

Stageverslag

Python Framework Installer

PIETER-JAN ROBRECHT

Student Industrieel ingenieur ICT
01/08/2016 - 19/08/2016

Inhoudsopgave

1	Inleiding	1
2	Analyse van het probleem	1
3	Oplossingen	2
3.1	Mogelijkheden	3
3.1.1	WiX [6]	3
3.1.2	NSIS [4]	3
3.1.3	Chocolatey [2]	3
3.2	Architectuur	3

1 Inleiding

In het kader van mijn masterproef was het mogelijk om bij Televic Rail een stage te lopen tijdens de zomer. Gedurende deze stage heb ik mijn masterproef voorbereid en onderzoek gedaan naar de verschillende mogelijkheden die er zijn voor het uitvoeren van mijn opdracht. De opdracht die voltooid moet worden is de volgende:

Televic Rail heeft een Python test framework ontworpen. Dit framework werd oorspronkelijk gebruikt om gebruikt te worden op een volledig uitgeruste testtoren, maar het framework werd aangepast zodanig dat het onafhankelijk van de testtoren gebruikt kan worden. Aangezien het framework gebruik maakt van verschillende niet-standaard Python bibliotheken, is het installeren van het framework op een nieuw systeem een hele klus. Het doel is dan ook het maken van een installer en een updater waarmee dit proces kan worden vergemakkelijkt zodanig dat er zo min mogelijk interactie van de gebruiker moet zijn.

Tijdens de duur van de stage werden de verschillende implementatie manieren onderzocht en werd er een algemene architectuur voor de installer-updater uitgedacht. In wat volgt wordt er beschreven welke stappen er werden ondernomen om het probleem onder te verdelen in enkele logische stappen en hoe deze problemen dan werden opgelost.

2 Analyse van het probleem

Tijdens het bespreken van het probleem werd het al snel duidelijk dat er verschillende scenario's zijn waarin het framework gebruikt wordt. Het framework wordt als een standalone programma op een laptop of desktop maar het zal ook moeten draaien op verschillende testtorens. Voor de verschillende omgevingen zal iedere keer een andere configuratie nodig zijn maar ze wel beiden Windows gebruiken als besturingssysteem. Er zullen dus geen grote veranderingen moeten

worden aangebracht in het maken van de installer aangezien beide omgevingen overweg kunnen met een .exe bestand. Uiteraard moet er rekening worden gehouden met de toekomst. Het framework zal in de nabije toekomst worden aangepast zodanig dat het mogelijks op een Raspberry Pi kan draaien. Dit zou wel voor verschillende problemen kunnen zorgen aangezien een Raspberry Pi een linux distributie zal gebruiken als zijn besturingssysteem. Aangezien linux distributies geen gebruik maken van .exe files voor het installeren van programma's zal er een alternatief worden gezocht zodanig dat het framework ook kan geïnstaleerd kan worden op deze distributies. Aangezien de uitbreiding naar de Raspberry Pi een toekomst plan is, gaan we hier niet de focus leggen.

Tijdens het installeren van het framework moeten we rekening houden met het feit dat iedere computer een andere configuratie en andere drivers zal nodig hebben. Er zal een manier moeten worden uitgedacht waardoor we kunnen achterhalen welke devices zijn geconnecteerd en welke drivers deze exact nodig heeft. Het probleem van de drivers beperkt zich jammer genoeg niet enkel tot dit. Als er nieuwe hardware ontwikkelt wordt, zullen er ook verschillende nieuwe drivers nodig zijn. Voor de installer is het dan ook belangrijk dat deze nieuwe drivers snel kunnen worden toegevoegd en dat dit geen grote problemen levert. Nieuwe versies van het framework zelf zullen ook snel en eenvoudig moeten worden gedistribueerd.

De installer is best ook een programma dat zo eenvoudig mogelijk uit te breiden is aangezien het in de toekomst nog kan worden aangepast. De interface naar de gebruiker toe is best ook zo eenvoudig mogelijk zodanig dat er geen problemen kunnen ontstaan tijdens de installatie van het framework. Uiteraard gaan we er vanuit moeten gaan dat er af en toe een probleem zal ontstaan tijdens de installatie van enkele componenten. Eventuele updates zullen dan in een afgesloten omgeving moeten gebeuren en na het correct uitvoeren pas in het grote geheel geplaatst worden. Ook hiermee zullen we rekening moeten houden.

3 Oplossingen

Aangezien we een algemeen beeld hebben van wat we juist allemaal moeten voorzien, kunnen we vervolgens aan de slag met het zoeken naar een goede oplossing voor alle problemen. In wat hierna volgt worden enkele mogelijkheden overlopen die gebruikt kunnen worden om de installer te implementeren. Alle verschillende opties gaan overlopen worden en de verschillende voor- en nadelen zullen besproken worden¹.

¹Uiteraard is het mogelijk dat tijdens de duur van de masterproef nog mogelijke oplossingen worden gevonden. In sectie 3.1 op de volgende pagina worden enkel de mogelijke oplossingen besproken die tijdens de stage zijn gevonden.

3.1 Mogelijkheden

3.1.1 WiX [6]

Windows Installer XML Toolset is een toolset waarmee Windows Installer packages gemaakt kunnen worden vanuit XML code. WiX valt onder de Microsoft Reciprocal License [5]. Met de toolset is het mogelijk om een installer met als extensie .msi te maken. Hiernaast is het ook mogelijk om patches te schrijven en code van andere te includeren dankzij merge modules[7]. Uitgaande van de eerste indruk, is het goed mogelijk dat deze genoeg opties aanbiedt om een degelijke installer te maken. Hiernaast moet het ook mogelijk zijn om dankzij het patch systeem updates uit te voeren. Er zal wel moeten gecontroleerd worden of dit systeem volautomatisch is. Als dit niet zo is, dan zal er gekeken moeten worden hoe een updater kan worden geïmplementeerd zodanig dat er geen nieuwe code moet geschreven worden bij iedere update.

3.1.2 NSIS [4]

Nullsoft Scriptable Install System is een opensource systeem waarmee Windows installers kunnen gemaakt worden. De overhead die NSIS zou creëren zou klein zijn maar dit zal uiteraard moeten gecontroleerd worden. NSIS is script-based en biedt verschillende features aan die handig kunnen zijn voor deze toepassing [3]. Net zoals WiX zal de het script van NSIS een installer generen en vervolgens kan deze dan gebruikt worden voor de initiële installatie.

3.1.3 Chocolatey [2]

Chocolatey is een Powershell execution engine dat gebruikt maakt van de NuGet packaging infrastructuur. Het systeem is te vergelijken met de package manager apt-get van linux. Chocolatey zal alle executables, zips, ... die nodig zijn voor een stuk software encapsuleren zodanig dat alles in één package zit. Het packaging framework verstaat ook alle versioning en dependency requirements, iets dat zeker nodig is bij dit probleem [1]. Chocolatey biedt verschillende mogelijkheden zodanig dat installatie, upgrades en verwijdering van het programma in de handen van de programmeur liggen. Een mogelijk nadeel van deze methode is het gebruik van Powershell. De gebruikers zullen overweg moeten kunnen met Powershell en dit kan voor problemen zorgen.

Nu dat alle verschillende opties globaal zijn besproken, is het belangrijk dat deze ook worden uitgetest zodanig dat er een duidelijk beeld kan gevormd worden over.

3.2 Architectuur

Referenties

[1] Chocolatey Documentation. <https://chocolatey.org/docs>, 2016. [On-

line; geraadpleegd 3-08-2016].

- [2] Chocolatey Main page. <https://chocolatey.org/>, 2016. [Online; geraadpleegd 3-08-2016].
- [3] NSIS Features. <http://nsis.sourceforge.net/Features>, 2016. [Online; geraadpleegd 2-08-2016].
- [4] NSIS Main page. http://nsis.sourceforge.net/Main_Page, 2016. [Online; geraadpleegd 3-08-2016].
- [5] WiX toolset License. <http://wixtoolset.org/documentation/manual/v3/main/license.html>, 2016. [Online; geraadpleegd 3-08-2016].
- [6] WiX toolset main page. <http://wixtoolset.org/>, 2016. [Online; geraadpleegd 3-08-2016].
- [7] WiX toolset Merge modules. <https://www.firegiant.com/wix/tutorial/upgrades-and-modularization/mergers/>, 2016. [Online; geraadpleegd 3-08-2016].