

QS-Barcamp, 05.09.2020

Infrastructure as Code - Muss man nicht  
testen, Hauptsache es läuft

Sandra Parsick

@SandraParsick  
mail@sandra-parsick.de

# Wer bin ich?

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Trainings
- Workshops

 mail@sandra-parsick.de

 @SandraParsick

 xing.to/sparsick

 <https://www.sandra-parsick.de>

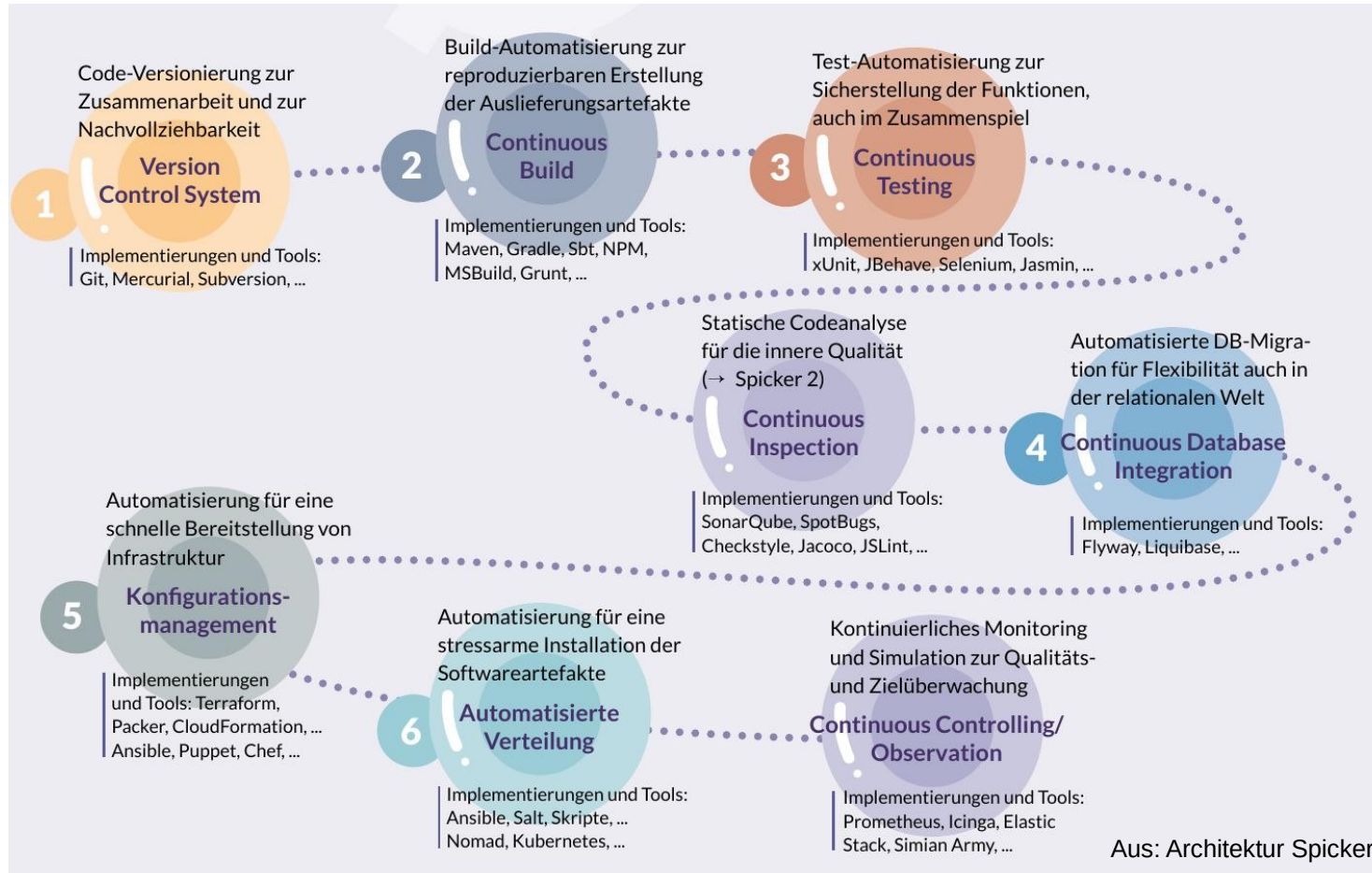


*„Du muss nicht jede Erfahrung selber machen, es kommt dir günstiger, wenn du aus Fehler, die Andere schon gemacht haben, lernst.“*

Mein Vater

Was hat das mit Infrastructure As Code zu tun?

# Continuous Delivery



# Infrastructure As Code



Automatisierung für eine schnelle Bereitstellung von Infrastruktur

5

## Konfigurationsmanagement

Implementierungen und Tools: Terraform, Packer, CloudFormation, ...  
Ansible, Puppet, Chef, ...

Automatisierung für eine stressarme Installation der Softwareartefakte

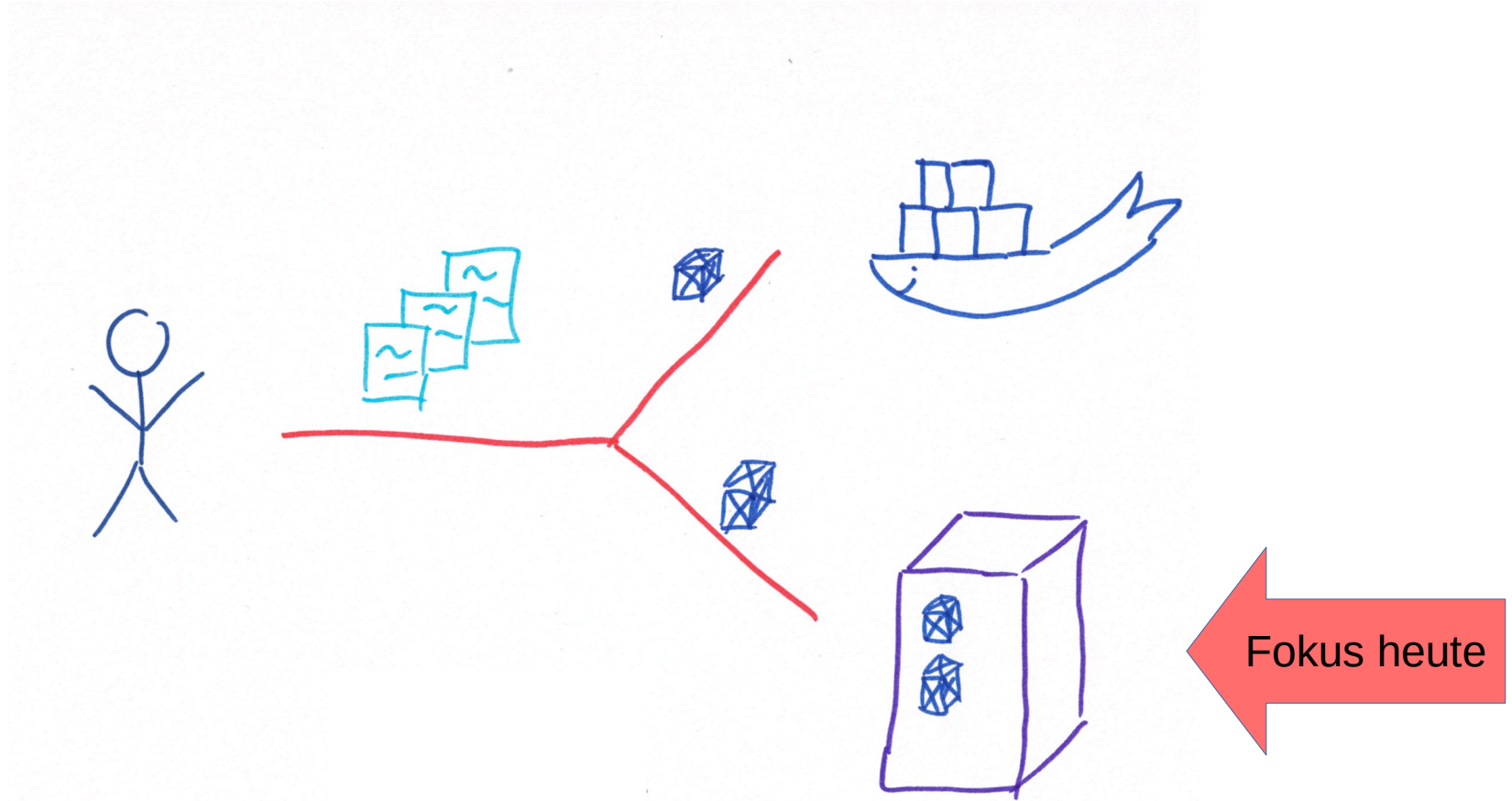
6

## Automatisierte Verteilung

Implementierungen und Tools: Ansible, Salt, Skripte, ...  
Nomad, Kubernetes, ...

In der klassischen IT:  
Ops-Tätigkeiten

# Infrastructure As Code (IaC)



# Beispiel für IaC Skripte: Ansible

```
1 - hosts: application_server
2   vars:
3     tomcat_version: 9.0.27
4     tomcat_base_name: apache-tomcat-{{ tomcat_version }}
5
6   tasks:
7     - name: install java
8       apt:
9         name: openjdk-8-jdk
10        state: present
11        become: yes
12        become_method: sudo
13
14    - name: Setup Group tomcat
15      group:
16        name: tomcat
17        become: yes
18        become_method: sudo
19
20    - name: Setup User tomcat
21      user:
22        name: tomcat
23        group: tomcat
24        become: yes
25        become_method: sudo
26
```



# Konsequenz für die Ops-Abteilung

- Ops brauchen ein Grundverständnis für das Programmieren (Algorithmen und Datenstrukturen)
- Ops stoßen auf ähnliche Probleme, die Devs auch hatten und für die sie Lösungen gefunden haben
- Ops brauchen auch einen **Entwicklungsprozess**

# Konsequenz

- Ops brauchen ein Grundverständnis für das Programmieren (Algorithmen und Datenstrukturen)
- Ops stoßen auf ähnliche Probleme, die Devs auch hatten und für die sie Lösungen gefunden haben
- Ops brauchen auch einen **Entwicklungsprozess**

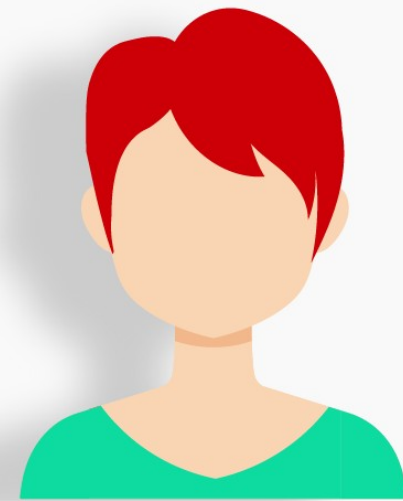
Wie sieht der Entwicklungsprozess  
bei den Devs aus,  
wenn sie nur mit ihren Code zu tun haben?



Was können die Ops aus diesem Prozess für sich übernehmen?



IaC Skripte liegen  
direkt auf dem Server.



Sie funktionieren  
gerade nicht.



Keine Ahnung,  
wer sie geändert hat

# Version Control System

Code-Versionierung zur  
Zusammenarbeit und zur  
Nachvollziehbarkeit

Version  
Control System

1

Implementierungen und Tools:  
Git, Mercurial, Subversion, ...

1

## Versionskontrollsystem

Ein Versionskontrollsystem (VCS) dient als Basis jeder CI- (und später CD-) Umgebung. Mit ihm kehrt ein Team bei Bedarf (z.B. im Fehlerfall) zuverlässig auf einen früheren, funktionierenden Stand zurück.

### BEST PRACTICES

- Das Projekt im VCS besitzt eine konsistente Ordnerstruktur.
- Source Code ist an einer zentralen/offiziellen Stelle abgelegt.
- Zum Source Code gehören: Build-Skripte, Programm-Code, Konfigurationen (zu Beginn, später im Konfigurationsmanagement), Deployment-Skripte, Datenbank-Skripte

### SO ARBEITEN EURE TEAMS

- Entwickler checken Code häufig ein.
- Teams entscheiden sich für einen VCS Workflow und passen ihre Arbeitsweise daran an.
- Jedes Team ist für seinen Quelltext gemeinsam verantwortlich (Collective Ownership).


















master ▾

[ansible-talk](#) / [ansible](#) /[Go to file](#)[Add file ▾](#)[Gitpod](#)**sparsick** fix ansible-lint issues

60891ce on 15 Oct 2019

**History**

..

	group_vars	add source samples, Vagrantfile for test enviroment and slides	5 years ago
	host_vars	add source samples, Vagrantfile for test enviroment and slides	5 years ago
	inventories	fix "bad characters in group names"	14 months ago
	roles	fix ansible-lint issues	10 months ago
	spec	adjust serverspec tets to systemd	2 years ago
	tests	add testinfra	2 years ago
	.rspec	add serverspec tests for setup's playbooks	5 years ago
	Jenkinsfile	update for cloud native night	2 years ago
	Rakefile	add serverspec tests for setup's playbooks	5 years ago
	demo-app-ansible-deploy-1.0-SNAPSHOT.war	add source samples, Vagrantfile for test enviroment and slides	5 years ago
	deploy-demo.yml	fix "bad characters in group names"	14 months ago
	setup-app-roles.yml	update tomcat 8 version	10 months ago
	setup-app.yml	update tomcat 8 version	10 months ago
	setup-db-roles.yml	fix "bad characters in group names"	14 months ago
	setup-db.yml	fix issues with update ansible 2.8	14 months ago



Commits on Oct 14, 2019

**update tomcat 8 version**



**sparsick** committed on 14 Oct 2019



[1f85617](#)



Commits on Jun 17, 2019

**fix "bad characters in group names"**



**sparsick** committed on 17 Jun 2019



[06fb6ce](#)



Commits on May 28, 2017

**replace init.d by systemd for tomcat setup**



**sparsick** committed on 28 May 2017



[c3c6952](#)



Commits on Nov 12, 2016

**update tomcat version to 8.5.8; update java version to 1.8**



**sparsick** committed on 12 Nov 2016



[367a959](#)



Commits on Feb 14, 2016

**adjust ansible code to version 2.0.0**



**sparsick** committed on 14 Feb 2016



[8e94efc](#)






Die Skripte in VCS  
laufen nicht durch.



Bei mir aber.



Letzten Mal hatten sie  
sogar  
Syntaxfehler.

# Continuous Build



## 2 Continuous Build

Der erste Automatisierungsschritt etabliert einen kontinuierlichen Build-Lauf. Jede Änderung im VCS zieht das Bauen nach sich. So bleibt die Software durchgängig mindestens kompilierfähig.

### BEST PRACTICES

- Keine manuellen Schritte – Builds laufen automatisiert ab.
- Das Abhängigkeitsmanagement erfolgt über das Buildwerkzeug.
- Das Resultat eines Build sind fertige, umgebungsunabhängige Deployment-Artefakte.
- Alleinige Wahrheit über den Zustand des Builds ist eine dedizierte Build Integration Maschine (CI-Server) – kein „It works on my machine.“

### SO ARBEITEN EURE TEAMS

- Sie fixen einen fehlgeschlagenen Build auf dem Server unverzüglich.
- Entwickler lassen den Build auch auf ihrem lokalen Rechner laufen.

# Continuous Build für Ops

- IaC Skripte können lokal ausgeführt werden.
- Beispiel: Vagrant + Virtualbox / Docker

# Demo Vagrant + Virtualbox

# Continuous Build für Ops

- Lokale Arbeitsrechner der Ops müssen dafür geeignet sein.
  - Zu oft müssen Ops wie auch Devs Office-Rechner benutzen, die nicht genügend Performance haben.
  - Falsche OS
  - Keine Admin-Rechte auf den Arbeitsrechner

**YES, I'M LOOKING  
ON YOU, MANAGEMENT**



# Continuous Build für Ops

- IaC Repository wird von CI Server auf Änderungen überwacht.




# Pipeline IaC CI Pipeline



[Recent Changes](#)

## Stage View


		Declarative: Checkout SCM	Ansible Lint Check	Ansible Playbook run with tests	Declarative: Post Actions
Average stage times:		267ms	1s	29s	2s
#5	Aug 23 17:12 No Changes	184ms	1s	23s failed	2s
#4	Aug 23 17:10 1 commit	245ms	1s	1min 1s	2s
#3	Aug 23 17:09 No Changes				
#2	Aug 23 16:57 No Changes	373ms	1s	2s failed	2s
#1	Aug 23 16:56 No Changes				



Die Skripte in VCS  
funktionieren schon wieder  
nicht wie erwartet.



Kann nicht sein.



Ups, dann ist wohl  
beim Refactoring  
was kaputt gegangen.

# Continuous Testing

Test-Automatisierung zur  
Sicherstellung der Funktionen,  
auch im Zusammenspiel

3

Continuous  
Testing

Implementierungen und Tools:  
xUnit, JBehave, Selenium, Jasmin, ...

3

## Continuous Testing

Automatisierte und regelmäßige Tests auf mehreren Ebenen verkürzen die Feedbackzeiten zum Zustand des Systems – über das bloße Bauen hinaus.

### BEST PRACTICES

- Tests sind automatisiert, wiederholbar und laufen voneinander unabhängig.
- Tests liegen entweder in eigenen Repositories oder parallel zum Source Code.
- Tests sind kategorisiert (siehe Testpyramide).
- Für frühes Feedback laufen schnelle Tests zuerst.

### SO ARBEITEN EURE TEAMS

- Entwickler lassen die Tests auch auf ihren lokalen Rechnern laufen.
- Sie schreiben automatisierte Entwicklertests und integrieren sie in den Build.
- Teammitglieder checken keine Tests ein, die fehlschlagen.
- Die Teams dokumentieren auftretende Softwarefehler anhand von Tests.

# Continuous Testing für Ops



```
def test_openjdk_is_installed(host):
    openjdk = host.package("openjdk-8-jdk")
    assert openjdk.is_installed

def test_tomcat_catalina_script_exist(host):
    assert host.file("/opt/tomcat/bin/catalina.sh").exists
```

# Continuous Testing für Ops




testinfra

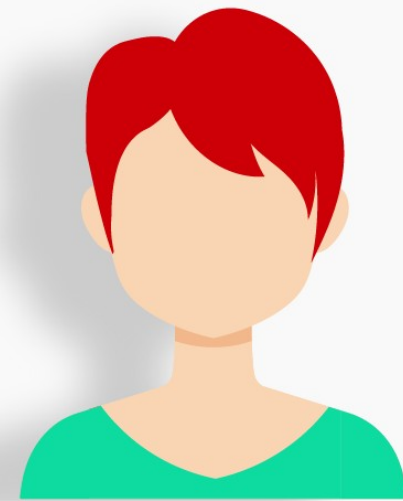
```
~/dev/workspace/iac-qa-talk/samples(master x) py.test --connection=ansible --ansi
ble-inventory inventory/test -v tests/*.py -vv
===== test session starts =====
platform linux -- Python 3.8.2, pytest-5.4.3, py-1.9.0, pluggy-0.13.1 -- /usr/bin
/python3
cachedir: .pytest_cache
rootdir: /home/sparsick/dev/workspace/iac-qa-talk/samples
plugins: testinfra-5.2.1
collected 2 items

tests/test_tomcat.py::test_openjdk_is_installed[ansible://192.168.33.10] PASSED [
 50%]
tests/test_tomcat.py::test_tomcat_catalina_script_exist[ansible://192.168.33.10]
PASSED [100%]

===== 2 passed in 3.91s =====
```



Die Skripte in VCS  
folgen schon wieder  
nicht unseren Style-Guide.

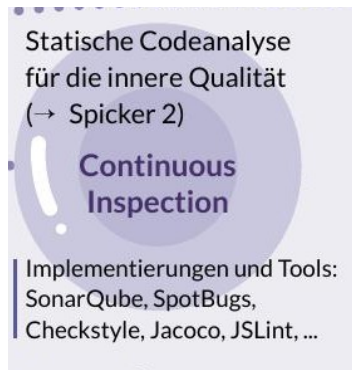


Wir haben eine  
Style-Guide?



Ja, Best Practices auch nicht

# Continuous Inspection



- Automatisierte statische Codeanalyse

## Best Practices

- Regelsatz orientiert sich an den Best Practices aus der Community
- Team einigt sich auf ein Regelsatz

## So arbeitet das Team

- Codeanalyse ist automatisiert und in den Build integriert.
- Verstöße gegen den Regelsatz werden sofort behoben.

# Continuous Inspection für Ops

```
~/dev/workspace/iac-qa-talk/samples(master x) ansible-lint *.yaml
[502] All tasks should be named
setup-tomcat.yaml:34
Task/Handler: file name=/opt mode=511 owner=tomcat group=tomcat __line__=35 __file__=setup-tomcat.yaml

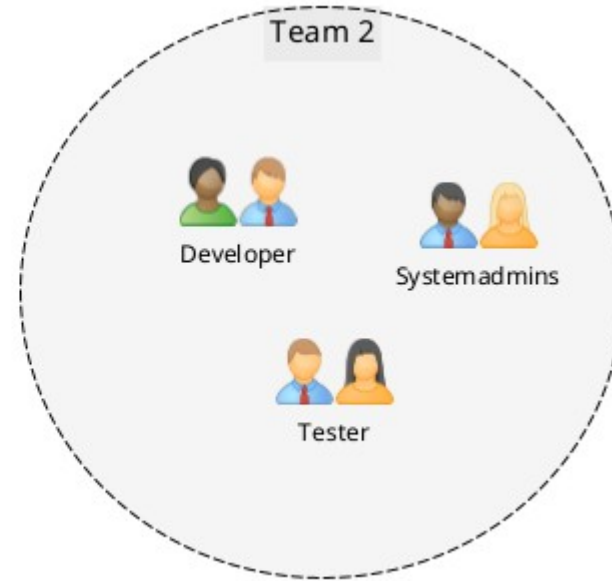
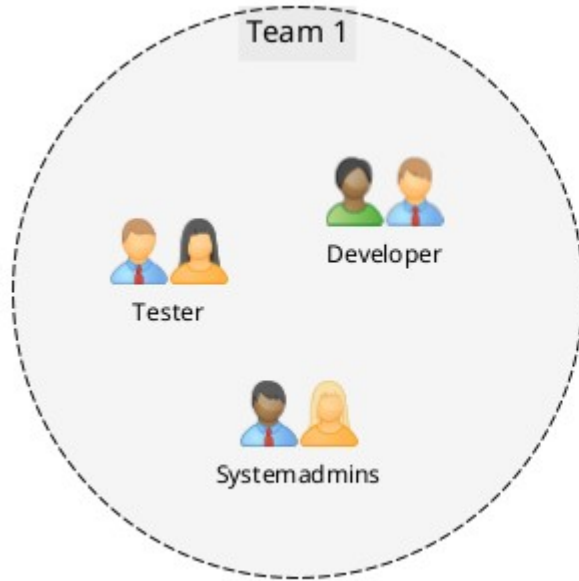
[502] All tasks should be named
setup-tomcat.yaml:63
Task/Handler: find paths=/opt/{{ tomcat_base_name }}/bin patterns=*.sh
```



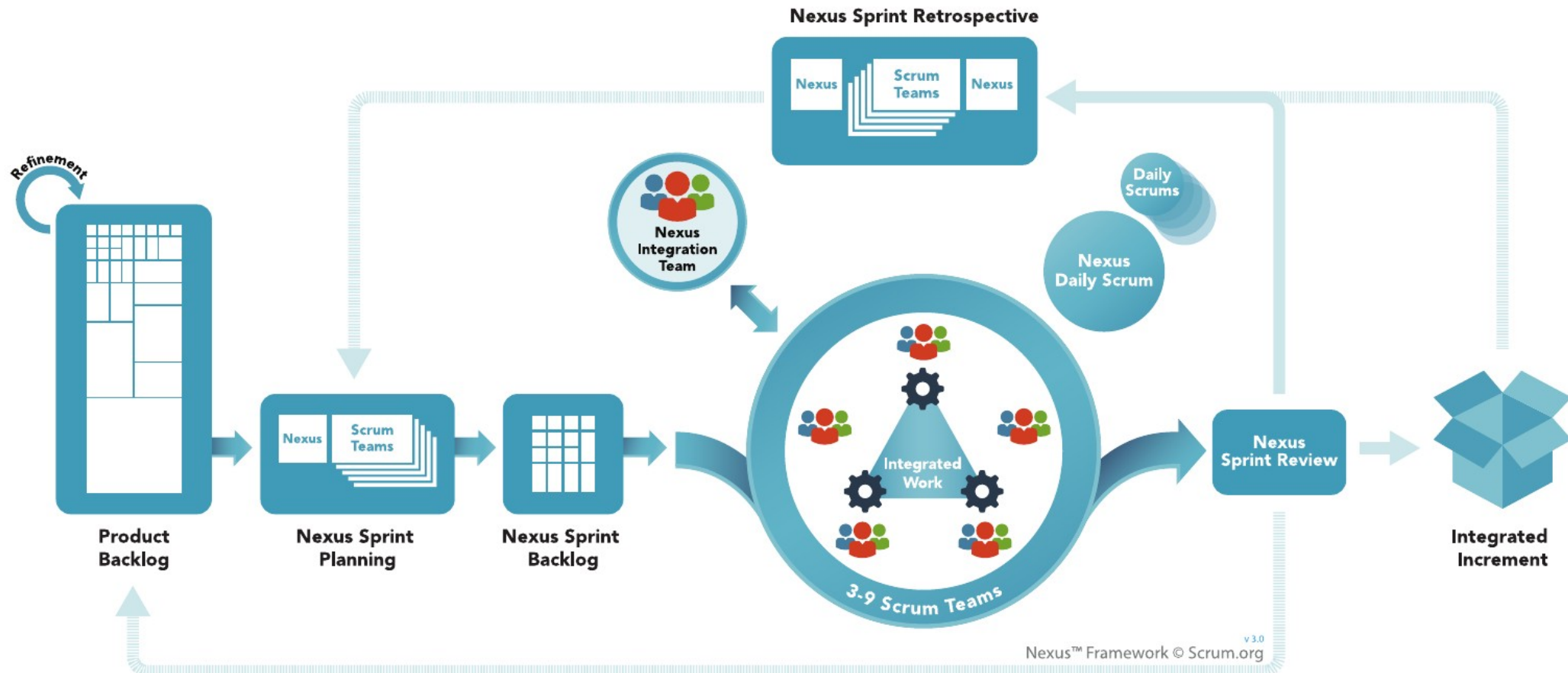


Diese technische Lösung bringen nichts, wenn sich die Arbeitsweise der Organisation nicht ändert.

# Idealbild



# NEXUS™ FRAMEWORK



## BUSINESS AGILITY

MEASURE &amp; GROW



## Enterprise Solution Delivery

- Apply Lean system engineering to build really big systems
- Coordinate and align the full supply chain
- Continually evolve live systems

Lean System and  
Solution EngineeringCoordinating Trains  
and SuppliersContinually Evolve  
Live Systems

## Lean Portfolio Management

- Align strategy, funding, and execution
- Optimize operations across the portfolio
- Lightweight governance empowers decentralized decision-making



## Agile Product Delivery

- The customer is the center of your product strategy
- Develop on cadence and release on demand
- Continuously explore, integrate, deploy, and innovate

Customer Centricity  
& Design ThinkingDevelop on Cadence  
Release on DemandDevOps and the Continuous  
Delivery Pipeline

## Organizational Agility

- Create an enterprise-wide, Lean-Agile mindset
- Lean out business operations
- Respond quickly to opportunities and threats

Lean-thinking People  
and Agile TeamsLean Business  
Operations

Strategy Agility

## Continuous Learning Culture

- Everyone in the organization learns and grows together
- Exploration and creativity are part of the organization's DNA
- Continuously improving solutions, services, and processes is everyone's responsibility

Learning  
OrganizationInnovation  
CultureRelentless  
Improvement

## Team And Technical Agility

- High-performing, cross-functional, Agile teams
- Business and technical teams build business solutions
- Quality business solutions delight customers



Agile Teams

Teams of  
Agile TeamsBuilt-In  
Quality

## Lean-Agile Leadership

- Inspire others by modeling desired behaviors
- Align mindset, words, and actions to Lean-Agile values and principles
- Actively lead the change and guide others to the new way of working



Leading by Example



Mindset &amp; Principles



Leading Change



### Select Configuration



Fazit

Fragen?

[mail@sandra-parsick.de](mailto:mail@sandra-parsick.de)

@SandraParsick

<https://github.com/sparsick/iac-qa-talk>



# Literatur

- Architektur Spicker 7 – Continuous Delivery  
<https://www.sandra-parsick.de/publication/architektur-spicker-cd/>
- Scaling Scrum with Nexus  
<https://www.scrum.org/resources/scaling-scrum>
- Scaled Agile Framework SAFe  
<https://www.scaledagileframework.com/>