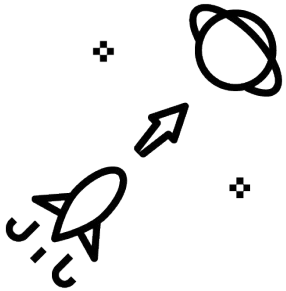
Agile Ops und modernes Puppet

Martin Alfke
<ma@betadots.de>



Martin Alfke
CEO/Consultant/Trainer
betadots GmbH - Berlin

- Platform Automation und Engineering
- Consulting und Training
- Agile Methoden, Scrum
- GIT, Foreman und Puppet Training
- tuxmea (Twitter, GitHub, Slack)

Agenda

Agile Ops

- Hintergründe Agile Entwicklung
- Problematik im Ops Umfeld
- ScrumBan Board

modernes Puppet

- Was ist neu in Puppet 8?
- Wie bekomme ich meinen Code für für das Neue?
- Neue Features
- Neue Anwendungen

Agenda

Workshop Umgebung

- https://puppet.slac.betadots.training (Puppet Enterprise)
- http://puppet.slac.betadots.training:3000 (Hiera Data Manager)
- http://puppet.slac.betadots.training:8089 (Puppet Board)
- http://login.slac.betadots.training (Guacamole Login)
- http://gitlab.slac.betadots.training (GitLab)

Agenda

Workshop Umgebung

Login für Puppet, Login, Gitlab:

Username: `student<N>` N=1-25

Password: `betastud<N>`

Login für HDM:

username: `user@domain.tld`

passwort: `1234567890`

Agenda

Workshop Login 1:

- http://login.slac.betadots.training
- Username: `student<N>` N=1-25
- Password: `betastud<N>`
- Shell öffnen, `ssh-keygen`, cat pub key

Agenda

Workshop Login 2:

http://gitlab.slac.betadots.training

Username: `student<N>` N=1-25

Password: `betastud<N>`

Preferences -> SSH Keys -> Pub Key pasten

Agenda

Workshop Login 3:

http://login.slac.betadots.training

Username: `student<N>` N=1-25

Password: `betastud<N>`

```
git clone git@gitlab:puppet/control-repo.git
```

Agile Ops

Agile Ops

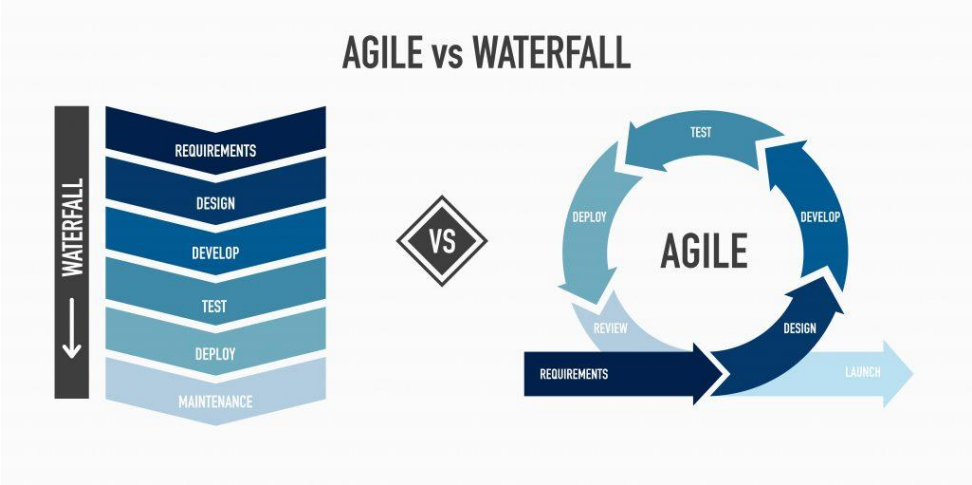
Probleme in der SW Entwicklung:

- Planung, Durchführung, Implementierung, Produktion - jeweils mehrere Monate
- Projekte werden nicht fertig
- Anforderungen ändern sich im Lebenszyklus des Projektes

Agile Ops

Lösung:

- agile Methoden
- Klein Schneiden von Projekten
- iterativer Entwicklungsprozess
- DevOps Prinzipien

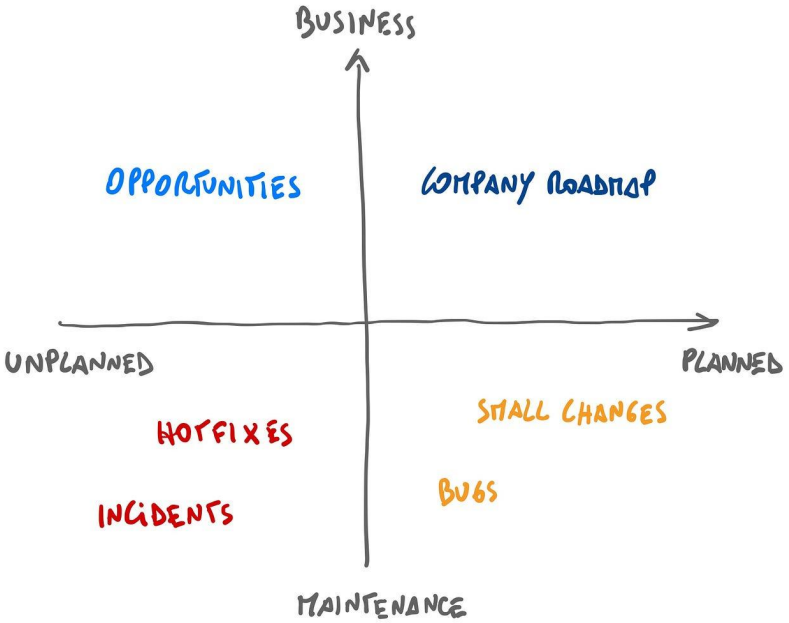


Irene Casucian - https://technologyadvice.com/blog/information-technology/what-is-agile-project-management/

Agile Ops

Aber:

- ungeplante Arbeiten
- Betrieb, Incidents, Probleme



Luca Rossi - https://refactoring.fm/p/the-four-types-of-work

Agile Ops

Team Größe?

- Agil bedeutet Kommunikation
- Mehr Menschen = mehr Kommunikations Overhead
- 6 Personen maximal

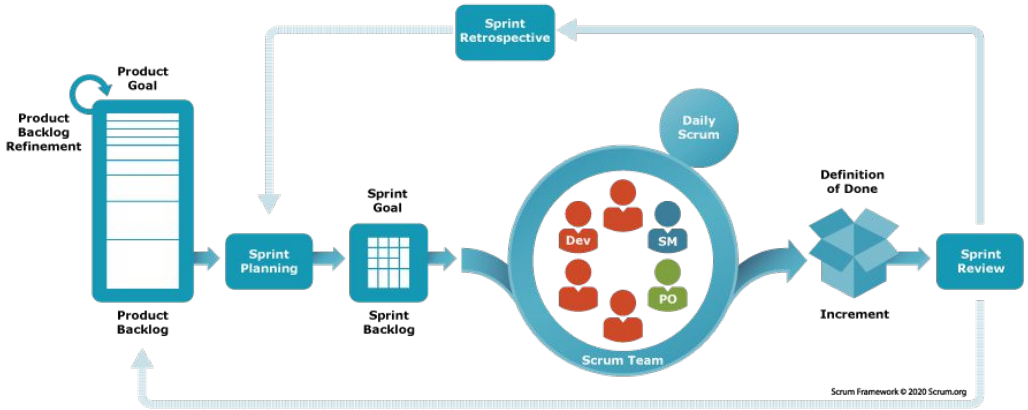
Agile Ops

SCRUM:

- Planung
- Durchführung (daily)
- Review
- Retrospektive

Geht super bei SW Entwicklung.

Wie wird ungeplantes eingebunden?



Scrum.org - https://www.scrum.org/resources/what-scrum-module

Agile Ops

Kanban:

- Swim Lanes
- Planung
- Durchführung
- Review
- Erledigt

Lanes können flexibel erweitert werden

Agile Ops

ScrumBan:

- Open - all anstehenden Arbeiten
- Planned - alle Arbeiten, die im Team abgesprochen sind
- Urgent/Incident - ungeplante Arbeit
- Work in Progress (WIP) - Limit 1 !
- Review - erledigte Arbeiten (vor Rollout)
- Done - erledigte Arbeiten (nach Rollout)

Agile Ops

1. Teams bilden
2. Aufgabe besprechen
3. Ungeplante Arbeit

Wer übernimmt ungeplante Arbeiten? Alle?

Agile Ops

Ihr wollte Projekte voranbringen und Dinge erledigen.

Wie soll das gehen, wenn man dauernd unterbrochen wird?

1h Analyse

Break

30 Minuten, um wieder an den Punkt zu kommen, wo man vorher war. 15 Minuten arbeiten

Break

...

Agile Ops

Schützt eure Projekte vor ungeplanter Arbeit.

Aber: Ignorieren geht nicht.

Vorschlag:

- Chief of the (day|week)

Der Co(D|W) ist Ansprechpartner für alle Team-Externen

NO C-LEVEL OVERRIDE!

Agile Ops

4. CoD bestimmen (im Daily)
5. einfach nur Arbeiten

Agile Ops

Planung, Aufgabe und Zuständigkeiten CoD:

- Tut Gutes und redet drüber - auch mit den Vorgesetzten und HR!
- Erklärt den CoD in der Firma, informiert, wie man mit euch arbeiten soll/kann/darf
- CoD ist Ansprechpartner für alle Team-Externen (egal welche Hierarchie-Ebene)
- Wenn der CoD nicht im Raum ist, ist er schon bei einem Team.
- Wenn der CoD nicht weiter weiss, kann er das Team aktivieren.

Agile Ops

Aufgaben und Zuständigkeiten des Team:

- Entfernt UDP Broadcast Probleme aus dem Raum
- Freundlich, aber bestimmt
- Sagt dem CoD, dass jemand da war (und wer)

Tip: Raumplan an der Tür mit CoD Pin (oder eigenen CoD Channel oder CoD im Title im Kommunikations Channel nennen)

Agile Ops

beladots

Trunk oder MR?

Je nach Anforderung.

Puppet:

- Control Repo: MR mit Branches - Branches werden Environments - Canary Nodes
- eigene Module: Tags im Trunk auf Basis von MR mit Branch - Puppetfile control-repo Branch Tracking

Scripte: Tags im Trunk - Commit ID testen auf Canary Nodes

Agile Ops

GIT muss sitzen!

- Branch, Merge und Rebase
- Cherry-pick und Squash
- Conflicts

Test Automatisierung (Pipelines) - CI/CD/CD

modernes Puppet

modernes Puppet

- Hiera v5
- Hiera basierte Node Klassifizierung
- Simple Puppet mit Hiera
- Puppet 8 Releases, Features und Deprecations
- Hiera Data Manager
- Puppet Operational Dashboard
- Puppet Server in Containern

Vorbereitung

Login: http://login.slac.betadots.training

Shell starten

```
git clone
git@gitlab.slac.betadots.training:puppet/control-repo.git

cd control-repo

git switch -c student<N>

sudo puppet config set --section agent environment
student<N>
```

Hiera v5

3 Layer:

- Global Data - `/etc/puppetlabs/puppet/hiera.yaml`
- Environment Data - `<environment>/hiera.yaml`
- Module Data - `<module>/hiera.yaml`

Jeder Layer kann andere Hierarchie Ebenen haben.

Empfehlung:

- keine global data, eigene Daten in Environment, im Modul OS spezifische Informationen

Hiera v5

Hiera YAML Backend

- basierend auf Facts (custom oder external)

Hiera lookup Verhalten

- entweder beim expliziten lookup Aufruf
- als Daten in Hiera ← Empfohlen

Hiera v5

Aufgabe: Analyse Hiera lookup Verhalten für den Key

```
infrastructure::base::packages::list
```

Hinzufügen eines Paketes zu der Liste:

- common.yaml: global
- Node yaml: nur für das eine System

Hiera v5

betadots

EYaml Integration

- public-private Key Paar
- Daten in GIT Verschlüsselt ablegen

Aufgabe: `eyaml encrypt --help`

Hiera basierte Node Klassifizierung

betadots

Lösung 1: Arrays

In YAML:

```
---

classes:

 -  'chrony'
 -  'profile::base::linux'
```

Hiera basierte Node Klassifizierung

Lösung 1: Arrays

In manifests/site.pp

```
lookup('classes', { 'value_type' => Array, 'default_value'
=> [] }).each |$c| {

  include $c

}
```

Hiera basierte Node Klassifizierung

Lösung 2: sortierte Arrays - Sortierung von global -> spezifisch

In manifests/site.pp

```
lookup('classes', { 'value_type' => Array, 'default_value'
=> [] }).reverse_each |$c| {

  include $c

}
```

Hiera basierte Node Klassifizierung

betadots

Lösung 3: Hashes

In Hiera:

```
---

classes_hash:

  'base': 'profile::base::linux'

  'time': 'chrony'
```

Hiera basierte Node Klassifizierung

betadots

Lösung 3: Hashes

In manifests.site.pp:

```
lookup('classes_hash', { 'value_type' => Hash, 'default_value' => {} }).each |$n, $c|
{
  if $c == '' {
    echo { "Class for identifier ${n} has been removed on node
$facts['networking']['fqdn']")
  } else {
    include $c
}
```

Hiera basierte Node Klassifizierung

Lösung 3: Hashes überschreiben oder entfernen

In Hiera:

```
---

classes_hash:

  'base': 'profile::base::windows'

  'time': ''
```

Hiera basierte Node Klassifizierung

betadots

Lösung 4: sortierte Hashes

In Hiera:

```
---

classes_hash:

  '00_base': 'profile::base::linux'

  '02_time': 'chrony'
```

Hiera basierte Node Klassifizierung

Lösung 4: sortierte Hashes - In manifests.site.pp:

```
$classes_hash = lookup('classes_hash', { 'value_type' => Hash,
'default_value' => {} })

$classes_hash.keys.sort.each |$key| {

  if $classes_hash[$key] == '' {

    echo { "Class for identifier ${key} has been removed on node
$facts['networking']['fqdn']")

  } else {

    include $classes_hash[$key]

}
```

Hiera basierte Node Klassifizierung

Lösung 4: Aufgabe

Auf Node Ebene den Identifier für `00_base_packages` auf leeren String setzen

```
git add; git commit; git push
```

Paket deinstallieren - `sudo dnf remove tig`

```
sudo puppet agent -t
```

Hiera basierte Node Klassifizierung

Lösung 4: Aufgabe

Auf Node Ebene den Identifier für `00_base_packages` komplett entfernen

```
git add; git commit; git push
```

```
sudo puppet agent -t
```

tig wird wieder installiert

Simple Puppet mit Hiera

stdlib::manage Klasse

Kein Puppet Coding - Hat Limitierungen

```
stdlib::manage::create_resources:
  'package':
    'nano':
      ensure: 'absent'
    'vim':
      ensure: 'present'
  'user':
    'monitoring':
      ensure: 'present'
```

Puppet 8

Releases

- Puppet 5 - Juli 2017 - letztes Release Oktober 2020 - EOL
- Puppet 6 - September 2018 - letztes Release August 2022 - EOL
- Puppet 7 - November 2020 - letztes Release April 2024 - Support bis Februar 2025
- Puppet 8 - April 2023 - letztes Release April 2024

Puppet 8

Releases

- Puppet 5 - Hiera v5
- Puppet 6 - Core Types/Provider in Modules, CA Management
- Puppet 7 - Performance
- Puppet 8 - Legacy facts, CA/Cert Refresh

Puppet 8

betadots

Deprecations

- Hiera v3
- Legacy Facts

Enhancements

- Strict mode (strict = error, strict_variable = true)
- Exclude unchanged resources im Report (exclude_unchanged_resources=true)
- Tägliches CRL und CA Refresh (crl_refresh_interval = 1d, ca_refresh_interval = 1d)
- Updates: Ruby 3.2 und OpenSSL 3.0
- Auto Renew Agent Cert (hostcert_renewal_interval = 30d)

Puppet 8

betadots

Legacy Facts

https://www.puppet.com/docs/pe/2023.6/upgrade_cautions#platform-upgrade-legacy-facts-upgrade-caution

```
cd site

pdk new module slac

cd slac

pdk new class slac
```

Legacy Facts

```
class slac (

) {

  notify { $fqdn: } # besser: echo anstelle von notify

}
```

Puppet 8

beladots

Legacy Facts

data/nodes/student<N>.yaml

```
---
classes_hash:
  99_slac: 'slac'
```

```
git add --all; git commit; git push origin student<N>
```

Legacy Facts

Fehler im Report !

Zurück zum Code:

```
pdk validate puppet

pdk validate puppet -a

git diff

git add --all; git commit; git push origin student<N>
```

Puppet 8

Ruby 3.2

Major Version Upgrade (von Puppet 7 zu Puppet 8)

File.exists? geht nicht mehr.

File.exist? ist richtig

Puppet 8

beladots

Client Cert Auto Renewal

Puppet Server 8.2 oder neuer

ca.conf Einstellung

Puppet Agent 8.2 oder neuer

puppet.conf Einstellung

Muss VOR dem Erstellen des CSR vorgenommen werden!

Puppet 8

Client Cert Auto Renewal /etc/puppetlabs/puppetserver/conf.d/ca.conf

```
certificate-authority: {
  allow-subject-alt-names: true
  allow-authorization-extensions: true
  enable-infra-crl: true
  ca-ttl: "5y"
  auto-renewal-cert-ttl: "90d"
  allow-auto-renewal: true
}
```

Puppet 8

Client Cert Auto Renewal

Agent:

/etc/puppetlabs/puppet.conf

```
hostcert_renewal_interval = 30d (default)
```

Client Cert Auto Renewal

```
puppet resource service puppet ensure=stopped
```

```
puppet config set --section agent hostcert_renewal_interval 5m
```

```
rm -fr /etc/puppetlabs/puppet/ssl/*
```

Warten bis die Zertifikate auf dem Puppet Server gelöscht wurden.

```
puppet agent -t
```

Warten bis die Zertifikate wieder signiert wurden.

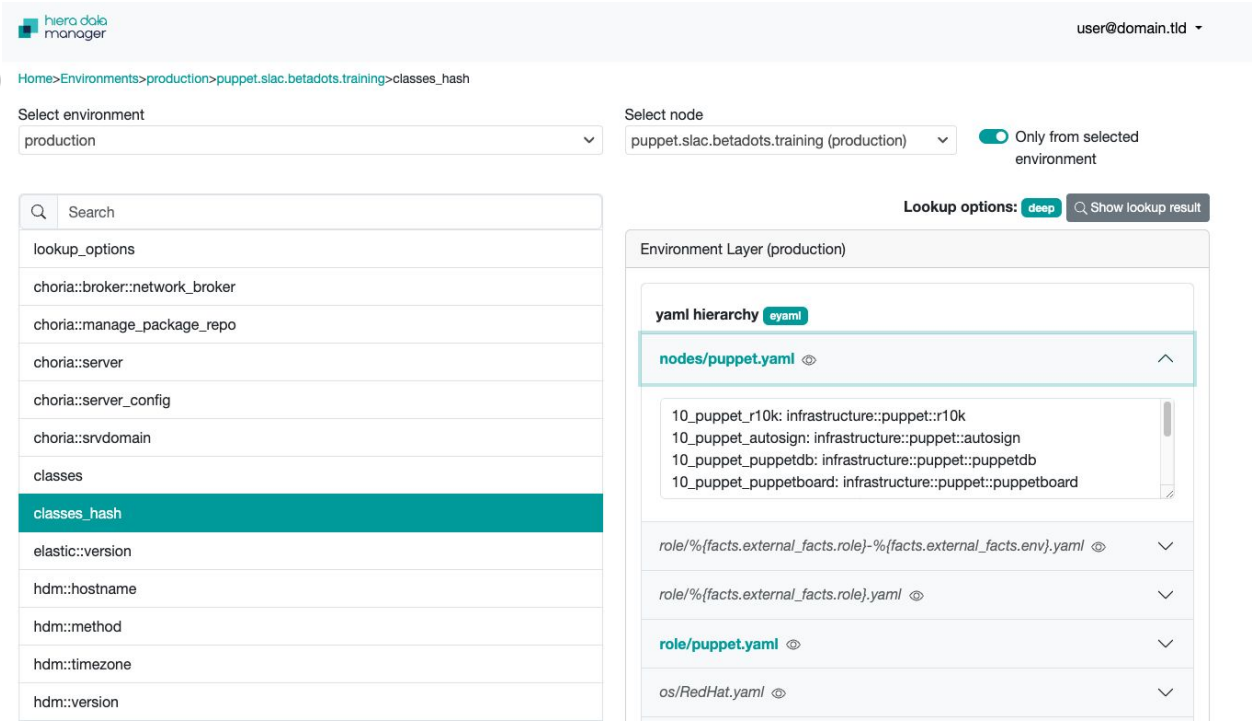
Hiera Data Manager

HDM (Hiera Data Manager) by betadots GmbH

Open Source Web UI zur Analyse von Hiera Daten

Ruby on Rails Application - braucht Ruby 3.3 !

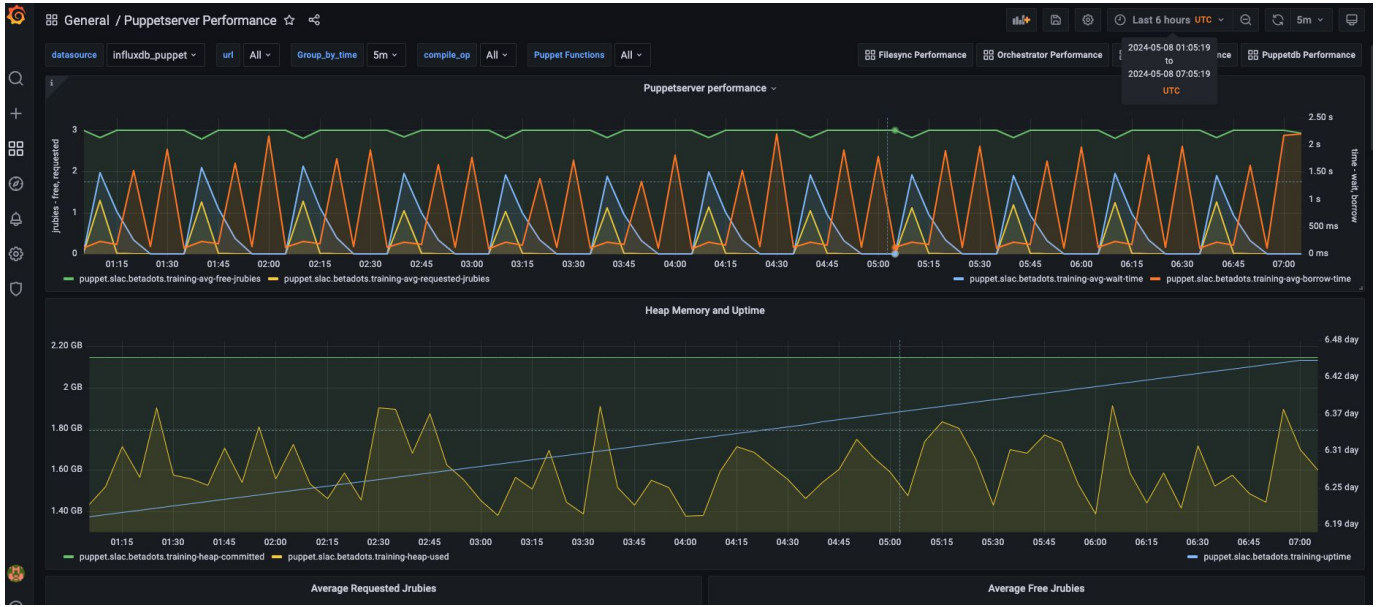
Wir nutzen die Container



Puppet Operational Dashboard

Telegraf -> InfluxDB -> Grafana

Telegraf pollt Metriken von den Puppet API's



Puppet in Containern

GitHub Projekt liegen in der Voxpupuli Organisation (Sicherstellen Open Source Lizenz)

Maintained durch betadots GmbH und Community.

https://github.com/voxpupuli/container-puppetserver/

https://github.com/voxpupuli/container-puppetdb

Beispiele für Docker Compose liegen im https://github.com/voxpupuli/crafty Repository

Puppet Testing

beladots

Unit Tests

- rspec-puppet
- puppet-lint

Acceptance Tests

- beaker oder litmus

Impact Analysis

- catalog-diff und catalog-diff viewer

Agile Ops und modernes Puppet

Martin Alfke
<ma@betadots.de>