



Hochschule für angewandte Wissenschaften Augsburg
Fakultät für Informatik

Exposé

Test-driven development mit Python

zur Erlangung des akademischen Grades
Bachelor of Science

Thema:	Test-driven development mit Python
Autor:	Maximilian Konter maximilian.konter@hs-augsburg.de MatNr. 951004
Version vom:	27. Januar 2019
1. Betreuer:	Dipl.-Inf. (FH), Dipl.-De Erich Seifert, MA
2. BetreuerIn:	Prof. Dr. X

Inhaltsverzeichnis

Abkürzungsverzeichnis	3
Glossar	3
1 Ausgangslage	4
2 Problembeschreibung	4
3 Fragestellung	4
4 Zielsetzung	5
5 Theoretische Grundlage	5
6 Konzept	6
6.1 Hypothese	6
6.2 Methodik	6
7 Motivation	6
Literaturverzeichnis	8

Glossar

Pythonista Jemand der Python als seine favoritisierte Sprache verwendet. 6

1 Ausgangslage

TDD wird in der heutigen Softwareentwicklung immer verbreiteter und beliebter. Die Ansprüche an Software sind in den letzten Jahren immer weiter gestiegen. Dies liegt vor allem an der Reichweite, die Software heute hat. So besitzen im Jahr 2018 bereits circa 66% aller Menschen ein Smartphone [Sch17], im Arbeitsleben ist ein PC meist gar nicht mehr weg zu denken. Doch mit den steigenden Nutzerzahlen steigen auch die Anforderungen, die die Nutzer an die Software stellen. Somit wird die Anzahl der gefundenen Bugs dementsprechend größer.

2 Problembeschreibung

Im weltweiten Markt gibt es viele große Unternehmen, die gegenseitig um die Nutzer kämpfen. Selbstverständlich präferieren die Nutzer denjenigen Anbieter, welcher die bessere Software bietet. Dies kann sich heute jedoch stetig ändern. Mit der steigenden Anzahl an Bugs, die gefunden werden, steigt auch die Anzahl der Nutzer, die von diesen Bugs betroffen sind. Diese Bugs sollen natürlich schnellstmöglich gefixt werden um so zu verhindern, dass die Nutzer die Software wechseln.

So schwer es ist seine Nutzer zu halten, umso schwerer ist es zum Start einer Software Nutzer zu akquirieren. Es gibt bereits andere Software, die ähnliche Services anbieten. So ist es noch schwerer dem Markt beizutreten. Die Anforderungen sind durch die anderen Firmen höher als die Anforderungen an ein neues Produkt und alte Fehler, die in anderer Software schon gelöst wurden, sollten möglichst nicht auftauchen.

Für Unternehmen sind diese Anforderungen meist schwer zu meistern weshalb Software meist mit Fehlern released wird, um diese dann von den Nutzern aufdecken zu lassen und zu fixen.

3 Fragestellung

Wie schaffen es Firmen ihre Entwicklung so zu optimieren, dass ihre Software zu jeder Zeit optimal funktioniert und alte Fehler und Bugs nicht wieder auftauchen? Am Besten sollte dies von Anfang an im Projekt der Fall sein und sich bis zur Wartung der Software durchziehen.

oder/und

Warum ist TDD die beste Lösung seine Software zu entwickeln und wie würde so eine Entwicklung ablaufen?

4 Zielsetzung

Das Ziel dieser Arbeit ist es, Möglichkeiten aufzuzeigen TDD in der Entwicklung mit Python zu betreiben und zusätzlich dazu auch Möglichkeiten aufzuzeigen, wie alle Schritte automatisiert werden können.

Dies soll anhand eines Dummy Projektes geschehen, welches zusammen mit der Arbeit geschrieben wird.

Zusätzlich soll jeder Schritt auch in ein bereits bestehendes Projekt integriert werden können. Dazu werden in jedem Schritt Möglichkeiten geboten, sodass auch bei einer möglichen Nicht-Betreibung von TDD, dennoch von den Automatisierungsschritten profitiert werden kann.

Zu jedem Schritt, bei dem es möglich ist, sollen auch Möglichkeiten dargestellt werden Python für TDD in einem Projekt mit einer anderen Programmiersprache zu verwenden.

Am Ende soll der Leser einen klaren Überblick haben, was TDD ist, wie er es in seinem aktuellen oder nächsten Projekt anwenden kann und welche Vorteile TDD dem Nutzer bieten.

5 Theoretische Grundlage

Hier aufgeführt findet sich einiges an Literatur, welche relevant für diese Arbeit sein könnte.

- 5-jährige Studie von IBM aus dem Jahr 2007 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.6319&rep=rep1&type=pdf>
- Diskussion zu einem Buch über Empirische Softwareentwicklung (Paywall) <https://www.infoq.com/news/2009/03/TDD-Improves-Quality>
- Studie über den Gebrauch von TDD und dem daraus resultierenden Design und Testverbesserungen <https://arxiv.org/pdf/1711.05082.pdf>

- TDD vs Test-Last https://www.researchgate.net/publication/315743099_An_Experimental_Evaluation_of_Test_Driven_Development_vs_Test-Last_Development_with_Industry_Professionals
- TDD vs nicht TDD auf Github <https://peerj.com/preprints/1920/>
- Beispiel für TDD mit Python <https://medium.freecodecamp.org/learning-to-test-with-python-997ace2d8abe>

6 Konzept

6.1 Hypothese

TDD ist einfach zu betreiben und sorgt dafür, dass die Software jederzeit in einem vorzeigbaren Zustand ist und keine Fehler, die vermeidbar wären oder schon einmal da waren, die Produktion behindern.

6.2 Methodik

Anhand des Dummy Projekts wird gezeigt, wie es möglich ist die Software immer stabil zu halten und im späteren Verlauf automatisch Fixes in die produktive Software zu deployen.

7 Motivation

Da ich selbst ein Pythonista bin, war es für mich naheliegend meine Bachelorarbeit dieser Sprache zu widmen. Durch mein Praxissemester und dem daraus resultierenden Werkstudentenjob, in dem ich seit eineinhalb Jahren an der Test- und Buildautomatisierung arbeite, wurde mein Interesse in die automatisierte Ausführung von Tests geweckt. Auch mein Bedürfnis, Tests für meine Programme zu schreiben, ist enorm gewachsen, da ich festgestellt habe, wie gut das Gefühl ist, wenn man etwas im Code ändert und alle Tests danach noch zu 100% durchlaufen.

Seitdem ist Testen ein Teil meiner Entwicklung und auch meines Denkens. TDD selbst habe ich jedoch noch nicht ausprobiert, aber der Reiz für mich ist hoch, da ich denke, dass Testen von Anfang an sich nur positiv auf die Software auswirken kann.

Auch mein Interesse für die Automatisierung möchte ich in dieser Bachelorarbeit widerspiegeln, da dies den Entwicklungsprozess erheblich erleichtert und schneller gestaltet.

Literaturverzeichnis

- [Sch17] SCHOBELT, Frauke: Weltweite Smartphone-Verbreitung steigt 2018 auf 66 Prozent. (2017). – Online erhältlich unter https://www.wuv.de/digital/weltweite_smartphone_verbreitung_steigt_2018_auf_66_prozent; abgerufen am 25. Januar 2019.