Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Informatikdienste

ETH Zürich
CH-8092 Zürich

Dr. Benno Luthiger
WEC E 17
Weinbergstrasse 11
Tel. +41 44 632 57 65
benno.luthiger@id.ethz.ch
www.foss.ethz.ch

Software Services (SWS)
Informatikdienste
ETH Zürich

Zürich, 10. Sept. 2012 BL

Europython 2012, Konferenz-Bericht

Die Europython 2012 in Florenz bot den Konferenzteilnehmern an den fünf Konferenztagen in fünf parallel geführten Tracks über 120 Vorträge. Mehrheitlich waren die Vorträge Python-spezifisch. Es gab aber auch etliche Präsentationen über Technologien aus verwandten Gebieten (z.B. A. Willmer über „Full Text Search for Trac with Apache Solr"), über die Open-Source-Thematik allgemein (z.B. E. Andrei „How Brazil is building a Digital Nation with Open Source and Python") sowie solchen, welche den künstlerischen Bereich berührten (z.B. N. Tollervey „Music theory, genetic Algorithms and Python").

Die in der folgenden Liste aufgeführten Europython-Präsentationen haben eine gewisse Relevanz für die Software-Entwicklung an der ETH Zürich:

Alex Martelli „Permission or Forgiveness?":
Von Grace Murray Hopper (amerikanische Computer-Pionierin und Konter-Admiralin der Marine) stammt das Motto "Es ist besser, um Verzeihung zu fragen als um Erlaubnis". Gemäss Hopper hat das mittlere Kader einer grossen Organisation eine natürliche Risiko-Abneigung. Aus diesem Grund handelt es übervorsichtig, wenn von Mitarbeitern innovative Ideen an dieses herangetragen werden. Damit eine solche Organisation ihre Innovationsfähigkeit nicht verliert, ist es wichtig, dass die Mitarbeiter auf der untersten Stufe Ideen ohne Nachfrage (und damit ohne Erlaubnis) umsetzen, sich aber darauf vorbereiten, bei Bedarf um Verzeihung bitten zu müssen.
Dieses Prinzip kann auf das Programmieren übertragen werden. Python hat ein leichtgewichtiges Exception-Handling. In Python ist es generell besser, eine Aktion auszuführen und allfällige Fehler aufzufangen als vorgängig die Bedingungen zu überprüfen.
Optimistic locking ist eine Anwendung dieses Prinzips im Persistenzbereich.
Generell sind Anwendungen, welche *forgiveness* umsetzen, erfolgreicher als Anwendungen aus dem gleichen Bereich, welche das *permission*-Prinzip implementieren. Z.B. SVN oder CVS (*forgiveness*) gegenüber VSS, ClearCase (*permission*), Ethernet (*forgiveness*) gegenüber Tokenring (*permission*), iterative-agil (*forgiveness*) gegenüber Wasserfall (*permission*).
Es gibt allerdings einige Themenbereiche, bei welchen das Prinzip „Erlaubnis" prioritär ist: Die Thematik *Privacy/Security* muss prioritär behandelt werden. Auch Code- und Design-Reviews dürfen nicht im Nachhinein erfolgen.

Pádraig Brady „OpenStack overview – operation details of a large Python project":
OpenStack ist ein Cloud-Service (IaaS) wie Amazon WebService, aber vollständig Open Source. OpenStack wird von folgenden Organisationen und Projekten verwendet: NASA, CERN (HEPIX), ISI.edu, Wikimedia, Rackspace. Innerhalb des OpenStack-Projekts wird OpenShift als PaaS (Platform as a

service) entwickelt. OpenShift läuft auf OpenStack.
OpenStack könnte eventuell für die ID-SD interessant sein.

URL: https://ep2012.europython.eu/media/conference/slides/openstack-overview-operational-details-large-python-project.pdf

Alexandros Kanterakis „PyPedia: a Python development environment on a Wiki":
Das Ziel des PyPedia-Projekts ist es, eine Sammlung von ausführbaren, dokumentierten und verifizierten Python-Implemetierungen von allgemeinen Computer- und Berechnungsproblemen zu erzeugen. Nutzer von PyPedia-Beiträgen sollen mit konkreten technischen Problemen die PyPedia-Seiten abfragen können und bei Treffern mit geringstem Aufwand austesten können, ob die angebotene Implementierung für die Problemlösung im konkreten Projekt tauglich ist.
Jeder PyPedia-Beitrag besteht aus Python-Quellcode, aus Unit-Tests und aus Dokumentation. Der Benutzer kann den Quellcode herunterladen, um ihn lokal laufen zu lassen. Die PyPedia-Schnittstellen erlauben es aber auch, dass die im PyPedia-Beitrag definierten Funktionen direkt über das Internet in eine lokale Python-Applikation importiert werden können, so dass der PyPedia-Code auch ohne Download des Quellcodes ausgeführt werden kann.
Wichtig für das PyPedia-Projekt ist nicht nur die technisch gelungene Umsetzung der Projektvision, sondern auch, dass eine Community von Beitragsleistern (ähnlich wie bei der Wikipedia) gebildet werden kann. Das PyPedia-Projekt plant entsprechend Präsentationen bei der Bioinformatics Open Source Conference 2012 (BOSC) in Los Angeles sowie der EuroSciPy in Brüssel. Möglicherweise könnten auch Wissenschaftler an der ETH an PyPedia-Projekt interessiert sein.

URL: http://www.slideshare.net/jandot/a-kanterakis-pypedia-a-python-crowdsourcing-development-environment-for-bioinformatics-and-computational-biology

Lynn Root „Increasing women engagement in the Python community":
In den USA wurden an verschiedenen Orten PyLadies-Gruppen gebildet mit dem Ziel, den Frauenanteil in der Python-Community zu erhöhen. Solche Gruppen bieten Python-Neueinsteigerinnen einen sicheren Raum, wo sie ohne die Angst, sich als Newbie zu outen, Fragen stellen können. Weiter intensivieren solche Gruppen den Erfahrungsaustausch unter Frauen und bieten eine Willkommens-Atmosphäre für Frauen allgemein.

Alan Franzoni „Language alone won't pay your bills":
Python ist eine elegante Programmiersprache, welche es erlaubt, produktiven Software-Code in kurzer Zeit zu schreiben. In einem grösseren Software-Projekt-Kontext geht es aber nicht nur darum, Code zu entwickeln, sondern funktionierenden Code beim Kunden zu installieren und dort zu warten. In einem grösseren Software-Projekt-Kontext schneidet Python nicht optimal ab.
Das Paketieren in Python lässt viele Freiheiten. Das führt dazu, dass die Installation eines Projekts mit vielen Abhängigkeiten häufig zu unerwarteten Ergebnissen führt. Java (mit Maven) führt zu deutlich absehbareren Zuständen.
Code-Reuse in Java ist einfacher als in Python.
Die Tool-Integration (Refactoring, Debugger, Continuous Integration etc.) in der Java-Welt ist besser als in Python.

URL: https://ep2012.europython.eu/media/conference/slides/language-alone-wont-pay-your-bills.pdf

Benno Luthiger, Sept. 2012