boven.

Geautomatiseerde testen kunnen eenvoudig meermaals worden uitgevoerd aan een relatief lage kost. Dit komende door het feit dat een test eenvoudig kan lopen zonder dat een ontwikkelaar of tester deze testen manueel moet uitvoeren en dus werkuren hieraan besteed (Hanssen & Haugset, 2009). Wat echter nog niet duidelijk is of de kost voor het aanmaken en opzetten van de geautomatiseerde testen effectief lager ligt vergeleken met de kost van klassiek testen.

Wanneer we processen automatiseren is dit met een vermindering in workload als doel. Ditzelfde principe is geldig voor het automatiseren van acceptatie testen. Volgens de studie van Balfe e.a. (2015) is er een significant verschil in workload vermindering na automatisatie van verschillende processen. Dit zowel in normale omstandigheden als in omstandigheden waar het normale procesverloop verstoord is. Deze significante vermindering ging niet ten nadele aan kwaliteit en performantie. Daarnaast geeft dit aan dat met automatisatie er een grote winst geboekt kan worden op het vlak van workload vermindering wanneer gewerkt wordt in een volledig geautomatiseerde setting als ook in een deels geautomatiseerde setting. Workload kan worden opgesplitst in de fysieke workload als ook de mentale workload. Een automatisatie kan ervoor zorgen dat de fysieke workload die samen met een proces komt sterk verminderd. Daartegenover kan dit wel voor een verhoogde mentale workload zorgen. Dit dankzij een grotere vraag naar het vooruitdenken tijdens de processen. Men kijkt naar de acties van de automatisatie en de manier waarop er mee moet worden omgegaan (Balfe e.a., 2012).

De bovenstaande bevindingen zijn gebaseerd op het automatiseren van verschillende pro-

cessen. Dit is geen garantie dat de workload van een geautomatiseerde acceptatie test zich gelijkaardig zal gedragen. Voor zowel de fysieke als mentale workload is nog geen voldoende onderzoek beschikbaar. Beiden zaken zijn interessant om op te nemen in verder onderzoek.

Een ander doel van automatisatie is het verhogen van de frequentie waarmee een proces uitgevoerd wordt. Een automatisatie kan steeds worden uitgevoerd volgens een vooropgesteld schema of via een handmatige executie. De frequentie waarmee een automatisatie zal worden uitgevoerd is afhankelijk van klantwensen en frequentie van update releases die een invloed kunnen hebben op de werking van de geautomatiseerde processen. Dit is ook het geval bij geautomatiseerde acceptatie testen. Zo zal bij elke wijziging van zowel het te testen proces als ook bij elke Microsoft One Version¹ update de automatisatie worden uitgevoerd. Microsoft One Version laat gebruikers toe steeds de laatste versie van Finance & Operations te gebruiken op een consistente, voorspelbare en geïntegreerde manier.

2.6 Azure Devops

Het is natuurlijk ook belangrijk dat we de verschillende testen op een gemeenschappelijke plaats kunnen uitvoeren en de resultaten ervan bekijken. Dit is ook zeker en vast van toepassing voor gebruiker acceptatie testen. Azure DevOps is een software als een service (SaaS) platform van Microsoft. Het geeft je de mogelijkheid om een volledige end-to-end toolset te gebruiken tijdens en na het ontwikkelen van software. DevOps is een samenstelling van ontwikkelen (dev) en bedrijfsactiviteiten (ops) het staat voor de samenstelling van mensen en technologie om op die manier extra voordeel en omzet voor klanten te genereren (Microsoft, 2021).

Een groot deel van bedrijven maken gebruik van processen tijdens de ontwikkeling van hun producten of diensten. Deze processen hoeven daarvoor niet neergeschreven te zijn. Er is een eigen aanpak tot het ontwikkelen van software, ze maken gebruik van een eigen procesopbouw. Langs de andere kant van het spectrum heb je bedrijven die de verschillende processen zeer intensief uitdenken en beschrijven. Zo formuleren deze naam voorschriften, bestandslocaties en meer van dergelijke regels. Echter hebben ze het vaak moeilijk deze processen in de praktijk door de project deelnemers te laten gebruiken. Dit zorgt ervoor dat projectmedewerkers eigen regels hanteren en dat er geen uniformiteit meer aanwezig is binnen de organisatie. Dit zorgt niet enkel voor verwarring tussen de verschillende partijen maar ook voor een groot verlies aan efficiëntie. Voor dergelijke problemen kan Azure Devops een oplossing bieden. In Azure Devops kan een bepaald proces worden geïmplementeerd in de vorm van een template. Deze template kan vervolgens verplicht worden tijdens de aanmaak van een nieuw project. Op deze manier worden processen steeds correct gevolgd, er is minder verwarring en de efficiëntie verhoogd.

Een andere functionaliteit van Azure Devops dat intensief gebruikt wordt tijdens projec-

¹<https://docs.microsoft.com/en-us/business-applications-release-notes/april19/dynamics365-finance-operations/one-version-experience>

ten is de mogelijkheid tot het aanmaken van verschillende rapporten. Deze rapporten zijn beschikbaar voor elk nieuw project. Ze bieden een diepere inzicht in hoe een project ervoor staat en wat de huidige en voorgaande status van het project is. Dit kan gaan van het aantal openstaande bugs tot het aantal vervulde taken. Naast deze rapportering tool zorgen Work items ervoor dat projecten effectiever verlopen. Een work item stelt je in staat gelijk welk onderdeel van een project te volgen. Dit kan gaan van een bug tot een verplichting. Op deze manier kan je verschillende zaken visualiseren doorheen de projectcyclus. Dit zijn maar enkele voordelen van de verschillende onderdelen van een template. Deze templates zijn een flexibel geheel wat ervoor zorgt dat ze eenvoudig en snel aan de noden van de klant kunnen worden aangepast. Dergelijke templates worden aangeboden door zowel Microsoft als externe partijen of deze kunnen zelf worden samengesteld (Rossberg, 2019).

Een ander onderdeel van Azure Devops is Azure Test Plans. In deze tool kan je testplannen voor verschillende testscenarios die je project moet doorlopen aanmaken. Vermits gebruiker acceptatie testen nog vaak door werknemers worden uitgevoerd is het belangrijk een Test Plan te volgen. Op deze manier verminder je de kans op fouten. Wanneer we dergelijke testen automatiseren is het belangrijk tijdens het opzetten van de automatisatie een strikt plan te volgen.

2.7 Regression Suite Automation Tool

RSAT of Regression Suite Automation Tool is een tool van Microsoft voor het automatiseren van verschillende acceptatie testen. Deze testen controleren of de geïmplementeerde processen in Microsoft Dynamics 365 Finance & Operations voldoen aan de verwachtingen van de sponsor. RSAT is veelal gebruikt voor het testen van custom code, met verschillende regressies gebruik makend van eenzelfde basis code. Dit wanneer een incrementele ontwikkel methodologie gehandhaafd wordt (Donepudi, 2020).

Wanneer we vervolgens kijken naar het automatiseren van de acceptatie testen gebruik makende van RSAT zien we dat dit op een zeer eenvoudige manier gebeurt. De ontwikkelaar neemt een scenario op aan de hand van de recorder. Deze opgenomen test case kan vervolgens herhaaldelijk worden uitgevoerd door RSAT.

Tijdens de literatuurstudie zijn een aantal zaken naar boven gekomen die verder onderzoek vereisen. Zo is er momenteel weinig tot geen kennis neergeschreven betreffende de invloed op de workload na het automatiseren van acceptatie testen. Hiervoor kijken we naar de workload die zal ondervonden worden tijdens het automatiseren van testen en uitvoeren van de geautomatiseerde testen in RSAT.

RSAT is nog een relatief nieuwe tool, hierdoor is literatuur beperkt en gedetailleerde kennis niet overal beschikbaar. Tijdens verder onderzoek zullen verschillende zaken toegelicht worden die niet neergepend zijn in bestaande literatuur.

3. Methodologie

In dit onderdeel wordt de aanpak van het onderzoek besproken. Als eerste wordt besproken hoe de Regression Suite Automation Tool geconfigureerd wordt. Vervolgens wordt het aanmaken van de verschillende testen besproken. Na het aanmaken van de testen wordt de uitvoering ervan besproken om ten slotte de bekomen resultaten te bespreken. Het doel is om aan de hand van de gebruikte methodologie een sluitend antwoord te bekomen op de onderzoeksvraag en de verschillende deelvragen.

3.1 Configureren van RSAT

In dit deel van het onderzoek wordt nagegaan welke zaken nodig zijn voor de configuratie van de Regression Suite Automation Tool en met uitbreiding het uitvoeren van geautomatiseerde gebruiker acceptatie testen. Meer specifiek worden alle verschillende componenten overlopen. Dit gaande van de verschillende applicaties, de omgeving maar ook welke elementen geïnstalleerd moeten zijn als ook welke gegevens gekend moeten zijn alvorens een test kan aangemaakt worden.

Dit is een belangrijke onderdeel voor het opzetten en uitvoeren van de verschillende geautomatiseerde testen. Wanneer niet correcte gegevens gebruikt worden voor het opzetten van de Regression Suite Automation Tool zal dit ervoor zorgen dat een test niet kan worden uitgevoerd.

3.2 Aanmaken van testen

In deze fase van het onderzoek wordt nagegaan hoe een effectieve test moet worden aangemaakt. Er wordt gekeken naar de setup, de kennis nodig betreffende het systeem als ook de technische kennis om een test op te nemen. Het opnemen van het test scenario is slechts een onderdeel bij het aanmaken van een test. Zo zal er ook onderzocht worden welke zaken zeker voldaan moeten worden. Dit betreffende de naamgeving, het inladen van een test, het bewaren van een test,

De zaken die in dit deel van het onderzoek worden onderzocht zijn cruciaal voor het slagen van de test. Wanneer een test op een foutieve of onvolledige manier aangemaakt wordt zal dit leiden tot het falen van de test. De verschillende elementen worden door het intensief gebruiken van de tool onderzocht. Zo zullen verschillende scenarios worden aangemaakt. Onder deze scenarios zullen ook met opzet foutieve zaken worden uitgevoerd. Op deze manier achterhalen we welke fout in het scenario welke foutboodschap weergeeft bij het uitvoeren ervan. Het is een proces van trial en error, dit om steeds de resultaten van de gekozen opbouw te kunnen bekijken en vervolgens te vergelijken.

Het opnemen van verschillende scenarios is slechts een beperkt deel van het proces om verschillende testen aan te maken. Het is ook belangrijk dat een test op de correcte manier aan Azure DevOps zal worden toegekend. Zo zal worden onderzocht wat de exacte naamgeving van een testgeval moet zijn. Hoe testen moeten worden gerangschikt en op basis van wat een test wordt geïdentificeerd. Naast het toevoegen van testen in Azure DevOps zal ook onderzocht worden hoe een test aan de Regression Suite Automation Tool moet worden toegevoegd. Hier kijken we naar de opbouw en hoe deze opbouw overeenstemt met de opbouw van een Test Plan en meer specifiek de verschillende Test Suites. Er wordt onderzocht of deze opbouw gelijk is aan deze in Azure DevOps waaruit de testen worden geïmporteerd.

Ten slotte is het linken van verschillende testgevallen een belangrijk onderdeel van geautomatiseerd gebruiker acceptatie testen. In dit deel van het onderzoek wordt gekeken naar hoe de ene test aan een andere test gelinkt wordt. Vervolgens wordt ook gekeken of het al dan niet belangrijk is deze test achtereenvolgens uit te voeren. Dit zodat de samenhang van deze gelinkte testen onderzocht kan worden. Naast het linken van testen wordt ook onderzocht hoe een unieke waarde wordt voorgesteld. Zo kan het zijn dat een product met productnummer wordt aangemaakt, telkens een geautomatiseerde test uitgevoerd wordt moet dit nummer verschillend zijn. Ten slotte is het ook belangrijk te onderzoeken hoe bepaalde waarden gevalideerd worden. Kan je na het wijzigen de validatie automatisch laten uitvoeren of moet dit ook handmatig gebeuren?

Deze fase van het onderzoek, waar naar de aanmaak van de verschillende testen wordt gekeken, is tevens ook de belangrijkste fase van het onderzoek. Hier is het belangrijk om weten hoe een goede test case eruitziet en waaraan voldaan moet worden. Daarnaast is het ook belangrijk te onderzoeken welke zaken zeker niet mogen worden uitgevoerd tijdens de aanmaak van een test case. Wanneer een testgeval niet ten goede opgebouwd is kan je het resultaat ook niet met volle zekerheid voor waar aannemen.

3.3 Uitvoeren en resultaten van testen

In deze fase van het onderzoek wordt een diepere kijk genomen naar het uitvoeren van de verschillende testen. Het uitvoeren van een test op zich vraagt niet zo veel onderzoek werk. Het is eerder belangrijk te onderzoeken hoe een positief testresultaat eruitziet en zeker en vast hoe een negatief testresultaat er zal uitzien.

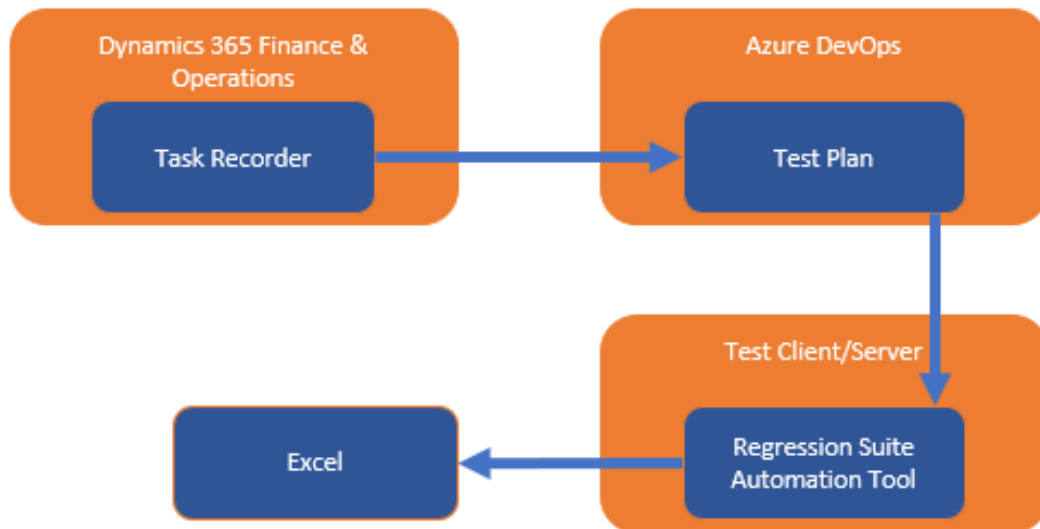
Bij een positieve test zal gekeken worden naar de verschillende elementen die aanduiden dat de test positief is. Zijn er elementen die belangrijk zijn voor andere testen, zijn er elementen die een invloed geven op andere testen, Dit zal eerder een beperkt onderdeel van het onderzoek zijn.

Het onderzoeken van een negatief testresultaat is een belangrijk onderdeel. Door het onderzoeken van een gefaalde test zal trachten achterhaald te worden wat de oorzaak voor het falen is, de stap waarin de test gefaald is, Daarnaast is het ook belangrijk te onderzoeken of deze elementen voldoende informatie verschaffen om te achterhalen of de reden tot het falen van de test zich in de test zelf bevindt of eerder in de applicatie. Zo is er een belangrijk verschil tussen een testgeval die faalt omdat de test zelf foutief is en een testgeval die faalt omdat de applicatie niet langer voldoet aan de test case.

4. Regression Suite Automation Tool

"The Regression suite automation tool (RSAT) significantly reduces the time and cost of user acceptance testing (UAT) of Finance and Operations apps." — Microsoft

In onderstaande paragraaf nemen we een kijk naar de bekomen resultaten betreffende de Regression Suite Automation Tool. Als eerste wordt de configuratie van de tool besproken, wat is essentieel nodig voor een correcte werking van RSAT. Vervolgens bespreken we het aanmaken van de effectieve testen. Waar moet zeker aandacht besteed worden en waar niet, voor het welslagen van een testgeval. Na het opstellen van de testen moeten deze ook worden uitgevoerd. In dit onderdeel kijken we hoe je een test uitvoert. Ten slotte wordt de uitvoer van de testen besproken. Hoe zie je of een test succesvol is en waar kan je foutmeldingen terugvinden.



Figuur 4.1: Grafische voorstelling relaties tussen verschillende applicaties.

4.1 Configuratie

4.1.1 Nodige applicaties


Het configureren van RSAT is eenvoudig en kan snel gebeuren. Dit is ook essentieel gezien RSAT ontwikkeld is met als doel gebruikt te worden door de opdrachtgever en gebruiker van de Finance & Operations applicatie. Onderstaand zijn de verschillende onderdelen die nodig zijn voor het geautomatiseerd gebruiker acceptatie testen opgelijst:

- Excel
- Regression Suite Automation Tool
- Azure DevOps
- Dynamics 365 Finance & Operations

Onderling verband tussen de verschillende applicaties kan teruggevonden worden op afbeelding 4.1.

Sinds Microsofts stap naar de Cloud (Luszczak, 2019) zijn verschillende van bovenstaande applicaties toegankelijk via de browser. Dit wil niet zeggen dat bij het geautomatiseerd gebruiker acceptatie testen geen desktopapplicatie gebruikt wordt. Het is noodzakelijk Excel en de Regression Suite Automation Tool te downloaden als desktopapplicaties. Daarnaast is het ook belangrijk dat beide applicaties op dezelfde fysieke machine geïnstalleerd staan. De reden hiervoor is dat de Regression Suite Automation Tool intensief gebruik maakt van Excel voor het specificeren van verschillende testinstellingen als ook voor het definiëren van test specifieke parameters.

Uitsluitend gebruik makend van Excel en de Regression Suite Automation Tool kan je niet



Figuur 4.2: Azure Devops Configuration.

succesvol een test opstellen en uitvoeren. Ook Azure DevOps is een essentieel deel van de configuratie. Hierin zullen de verschillende Test Plannen en Test Suites weergegeven worden. Dit is ook de applicatie waar een task opname toegekend wordt aan een test. In sectie 4.2.1 kan je meer info terugvinden betreffende de task recorder en het toekennen van deze opname aan een test binnen een Test Suite.

Bovenvermelde applicaties worden gebruikt voor het correct opslaan van de verschillende testen, het instellen van de testen en het uitvoeren ervan. Natuurlijk hebben we ook een omgeving nodig die we testen. Namelijk de Microsoft Dynamics 365 Finance & Operations applicatie op zich. Op de Finance & Operations applicatie gaan we de verschillende te doorlopen stappen opnemen aan de hand van de task recorder, meer informatie betreffende de task recorder kan je terugvinden in sectie 4.2.1.

Bovenvermelde applicaties worden als een geïntegreerd geheel gebruikt voor het opstellen, bewaren en uitvoeren van geautomatiseerde gebruiker acceptatie testen.

4.1.2 Configureren van RSAT

Het configureren van RSAT is opgesplitst in 3 grotere onderdelen:

- Azure DevOps
- Finance and Operations Test Environment
- Run Settings

Elk van deze onderdelen hebben een subset aan gegevens nodig. Bij de sectie voor de Azure DevOps configuratie is het belangrijk de juiste connectie te leggen tussen RSAT en Azure DevOps. Deze configuratie is van essentieel belang voor het correct inladen van de verschillende Test Plans en Test Suites. Deze configuratie is van essentieel belang omdat de verschillende testgevallen opgeslagen zijn in Azure DevOps. Wanneer een gebruiker de testgevallen binnenhaalt in RSAT zijn deze afkomstig van Azure DevOps. Op onderstaande afbeelding kan je de verschillende te configureren onderdelen terugzien.

We overlopen elk van de te configureren onderdelen zoals je kan zien in afbeelding 4.2 en lichten hun werking kort toe. De gebruikte waarden in dit onderzoek zijn verborgen.

Figuur 4.3: Finance and Operations Test Environment.

- Azure DevOps Url: deze url verwijst naar de correcte Azure DevOps omgeving die gebruikt zal worden voor het toevoegen van de testen.
- Access Token: dit token geeft de Regression Suite Automation Tool toegang tot de Azure DevOps omgeving. Er moet steeds een individueel token worden aangemaakt. Hoe dit token wordt aangemaakt bespreken we niet in dit onderzoek.
- Project Name: deze waarde geeft de naam van het project aan. Alle aangemaakte testgevallen zullen in dit project worden opgeslagen.
- Test Plan: deze waarde geeft het Test Plan aan waar de verschillende testen worden opgeslagen. Alle aangemaakte testgevallen zullen in dit Test Plan worden opgeslagen.

Het tweede onderdeel dat geconfigureerd moet worden is de Finance & Operations Test Environment. Deze configuratie omschrijft waar de geautomatiseerde testen moeten worden uitgevoerd. Hier wordt de link naar de juiste Finance & Operations omgeving gelegd. Op onderstaande afbeelding kan je de verschillende te configureren onderdelen terugzien.

We overlopen elk van de te configureren onderdelen zoals weergegeven in afbeelding 4.3 en lichten hun werking toe. De gebruikte waarden in dit onderzoek zijn verborgen.

- Hostname: de naam van het systeem waarop de verschillende testen uitgevoerd worden. Het is belangrijk dat je `http://` of `https://` niet toevoegt. Een voorbeeld van een hostname is: `myhost.cloudax.dynamics.com` ¹.
- SOAP Hostname: Simple Object Access Protocol of SOAP is een protocol gebruikt voor xml- gebaseerde communicatie. Dit is essentieel gezien de verschillende testen gebaseerd zijn op een xml-bestand ².
- Admin user name: het mailadres van een administrator binnen de test omgeving. Dit mailadres moet toegekend zijn aan de System Administrator rol op de Finance & Operations testomgeving waarmee connectie gemaakt wordt.
- Thumbprint: is de vingerafdruk of de specifieke code die gebruikt wordt als authenticatie certificaat voor het connecteren met de test omgeving. Je kan eenvoudig een

¹<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/perf-test/rsat/rsat-install-configure>

²<https://d365solution.blogspot.com/2019/02/soap-and-rest-in-d365-explanation-web.html>



Figuur 4.4: Run Settings.

nieuwe waarde aanmaken door op New te klikken naast het vak.

- Company name: deze waarde verwijst naar de bedrijfsnaam die standaard gebruikt zal worden tijdens het testen. Er kan tussen company gewisseld worden indien de setup van beide entiteiten volledig gelijklopend is.

Het laatste onderdeel van de configuratie is het toekennen van de gewenste bestandslocatie en het selecteren van de te gebruiken browser. De gebruikte waarden in dit onderzoek zijn verborgen.

We overlopen elk van de te configureren onderdelen zoals weergegeven in afbeelding 4.4 en lichten hun werking toe.

- Working directory: bestandslocatie waar verschillende configuratie en parameterbestanden worden bewaard.
- Default browser: standaard browser die gebruikt zal worden tijdens het uitvoeren van de verschillende geautomatiseerde testen.

Het configureren van bovenvermelde elementen is een belangrijk onderdeel wanneer het gebruik van de Regression Suite Automation Tool opgezet wordt.

4.2 Aanmaken van testen

Na het configureren van de omgeving waar de testen worden opgenomen en uitgevoerd gaan we over tot het opstellen van de effectieve testgevallen. In deze sectie kijken we hoe een test opgenomen wordt, aan een test plan toegevoegd wordt en vervolgens in RSAT klaargezet wordt voor het uitvoeren ervan.

Het opstellen van een test verloopt steeds volgens eenzelfde stappenplan. Het eerste onderdeel van dit stappenplan is het uitwerken van een scenario die je wenst te testen. Een voorbeeld hiervan is het aanmaken van een nieuwe verdor in Finance & Operations. Vervolgens neem je het scenario op aan de hand van de Task Recorder. Na opname wordt de test case toegevoegd aan Azure DevOps alvorens deze test wordt ingeladen in RSAT. In secties 4.2.1, 4.2.2 en 4.2.3 gaan we dieper in op elk van deze onderdelen.

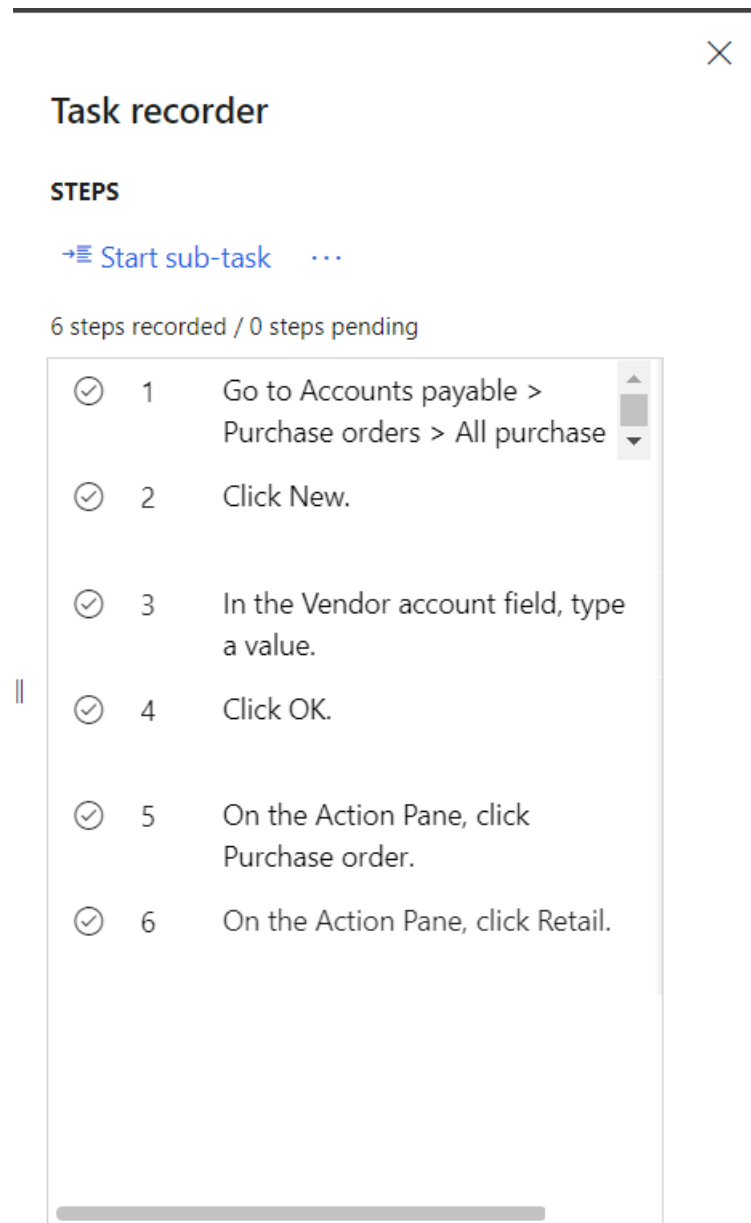
4.2.1 Scenario opnemen

Een gebruiker acceptatie test is in essentie het doorlopen van bepaalde zaken van de applicatie en vervolgens evalueren of dit volgens de verwachtingen verlopen is. Wanneer we geautomatiseerd gaan testen aan de hand van de Regression Suite Automation Tool volgen we hetzelfde principe. Met als verschil dat we de stappen opnemen aan de hand van de Task Recorder en vervolgens geautomatiseerd laten afspelen.

De Task Recorder laat ons toe een scenario op te nemen. Alvorens we een bepaald scenario doorlopen starten we een nieuwe opname. Deze opname zal alle stappen opslaan die we uitvoeren tijdens het volgen van het opgestelde scenario. Wanneer we een scenario opnemen is het belangrijk met bepaalde aspecten rekening te houden. Het is van essentieel belang dat elke opname start op het beginscherm van de Finance & Operations applicatie. Indien dit niet het geval is zal de test niet succesvol uitgevoerd kunnen worden. De reden hiervoor lichten we verder toe in sectie 4.3.2. Een ander belangrijk aspect om rekening mee te houden is het specificeren van zoekopdrachten. We wensen de mogelijkheid op foutieve waarden zoveel mogelijk te beperken. Zoekwaarden of filterwaarden zo gericht mogelijk houden is dus een zeer belangrijk aspect tijdens het opstellen van een goede test. De reden hiertoe is verder toegelicht in sectie 4.3.2. Een laatste aspect waar rekening mee moet genomen worden is het duidelijk specificeren van de aangeklikte knop. Wanneer een bepaald venster geopend wordt kunnen vooraf uitgevoerde taken ervoor zorgen dat dit venster, tijdens het opnemen van de test, verschillend is vergeleken met het venster dat bij de geautomatiseerde test geopend wordt. Dit verschil zal ervoor zorgen dat de test hoogstwaarschijnlijk niet succesvol zal verlopen. Dit terwijl de applicatie wel correct geïmplementeerd is.

Tijdens het uitvoeren van de verschillende stappen zal de task recorder steeds de uitgevoerde stappen weergeven, zoals te zien is in afbeelding 4.5. Dit zorgt voor een duidelijke weergave van hetgeen door de task recorder zal worden opgeslagen. Het is op basis van deze stappen dat de Regression Suite Automation Tool vervolgens zal weten wat hij juist moet uitvoeren. Zoals eerder vermeld zal de task recorder niet onthouden op welk scherm je aan het werk bent. Enkel de zaken die de gebruiker expliciet uitvoert worden onthouden.

Tijdens het opnemen van een scenario kunnen bepaalde waarden worden gemarkeerd. Zo kan je een product ID kopiëren. Dit is belangrijk wanneer je verschillende testen aan hetzelfde product wenst te linken. Dit principe noemt men ook wel het chainen van testgevallen. Dit is een belangrijk onderdeel van het uitvoeren van gebruiker acceptatie testen. Door het aan elkaar koppelen van testen kan je een bepaald product, een bepaalde vendor of een bepaalde purchase volgen. Zo kan je bijvoorbeeld in een eerste test een nieuwe vendor aanmaken, om vervolgens een purchase order aan te maken voor deze vendor. Dit zijn al 2 verschillende testgevallen, dit vermits ze een ander doel testen. Maar gezien in beide gevallen dezelfde vendor gebruikt wordt kunnen beide testen aan elkaar gekoppeld worden. Als je vervolgens in een laatste testgeval deze purchase order wenst te bevestigen is het voordelig van dit ook mee in de ketting van testen te betrekken. Dankzij deze variabelen kan de uitvoering en het onderhoud van de verschillende testgevallen sterk verbeterd worden.



Figuur 4.5: Weergave uitgevoerde stappen.

Naast het kopiëren van een bepaalde waarde kan ook worden gemarkeerd indien een waarde gevalideerd moet worden. Het is belangrijk dat deze zaken correct worden toegekend zodat een test later opnieuw kan worden uitgevoerd.

Een opname kan je na het doorlopen van het scenario stoppen en vervolgens opslaan als Developer Recording. Dit zal een xml-bestand naar de gekozen bestandslocatie downloaden. Het is vervolgens op basis van dit xml-bestand dat Azure DevOps de test case zal opmaken. Meer info betreffende het aanmaken van een test case kan je terugvinden in sectie 4.2.2.

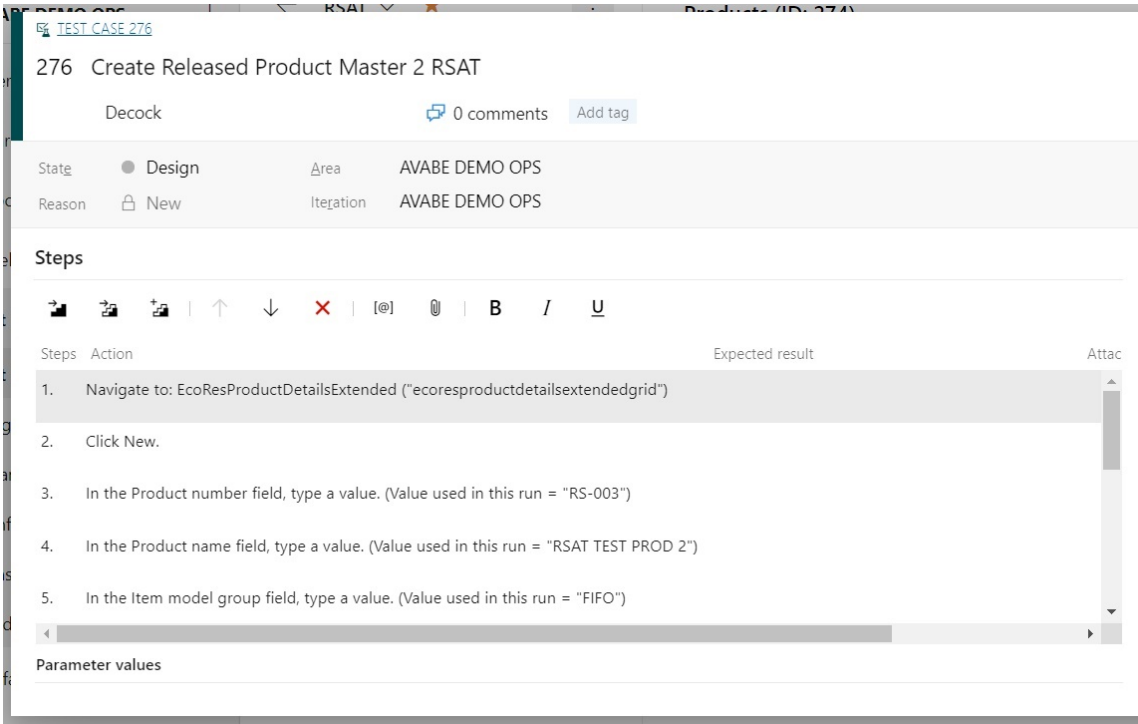
Wanneer een fout wordt gemaakt tijdens het opnemen van een test kan je deze stap schrappen. Uit het gebruik van de tool is gebleken dat dit niet de aangeraden manier van werken is. Vaak zal RSAT problemen geven wanneer een test met doorschrapte stappen wordt uitgevoerd. De reden voor dit probleem kan liggen bij het feit dat een bepaald scherm geopend wordt waarop verdere stappen worden uitgevoerd. Wanneer de stap waarin het nieuwe scherm geopend wordt, geschrapt is zullen de verdere stappen niet langer succesvol verlopen. Ook kan een bepaalde stap een actie op een waarde uitvoeren. Wanneer deze stap vervolgens geschrapt wordt zal de waarde gebruikt tijdens de opname niet langer overeenkomen met de waarde gebruikt bij de automatisatie.

4.2.2 Test toevoegen aan Azure DevOps

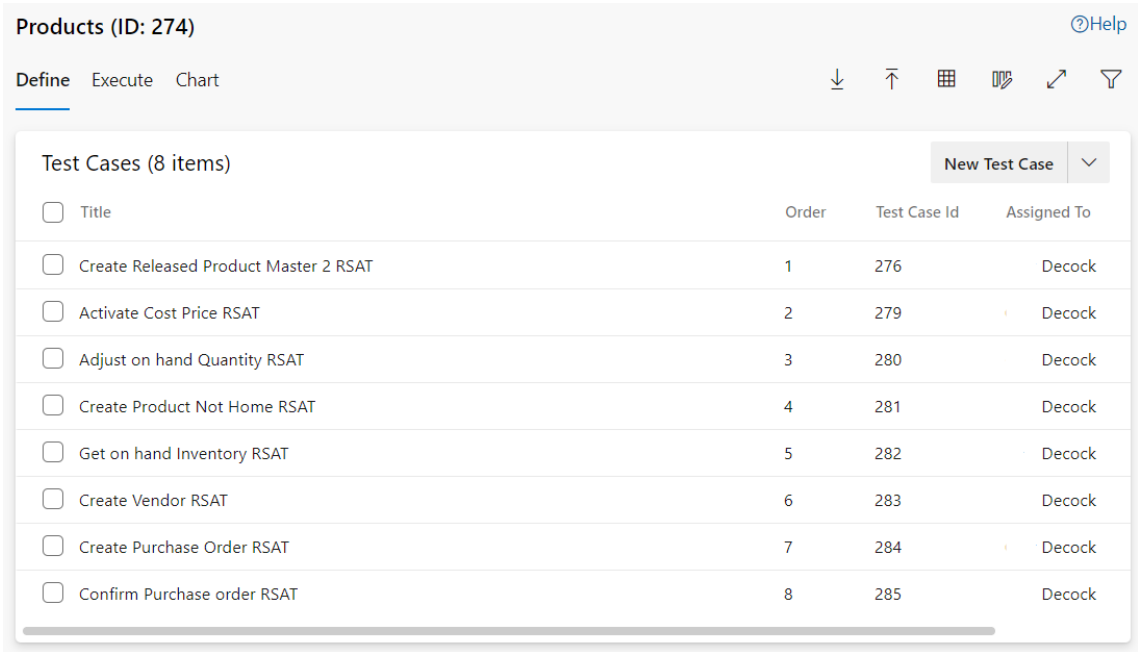
De opname van het uitgevoerde scenario toevoegen aan een Azure DevOps Test plan is de volgende stap in het stappenplan die elke geautomatiseerde test moet doorlopen. Hier verdelen we de verschillende testen op naargelang het scenario die ze beschrijven. Binnen een Test plan maken we verschillende Test Suites aan. Dit kan je zien als een sub map van het Test plan. Binnen een Test Suite kan vervolgens een test worden toegevoegd.

Om een nieuwe test toe te voegen dien je in de Define sectie van het scherm op 'New Test Case' te klikken. Het is belangrijk dat de naam van de test case exact overeenkomt met de naam van de eerder gemaakte opname. Wanneer de opname ingeladen is zie je meteen ook de verschillende uitgevoerde stappen. Deze stappen komen overeen met de eerdere stappen weergegeven in sectie 4.2.1. Bovenstaande zaken kan je terugzien op afbeelding 4.6.

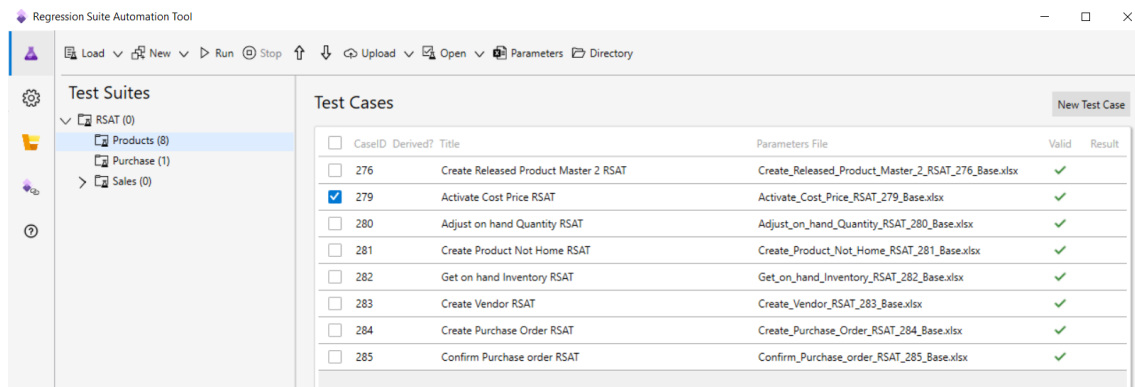
Na het toevoegen van de verschillende individuele testen kan je een overzicht terugzien van de verschillende testgevallen in de Test Suite. Elke test krijgt een uniek Test Case Id en een volgorde. Deze volgorde is belangrijk voor het uitvoeren van de verschillende testen. De reden hiervoor is dat testen vaak afhankelijk zijn van elkaar. Wanneer bijvoorbeeld een test scenario het aanmaken van een product beschrijft en een volgend scenario het activeren van dit product is het belangrijk dat de Regression Suite Automation Tool weet over welk product dit gaat. Echter wanneer deze volgorde niet duidelijk gedefinieerd staat zal de test falen. Het is mogelijk testen eenvoudig volgens een andere volgorde in te delen. Ten slotte kan een test ook worden toegekend aan een persoon. Bij creatie is de test steeds toegekend aan de persoon die de test in Azure DevOps aangemaakt heeft. Bovenstaande zaken kan je terugzien op afbeelding 4.7.



Figuur 4.6: Toevoegen test in Azure DevOps.



Figuur 4.7: Overzicht testen in Azure DevOps Test Suite.



Figuur 4.8: Overzicht testen in de Regression Suite Automation Tool.

Dankzij de grote hoeveelheid aan functionaliteit binnen Azure DevOps kunnen testgeval-
len verder worden opgevolgd. Deze verdere opvolging zal niet verder worden besproken
in deze scriptie.

4.2.3 Test toevoegen aan RSAT en testparameters wijzigen

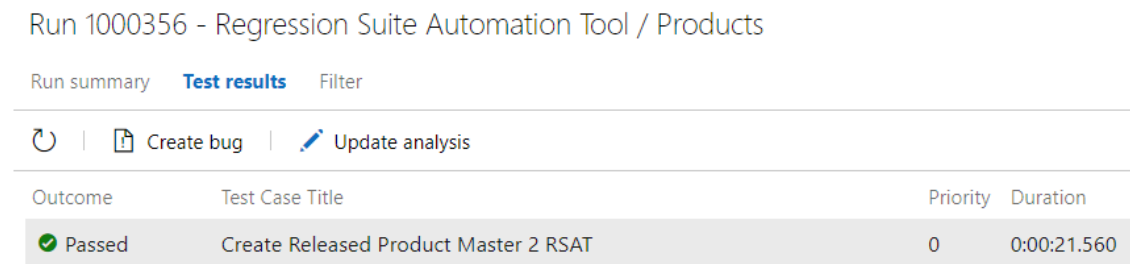
Een test toevoegen aan de Regression Suite Automation Tool is eenvoudig en kan snel
voltooid worden. De reden hiervoor is dat de verschillende testen reeds juist gedeclareerd
en onderverdeeld zijn in Azure DevOps. In RSAT kan een volledig Test plan worden inge-
laden. Wanneer men deze taak uitvoert zullen de verschillende Test suites en bijhorende
testen worden ingeladen. De volgende stap is het genereren van de executie bestanden als
ook de parameter bestanden.

Elke test heeft zijn eigen parameter bestand. Zoals je kan zien op afbeelding 4.8. Dit is
een Excel bestand waar de verschillende stappen die een test doorloopt beschreven staan
als ook de verschillende waarden die gebruikt worden tijdens de test. Deze parameters
kunnen gewijzigd worden om op deze manier verschillende elementen te testen.

Telkens je nieuwe testen wenst in te laden is het enkel mogelijk het volledige Test Plan
in te laden of de specifieke Test Suite. Dit zal resulteren dat de executie bestanden en
parameter bestanden van elke test in de Test Suite opnieuw moeten worden gegenereerd.
Hierdoor zullen ook de wijzigingen aangebracht aan de parameters niet meer aanwezig
zijn. Dit verhoogt de workload telkens een extra test aan de suite wordt toegevoegd. Het
is dus belangrijk alle testen reeds in Azure DevOps aan te maken alvorens deze in RSAT
op te halen. Het is echter wel mogelijk een individuele test te herladen. Wanneer een
bepaalde test niet succesvol geconfigureerd is kan je deze opnieuw toevoegen zonder het
herladen van alle andere testen.



Figuur 4.9: Succesvolle test weergegeven in de Regression Suite Automation Tool.



Figuur 4.10: Succesvolle test weergegeven in Azure DevOps.

4.3 Uitvoeren van testen

Het uitvoeren van een test of van verschillende testen tegelijk is zeer eenvoudig. Je selecteert de test of testen die je wenst uit te voeren en klikt run. De testen worden uitgevoerd volgens de gedeclareerde volgorde. Wanneer een test wordt uitgevoerd neemt de Regression Suite Automation Tool de controle van het scherm over. Een console scherm wordt opgestart en de voorkeur browser opent de Finance & Operations applicatie. Vervolgens worden de verschillende opgestelde stappen doornomen. Wanneer de test ten einde is worden alle schermen gesloten en geeft RSAT weer of de test al dan niet succesvol was.

4.3.1 Geslaagde test

Een succesvolle test bekom je wanneer de Regression Suite Automation alle stappen van de test correct kan uitvoeren. RSAT geeft een succesvolle test aan door het plaatsen van een groen vinkje aan het einde van de lijn waar de test gedeclareerd is, zoals je kan zien in afbeelding 4.9. Daarnaast kan je ook in Azure DevOps het testresultaat met extra informatie bekijken. In DevOps krijg je extra informatie te zien over de uitgevoerde test. Zo kan je zien wat de test uitkomst is als ook de uitvoertijd daarnaast is het ook mogelijke een grafische voorstelling met de verhouding succesvolle, niet-succesvolle testen op te vragen. In afbeelding 4.10 kan je zien dat de uitvoertijd voor de uitgevoerde test 21 seconden bedraagt. Er is geen prioriteit aan deze test toegekend. Ook dit is een element die je kan toepassen maar niet in deze scriptie behandeld wordt.

4.3.2 Gefaalde test

Het is steeds mogelijk dat een testgeval geen positief resultaat oplevert en dus faalt. In dit geval is het belangrijk te achterhalen wat de oorzaak voor het falen van deze test juist is. Ligt de oorzaak bij een foute implementatie van het systeem of ligt de fout eerder bij het foutief opstellen van de test zelf? Wanneer een test een negatief resultaat teruggeeft



Figuur 4.11: Gefaalde test weergegeven in de Regression Suite Automation Tool.

10.	On the Action Pane, click General.
11.	On the Action Pane, click Options.
12.	Click Add.
13.	In the Description field, type a value. (Value used in this run = "Test Vendor Contact")

Figuur 4.12: Gefaalde test weergegeven in Azure DevOps.

zal dit aangegeven worden in de Regression Suite Automation Tool aan de hand van een rood kruis bij het Result vakje. Zoals je kan terugzien in afbeelding 4.11. Dit toont aan dat de test niet succesvol is. Een meer gedetailleerde boodschap betreffende de oorzaak voor het falen van de test, kan teruggevonden worden in Azure DevOps. Hier krijg je een gedetailleerde weergave van welke stappen een test succesvol doorlopen heeft en bij welke stap de test een fout geeft. Zoals je kan zien op afbeelding 4.12 is er bij deze test in stap 13 iets mislopen. Echter geeft deze weergave niet voldoende info om te achterhalen wat juist de oorzaak voor het falen van de test is. Enkel de foutieve stap wordt afgebeeld.

De errorlog die bij een gefaalde test gegenereerd wordt geeft je extra informatie betreffende de oorzaak voor het falen van deze test. Hier worden alle zaken die door de Regression Suite Automation Tool worden uitgevoerd neergeschreven, zoals in onderstaand logbestand 4.3.2 weergegeven. In dit logbestand zien we dat de eerste 12 stappen van de test succesvol zijn verlopen. Wanneer de 13de stap van de test uitgevoerd wordt faalt de test, en wordt als reden een exceptie meegegeven. Op basis van deze exceptie kan je achterhalen wat de fout juist is. In dit geval hebben we te maken met een 'DuplicateKey-Exception' wat wijst op een slecht geconfigureerde test en niet op een foutief ontworpen systeem.

Status: Test case failed.

(Details are also in the attached base log file
Create_Vendor_RSAT_283_BaseLog.txt)

Steps:

Navigate to: VendTable ("vendtablelistpage")

Click New.

In the Vendor account field, type a value.

(Value used in this run = "RSATVE001")

In the Name field, type a value.

(Value used in this run = "RSAT TEST VENDOR 01")

Click Cancel.

In the Group field, type a value.

(Value used in this run = "40")

Note the value in the Vendor account field to reference later

(Value saved in this run = "RSATVE001")

On the Action Pane, click Procurement.

On the Action Pane, click Invoice.

On the Action Pane, click General.

On the Action Pane, click Options.

Click Add.

DuplicateKeyException: Cannot create a record in Vendors (VendTable).

Vendor account: RSATVE001.

The record already exists.

In the Description field, type a value.

(Value used in this run = "Test Vendor Contact")

ERROR:

<Message>Setting the value of the control ContactInfo_Description

(Type: Input) failed. This step expects an editable control,

but the control was either read-only,

disabled or not visible at the time of test execution

For debug purpose:

For Form VendTable's current state: IsInViewMode: false;

Preloaded: false; Caption: All vendors;

ActiveControl: SystemDefinedOptions; CanSwitchViewMode: true;

ForceLegacyGrid: 0;

For AppBarSection ActionPane's current state: Enabled: true;

Visible: true; TypeName: AppBarSection;

For ActionGroup SystemDefinedPage's current state: Enabled: true;

Visible: true; TypeName: ActionGroup; FrameOptionButton: None;

FrameOptionValue: 1;

For CommandButton SystemDefinedSaveButton's current state: Enabled: true;

Visible: true; TypeName: CommandButton; Label: Save;

For ActionGroup PrioritizedPage's current state: Enabled: true;

Visible: true; TypeName: ActionGroup; FrameOptionButton: None;

FrameOptionValue: 1;

For CommandButton SystemDefinedNewButton's current state: Enabled: true;

Visible: true; TypeName: CommandButton; Label: New;

For CommandButton SystemDefinedDeleteButton's current state: Enabled: true;

Visible: true; TypeName: CommandButton; Label: Delete;

For AppBarTab VendorTab's current state: Enabled: true;

Visible: true; TypeName: AppBarTab; Label: Vendor;

For AppBarTab Procurement's current state: Enabled: true;

Visible: true; TypeName: AppBarTab; Label: Procurement;

For AppBarTab Invoice's current state: Enabled: true;

```

Visible: true; TypeName: AppBarTab; Label: Invoice;
For AppBarTab General's current state: Enabled: true;
Visible: true; TypeName: AppBarTab; Label: General;
For AppBarTab SystemDefinedOptions's current state: Enabled: true;
Visible: true; TypeName: AppBarTab; Label: Options;
For AppBarSearchField AppBarSearch's current state: Enabled: true;
Visible: true; TypeName: AppBarSearchField;
For ActionGroup SystemDefinedButtonsButtonGroup_Form's current state:
Enabled: true;
Visible: true; TypeName: ActionGroup; FrameOptionButton: None;
FrameOptionValue: 1;
For MenuButton SysPAMenuButton's current state: Enabled: true;
Visible: true; TypeName: MenuButton; Label: Power Apps;
For MenuButton SystemDefinedOfficeButton's current state: Enabled: true;
Visible: true; TypeName: MenuButton; Label: Open in Microsoft Office;
For CommandButton SystemDefinedAttachButton's current state: Enabled: true;
Visible: true; TypeName: CommandButton; Label: 0;
For CommandButton SystemDefinedRefreshButton's current state: Enabled: true;
Visible: true; TypeName: CommandButton; Label: Refresh;
For CommandButton SystemDefinedPopoutButton's current state: Enabled: true;
Visible: true; TypeName: CommandButton; Label: Open in new window;
For CommandButton SystemDefinedCloseButton's current state: Enabled: true;
Visible: true; TypeName: CommandButton; Label: Close;
For AppBarGroup SystemDefinedPersonalizeGroup's current state: Enabled: true;
Visible: true; TypeName: AppBarGroup;
For Button SystemDefinedAlwaysEditModeButton's current state: Enabled: true;
Visible: true; Type

```

Het niet correct specificeren van zoek en filter waarden is een regelmatig voorkomende reden voor het falen van een test. Wanneer een test een bepaalde actie beschrijft voor een specifiek item is het belangrijk dit item zo specifiek mogelijk te beschrijven. Een veel voorkomend gegeven is dat een bepaald item in een lijst van verschillende items aangeklikt dient te worden. Aan de hand van een filter wordt ervoor gezorgd dat slechts het specifieke item overblijft. Indien dit niet het geval is kan dit ervoor zorgen dat de test niet succesvol afgewerkt kan worden. Een voorbeeld hierbij is indien 2 waarden beschikbaar zijn bij het opstellen van de test. Wanneer in een later fase nieuwe waarden aan de mogelijke data toegevoegd wordt, kan het zijn dat het verwachte item niet langer op dezelfde plaats in de lijst terug te vinden is. De test zal een fout item selecteren en volgende bewerking zullen niet langer succesvol kunnen worden uitgevoerd of zullen niet het verwachte resultaat opleveren. Het correct en specifiek specificeren van de te testen zaken is een cruciaal onderdeel in het opstellen van een correcte test en het welslagen van deze test.

In bovenstaand geval geeft de errorlog een duidelijke boodschap betreffende de oorzaak voor het falen van de test. Dit is niet steeds het geval. Wanneer bijvoorbeeld een bepaalde knop niet aanklikbaar is tijdens de geautomatiseerde test zal dit de test doen falen. De

boodschap die in dergelijke gevallen gegeven wordt, geeft dit ook aan als reden. Hiermee weet de tester niet wat de oorzaak voor het niet aanklikbaar zijn van deze knop is. Tijdens het opstellen van de test was deze knop wel aanklikbaar. Deze moeilijkheid toont aan dat het opstellen van geautomatiseerde testen zeer belangrijk is en delicaat is. De testen moeten steeds zo specifiek mogelijk worden geconfigureerd om op deze manier mogelijke verwarring te vermijden. Hieruit blijkt ook dat het belangrijk is korte specifieke testgevallen aan te maken. Op deze manier wordt de kans op een fout kleiner en blijft de kost indien een testgeval foutief geconfigureerd is relatief laag en kan deze eenvoudig opnieuw worden aangemaakt.

Het schrappen van een foutieve stap tijdens het opnemen van een nieuw testscenario kan snel leiden tot het foutief lopen van deze test. Dit komt omdat de test kan worden uitgevoerd, maar dat mogelijk foutieve waarden worden gebruikt. Het kan er ook voor zorgen dat de test niet alle gedefinieerde stappen kan uitvoeren en dus zal falen. Het is belangrijk dat testen steeds eenvoudig en correct geconfigureerd zijn. Indien een fout gemaakt wordt tijdens het opnemen van de verschillende stappen, begin je beter helemaal opnieuw met het opnemen van deze test. Dit om bovenvermelde problemen te voorkomen en de correctheid van de test te garanderen. Om eerder vermelde redenen is het ook belangrijk steeds een nieuwe test te starten vanaf het dashboard van de Dynamics 365 Finance & Operations applicatie.

5. Kritische noot

De technische uitwerking van hoe verschillende geautomatiseerde gebruiker acceptatie testen moeten worden opgezet aan de hand van de Regression Suite Automation Tool is slecht een deel van het succesvol implementeren van deze theorie. Het is belangrijk dat de theorie ook daadwerkelijk in de realiteit kan worden toegepast. Dit is echter makkelijker gezegd dan gedaan. Het gebruik van deze nieuwe tool is nog niet sterk vertegenwoordigd over projecten, een reden hiervoor is de jeugdigheid van de technologie maar ook het gebrek aan ervaring. Tot op heden heeft Avanade zelf nog geen projecten lopen waarin de Regression Suite Automation Tool gebruikt wordt.

In een interview met Stijn Colson¹, Solution Architect bij Avanade, wordt dieper ingegaan op hoe de Regression Suite Automation Tool momenteel in de praktijk toegepast wordt en wat de struikelblokken zijn. Stijn werkt al meer dan 12 jaar bij Avanade en heeft meer dan 16 jaar ervaring met Microsoft Dynamics. Dankzij zijn expertise in de sector kunnen we een concreet beeld scheppen van hoe de theorie en praktijk al dan niet met elkaar overeenkomen.

¹ <https://www.linkedin.com/in/stijn-colson-7a5b296/>

Het lastige met RSAT is dat je de theorie hebt en dan heb je de praktijk. Er zijn momenteel niet heel veel projecten die gebruik maken van RSAT. Met uitzondering van één project waar de klant zelf RSAT heeft opgezet. Dit is deels het probleem van RSAT, je hebt het technologische deel van de tool, het technisch configureren van de testen maar dan heb je ook de andere zijde, de business input. Deze business is het bepalen welke scenarios je wenst te testen, in deze fase zit het meeste werk. Het is daarbij ook het onderdeel waar de klant de meeste input moet geven. Daar deze reden is het voorzien dat de Regression Suite Automation tool door een klant zelf geconfigureerd wordt. Natuurlijk kunnen wij deze scenarios ook implementeren voor de klant, maar dit is eerder een duur gegeven.

Klanten verwachten dat RSAT een eenvoudig te implementeren tool is met een eenmalige opzet. In de praktijk is dit niet het geval, er is een mismatch aan verwachtingen. Microsoft brengt dit in de markt als een eenvoudige, lage kost oplossing vergeleken met het manueel gebruiker acceptatie testen. Maar dit is niet het geval, deze kost is zeer afhankelijk van omgeving tot omgeving. Er zijn bepaalde parameters die hier een sterke invloed op hebben. Maatwerk is een voorbeeld van een doorslaggevende parameter in het verhaal rond de Regression Suite Automation Tool. Hoe meer maatwerk aan een applicatie te pas komt, hoe meer je moet testen. Het aantal scenarios, het aantal versies van processen, nemen toe in deze context. Hoe meer scenarios je hebt hoe meer cases je moet gaan opzetten hoe meer effort het kost om al deze testgevallen op te zetten. Ook naar onderhoud toe zal dit steeds meer effort kosten.

Het opzetten van geautomatiseerde testen kost, vergeleken met standaardtesten, een pak meer effort. Dit gaande van het definiëren van deze testen tot het onderhouden ervan. De grote vraag is dan natuurlijk of deze effort er ook daadwerkelijk uitgehaald kan worden. Dit is de eeuwige vraag en oefening die je bij elk geval opnieuw moet gaan voeren. Het opzetten van de processen is in feite de basis die je bij elke projectimplementatie moet doen. Normaal heb je deze zaken. Deze zaken zou je vervolgens moeten kunnen hergebruiken als basis van RSAT en geautomatiseerd testen. De Total Cost of Ownership, de totale kost van het project, zal sterk afhangen van de winst die je eruit zal halen. Deze winst is gebaseerd op de geleverde effort bij het opzetten van de testen maar ook gezien de frequentie waarmee de verschillende testen zullen worden uitgevoerd. Wanneer je denkt in termen van Microsoft Cloud en Dynamics, moet je minstens 3 maal per jaar mee met de frequente releases dankzij Microsoft One Version. Er zijn natuurlijk meer releases per jaar en hoe vaker je deze releases doorvoert, hoe vaker je verschillende testen uitgevoerd moet worden, dus ook hoe meer benefit je eruit zal halen. Maar om eerlijk te zijn is dit in de praktijk geen eenvoudige oefening. Je kan niet zomaar al de verschillende parameters die een invloed hebben bepalen. Je moet dit case per case bekijken. Wanneer wij, binnen Avanade hiernaar vragen, schrikken we toch vaak hoeveel effort er nodig is om dit initieel op te zetten en draaiende te houden.

Dit wil niet zeggen dat we RSAT niet steeds vaker zien opduiken. Wanneer een klant een Request for Proposal doet, er steeds vaker gevraagd wordt naar, ten eerste wat zijn de opties en ten tweede wat is er daadwerkelijk haalbaar. Deze vragen zijn wel vaker puur informatief en minder, we willen het effectief toepassen, geef ons er een budget voor. Ook dit hangt terug sterk af van klant tot klant. Klanten die net de stap naar de Cloud zetten, beseffen nog niet waarom het een goed idee is van geautomatiseerde testen te gebruiken.

Dit omdat ze nog niet mee zijn in de correcte overtuiging.

Wanneer we kijken naar de Total Cost of Ownership, dan kunnen we stellen dat het vermijden van maatwerk en het zoveel mogelijk standaard houden, een veel grotere winst levert vergeleken met het toepassen van een hoop maatwerk en vervolgens aan geautomatiseerd testen te doen. Dit is in vele situaties een no brainer. We zorgen ervoor dat de klant richting de standaard implementatie geduwd wordt. Op deze manier is er functioneel gezien ook veel minder noot aan testing. Dit vermits we de zaken zeer high level kunnen testen, met de eigenaar van het product als verantwoordelijke. Er zullen natuurlijk ook geen zaken vrijgegeven worden waarvan we weten dat deze totaal niet zullen werken. Op die manier heb je ook meer leverage en zit de druk niet volledig op je maatwerk. Wanneer je natuurlijk in een omgeving zit met veel integraties of veel maatwerk zit je al snel in een heel andere situatie. Wat vervolgens terug een hele discussie op zich is. In theorie zou je steeds al je processen elke keer opnieuw moeten testen. Maar op de fundamentele, standaard processen mag je er eigenlijk vanuit gaan dat Microsoft niks zal doen dat ervoor zorgt dat deze niet meer zullen werken. Ook processen die fundamenteel zullen veranderen, weten we al lang op voorhand dat dit zal gebeuren en zijn er reeds migratie trajecten opgezet. En dan kunnen we dit als onderneming opvangen. De focus bij testen ligt op maatwerk en minder op de verschillende standaard processen.

Regression Suite Automation Tool als Microsoft tool zit al redelijk goed. Het heeft reeds een lang traject afgelegd en is naar gebruik toe steeds aan het evolueren in de positieve richting. Dit dan gericht op het initieel opzetten en beheren ervan. Maar de vraag zit hem vooral in waar zit de bepaalde benefit voor een bepaalde klant in een bepaalde omgeving. Dit is een oefening dat eigenlijk case per case en per omgeving samen met de klant moet bekeken worden. In de theorie kan je zeggen dat in elke Cloudomgeving naar een bepaalde vorm van geautomatiseerd testen moeten gaan. De vraag is echter, hoe ver ga je daarin. In een ideale wereld heb je bij elke klant een center of excellence met daar enkele experts werkzaam. Je zou dan verwachten dat een van deze experts enkel bezig is met geautomatiseerde testen, waardoor de organisatie zelf de verschillende testen perfect in orde en up to date kan houden. Dit in samenloop met de Microsoft One Version cyclus. Maar de realiteit is vaak heel anders. Als de klant zelf deze testen kan beheren zal de tijd en kost redelijk beheersbaar zijn. Als je deze zaken laat uitvoeren door een system integrator of iemand verantwoordelijk voor de Application Management Services, dan zal dit momenteel nog redelijk duur zijn. De ervaring uit onze Belgische entiteit naar wie de geautomatiseerde testen nu juist opzet is eerder beperkt. Binnen onze organisatie maken we weldegelijk gebruik van de tool. Maar we merken dat de initiële effort voor het opzetten van de cases en het zorgen voor testgegevens, redelijk serieus is. Als dit dan gedaan moet worden door experts is dat redelijk duur. Wanneer het samen met de klant gedaan worden zal de prijs lager liggen maar zeker en vast niet laag. Binnen Avanade hebben we momenteel niet genoeg ervaring om hier een sluitende conclusie rond te trekken.

Voor bestaande projecten zijn de verschillende gebruiker acceptatie testen reeds opgesteld. Gezien de Regression Suite Automation Tool nog niet lang beschikbaar is, is de vraag of deze bestaande testen geautomatiseerd kunnen worden voor gebruik met RSAT. In de projectuitvoering die we nu hanteren, namelijk de agile aanpak waar gewerkt wordt met een user en een scenario per proces en wanneer de oefening met de klant goed opgezet is,

zou het bijna 1 op 1 mogelijk zijn om deze testen over te zetten naar geautomatiseerde testen. Als ik dan kijk naar sommige van onze klanten, als zij alle info beschikbaar stellen gecombineerd met onze input, kunnen deze testen als een goed startpunt dienen. Dit wil natuurlijk niet zeggen dat elke test ook weldegelijk geautomatiseerd getest zal kunnen worden. Rond integratie heb je al een hogere complexiteit dit vermits verschillende systemen met elkaar moeten communiceren. Deze zaken zijn belangrijk dat het proces end to end getest kan worden. Of het mogelijk is deze testen ook door enkel gebruik te maken van de Regression Suite Automation Tool te automatiseren weet ik niet. Dit vermits er over verschillende systemen heen getest moet worden. Het gaat hier ook meer over integratie testen en minder over gebruiker acceptatie testen.

Een ander vaak voorkomend gegeven is dat veel organisaties blijven steken in het midden, er is een voordeel bij het gebruik van geautomatiseerd testen maar dit voordeel is niet substantieel genoeg om voor volledig automatisatie te kiezen. Momenteel is geen duidelijke reden waarom dit zo is. Als we op termijn meer feedback hebben betreffende wat de benefits kunnen zijn bij automatisatie, zal er ook gemakkelijker geschakeld worden. Dit omdat je vervolgens al rechtstreeks vanuit het project de automatisatie kan opnemen en hier genoeg tijd en energie aan kan besteden. Nu heb je vaak dat een klant de gebruiker acceptatie testen zelf opzet, wanneer je dan vervolgens wil schakelen naar het gebruik van RSAT kan het zijn dat de helft van de scenarios opnieuw moet gaan aanmaken. In dergelijke situaties zal er niet veel interesse naar het gebruik van de Regression Suite Automation Tool zijn. Als op deze manier te werk moet gegaan worden, zal er al een grote meerwaarde aan verbonden moeten zijn om het er nog uit te halen.

De zaken uit voorgaande paragraaf geven wel een goed punt aan. Als je RSAT wenst te gebruiken dat je het moet gaan integreren vanaf dag 1 van het project. Er moet meteen rekening mee gehouden worden en het gebruik moet goed voorbereid worden. Dit is een belangrijke factor. Veel klanten zeggen echter dat ze dit later willen toevoegen, dit zorgt er vervolgens voor dat het op de lange baan komt, er in een latere fase naar gekeken wordt. Hierdoor zal het gebruik van de Regression Suite Automation Tool al een pak minder efficiënt zijn en zullen de voordelen zo goed als verloren gegaan zijn. De reden voor het uitstellen van het gebruik van de tool kan zijn dat een bepaalde input niet gekend is. Wat is de huidige ervaring en waar toont dit of we RSAT al dan niet kunnen gebruiken? Wanneer werkt het wanneer niet, wat kan je automatiseren wat niet, wat kan de klant zelf doen wat niet? Momenteel missen we hier feedback die ons in staat stelt hierover te oordelen.

Een andere reden voor het niet doorstoten van RSAT kan bij de overtuiging van de klant liggen. Wanneer deze klant in een bepaalde overtuiging, van waarom zou ik deze tool nodig hebben, en vervolgens enkel de initiële kost van het plaatje bekijkt, kan hij al snel concluderen dat het gebruik van RSAT geen meerwaarde oplevert. Wanneer de klant buiten deze overtuiging zou stappen en geautomatiseerd testen op lange termijn bekijkt, kan hij mogelijks de voordelen van geautomatiseerd testen inzien. De praktijk stemt helaas nog niet overeen met de theorie. Wanneer puur naar de theorie gekeken wordt en je wenst als onderneming enkel gebruik te maken van de Cloud en dan ook steeds de laatste versie van het systeem wenst te gebruiken is het bijna een no brainer. Helaas is dit niet hetgeen we in de praktijk merken. Vaak vragen klanten juist het omgekeerde. Ze wensen zo weinig mogelijk updates te moeten doen, dit is natuurlijk ook het kip-en-ei-

probleem. Het kan goed zijn dat klanten juist geen updates willen omdat ze telkens deze update moeten uitvoeren en vervolgens alle checks en testen moeten overlopen. Als je ervoor kan zorgen dat RSAT goed geïmplementeerd is, kan deze druk wegvallen. Maar dan kom je terug op het punt van als je de implementatie zo standaard mogelijk houdt en enkel standaard integraties gebruikt zal dit er ook voor zorgen dat de automatisatie vlotter zal lopen. Deze zaken hangen allemaal vast aan elkaar.

Wat we nu eigenlijk kunnen concluderen is dat de tool op zich niet het probleem lijkt te zijn. Maar dat bij het opzetten, de effort nodig om de tool goed te laten werken eerder het struikelblok is. De initiële effort en de effort nodig om de tool goed te laten werken schrikt organisaties momenteel nog af. Het is niet iets dat je zomaar kan opzetten, er is een grote effort aan verbonden die vaak onderschat wordt. De content om het te laten werken op een klant omgeving is het moeilijkste onderdeel. Om te weten of het al dan niet haalbaar is om de Regression Suite Automation tool te implementeren zal afhangen van bepaalde parameters, eens we deze parameters hebben zal dit al een pak meer bijdragen aan het gebruik van RSAT. Deze parameters kan je enkel bekomen door ervaring op te doen en verschillende projecten uit te voeren. Er is geen discussie dat geautomatiseerd testen de toekomst is, er is enkel nog werk om het praktisch te gaan toepassen en gebruiken. De hoeveelheid effort nodig om te implementeren is momenteel te hoog, eens we de best practices hebben en de effort naar beneden krijgen is er een mooie toekomst weggelegd voor de Regression Suite Automation Tool en geautomatiseerd testen in het algemeen.

6. Conclusie

In deze studie hebben we onderzocht hoe je de Regression Suite Automation Tool implementeert voor het geautomatiseerd gebruiker acceptatie testen van Microsoft Dynamics 365 Finance & Operations applicaties. Hieruit is gebleken dat het opstellen van een correcte testen afhankelijk is van verschillende factoren. Zo is het belangrijk een juiste configuratie te hanteren. Naast de configuratie heb je ook verschillende punten waaraan tijdens het opnemen van een testgeval aandacht besteed moet worden. Ten slotte moet ook rekening gehouden worden met het correct toevoegen van de verschillende testen in Azure DevOps. In onderstaande secties concluderen we per onderdeel wat de belangrijkste bevindingen zijn van deze studie.

6.1 Hoe configureer je RSAT?

Uit het onderzoek kunnen we concluderen dat het opstellen en configureren van correcte testgevallen mogelijk is. Het configureren van de Regression Suite Automation Tool is slechts een klein onderdeel van het correct uitvoeren van geautomatiseerde gebruiker acceptatie testen. Tijdens de configuratie is het belangrijk dat elk onderdeel correct gespecificeerd is. Eens deze configuratie voltooid is zal deze ook steeds correct werken. Tijdens het configureren van RSAT is het ook belangrijk dat je zeker bent van elk element op zich. Zo kan je indien een test niet succesvol is verschillende elementen uitsluiten.

6.2 Hoe stel je een correcte test op?

Het belangrijkste en ook meest foutgevoelige onderdeel tijdens het opzetten van geautomatiseerde gebruiker acceptatie testen, is het opnemen van een testgeval op zich. Dit is de meest foutgevoelige fase en vraagt veel aandacht. Een correcte test neem je op door op verschillende elementen te letten. Onderstaand zie je een overzicht van de belangrijkste aandachtspunten.

- Starten vanaf dashboard
- Specifiek filteren
- Expliciet stappen definiëren
- Fouten vermijden
- Korte scenarios

Het starten van de Task Recording dient steeds vanaf het Finance & Operations dashboard te gebeuren. Van elk testgeval wordt verwacht dat deze vanaf dit dashboard start. Indien je de opname met de Task Recorder niet gestart bent op het dashboard, zal de test ook niet correct verlopen. De reden hiervoor is dat de Regression Suite Automation Tool de test steeds aanvangt vanop dit dashboard. Indien je test op een specifieke pagina begint op te nemen zullen de eerste stappen uitgevoerd worden op een verschillend scherm dan aangegeven met de task recorder, dit terwijl deze stappen hier helemaal niet kunnen worden uitgevoerd. Dit kan resulteren in een totaal verschillende test wat op zich foutieve gegevens met zich meebrengt. De test zal in deze gevallen niet relevant zijn. Het is dus zeer belangrijk tijdens het opstellen van een correcte test dat een nieuwe opname steeds vanaf het Finance & Operations dashboard gestart wordt.

Een volgend onderdeel dat een significante rol speelt in het opstellen van correcte testgevallen, is het gebruik van specifieke filters tijdens de opname. Indien geen gebruik gemaakt wordt van zeer specifieke filterwaarden kan dit resulteren in een falende of foutieve test. De reden hiertoe is dat over een bepaalde periode steeds nieuwe data wordt toegevoegd aan het systeem. Wanneer tijdens de creatie van de test slechts een beperkte hoeveelheid data aanwezig is zal de uitkomst van een zoekopdracht ook slechts deze beperkte hoeveelheid data weergeven. Wanneer vervolgens een waarde wordt aangeklikt zal de Task Recorder dit op basis van een index registreren. Wanneer vervolgens de test in een later fase nogmaals uitgevoerd wordt, kan het zijn dat deze index niet langer overeenkomt met het gewenste item. Indien steeds zeer specifiek gefilterd wordt zal enkel deze specifieke waarde teruggegeven worden. Op deze manier is het onmogelijk een foutieve waarde aan te klikken en zal de test op het correcte element de volgende stappen uitvoeren.

Een ander belangrijk onderdeel in het opstellen van correcte testgevallen is het expliciet definiëren van een aangeklikte knop. Vaak dien je een actie uit te voeren die zich in een submenu bevindt, wanneer je systeem reeds dit submenu heeft openstaan zal deze stap niet worden opgenomen. Dit heeft natuurlijk gevolgen wanneer de geautomatiseerde test wordt uitgevoerd. Het systeem zal niet langer dezelfde pagina openen en de verwachte actie kan niet worden voldaan. Dit zal vervolgens resulteren in het falen van de test. Het is dus belangrijk van expliciet elke actie uit te voeren aan de hand van een klik, dit zodat

deze actie wordt opgenomen in het stappenplan en de test vervolgens correct kan worden uitgevoerd.

Het vermijden van het maken van fouten tijdens de opname van een testgeval is ook een belangrijk onderdeel tot het opleveren van een correcte test. Dit grotendeels wegens dezelfde reden als bovenvermelde paragraaf. Tijdens het opnemen van een test kan een bepaalde stap geschrapt worden. Wanneer echter een stap geschrapt wordt die als actie het weergeven van de volgende stappen heeft, zal de test niet langer succesvol kunnen verlopen. Het is om deze reden belangrijk geen fouten te maken tijdens het opnemen van de verschillende testen. Indien toch een fout gemaakt wordt tijdens het opnemen van een test, is het aangeraden het opnemen van de test volledig opnieuw te starten.

Een laatste aandachtspunt tijdens het opnemen van een correcte test is het gebruik van korte scenarios. Dit sluit eigenlijk zeer dicht aan bij voorgaande paragraaf. Het gebruik van een kort scenario zorgt ervoor dat je veel parameters uitsluit die een invloed hebben op het resultaat. Op deze manier kan je gerichter een fout opsporen. Indien er een fout gemaakt wordt in het opnemen van de test, kost dit ook minder effort om het scenario vanaf nul opnieuw op te nemen.

6.3 Hoe voer je testen uit aan de hand van RSAT?

Een test succesvol uitvoeren aan de hand van de Regression Suite Automation Tool is afhankelijk van enkele factoren. De eerste factor is het opstellen van een correcte test op zich. Daarnaast is het ook belangrijk dat een test correct wordt toegevoegd aan Azure DevOps. Een test correct toevoegen aan Azure DevOps is afhankelijk van bepaalde factoren. Deze factoren zijn:

- Naamgeving
- Test Id
- Volgorde

De naamgeving van een test in Azure DevOps is een belangrijk onderdeel voor het uitvoeren van een test. Het is belangrijk dat de naam van de test overeenkomt met de naam van het bestand die na afloop van de opname gedownload wordt. Indien deze twee namen niet overeenkomen zal een test niet kunnen worden uitgevoerd.

Ook is het belangrijk dat elke test een uniek Id heeft. Op deze manier kunnen testen geïdentificeerd worden. Dit Id wordt bij het aanmaken van de test in Azure DevOps automatisch toegekend en vormt vaak niet de oorzaak tot het falen van een test. Het is natuurlijk wel een essentieel onderdeel voor het maken van correcte geautomatiseerde gebruiker acceptatie testen.

Een laatste belangrijk onderdeel voor het uitvoeren van een test is de volgorde van de verschillende testgevallen. Testen zijn vaak afhankelijk van data of stappen uitgevoerd door eerdere testen. Dergelijk proces wordt ook wel het chainen van testen genoemd. Wanneer verschillende verbonden testen niet in de juiste volgorde uitgevoerd worden zal

dit leiden tot niet succesvolle testgevallen. Het is dus essentieel dat elke test in de juiste volgorde gedefinieerd wordt.

Alvorens een test effectief uitgevoerd kan worden moet deze worden ingeladen in de Regression Suite Automation Tool. Eens deze testen correct gedefinieerd zijn in Azure DevOps is er niet veel werk in de Regression Suite Automation Tool om de test, test klaar te zetten. Het genereren van de uitvoerbestanden en parameter bestanden is een van de laatste belangrijke stappen. De laatste stap is het correct definiëren van de gegenereerde parameterbestanden. In deze Excel bestanden definieer je de waarde voor elke parameter. Deze parameters definiëren de waarden die zullen worden ingegeven in de Finance & Operations applicatie. Het is belangrijk rekening te houden met parameters die een unieke waarde omschrijven in de applicatie. Indien bij de creatie van een nieuwe vendor geen uniek nummer gespecificeerd is zal dit het falen van de test als gevolg hebben. Ook in deze parameterbestanden is het belangrijk de verschillende testen correct aan elkaar te linken. Eens deze zaken correct zijn uitgevoerd kunnen één of meerdere testen uitgevoerd worden.

6.4 Hoe worden de resultaten van een test weergegeven?

Na het uitvoeren van één of meerdere testgevallen zal in de Regression Suite Automation Tool reeds worden aangegeven indien een test al dan niet succesvol was. Dit is echter een zeer beperkte weergave. In Azure DevOps kan een meer gedetailleerde beschrijving van een test teruggevonden worden. In Azure DevOps wordt per test beschreven wat fout liep en wat correct liep. Daarnaast worden ook verschillende grafieken opgebouwd die het aantal succesvolle testen weergeeft ten op zichte van het aantal gefaalde testen. Op deze manier krijg je een gedetailleerd overzicht van de resultaten van elke test als ook een algemeen overzicht.

Geautomatiseerd gebruiker acceptatie testen aan de hand van de Regression Suite Automation Tool is een stap richting de toekomst. Het opstellen van correcte testen vraagt momenteel nog een zeer hoge workload. Daarnaast moeten deze geautomatiseerde testen ook regelmatig onderhouden worden. Dit alles vraagt veel tijd, geld en kan slechts voordelig zijn in bepaalde situaties. De frequentie waarmee deze testen worden uitgevoerd is een van deze bepalende factoren, als ook de hoeveelheid maatwerk aanwezig in de Finance & Operations applicatie. De theorie van de geautomatiseerd gebruiker acceptatie testen aan de hand van RSAT zit goed. De tool werkt vlot en de nodige componenten zijn aanwezig. Helaas verteld de praktijk momenteel nog een ander verhaal, en zal enkel projectervaring en het gebruik van de Regression Suite Automation Tool ervoor zorgen dat theorie en praktijk dichter naar elkaar toe groeien.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Het testen van ontwikkelde software is een van de belangrijkste onderdelen in het ontwikkelingsproces. Testen geven namelijk een inzicht in welke zaken correct verlopen maar veel belangrijker welke elementen niet correct verlopen. Echter is het schrijven van testen een tijdrovend en kostelijk proces. Wanneer we verschillende regressie- (bestaande software) en aanvaarding (nieuwe software) testen automatiseren kan er op verschillende van deze resources bespaard worden. Maar hoe automatiseren we juist deze testen? Aan de hand van de Regression Suite Automation Tool van Microsoft kan een deel van bovenvermelde testen geautomatiseerd worden. In dit onderzoek kijken we naar welke testen geautomatiseerd kunnen worden en welke invloed dit heeft op de ontwikkelsnelheid van Microsoft Dynamics 365 Finance & Operations projecten. Dit met als doelstelling duidelijke parameters te kunnen afbakenen. Deze parameters hebben een bepalende invloed op de tijd en de kost nodig voor het implementeren van geautomatiseerde testen vergeleken met klassiekere methoden (bv. V-Model) voor het uitvoeren van regressie- en aanvaardingstesten.

A.2 Stand van zaken

ERP implementaties hebben doorheen de jaren vaak moeten afrekenen met mislukkingen. Maar desondanks worden steeds meer technologieën ontwikkeld en toegepast binnen ERP. Een grote oorzaak voor het falen van deze projecten waren de te strakke deadlines en het hierdoor schrappen van verschillende belangrijke onderdelen. Een van deze geschrapte onderdelen is het testen van de ontwikkelde componenten. Zo werd de tijd voor testen verminderd en regressie testen werden slechts oppervlakkig aangehaald. (Murray & Coffin, 2001). Een grote verandering was het besef, dat een belangrijke factor voor het slagen van een project, het zetten van haalbare doelstellingen was. Zo ging opnieuw meer aandacht naar het testen van ontwikkelde modules. Wanneer bepaalde testen niet slaagden werd de oplevering van de software naar een latere datum verschoven en werden de nodige aanpassingen doorgevoerd. Men zag het belang van testen dus opnieuw in. Testen van een ERP systeem bestaat uit verschillende onderdelen. Integratie testen is een van de belangrijkste onderdelen binnen Microsoft Dynamics 365 Finance & Operations. Voorafgaand aan deze integratie testen zal men echter eerst enkele specifieke unit testen ondergaan. In deze testgevallen wordt nagegaan of de ontwikkelde code voldoet aan de gevraagde eigenschappen. Na het succesvol opleveren van bovenstaande testen zal men overgaan tot de integratie. Tijdens de integratie brengt men al de verschillende onderdelen samen tot een volwaardig ERP systeem. Nadat de integratie succesvol tot stand is gekomen zal men deze integratie ook aan verschillende testen onderwerpen. Dit zijn de verschillende integratie testen. Deze testen zijn noodzakelijk om te achterhalen of de verschillende zelfstandig werkende componenten ook als een geheel kunnen functioneren. Er zijn verschillende manieren om aan integratie testen te doen. Zo hebben we een top down strategie en een bottom-up strategie. Wanneer we kijken naar top down gaan we te werk met verschillende test stubs. Een stub vervangt een nog ontbrekende module. Deze stub zal op dezelfde manier reageren als de module die deze stub vervangt. Deze strategie wordt veelal gebruikt wanneer bepaalde modules nog aan unit testen onderworpen moeten worden (Singh & Singh, 2012). We plaatsen alle modules in de structuur van het programma. Hierna zullen de stubs vervangen worden door de effectieve modules. Op deze manier kunnen we de volledige integratie testen. Wanneer dit alles klaar is kunnen we aan de executie van de testen beginnen. Wanneer we kijken naar bottom up testen zien we een groot verschil. Bij deze strategie worden verschillende features samen gebracht in een cluster. Deze clusters gedragen zich als sub-functionaliteit van de module. Vervolgens zal men testen schrijven en uitvoeren voor de verschillende clusters. Deze methode is effectiever vergeleken met het top down integratie testen (Patel & Ali, g.d.).

Ten slotte is het ook belangrijk aandacht te hebben voor regressie testen. Zeker wanneer we spreken over ERP systemen. Het is namelijk van groot belang dat wanneer wijzigingen plaatsvinden in een bepaalde module, deze wijzigingen geen negatieve invloed zullen hebben op andere modules binnen het totale systeem. Een moeilijkheid bij regressie testen is het selecteren van een subset van testgevallen. Bij het uitvoeren van regressie testen is het dan ook belangrijk een juiste keuze te maken tussen bepaalde test technieken. Zo zijn efficiënte technieken om aan regressie testen te doen niet zeer nauwkeurig terwijl de betrouwbare testen dan weer zeer duur kunnen zijn wanneer men deze in grote systemen gebruikt (Orso e.a., 2004). Het is dus van groot belang de juiste balans te vinden zodat

nauwkeurig en gedetailleerd maar ook kostenefficiënt getest wordt. In deze balans zal automatisatie een steeds prominentere rol gaan spelen. Regression Suite Automation Tool (RSAT) is een tool van Microsoft voor het geautomatiseerd testen van business processen in het Microsoft Dynamics 365 Finance & Operations systeem. Deze tool laat organisaties en hun partners toe zowel custom als standaard implementaties van het systeem te testen. RSAT is veelal gebruikt voor het testen van custom code, met verschillende regressies gebruik makend van eenzelfde basis code. Dit wanneer een incrementele ontwikkelmethodologie gehandhaafd wordt (Donepudi, 2020).

A.3 Methodologie

Het beschrijven van een doeltreffende en goede methodologie zal een geleidelijk proces zijn. Dit wegens het constante leerproces en de complexiteit van de materie. Om te onderzoeken welke parameters een verschil maken dien je verschillende testmethoden te onderzoeken. Het schrijven van test gevallen alvorens de ontwikkelfase aanvangt of het schrijven/uitbreiden van testen na de ontwikkelfase. Vervolgens kan ook op een ontwikkeld systeem geautomatiseerde testen geschreven worden aan de hand van RSAT. Naast deze geautomatiseerde testen kan ook gekeken worden naar het uitvoeren van deze testen met de hand. Dit voor zowel regressie- als aanvaardingstesten. Concreet betekent dit dat verschillende testen zullen worden uitgeschreven. Testen die manueel worden uitgevoerd en testen die aan de hand van RSAT worden uitgevoerd. Dit voor eenvoudigere testgevallen in de beginfase maar vervolgens ook voor steeds complexere gevallen. Dit om een eventueel cut off punt in complexiteit te kunnen identificeren. Eens de verschillende testgevallen beschreven en uitgewerkt zijn kan data verzameld worden. Deze data kan zowel bepaalde cijfers zijn als ook bevindingen. Zo zal de implementatie effort, uitvoertijd, complexiteit, efficiëntie, nauwkeurigheid, uitvoertijd testen, worden bekeken en verzameld. Aan de hand van deze verzamelde data kan ten slotte een conclusie gevormd worden en kunnen de verschillende, bepalende parameters worden vastgelegd.

A.4 Verwachte resultaten

Regression Suite Automation Tool van Microsoft is nog een zeer jonge technologie. Vele testgevallen zullen nog niet optimaal uitgewerkt zijn binnen de tool. Niettegenstaande de leeftijd van RSAT kan deze technologie wel op een eenvoudige manier geïmplementeerd worden. Dit voor mensen met een sterke technische kennis maar ook voor mensen met een minder technische achtergrond. Ook kan je willicht een grote tijds winst boeken door gebruik te maken van RSAT vergeleken met klassieke testtechnieken. Het gebruik van RSAT als geautomatiseerde testtool zal niet voor elk bedrijf handig zijn. Bepaalde parameters zullen een sterke invloed hebben op hoe voordelig het gebruik van RSAT is. Dit kijkende naar implementatie effort, kosten en de gebruiksvriendelijkheid.

A.5 Verwachte conclusies

Regression Suite Automation Tool (RSAT) is een nieuwe technologie die ons in staat stelt om verschillende test scenarios van een Microsoft Dynamics 365 Finance & Operations systeem te automatiseren. Vermits deze technologie nog in zijn kinderschoenen staat ontbreken nog verschillende zaken (component- en unit testen). Zo kan je momenteel enkel aanvaardingstesten uitvoeren. Op deze manier wordt de business cycle getest. De gebruiksvriendelijkheid, inspanning en kost voor het implementeren van deze technologie zullen over de komende jaren enkel verbeteren. Dit kunnen we vergelijken met de evolutie van ERP systemen. Grote bedrijven sprongen snel op de kar maar na verloop van tijd wijzigden ERP systemen zodat ook middelgrote bedrijven hiervan gebruik konden maken (Van Everdingen e.a., 2000). Deze trend zal zich ook doorzetten wanneer het gaat over geautomatiseerd testen. Daarnaast zal ook de functionaliteit toenemen. Wanneer men de verschillende tekortkomingen die nu nog aanwezig zijn kan ontwikkelen en toepassen zal het gebruik en de effectiviteit van de tool stijgen. Het automatiseren van testen zal in de toekomst een steeds belangrijkere rol spelen. Zelf tot in die mate dat het merendeel van de testen geautomatiseerd zal zijn.

B. Kritische noot: gestelde vragen

Onderstaand kunnen de gestelde vragen tijdens het interview met Stijn Colson teruggevonden worden.

1. Wat is je ervaring in het gebruiken van RSAT?
2. Zie je een voordeel in het gebruiken van RSAT?
 - (a) Zo ja, voor wat soort projecten.
 - (b) Zo nee, wat is de reden hiertoe?
3. Op basis van welke elementen wordt bepaald of het gebruik van RSAT een voordeel oplevert?
4. Wat is volgens u het grote voordeel/nadelen verbonden aan RSAT?
 - (a) Hoe loopt dit in de praktijk?
5. Zijn klanten happig naar het gebruik van RSAT?
6. Adviseren jullie klanten om RSAT te gebruiken bij nieuwe en of bestaande projecten?
7. Indien RSAT geïmplementeerd wordt bij bestaande projecten wat is de effort nodig om RSAT te gebruiken voor deze reeds bestaande test gevallen?
 - (a) Totaal nieuwe testen maken?
 - (b) Kan dit overgezet worden?
 - (c) Kost verbonden hieraan?
8. Welke wijzigingen zijn er volgens u nodig om RSAT voordelig te maken voor elk project?

Bibliografie

- Balaji, S. & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26–30.
- Balfe, N., Sharples, S. & Wilson, J. R. (2015). Impact of automation: Measurement of performance, workload and behaviour in a complex control environment. *Applied ergonomics*, 47, 52–64.
- Balfe, N., Wilson, J. R., Sharples, S. & Clarke, T. (2012). Development of design principles for automated systems in transport control. *Ergonomics*, 55(1), 37–54.
- Cimperman, R. (2006). *UAT Defined: A Guide to Practical User Acceptance Testing (Digital Short Cut)*. Pearson Education.
- Collins, E. F. & de Lucena, V. F. (2012). Software Test Automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test (AST)*, 57–63. <https://doi.org/10.1109/IWAST.2012.6228991>
- Donepudi, V. P. (2020). Driving a successful implementation using Microsoft Dynamics 365 Finance and Operations and RSAT. <https://msdynamicsworld.com/story/driving-successful-implementation-using-microsoft-dynamics-365-finance-and-operations-and-rsat>
- Hanssen, G. K. & Haugset, B. (2009). Automated acceptance testing using fit. *2009 42nd Hawaii international conference on system sciences*, 1–8.
- Haugset, B. & Hanssen, G. K. (2008). Automated acceptance testing: A literature review and an industrial case study. *Agile 2008 Conference*, 27–38.
- Itkonen, J., Mäntylä, M. V. & Lassenius, C. (2013). The Role of the Tester's Knowledge in Exploratory Software Testing. *IEEE Transactions on Software Engineering*, 39(5), 707–724. <https://doi.org/10.1109/TSE.2012.55>
- Kupersmith, J. (2016). Putting the User Back in User Acceptance Testing.

- Luszczak, A. (2019). What is Microsoft Dynamics 365/AX? *Using Microsoft Dynamics 365 for Finance and Operations* (pp. 1–4). Springer.
- Meyer, B. (2014). *Agile The Good, the Hype and the Ugly*.
- Microsoft. (2021). *Wat is DevOps* (M. Azure, Red.). <https://azure.microsoft.com/nl-nl/overview/what-is-devops/>
- Murray, M. & Coffin, G. (2001). A case study analysis of factors for success in ERP system implementations. *AMCIS 2001 Proceedings*, 196.
- Orso, A., Shi, N. & Harrold, M. J. (2004). Scaling regression testing to large software systems. *ACM SIGSOFT Software Engineering Notes*, 29(6), 241–251.
- Otaduy, I. & Diaz, O. (2017). User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. *Journal of Systems and Software*, 133, 212–229. <https://doi.org/https://doi.org/10.1016/j.jss.2017.01.002>
- Patel, M. P. & Ali, S. (g.d.). Automation Integration Testing-Microsoft Dynamics 365 Finance and Operations.
- Rossberg, J. (2019). An overview of azure devops. *Agile Project Management with Azure DevOps*, 37–66.
- Singh, S. K. & Singh, A. (2012). *Software testing*. Vandana Publications.
- Van Everdingen, Y., Van Hillegersberg, J. & Waarts, E. (2000). Enterprise resource planning: ERP adoption by European midsize companies. *Communications of the ACM*, 43(4), 27–31.
- Yadav, R. S. (2012). Improvement in the V-Model. *International Journal of Scientific & Engineering Research*, 3(1), 2229–5518.