

# Workflow Modellierung mit Roundup

Dr. Ralf Schlatterbeck

Email: [rsc@runtux.com](mailto:rsc@runtux.com)  
Tel. +43/650/621 40 17

## Inhalt

Abbildung von Workflows	3
Roundup: Issue Tracking	4
Roundup Eigenschaften	5
Rollen und Rechtevergabe	7
Roundup Hyperdb	8
Trigger: Detector Interface	9
Beliebige Schnittstellen über Python-API	10
Fazit: Roundup + Workflow	11

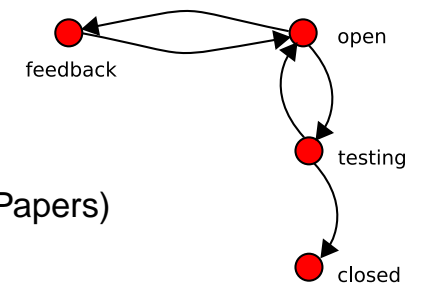
## Abbildung von Workflows

- Prozessdefinition ist einfach möglich:  
wer, wann, wie, was
- Rapid Prototyping
- Schrittweise Verfeinerung
- Flexible Rollendefinition
- Flexible Rechtevergabe
- Einfache Definition von Prozesseinschränkungen

## Roundup: Issue Tracking

Standard Anwendung „Issue Tracking“  
Nr. 1 Google Hit für „Issue Tracker“

- Trouble-Tickets
- Software Bug-Tracking
- IT-Helpdesk
- Konferenz Einreichungen (Papers)
- Sales Opportunities
- ...



## Roundup Eigenschaften

- Läuft überall wo Python läuft
- Web, Email, Kommandozeilen Schnittstelle
- Web-Interface als CGI, eigener Web-Server oder eingebettet in Zope
- Email: Mailbox, IMAP, POP, sendmail pipe
- Eigene Mini-Mailingliste für jedes „Issue“

## Roundup Eigenschaften

- Anwender-Dokumentation (Issue Tracker)
- Entwickler-Dokumentation
- Einfache Administration, Installation
- Python-API (Programmierschnittstelle)
  - Statistiken, Reports, Erinnerungs-Email, ...
- Import/Export Format und API
  - Datensicherung
  - Migration von anderen Systemen

## Rollen und Rechtevergabe

- Zuordnung von Rollen zu Benutzern
- Rechtevergabe nach Rollen auf Tabellen- oder Spaltenebene
- Berechnete Rechte möglich
- Web: Log-In für jeden Benutzer
- Email: Authorisierung über Email Adresse
- Optional Web- oder Email Registrierung
- Mehrere Email-Adressen pro Benutzer

## Roundup Hyperdb

- Verschiedene Datenbank Backends (z. B. DBM, SQLite, MySQL, PostgreSQL)
- Einfache Schema-Definition in Python
  - Backend-unabhängig
- Änderung Datenbank-Schema im Betrieb
- Automatische Attribute (creator, creation, ...)
- Änderungshistorie (Journal)
- Indizierung und Volltextsuche

## Trigger: Detector Interface

- Detector: Auditor oder Reactor
- Zugriff auf aktuellen Datensatz
- Zugriff auf ganze Datenbank möglich
- Auditor: „Veto“ auf Änderungen
  - Anwendungsspezifische Einschränkungen
  - Automatische Updates anderer Datensätze
- Reactor: Aktionen nach Commit in Datenbank
  - Email-Versenden

## Beliebige Schnittstellen über Python-API

- Schnittstelle zu OpenOffice.org (OOoPy)
  - Generierung von OOo Dokumenten
    - Auswertungen
    - Reports
    - Inhalte
- Erweiterungen z. B. Projektplanung Gantt Charts (PyGantt)

## Fazit: Roundup + Workflow

- Vorhandene Eigenschaften für die Web-basierte **Abbildung von Workflows**
  - Rapid Prototyping, Änderung von Schemata
  - Rollendefinition und Rechtevergabe eingebaut
  - Prozesseinschränkungen mit Detector
- Web Anwendungs-Framework, nicht exklusiv für Issue Tracking
- Konzentration auf Anforderungsanalyse und Prozessoptimierung