

Legacy & Innovation, 22.11.2023

Raus aus der Dependency-Wartungshölle  
... zumindest ein bißchen

Sandra Parsick

@SandraParsick

@sparsick@mastodon.social

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

🐙 @sparsick@mastodon.social

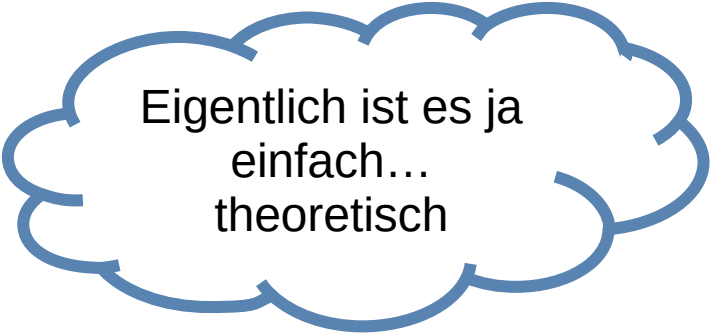
📡 <https://www.sandra-parsick.de>

📻 <https://ready-for-review.dev>



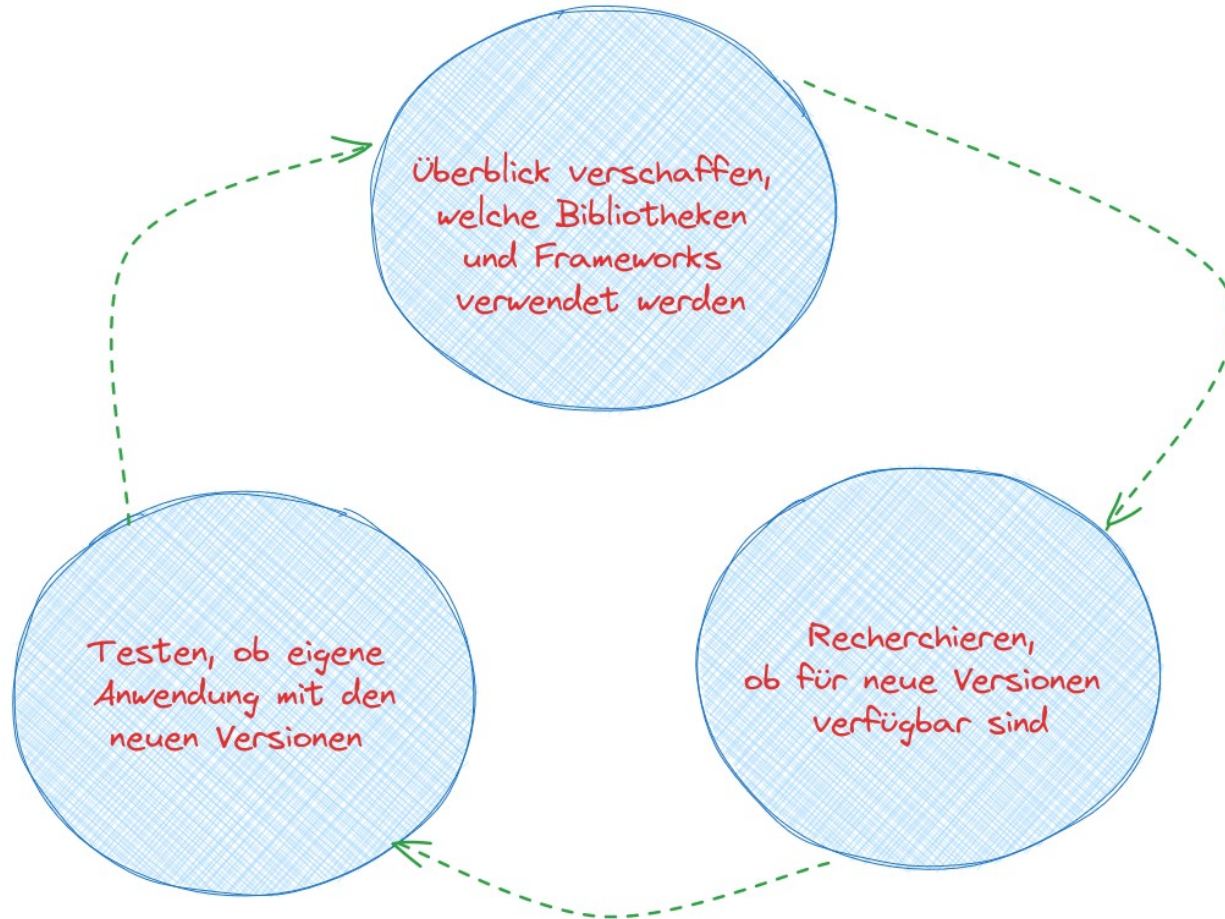
# Willkommen in der Dependency-Hell

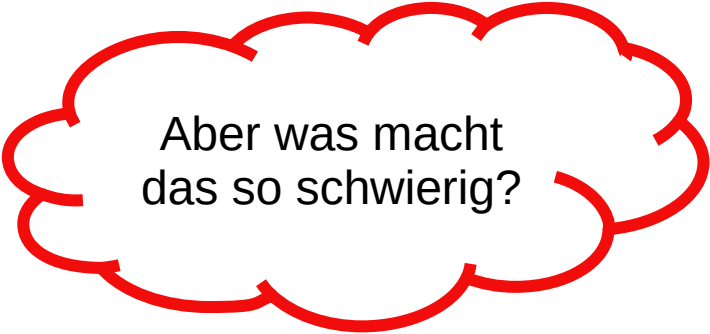




Eigentlich ist es ja  
einfach...  
theoretisch

# Dependency Updates





Aber was macht  
das so schwierig?

# Überblick verschaffen

- Unter Umständen viele Stellen, wo Abhängigkeiten definiert werden:
  - Pom.xml /gradle.kt
  - Package.json
  - Dockerfile
  - Helm Charts
  - Pipeline Scripte (Bsp. Github Actions)

# Recherche

- Unter Umständen, viele Stellen zum Nachschlagen:
  - Maven Central
  - NPM
  - Einzelne Webseiten
- Wie bekomme ich Updates mit?



# Testen

- Habe ich genügend automatisierte Tests?
- Wie groß ist das Vertrauen in den automatisierten Tests?
- Wie hoch ist der manuelle Aufwand?

# Wie Entwickler diese Tätigkeit einstufen?

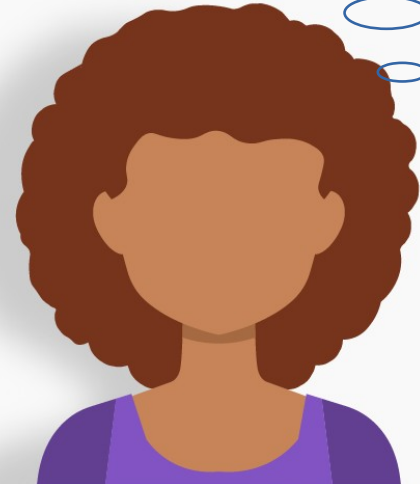


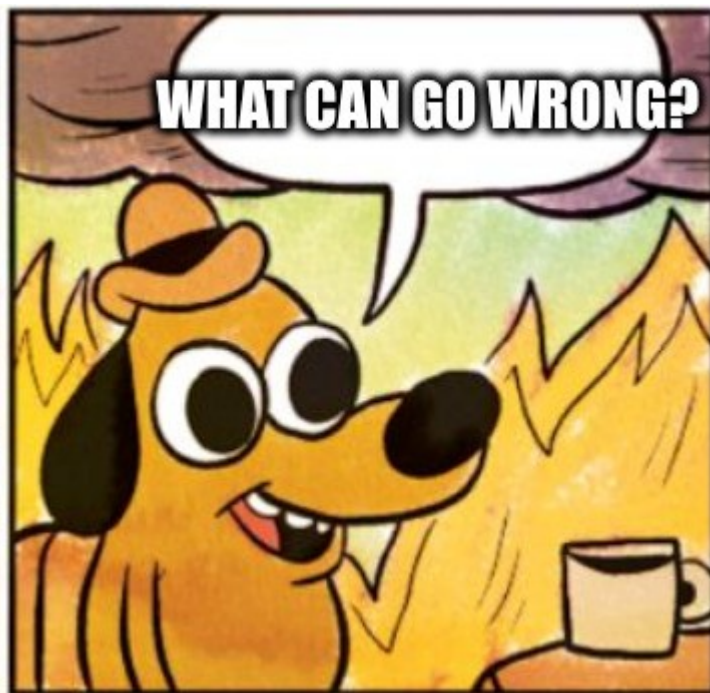
# Was sagt das Management dazu?

Zu aufwendig

Kein messbaren  
Business Value

Die Zeit können wir auch in  
neue Feature stecken





# Konsequenzen

- Unnötige Sicherheitslücken in der eigenen Anwendung wegen veralteten Bibliotheken
- Aufwände unter Umständen in der Zukunft höher, wenn über mehrere Versionen hinweg aktualisiert werden müssen
- Stresspegel höher, wenn unter Druck ein Update gemacht werden muss bei kritischen Sicherheitslücken

# Lösung


Automation is Queen :)




## Dependabot

Automated dependency updates built into GitHub

 **593** followers

 United States of America

 <https://github.com/features/security>

 [support@github.com](mailto:support@github.com)

Verified



MEND  
**Renovate**

Demo



Was ist bei Updates, wo ich viel Code anpassen  
muss/will?

Java 8 → Java 11  
JUnit4 → JUnit5



# Large-Scale Automated Source Code Refactoring

Demo

# Grenzen

- Ramp up bei Projekten, wo Aktualisierungen lange nicht gemacht wurden
- Testen
- Projektsonderlocken

# Fazit

- Automatismen können den Aufwand reduzieren
- Fokus auf harte Nüsse
- Fertige Werkzeuge, so dass Eigenentwicklung sich in Grenzen hält

Fragen?

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

@SandraParsick

@sparsick@mastodon.social

<https://github.com/sparsick/dependency-hell-talk>

# Referenzen

- <https://github.com/spring-projects/spring-petclinic>
- <https://spring-petclinic.github.io/>
- <https://www.arc42.de/overview/>
- <https://adr.github.io/>
- <https://www.unusual.vc/post/ai-software-development>

# Bildnachweise

- [https://pxhere.com/de/photo/970825?utm\\_content=shareClip&utm\\_medium=referral&utm\\_source=pxhere](https://pxhere.com/de/photo/970825?utm_content=shareClip&utm_medium=referral&utm_source=pxhere)
- [https://pxhere.com/de/photo/867723?utm\\_content=shareClip&utm\\_medium=referral&utm\\_source=pxhere](https://pxhere.com/de/photo/867723?utm_content=shareClip&utm_medium=referral&utm_source=pxhere)
- [https://pxhere.com/de/photo/1409158?utm\\_content=shareClip&utm\\_medium=referral&utm\\_source=pxhere](https://pxhere.com/de/photo/1409158?utm_content=shareClip&utm_medium=referral&utm_source=pxhere)
- <https://www.pexels.com/de-de/foto/hausronovio>