

Ontwerp en ontwikkeling van een testraamwerk installer

Pieter-Jan ROBRECHT

Promotor: Annemie Vorstermans

Co-promotor: Carl Eeckhout

Masterproef ingediend tot het behalen van
de graad van master of Science in de
industriële wetenschappen: master of Science
in de industriële wetenschappen ICT
Advanced Communicatie Technologies

©Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologicampus Gent, Gebroeders De Smetstraat 1, B-9000 Gent, +32 92 65 86 10 of via e-mail iiw.gent@kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Dankwoord

Dank aan mezelf

Dank aan een ander

Abstract

Televic Rail ontwikkelde een Python testraamwerk voor het testen van verschillende producten. De ontwikkelde software, die op verschillende platformen moet draaien, gebruikt verschillende drivers en bibliotheken. Om producten correct te kunnen testen, wordt het raamwerk vaak geüpdatet, bijvoorbeeld bij het uitbrengen van een nieuwe driver, bibliotheek of om nieuwe producten te ondersteunen. Het installatieproces is tijdrovend, foutgevoelig en wordt best geautomatiseerd. Door dit proces te automatiseren wordt het mogelijk om informatie over het installatie- en updateproces te verzamelen. Het doel van de thesis is het uitvoeren van onderzoek naar een efficiënte oplossing en het ontwikkelen van een prototype. Dit prototype wordt onderverdeeld in drie componenten: een packager, een deployment server en een deployment omgeving. In een eerste fase wordt de packager ontworpen. Deze staat in voor het samenvoegen van de software componenten. Fase twee van de thesis bestaat uit het uitwerken van de deployment server. Met de server worden de verschillende installers verspreid en wordt er informatie verzameld over de deployment environments. Als laatste wordt dan de deployment environment behandeld. In deze geïsoleerde omgeving kan het installatie- en updateproces veilig gebeuren. Na een grondige evaluatie van een eerste basisprototype wordt het ontwerp eventueel aangepast. Het prototype wordt in een laatste fase uitgebreid zodat een rapportering over geïnstalleerde versies, deployment status, . . . beschikbaar wordt voor het bedrijf.

Trefwoorden: Automatische – installer – testraamwerk - Python

Inhoudsopgave

1	Situering	1
1.1	Het probleem	1
1.2	Werkwijze	2
2	Bespreking	3
2.1	Packager	3
2.1.1	WiX Toolset	3
2.1.2	NSIS	4
2.1.3	Chocolatey	4
2.1.4	Qt Installer Framework	4
2.2	Database	4
3	Prototype	5
3.1	Uitwerking	5
3.2	Testen	5
4	Eind product	6
4.1	Uitwerking	6
4.2	Testen	6
5	Conclusie	7
A	Een aanhangsel	10
B	Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel	11
C	Poster	12

Lijst van figuren

1.1	Overzichtsdiagram van de algemene structuur	2
2.1	Ontwerp van de databank	4

Hoofdstuk 1

Situering

Televic Rail heeft een Python test framework ontworpen waarmee zij in staat zijn om verschillende producten te onderwerpen aan verschillende testscenario's. Het framework werd ontworpen om gebruikt te worden op verschillende testtorens en werd later aangepast om bruikbaar te zijn op gewone computers. Dit framework wordt intensief gebruikt tijdens het productieproces en is een spilfiguur voor het afleveren van producten die voldoen aan strenge veiligheidsnormen.

1.1 Het probleem

De applicatie gebruikt verschillende Python bibliotheken en hardware drivers om correct te functioneren. Het installatieproces is tijdrovend en foutgevoelig. Bij het uitbrengen van een nieuwe versie van de applicatie, bijvoorbeeld bij het uitbrengen van een nieuwe driver, bibliotheek of om nieuwe producten te ondersteunen, moet de applicatie geüpdatet worden. Dit proces lijdt aan dezelfde gebreken als het installatieproces. Het doel van deze thesis is zoeken van een langdurige oplossing voor dit probleem. Na een onderzoek zal een prototype geproduceerd worden die als oplossing aangeboden wordt aan het bedrijf.

Een probleemanalyse onthulde al snel dat dit probleem onder te verdelen is in verschillende deelproblemen. Het framework bestaat uit verschillende componenten, hieronder vallen de drivers en bibliotheken. Elke component heeft een unieke installatiewijze en de componenten moeten in een specifieke volgorde geïnstalleerd worden. Hiernaast moeten verscheidene componenten geconfigureerd worden aan de hand van een configuratiebestand. Dit configuratiebestand is doelsysteem specifiek. Tijdens het onderzoek zal er onderzocht welke technologieën gebruikt zal worden om al deze componenten te combineren tot één uitvoerbaar bestand. Bij het selectieproces moet er rekening gehouden worden met de toekomst. Er wordt best een technologie gekozen die cross-platform is, zodanig dat het bedrijf niet beperkt wordt in de toekomst.

De probleemanalyse onthulde ook dat de verschillende executables verspreidt moeten worden. Door dit proces te automatiseren, is het mogelijk om waardevolle informatie te verzamelen. Hiermee zouden rapporten gegenereerd kunnen worden voor het bedrijf. Tijdens het ontwerpen zal schaalbaarheid een belangrijke factor spelen. Naar de toekomst toe zal het aantal systemen waarop de applicatie geïnstalleerd wordt toenemen.

Zoals reeds hierboven vermeld, is het installatieproces foutgevoelig. Na de foutanalyse bleek dat hiervoor een schaalbare oplossing gevonden moet worden. Bij het finaliseren van een component installatie, moeten de functionaliteiten van de component gecontroleerd worden. Mocht

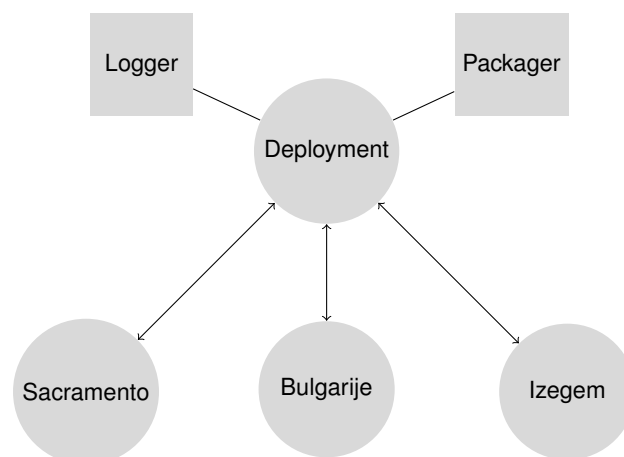
deze incorrect functioneren dan moet een gepaste actie, eventueel een rollback, gebeuren. Ook na de volledige installatie moet gecontroleerd worden of het geheel correct functioneert. Hierdoor wordt er voorkomen dat er nodeloos veel tijd verloren gaat in het herinstalleren van de applicatie. Robuustheid zal een belangrijke component zijn tijdens de ontwerpfase. Bij het ontwerpen moet gebruiksvriendelijkheid in het achterhoofd gehouden worden.

1.2 Werkwijze

In Sectie 1.1 werd opgelijst wat de problemen zijn. In wat volgt zullen verschillende oplossingswijzes opgesomd worden met vervolgens de werkwijze.

Na de probleemanalyse werd het al snel duidelijk dat het werk op te delen valt in drie grote componenten. Deze drie onderdelen zullen de basis vormen voor de architectuur en zullen gebruikt worden als leidraad. Het eerste onderdeel zal bestaan uit een packager met als doel het inpakken van de nodige drivers, bibliotheken, Naast de packager is er een deployment server nodig die instaat voor het verspreiden van de installers die de packager aflevert. Dankzij het automatiseren van de deployment, kan het installatieproces per client getrackt worden. Om deze stap te vereenvoudigen, moet aan de client-side een deployment environment voorzien worden. In die omgeving wordt de installatie geïsoleerd. Mocht een rollback nodig zijn, dan kan deze op een eenvoudige manier gebeuren.

Gedurende deze thesis wordt een prototype ontworpen. In Figuur 1.1 wordt de algemene structuur van de applicatie weergegeven. Met behulp van deze basis is het mogelijk om een demo te produceren die de verschillende problemen verhelpt.



Figuur 1.1: Overzichtsdiagram van de algemene structuur

Hoofdstuk 2

Bespreking

De structuur die in Hoofdstuk 1 werd opgebouwd In de loop van dit hoofdstuk zullen de verschillende technologieën besproken worden die in aanmerking komen. Naar het einde van het hoofdstuk toe, zal een keuze gemaakt worden en zal voor iedere component de meest geschikte technologie gekozen worden.

2.1 Packager

De eerste component die besproken wordt is de packager. Zoals reeds besproken, moet tijdens het ontwerpen van deze component rekening gehouden worden met enkele eigenschappen. Alvorens bepaald wordt welke technologie geschikt is, moet onderzocht worden wat ingepakt moet worden en op welke manier. Om een correct werkende applicatie te hebben, heeft het Python raamwerk verschillende drivers en bibliotheken nodig. Obreshkov e.a. (2008) bespreekt hoe het ATLAS project te werk gaat bij het inpakken van alle nodige software. Het ATLAS project gebruikt CMT als configuratie manager. Met behulp van een configuratie bestand weten verscheidene tools hoe ze een pakket moeten afhandelen (Obreshkov e.a., 2008). Een gelijkaardige structuur is wenselijk voor het probleem. Net zoals Obreshkov e.a. (2008) zou er voor ieder bibliotheek en driver een pakket voorzien worden. Hierdoor wordt de herbruikbaarheid bevordert en kan voor ieder pakket apart behandeld worden. Rybkin (2012) spreekt ook over CMT als informatiebron voor het ophalen van meta-data. Aan de hand van deze data kan een Pacman pakket geproduceerd worden. Met behulp van een "Pacman fileïs geweten hoe de ingepakte software behandelt moet worden.

In voorbereiding op het academiejaar, was het mogelijk om een stage te lopen bij het bedrijf. Het doel van deze stage was het voorbereiden van de thesis. Tijdens deze voorbereiding, werd er vooral gezocht naar een goede technologie die gebruikt kon worden voor de packager. In wat volgt, gaat er een opsomming volgen van de onderzochte technologieën samen met hun voor- en nadelen.

2.1.1 WiX Toolset

Windows installer XML Toolset is een set van build tools waarmee Windows Installer packages gemaakt kunnen worden. Bron code wordt gecompileerd en vervolgens gelinkt om een executable te maken („WiX Toolset", 2016).

2.1.2 NSIS

2.1.3 Chocolatey

2.1.4 Qt Installer Framework

2.2 Database

Het aantal pakketten die gebruikt gaan worden, gaan alleen toenemen. Naast deze groei zullen ook het aantal gebruikers toenemen. Daarom werd er geopteerd om een databank te ontwerpen. In overleg met het bedrijf werd ervoor gekozen om MySQL te gebruiken als managementsysteem. Het ontwerp van de databank is terug te vinden in Figuur 2.1.



Figuur 2.1: Ontwerp van de databank

Hoofdstuk 3

Prototype

3.1 Uitwerking

3.2 Testen

Hoofdstuk 4

Eind product

4.1 Uitwerking

4.2 Testen

Hoofdstuk 5

Conclusie

Referenties

- Obreshkov, E., Albrand, S., Collot, J., Fulachier, J., Lambert, F., Adam-Bourdarios, C., . . . Goldfarb, S. (2008). Organization and management of {ATLAS} offline software releases. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 584(1), 244–251. doi:<http://dx.doi.org/10.1016/j.nima.2007.10.002>
- Rybkin, G. (2012). ATLAS software packaging. *Journal of Physics: Conference Series*, 396(5), 4.

Bibliografie

- Castleman, K. R., Schulze, M. A. & Wu, Q. (1998). Simplified Design of Steerable Pyramid Filters. In *Proc. IEEE ISCAS*.
- Granlund, G. & Knutsson, H. (1995). *Signal Processing for Computer Vision*. Kluwer Academic Publishers.
- Holmes, T., Bhattacharyya, S., Cooper, J., Hanzel, D. & Krishnamurti, V. (1995). Handbook of biological Confocal Microscopy (2nd ed.) In J. Pawley (Red.), (Hfdstk. Light Microscopic Images reconstructed by Maximum Likelihood deconvolution, pp. 389–402). Plenum Press, New York.
- Hossack, W. (WWW). Digital Image Analysis and Image Processing 1. <http://www.ph.ed.ac.uk/~wjh/teaching/dia/index.html>. University of Edinburgh, School of Physics.
- Plášek, J. & Reischig, J. (1998). Transmitted-light Microscopy for Biology: A Physicist's Point of View Part I. *Proceedings of the Royal Microscopical Society*, 33(2), 121–127.
- van der Voort, H. (1989). *Three dimensional Image Formation and Processing in Confocal Microscopy* (proefschrift, University of Amsterdam).
- WiX Toolset. (2016). <http://wixtoolset.org/>. [Online; geraadpleegd 3-08-2016].

Bijlage A

Een aanhangsel

sdfsffqsfsf

Bijlage B

Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel

Bijlage C

Poster

FACULTEIT INDUSTRIELE INGENIEURSWETENSCHAPPEN
TECHNOLOGIECAMPUS GENT
Gebroeders De Smetstraat 1
9000 GENT, België
tel. + 32 92 65 86 10
fax + 32 92 25 62 69
iiw.gent@kuleuven.be
www.iw.kuleuven.be



LID VAN **ASSOCIATIE
KU LEUVEN**