DEV-OPS-TECHNOLOGIES – CHAOS ENGINEERING

offen

Quelle: Netflix - https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png, Apache License 2.0

© Fraunhofer

FRAUNHOFER-INSTITUT ENERGIEWIRTSCHAFT UND ENERGIESYSTEMTECHNIK

Das Fraunhofer IEE in Kassel forscht in den Bereichen Energiewirtschaft und Energiesystemtechnik.

Wir entwickeln Lösungen für wirtschaftliche und technische Herausforderungen bei der Transformation der Energiesysteme.

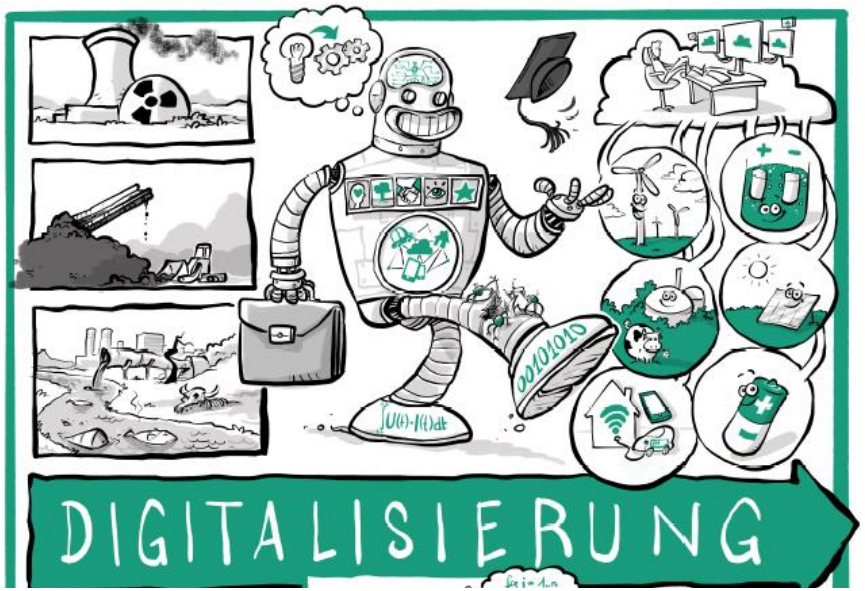
- Mitarbeiter: rund 360
- Jahresbudget: ca. 25 Mio EUR
- Leitung: Prof. Dr. Clemens Hoffmann

www.iee.fraunhofer.de

© Fraunhofer

Fraunhofer IEE

Energieinformatik



Monitoring, Aggregation,

Steuerung von Anlagen

Optimierung Fahrplänen,

Automatisierter Handel

Prognosesysteme: Wind,

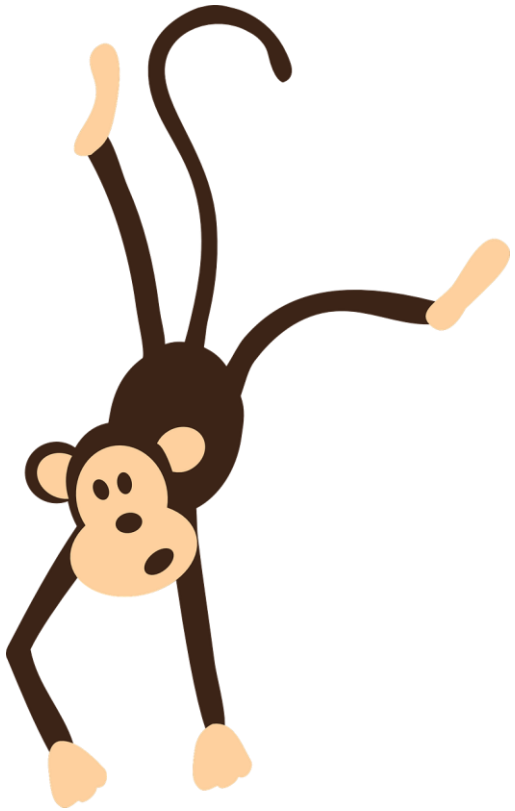
PV, Biomasse

© Fraunhofer

Fraunhofer IEE

Endspurt

Heute: Theoretischer Teil



Letzte Veranstaltung: Hands on

offen

Quellen: Links, Bild von Dmitry Abramov auf Pixabay, Rechts: Bild von Clker-Free-Vector-Images auf Pixabay

© Fraunhofer

Fraunhofer IEE

AGENDA

Motivation

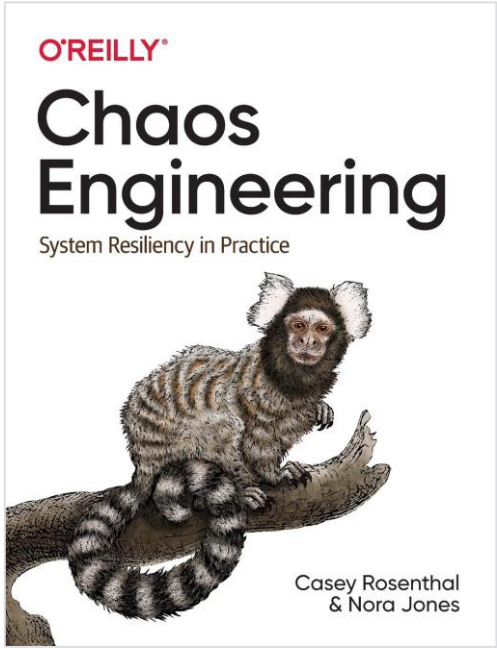
Woher kommt Chaos Engineering?

Was ist Chaos Engineering ?

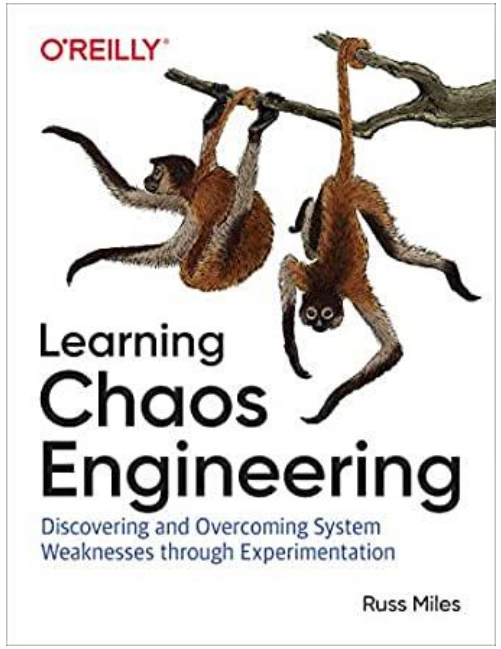
Vorgehen des Chaos Engineering?

© Fraunhofer

Quellen / Literatur



ISBN: 978-1-492-04386-7



ISBN: 978-1-492-05100-8

ISBN: 978-1-68050-239-8

- https://netflixtechblog.com/
- https://principlesofchaos.org/

© Fraunhofer

Fraunhofer IEE

Mitarbeiter

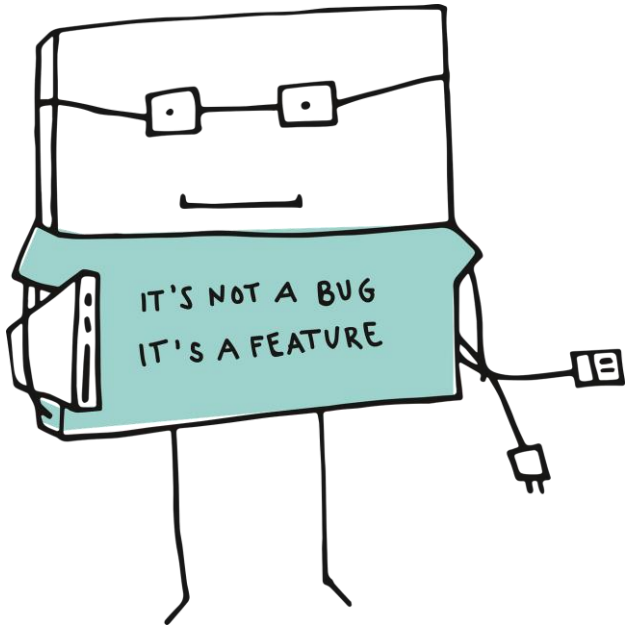
offen

Quelle: Bilder von Manfred Steger auf Pixabay, Bild von Wolfgang Eckert auf Pixabay

© Fraunhofer

Fraunhofer IEE

Mitarbeiter



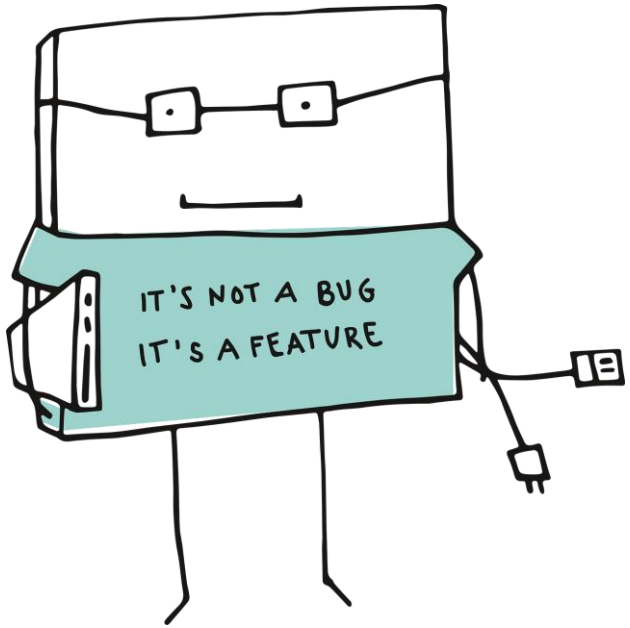
Developer

Quelle: Bilder von Manfred Steger auf Pixabay, Bild von Wolfgang Eckert auf Pixabay

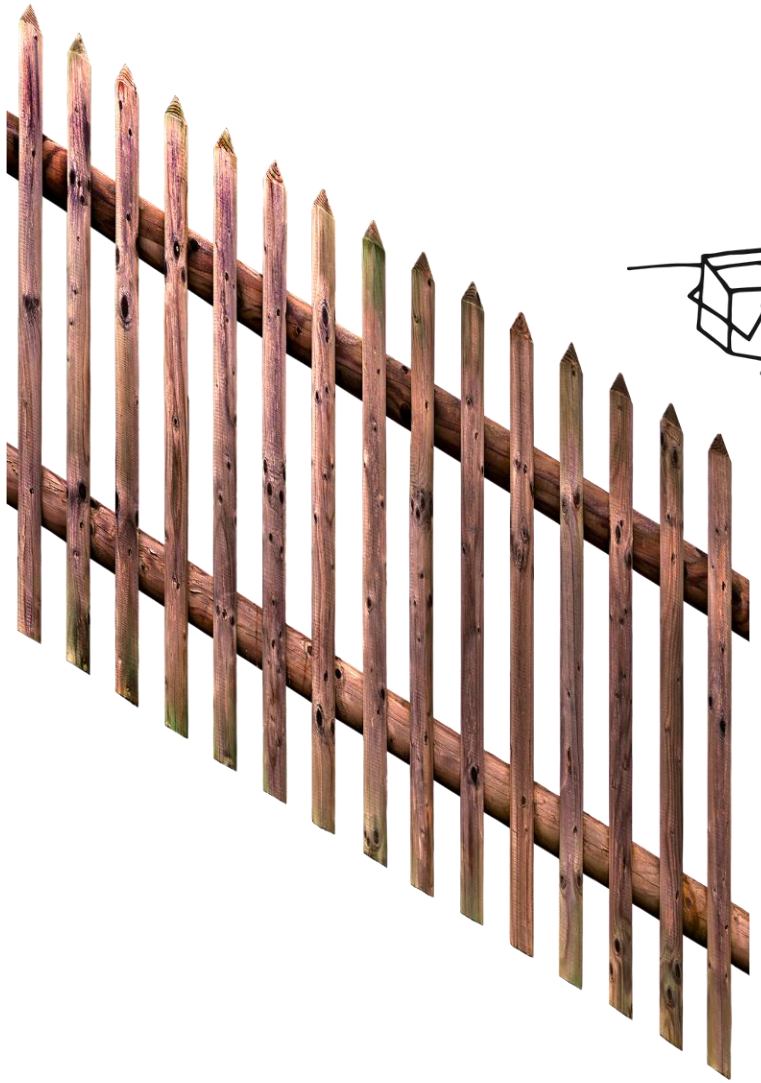
© Fraunhofer

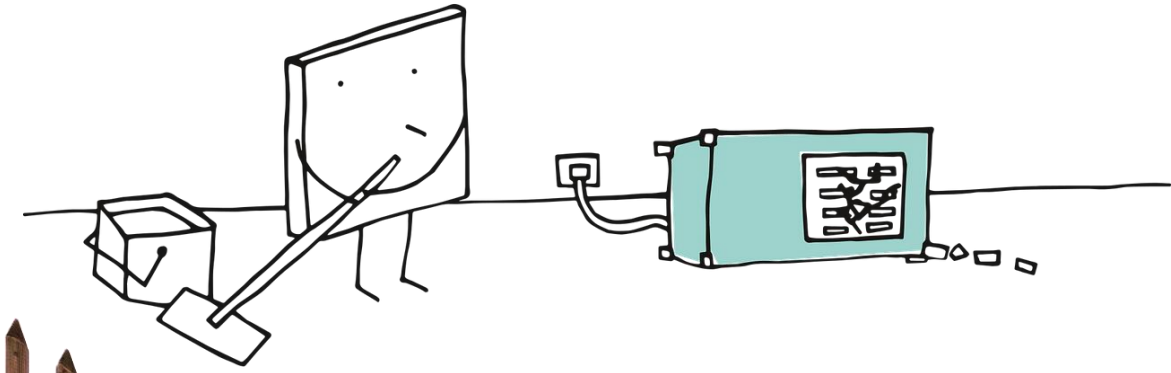
Fraunhofer IEE

Mitarbeiter



Developer





Admin

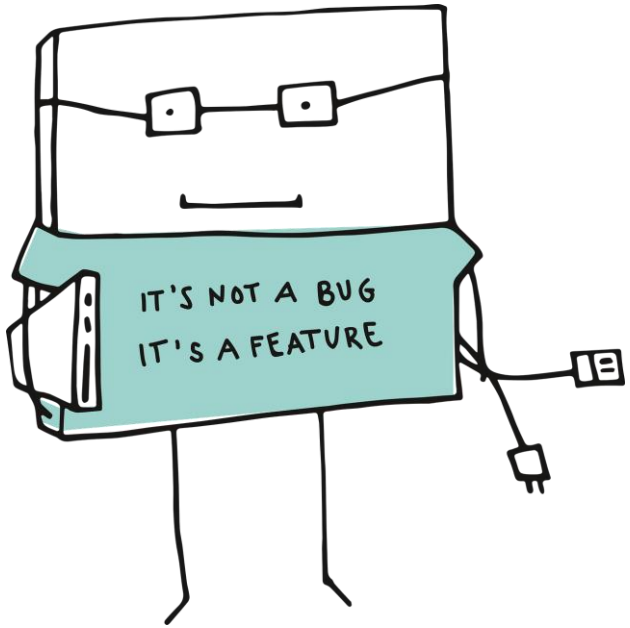
Quelle: Bilder von Manfred Steger auf Pixabay, Bild von Wolfgang Eckert auf Pixabay

© Fraunhofer

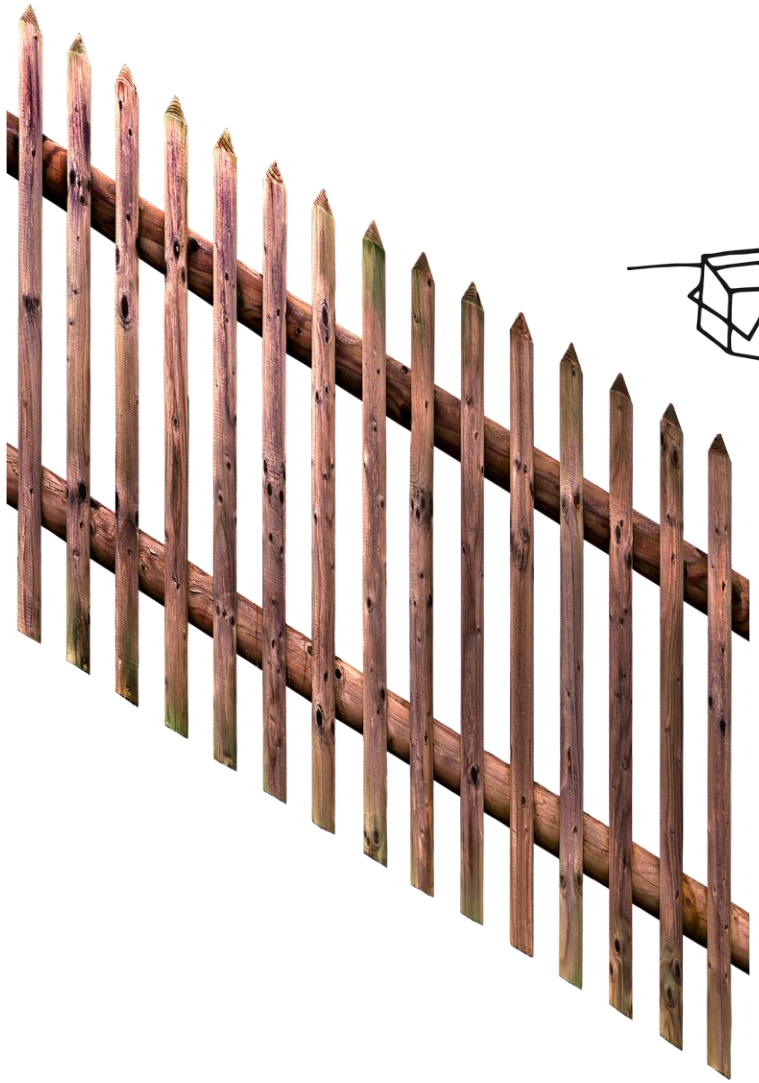
Fraunhofer IEE

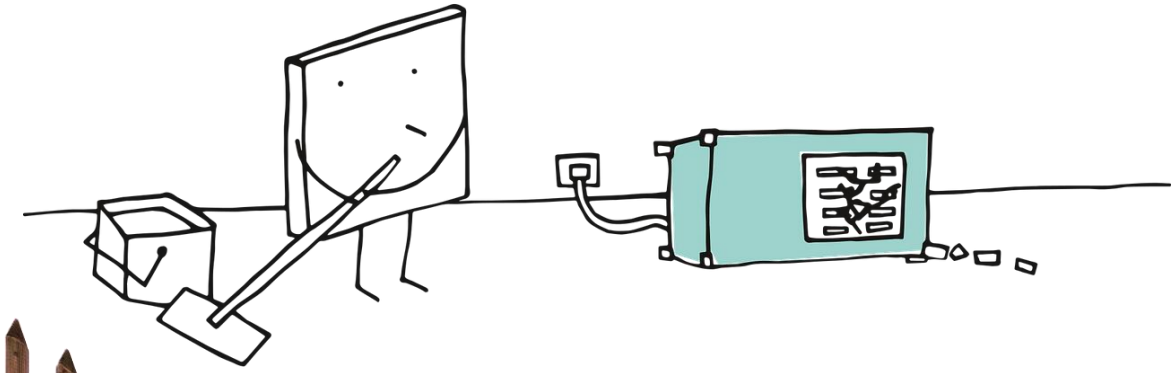
Mitarbeiter

Software



Developer





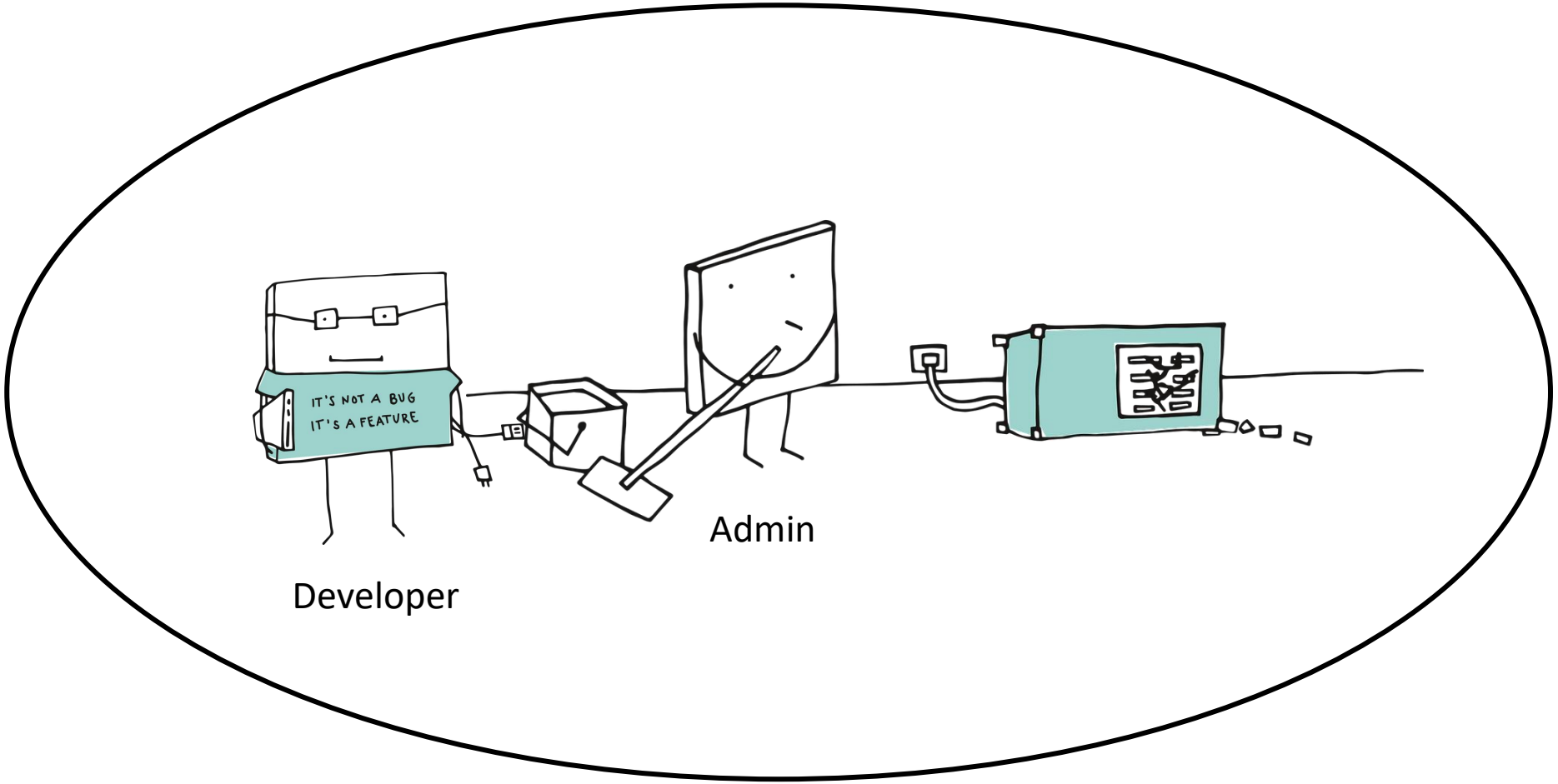
Admin

Quelle: Bilder von Manfred Steger auf Pixabay, Bild von Wolfgang Eckert auf Pixabay

© Fraunhofer

Fraunhofer IEE

Dev – Ops Mitarbeiter



Gemeinsame End – to – End Verantwortung

Quelle: Bilder von Manfred Steger auf Pixabay, Bild von Wolfgang Eckert auf Pixabay

© Fraunhofer

Fraunhofer IEE

Geschichte von Margaret Hamilton

- Margaret Hamilton: „Director of Software Programming“ MIT Instrumentation Laboratory.
- Leitete ein Team für On-Board-Flugsoftware für Apollo Programm
- Tochter Lauren entdeckte Fehler in er Software
- Priorisierungs-Display (Scheduling) ➔ Mondlandung Apollo11
- 2016 verlieh ihr Präsident Barack Obama die Presidential Medal of Freedom.

https://www.tagesspiegel.de/wissen/software-pionierin-margaret-hamilton-ihr-code-brachte-die-apollo-astronauten-sicher-zum-mond/24679884.html

Quelle Bild: Wikipedia – https://commons.wikimedia.org/wiki/File:Margaret_Hamilton_-_restoration.jpg

© Fraunhofer

Chaos Engineering

- 2008 Netflix: Data Center → Cloud
- VMs gingen öfter kaputt.
- Viele selbstorganisierende Teams

➔ Management Instrument als laufende Software

- Weihnachten 2012 → ELB AWS Probleme (https://netflixtechblog.com/a-closer-look-at-the-christmas-eve-outage-d7b409a529ee)
- Ausfälle ganzer Regionen
- Chaos Kong

Quelle Netflix: https://github.com/Netflix/chaosmonkey/blob/master/logo.png, , Apache License 2.0,
Netflix - https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png, Apache License 2.0

Definition Chaos Engineering

“Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production.”

Quelle: https://principlesofchaos.org/

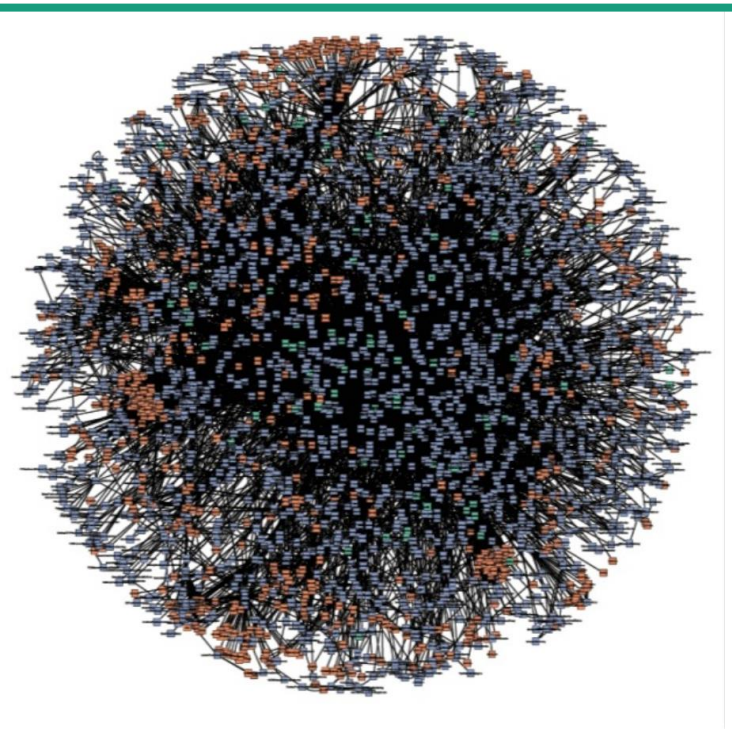
© Fraunhofer

Chaos Engineering – Advanced Principles

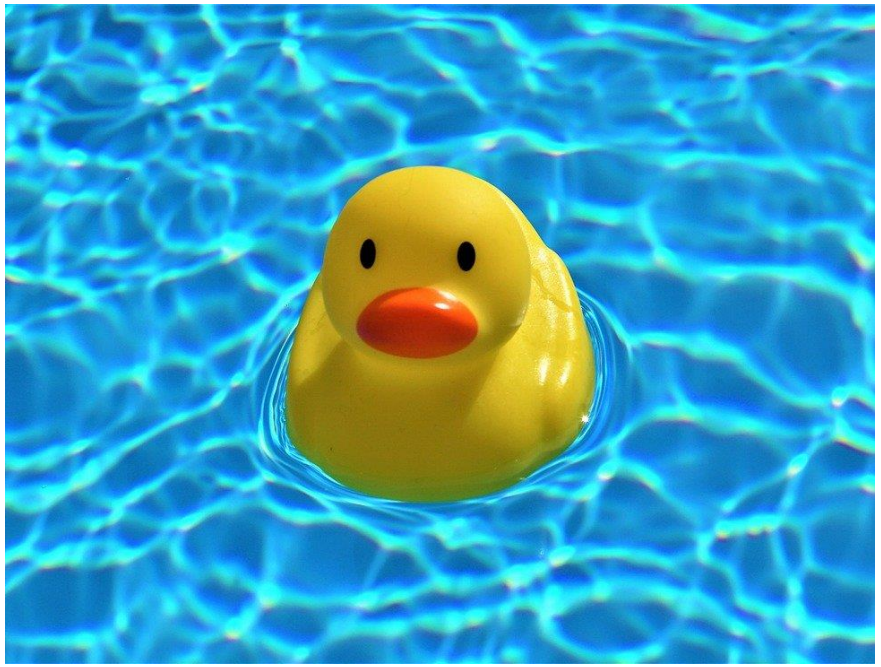
- **Build a Hypothesis around Steady State Behavior**
  - **Fokussierung auf Messbaren Output des Systems**
  - **Das es läuft ist relevant, nicht wie es läuft**
- **Vary Real-world Events**
  - **Variablen sollten Ereignisse aus der realen Welt reflektieren**
  - **Perspektive eines Verteilten Systems – Verfübarkeit = Latency / Status Code**
- **Run Experiments in Production**
  - **Experimente in Staging ➔ Aussagen über Staging**
  - **Staging != Produktion**
- **Automate Experiments to Run Continuously**
  - **Mehr Experimente und Beobachtungen über die Zeit**
- **Minimize Blast Radius**
  - **Minimiere die negativen Effekte bei gleicher Aussagekraft des Experiments**

© Fraunhofer

Chaos Engineering - Kontext



- ✓ Verteilten System / Microservice Architekturen
- ✓ Hohe Komplexität
- ✓ 24/7h Betrieb der Anwendung
- ✓ Abhängigkeiten zu Drittsystemen



- – Einzelne Services
- – Ein Entwickler
- – Dev = SI = P
- – Keine Externen Schnittstellen

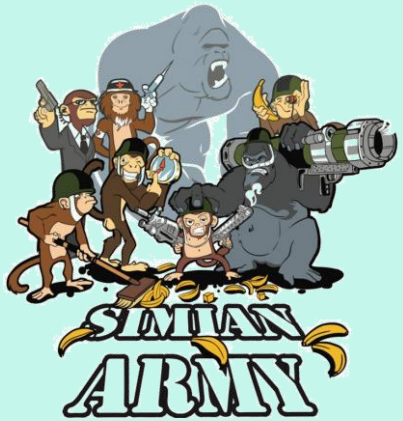
Quelle https://www.slideshare.net/apigee/i-love-apis-2015-microservices-at-amazon-54487258, Chris Munns, CC Attribution License

,Bild von Mylene2401 auf Pixabay

© Fraunhofer

Fraunhofer IEE

Breaking Stuff vs. Fixing Stuff



Chaos Engineering

- Ziel: Wissen über Schwachstellen generieren
- Systematisches Vorgehen um resilientes Gesamtsystem aufzubauen
- Fokus auf Messbarkeit

ein Chaos Engineering

- Lustig die eigene Infrastruktur stören
- Chaos Experimente machen bei denen das Ergebnis klar
- Kein Monitoring der wichtigen Variablen

Quelle: Netflix - https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png, Apache License 2.0

© Fraunhofer

Fraunhofer IEE

Gesamtes System

- **Chaos Engineering kommt aus einer Infrastrukturbetrachtung ist aber mehr:**

- **Betrachtetes Soziotechnisches System (sociotechnical system) besteht aus:**

Nicht technisch:

- People
- Practices
- Processes

Technisch:

- Plattform
- Infrastructure
- Applications

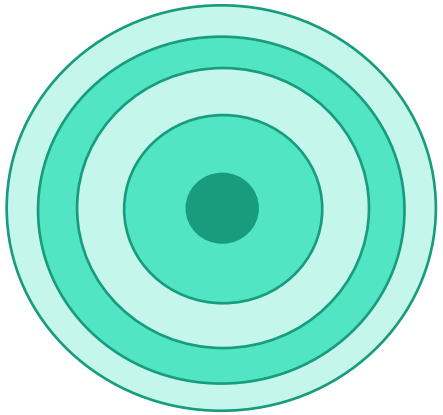
- **Schwachstellen werden daher in all diesen Systemteilen gesucht.**

© Fraunhofer

Chaos Engineering

- Erkenntnisse über das Verhalten von Systemen in turbulenten Situation erlangen!
- Schwachstellen erkennen und ausmerzen ➔ Bevor sie zu Störfällen und kritischen Problemen führen

- Wissenschaftliches Vorgehen beim Chaos Engineering



Wissen generieren !

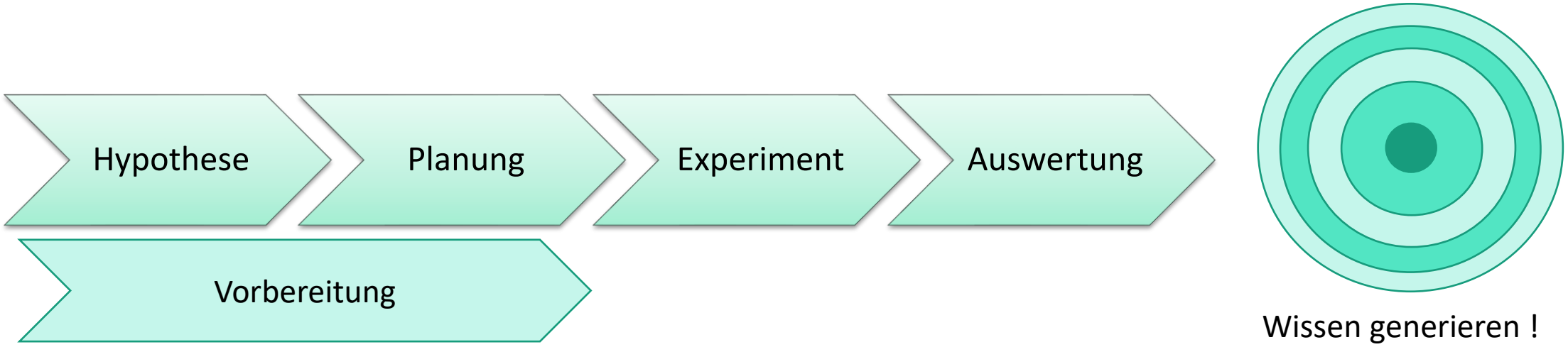
© Fraunhofer

Fraunhofer IEE

Chaos Engineering

- Erkenntnisse über das Verhalten von Systemen in turbulenten Situation erlangen!
- Schwachstellen erkennen und ausmerzen ➔ Bevor sie zu Störfällen und kritischen Problemen führen

- Wissenschaftliches Vorgehen beim Chaos Engineering



© Fraunhofer

Fraunhofer IEE

Experimentation vs. Testing

Testing

- Ziel: Prüfung ob die Anforderungen erfüllt werden
- Assertions basieren auf existierenden Wissen
- Klassisches Qualitätsmanagement
- Ergebnis = Passed oder Failed
- Kontext ➔ Software / SW/HW

Experimation

- Ziel: Wissen über Schwachstellen erlangen
- Experimente erzeugen neues Wissen
- Wissenschaftliches Vorgehen
- Ergebnis = Erkenntnisgewinn
- Kontext ➔ sozioökonomische Gesamtsystem

© Fraunhofer

Production vs. Sandboxing

- Chaos Experimente in Staging generieren Wissen über Schwachtellen im Statging

- Staging != Production ➔ Schwachstellen in Staging != Schwachstellen in Production

- Mögliche Unterschiede
  - ➔ Systemlast
  - ➔ Hardware (Beispiel Windenergieanlagen….)
  - ➔ Infrastruktur
  - ➔ Reale User

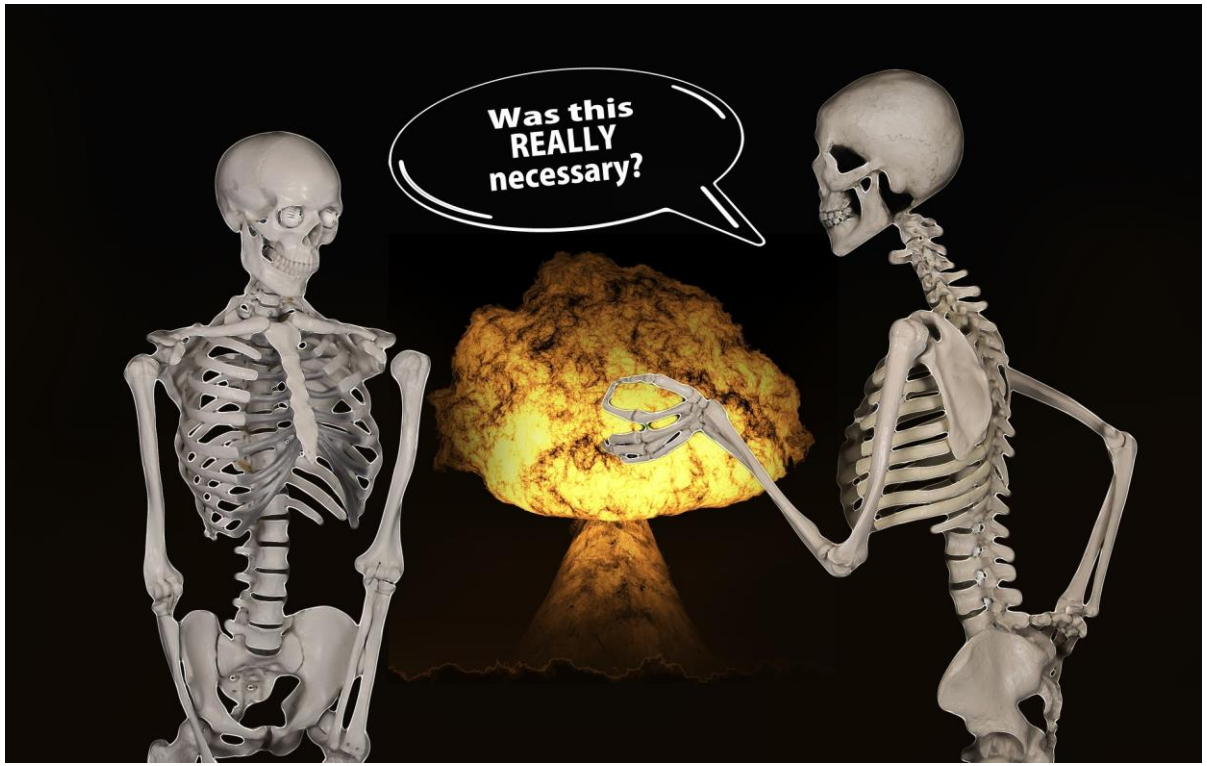
❖ Um den „Blast Radius“ zu reduzieren ist es oft sinnvoll in Staging anzufangen.

© Fraunhofer

Minimize Blast Radius

- Es geht um Erkenntnisgewinn, nicht um Zerstörung!

- Der Geschäftsbetrieb sollte nicht zum erliegen kommen

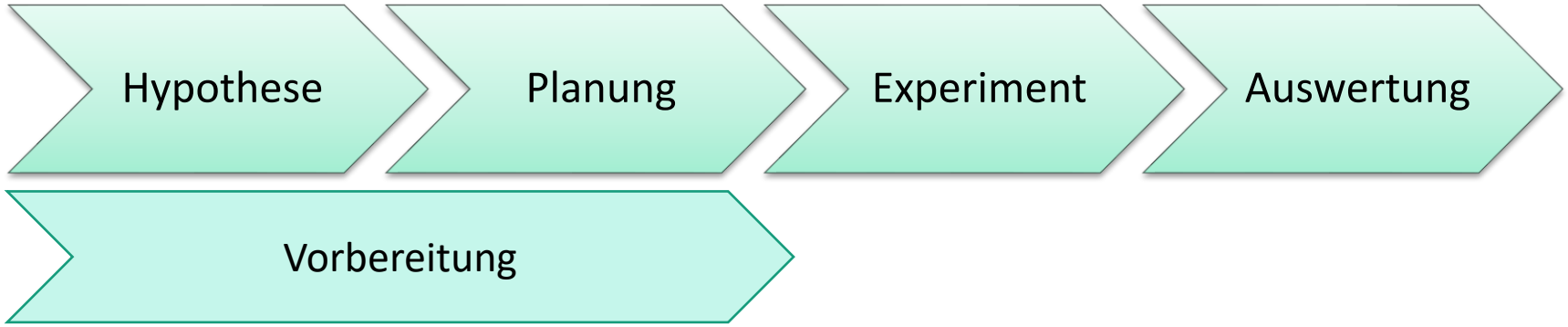


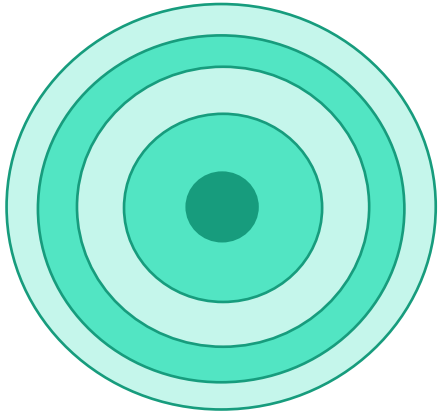
- Den sog. Blast Radius minimieren
- Starte nicht in der P-Umgebung!
- Canary Experiments – Nur ein Teil des Datenverkehrs für das Experiment benutzen
- Ggf. auf Business Objekten Filtern z.B. nur Transaktionen mit einem Wert < 100 €.
- Denk an Rollbacks!

Quelle: Bild von Gerd Altmann auf Pixabay

© Fraunhofer

Vorgehen





Wissen generieren !

© Fraunhofer

Fraunhofer IEE

Hypotheses Backlog

Backlog ähnlich wie User Story Backlog in SCRUM

Steady – State Hypotheses zum Verhalten in Turbulenten Situationen

Mögliche Quellen:

1. Vergangene Fehler
2. „Was könnte passieren wenn “

Normalerweise keine vollständige Liste

Hypothesen zu: People, Practices, Processes, Applications, Plattform & Infrastructure

Tags:

Reliability

Durability

Security

Safety

© Fraunhofer

Hypotheses Backlog - Beispiel

Das System bleibt erreichbar wenn…

Das Datenbankcluster ausfällt.

Das System liefert regelmäßige Messdaten auch wenn…

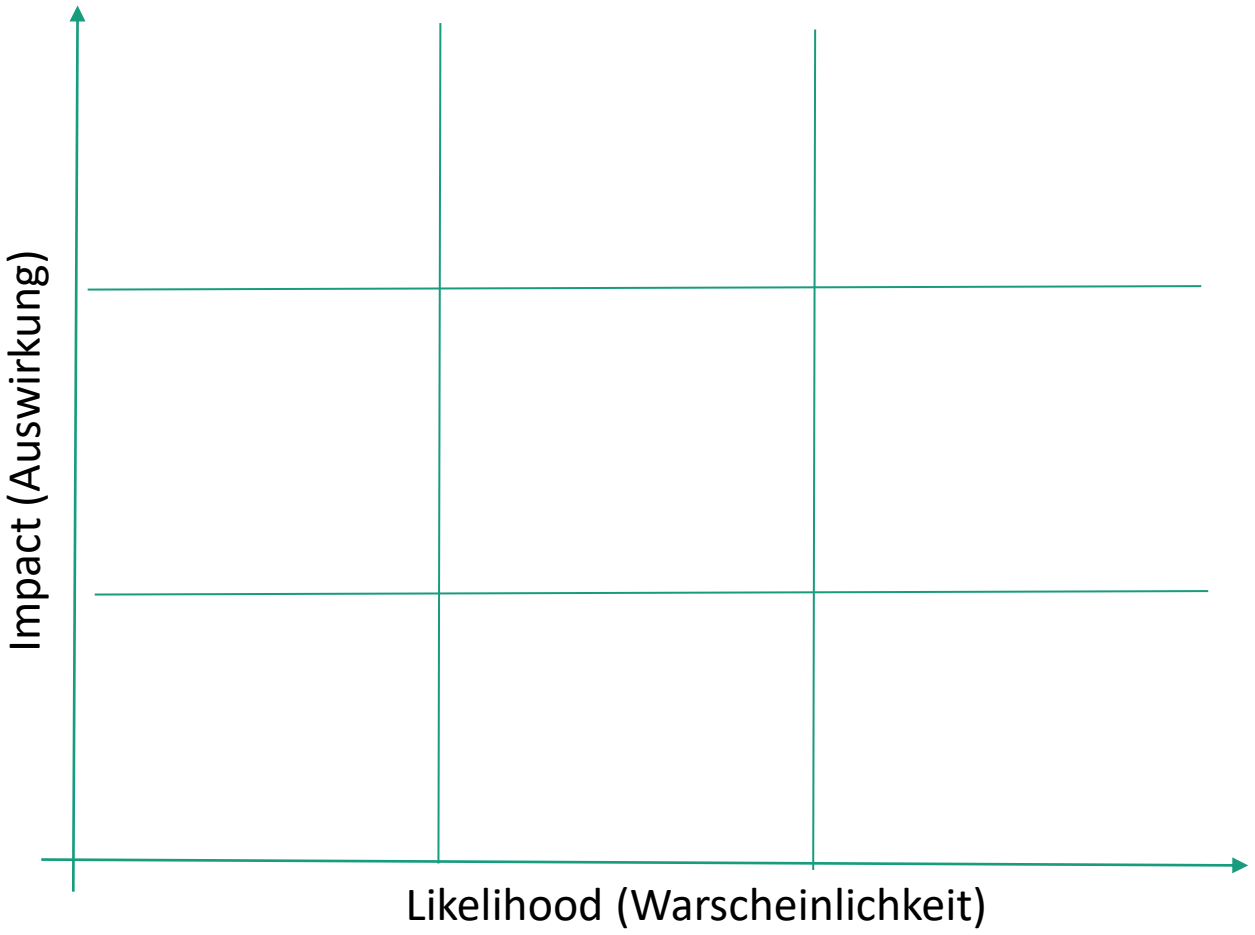
die Latenz mehrere IOT Devices erheblich steigt

Das System verliert keine Messwerte wenn …

Das Kafka Cluster ausfällt

© Fraunhofer

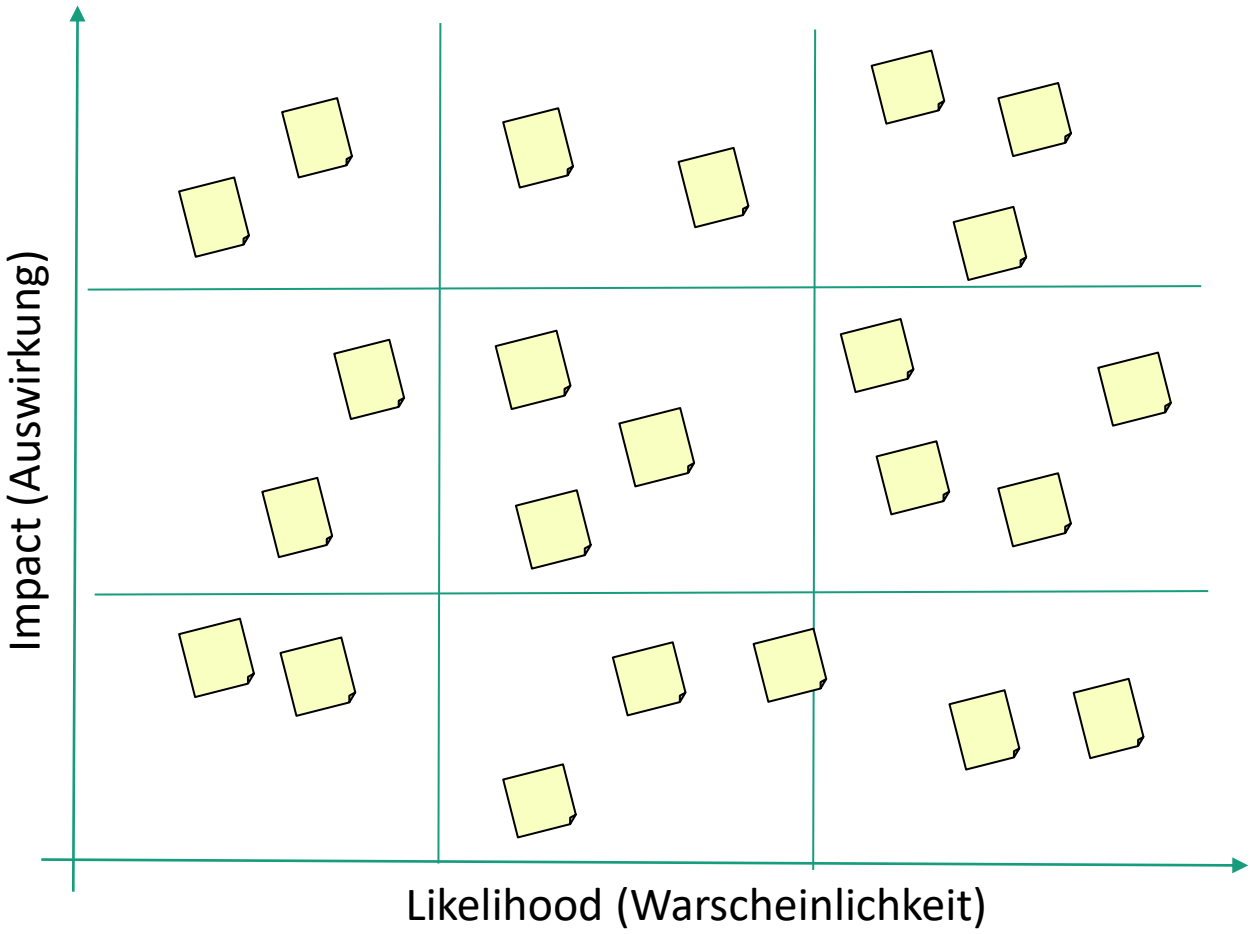
Hypotheses Backlog



© Fraunhofer

Fraunhofer IEE

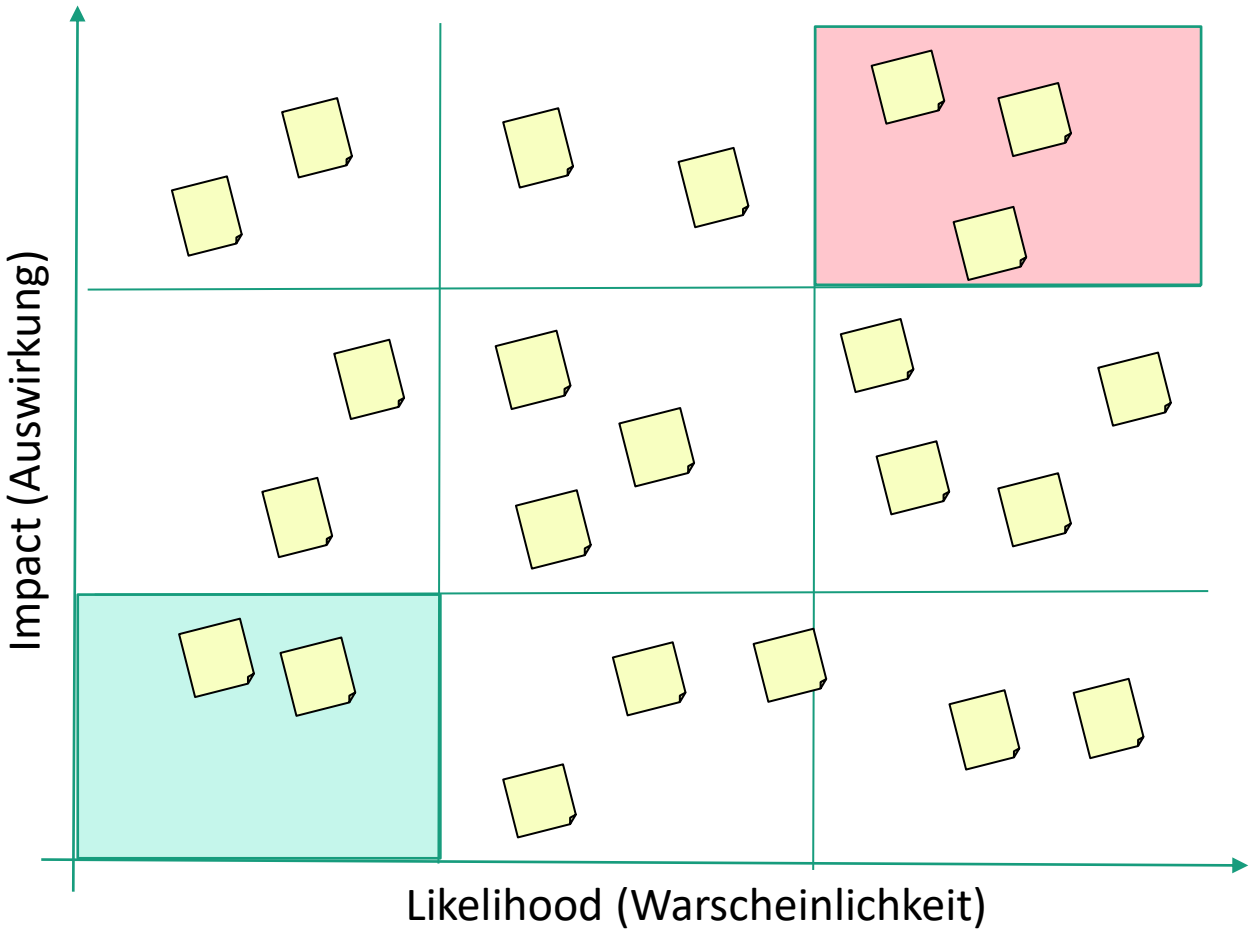
Hypotheses Backlog



© Fraunhofer

Fraunhofer IEE

Hypotheses Backlog



© Fraunhofer

Fraunhofer IEE

Planung des Experiments

Unterstützende Fragen (vgl. Chaos Engineerin, Casey Rosenthal & Nora Jones → Literatur)

- Wie definieren wir „normaler“ Systemzustand ?

- Wie ist der Scope / Wie wird der Fehler injeziert ?
  - Warum haben wir den Scope so gewählt ?
  - Ist es Angst getrieben ?
  - Haben wir keine Ahnung wie sich das System dabei verhält ?
- Was glauben wir was passiert wenn der Fehler auftritt ?

- Woran erkennen wir, das dass System in einem ?

  ➔ Steady State ggf. Überdenken

© Fraunhofer

Planung des Experiments

1. Hypothese aussuchen → Prioritäten, Blast Radius
2. Beschreibung des Experiments
   1. Steady State Hypothese

      Alle Messwerte die anzeigen das ein System sich im stabilen Zustand befinden

      (Zusätzlich ggf. noch was will ich noch Beobachten / Monitoren?)
   2. Methode

      Liste an Schritten um eine „turbulente Situation“ zu erzeugen
   3. Rollbacks

      Liste an Schritten wieder in den Normalzustand zurückzukehren

© Fraunhofer

Versuchsausbau

Das System bleibt erreichbar wenn…

Das Datenbankcluster ausfällt.

Erreichbar ➔ Get / Responsetime < 1 s. && HTTP.Status == 200

© Fraunhofer

Versuchsausbau

Das System bleibt erreichbar wenn…

Das Datenbankcluster ausfällt.

Erreichbar ➔ Get / Responsetime < 1 s. && HTTP.Status == 200

Was möchte ich ggf. noch beobachten ?

- Fallen andere Systeme kaskadierend aus
- Wie ist meine MongoDB Connection wirklich (Funktioniert das Experiment.

© Fraunhofer

Versuchsausbau

Das System bleibt erreichbar wenn…

Das Datenbankcluster ausfällt.

Erreichbar ➔ Get / Responsetime < 1 s. && HTTP.Status == 200

Was möchte ich ggf. noch beobachten ?

- Fallen andere Systeme kaskadierend aus
- Wie ist meine MongoDB Connection wirklich (Funktioniert das Experiment.

Datenbank Cluster fällt aus

➔ Shutdown MongoDB (alle Instanzen!)

Oder

➔ Change Routing to MongoDB

© Fraunhofer

Versuchsausbau

Das System bleibt erreichbar wenn…

Das Datenbankcluster ausfällt.

Erreichbar ➔ Get / Responsetime < 1 s. && HTTP.Status == 200

Was möchte ich ggf. noch beobachten ?

- Fallen andere Systeme kaskadierend aus
- Wie ist meine MongoDB Connection wirklich (Funktioniert das Experiment.

Datenbank Cluster fällt aus

➔ Shutdown MongoDB (alle Instanzen!)

Oder

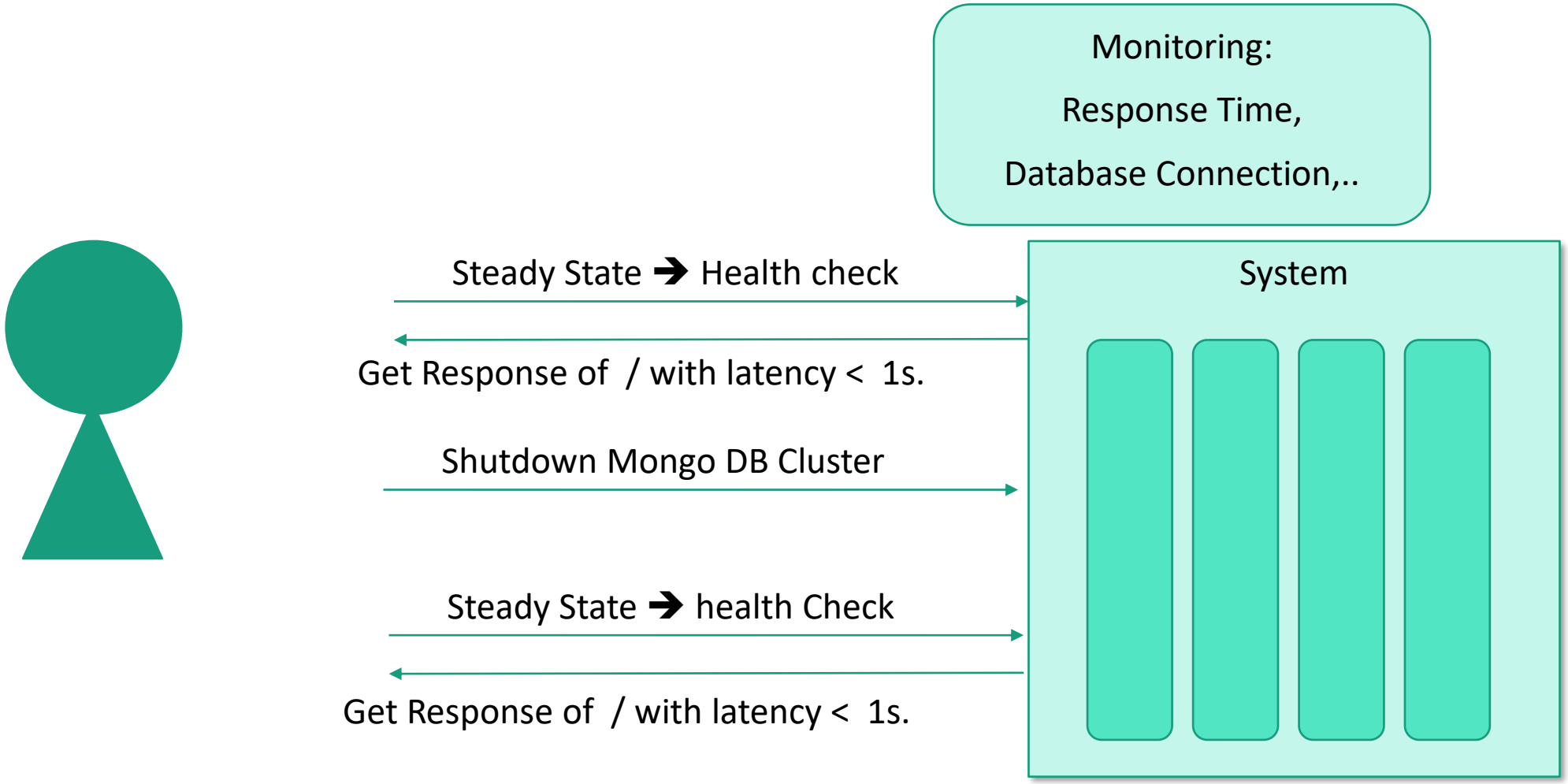
➔ Change Routing to MongoDB

Rollback:

Fahr das DB- Cluster wieder hoch.
Ggf. Services neu-starten.

© Fraunhofer

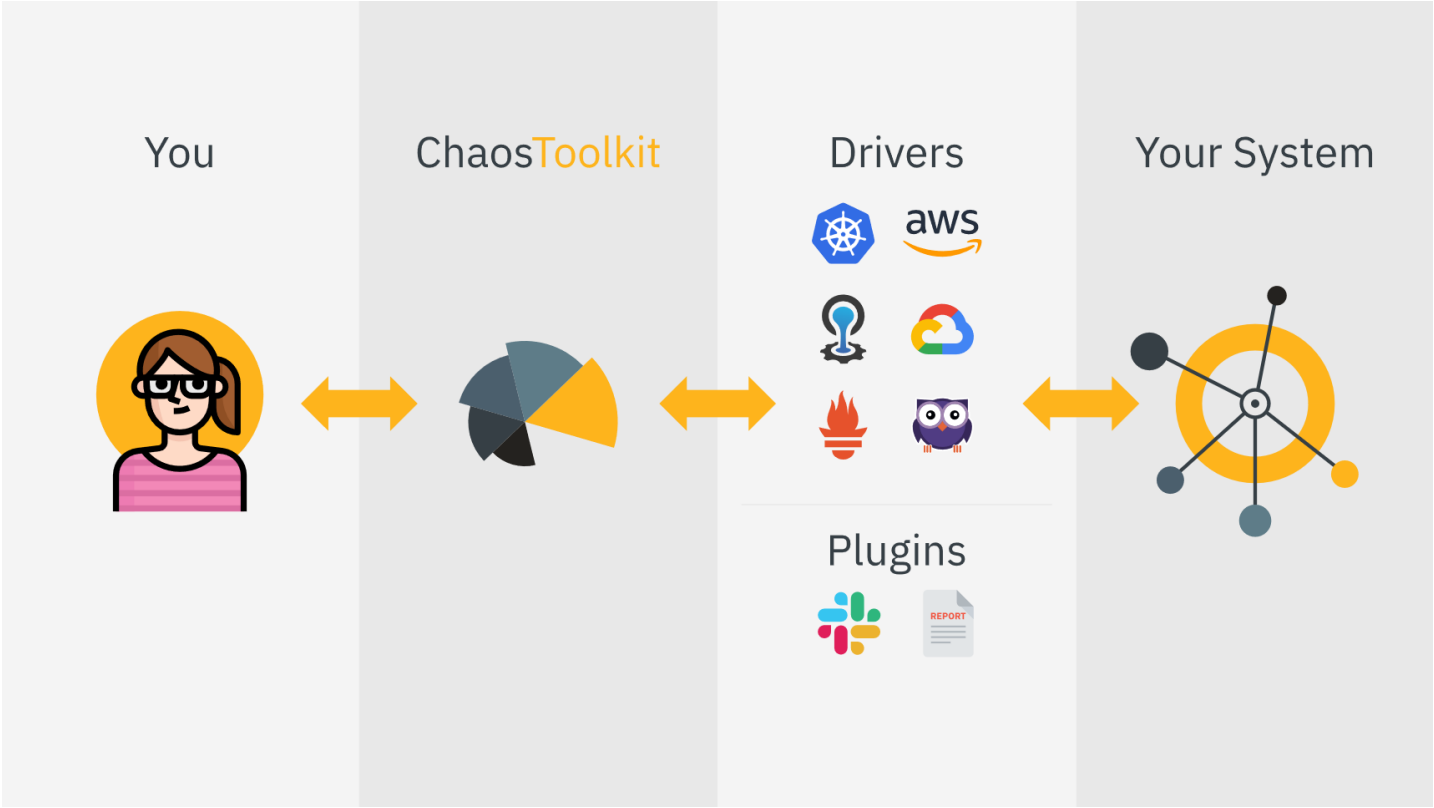
Versuchsausbau



© Fraunhofer

Toolunterstützung

https://chaostoolkit.org/



Nächste Vorlesung

© Fraunhofer

Quelle: https://github.com/chaostoolkit/chaostoolkit-documentation/blob/master/sources/static/images/schema-1920.svg, Lizenz: Apache 2.0

Game Days / Chaos Days

„The absolute worst time to try and learn about the weakness in your sociotechnical system is during an incident“ Russ Miles, Learning Choas Engineering

➔ Eine Mögliche Form des Experiments

➔ Möglichkeit mit Chaos Engineering zu starten – Game Day

➔ Insbesondere für Chaos Experimente bei denen nicht nur technische Schwachstellen analysiert werden

Game Day: Experiment wird gemeinsam an einem Tag (oder mehreren Stunden) durchgeführt.

Quelle: Bild von Hans Braxmeier auf Pixabay

© Fraunhofer

Game Days / Chaos Days

- Rollen im Festlegen
  - Teilnehmer, Beobachter, Spielleiter, (Safety-Monitor), etc.
- Informationen zu dem Game Day
  - Keine Infos, „Dungeons & Dragons“, Informiert im Vorraus
- Minimierung des Blast Radius (meist nicht auf Production)
- Ort und Zeitraum festlegen
- Zustimmung einholen! ☺

Quelle: Bild von Hans Braxmeier auf Pixabay

© Fraunhofer

Fraunhofer IEE

Auswertung der Ergebnisse

- Technische Auswertung hängt von den Informationen und dem genauen Experiment ab

- Zusätzliche Retrospektive „ähnliches“ Event

Unterstützende Fragen (vgl. Chaos Engineerin, Casey Rosenthal & Nora Jones → Literatur)

  - Was haben wir gelernt?
  - Was haben wir gelernt um unsere Verständnis zu verbessern und das Experiment zu wiederholen?
  - Musste irgendein unbeteiligter während des Experiments eingebunden werden?
  - Sind irgendwelche Probleme / Fehler aufgetreten?
  - Welche Teile des Experiments haben sich anders als erwartet verhalten?
  - Was machen wir als nächstes für ein Experiment?
  - …

© Fraunhofer

Automate Chaos Experiments

- Warum automatisieren?
  - Einfache Wiederholbarkeit
  - Änderungen über die Zeit mitbekommen
- Tool Support
  - https://chaostoolkit.org/
  - https://netflix.github.io/chaosmonkey/
- CI /CD Umgebung als Unterstützung
- Nicht alles ist vollständig automatisierbar.

© Fraunhofer

Zusammenfassung

- Chaos Engineering ist eine Methode den Erkenntnissgewinn über Schwachstellen im System zu steigern
- Dabei wird eine wissenschaftliches Vorgehen mit Hypothesen, Experimenten und der Auswertung genutzt um neues Wissen zu generieren
- Chaos Engineering wird verwendet um ein System resilienter zu machen ➔ Nicht Fehler vermeiden sondern mit Fehlern umgehen!
- Fokus ist dabei nicht „nur“ das Softwaresystem sondern das Gesamtsystem bestehend aus Personen, Praktiken, Prozessen, Plattformen, Infrastruktur und der Anwendungen.

© Fraunhofer

Vielen Dank für die Aufmerksamkeit

Manuel Wickert M. Sc.,

Abteilungsleiter Energieinformatik und Informationssysteme

Fraunhofer IEE

Goethestr. 27 | 34119 Kassel (Besuchsadresse)
Königstor 59 | 34119 Kassel (Postadresse)

mailto:manuel.wickert@iee.fraunhofer.de | Telefon 0561 7294-369

http://www.iee.fraunhofer.de

© Fraunhofer

Fraunhofer IEE