

# Happy Birthday Zope

## 10 Jahre Zope – ein Grund zum Feiern

Andreas Jung | [www.zopyx.de](http://www.zopyx.de) | [info@zopyx.com](mailto:info@zopyx.com)

### 10 Jahre Zope – ein Grund zum Rückblick

Historisch gesehen spielte Zope immer eine herausragende Rolle im Bereich der professionellen Web-Frameworks Seite an Seite mit Frameworks wie J2EE oder Coldfusion. Alleinstellungsmerkmale wie eine eingebaute Suchmaschine, mehrere Template-Sprachen und die Unterstützung für *through-the-web*-Entwicklung verschafften Zope einen besonderen Platz unter den Webframeworks. Insbesondere war Zope in der Python-Welt bis 2003 die einzige brauchbare Lösung zur Erstellung großer Web-Lösungen und Portale. Das Erscheinen von *Ruby* und *Ruby-on-Rails* (ROR) um das Jahr 2003 war von essentieller Bedeutung für alle Web-Frameworks. Zope war nie ein einfaches Framework (bedingt durch die komplexe Architektur, unzureichende und unstrukturierte Dokumentation und eine steile Lernkurve). ROR zeigte auf beeindruckende Weise, wie Webapplikationen von Nicht-Webentwicklern mit geringem Aufwand realisiert werden konnten. Die konsequente Umsetzung der Prinzipien *Don't repeat yourself* (DRY) und *Configuration through convention* vereinfachten die Realisierung von Webprojekten radikal. Mit dem Erscheinen von ROR wurde es auch unübersichtlich in der Python-Welt. Python-basierte Web-Frameworks sprossen wie Pilze aus dem Boden („*There are more Python web frameworks than Python keywords*“). Zope sah sich plötzlich in Konkurrenz zu Projekten wie *Django*, *Turbogears* oder *Pylons*. Viele Python-Entwickler gaben in der Folge diesen Projekten den Vorzug oder wendeten sich von Zope

*In diesem Jahr feierte der Applikationsserver Zope seinen zehnten Geburtstag. Dies ist fast schon ein biblisches Alter, wenn man bedenkt, dass das World Wide Web nur ein paar Jahre älter ist. Zieht man noch Zopes Vorgänger Principia und Bobo in die historische Betrachtung ein, so reicht der Ursprung von Zope zurück bis zu den Ursprüngen des WWW. Dies gilt nur für eine Handvoll Frameworks, die auch noch heute in Gebrauch sind.*

ab. Im Vergleich zu Zope kommt man mit diesen Frameworks oft einfacher zum Ziel – auch wenn deren Funktionalität bis heute nicht an das hererreicht, was Zope „out of the box“ zur Verfügung stellt.

### 10 Jahre Zope – Zope in den Wechseljahren

Die Zeit zwischen 2004 und 2006 war keine einfache Zeit für Zope. Einerseits bekam Zope in dieser Zeit massive Konkurrenz durch die anderen Python Webframeworks – zum anderen hat sich die Zope Community mit der Entwicklung von Zope 3 selbst Konkurrenz im eigenen Lager geschaffen. Zope 3 führte eine komponentenbasierte Architektur ein, wobei die einzelnen Komponenten über ZCML (einen XML-Dialekt zur Konfiguration der Komponenten) verbunden wurden. Die Einführung von Zope 3 bedeutete einen Paradigmenwechsel. Weg von dem monolithischen Applikationsserver (mit sehr viel impliziter Magie unter der Haube, z.B. Acquisition) hin zu einem System aus wohldefinierten Komponenten, deren Interaktion explizit konfiguriert werden musste. Die Koexistenz von Zope 2 und Zope 3 führte zu diversen Problemen. Einerseits impliziert die Benennung, dass Zope 3 der Nachfolger von Zope 2 sei (was aber aufgrund der unterschiedlichen Architekturen und der Inkompatibilität beider Systeme nicht sein kann). Zum anderen hatte die Zope-Community keine Vision bzgl. der weiteren Entwicklungen von Zope 2 und Zope 3 (weitere Parallelentwicklung bei-

der Zope-Versionen vs. Verschmelzung beider Technologien vs. Aufgabe der Zope 2-Entwicklung für Zope 3).

## 10 Jahre Zope – ein Blick in die nahe und ferne Zukunft

Diese Zope-Identitätskrise scheint aktuell überwunden zu sein. Jim Fulton, der Erfinder von Zope, präsentierte 2006 in seinem „Two visions“-Posting seine Idee davon, dass Zope 2 und Zope 3 in der Zukunft sich technologisch soweit angleichen würden und irgendwann verschmelzen könnten. Tres Seaver – der Vater von CMF – erläuterte 2007 im Rahmen seines Vortrags „What’s behind door #3“<sup>1</sup> die wesentlichen Thesen, die aktuell von der Zope-Entwicklergemeinschaft als Ziele für die Zukunft akzeptiert wurden:

- Umbenennung von „Zope 3“ in „Zope component library for web applications“,
- wir reden von „Zope 2“ als dem „Zope application server“,
- Zope 2 wird als Applikationsserver weiter gepflegt,
- die Zope 3-Komponenten werden als Komponenten unabhängig weitergepflegt und nach und nach in den Zope 2 Applikationsserver eingepflegt.

1 <http://tinyurl.com/2ehatn>

2 <http://tinyurl.com/3egfid>

Die Integration der Zope 3-Komponenten in den Zope 2-Applikationsserver geht langsam aber stetig voran. Priorität 1 ist Rückwärtskompatibilität. Die große Anzahl an Zope 2-Installationen muss bei jeder Änderung oder Erweiterung des Zope (2)-Kerns berücksichtigt werden. Neben der traditionellen Zope 2/3-Welt hat sich in den letzten beiden Jahren auch das Zope-basierte Framework GROK<sup>3</sup> etabliert. GROK ist die Antwort der Zope-Community auf die anderen pythonbasierten Webframeworks. Es ist sehr einfach zu erlernen und folgt den Ruby on rails-Paradigmen DRY und Configuration through convention. GROK konnte durch seine Einfachheit und Eleganz viele neue Entwickler gewinnen. Trotz der Einfachheit hat man vollen Zugriff auf alle Zope-Komponenten und kann mächtige Web-Applikationen auf Basis von GROK realisieren. Aus diesen Gründen soll GROK in der Zukunft als das Framework für den Einstieg in die Zope-Welt propagiert werden.

## 10 Jahre Zope – 5 Jahre Plone

Eng mit der Geschichte von Zope ist auch die Geschichte von Plone verbunden, welches ebenfalls bereits mehr als fünf Jahre als Content-Management-System auf Basis von Zope verfügbar

3 <http://grok.zope.org>

ist. Von den Plone-Erfindern (Alexander Limi und Alan Runyan) anfangs nur als schönerer Skin für CMF konzipiert, hat sich Plone in den letzten Jahren als ernsthaftes CMS etabliert und braucht den Vergleich zu anderen Content-Management-Systemen – kommerziell und Open Source – nicht zu scheuen. Die einfache Installierbarkeit, die einfache Erlernbarkeit und die Funktionalitäten, die Plone out of the box mitbringt prädestinieren Plone u. a. für den Einsatz in nicht-technischen Bereichen (NGO, non-Profit Organisationen, Bildungsbereich). Die Plone-Community hat in den letzten Jahren eine wichtige Rolle bei der technischen Weiterentwicklung im Zope-Bereich gehabt. Der gezielte Aufbau einer Entwickler- und Benutzercommunity hat entscheidend dazu beigetragen, dass das Zope-Projekt weiter vorangekommen ist.

## 10 Jahre Zope – voll unter Dampf

Zope befindet sich wieder in der richtigen Spur. Die Weichen für den Zope-Applikationsserver und die Zope-Komponenten sind gestellt. Mit GROK verfügt Zope über eine starke Lokomotive, die hilft, die Zope-Community in die Zukunft zu schieben. Zope mit all seinen Teilprojekten setzte und setzt technologische Maßstäbe in der Python-Welt. Der Zope-Zug rollt zuverlässig weiter.