

# SEUH 2023

Software Engineering im Unterricht der Hochschulen

## Nachhaltigkeit als Qualitätskriterium von Software

Den Blick auf ressourcen-sparsame  
Softwareentwicklung schärfen

# Motivation



Oliver Radfelder

Softwareentwicklung mit Java  
DevOps und Monitoring  
Testautomatisierung

Karin Vosseberg

Analytische Verfahren der QS  
QS in Vorgehensmodellen  
Testautomatisierung

# Software Engineering an der HS Bremerhaven

- Einführung in Informatik/Wirtschaftsinformatik

  - Einführung in Linux-basierte Serverumgebung

  - Kleine Automatisierungsketten

- Software Engineering (1-3)

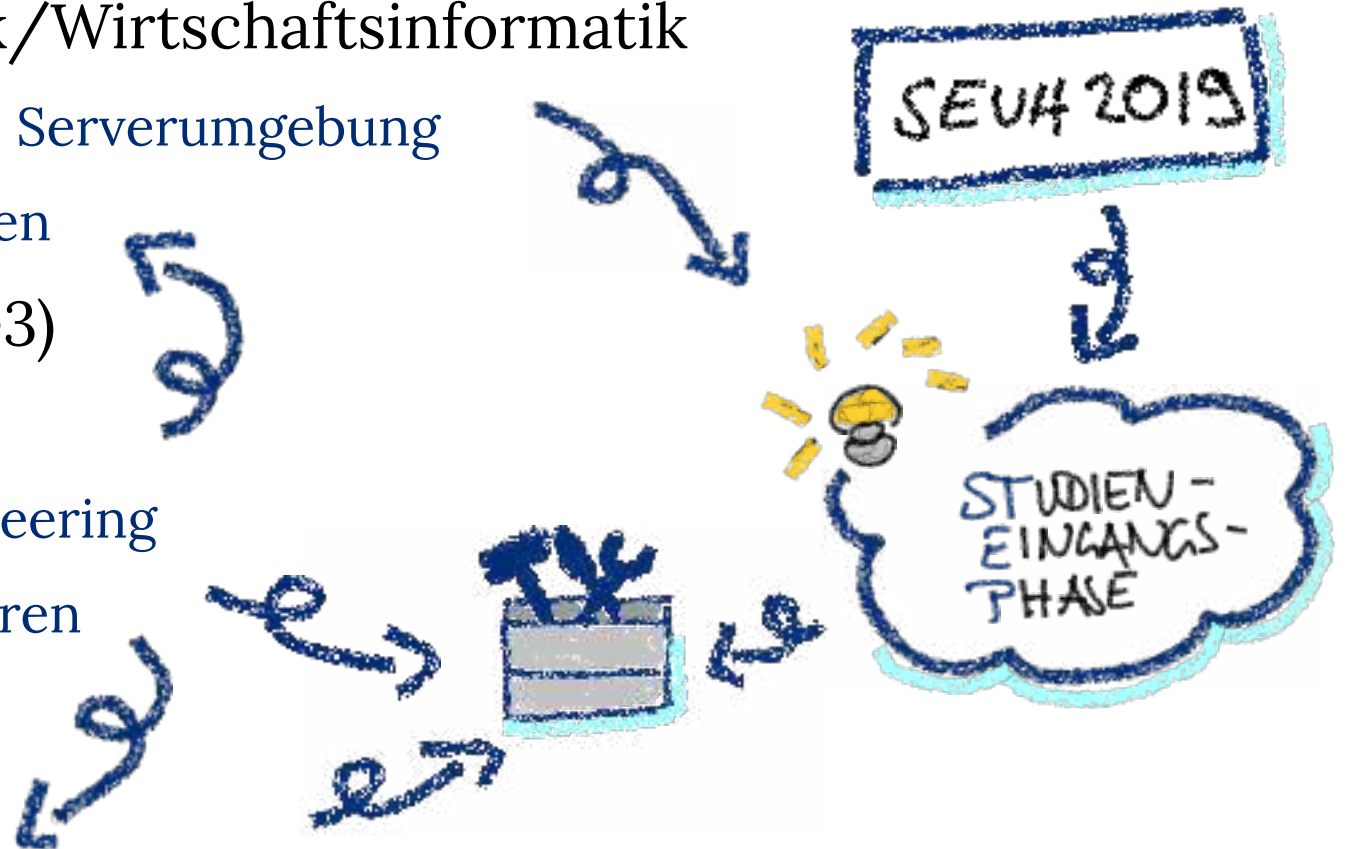
  - SWE 1 - Modellierung

  - SWE 2 - Requirements Engineering

  - SWE 3 - Software Architekturen

- Programmierung (1-2)

  - JAVA - als Hauptsprache



# IT-Infrastruktur des Informatikbereichs

## Linux-basierte Server-Infrastruktur

Docker-Container für jeden User

Git-Repository

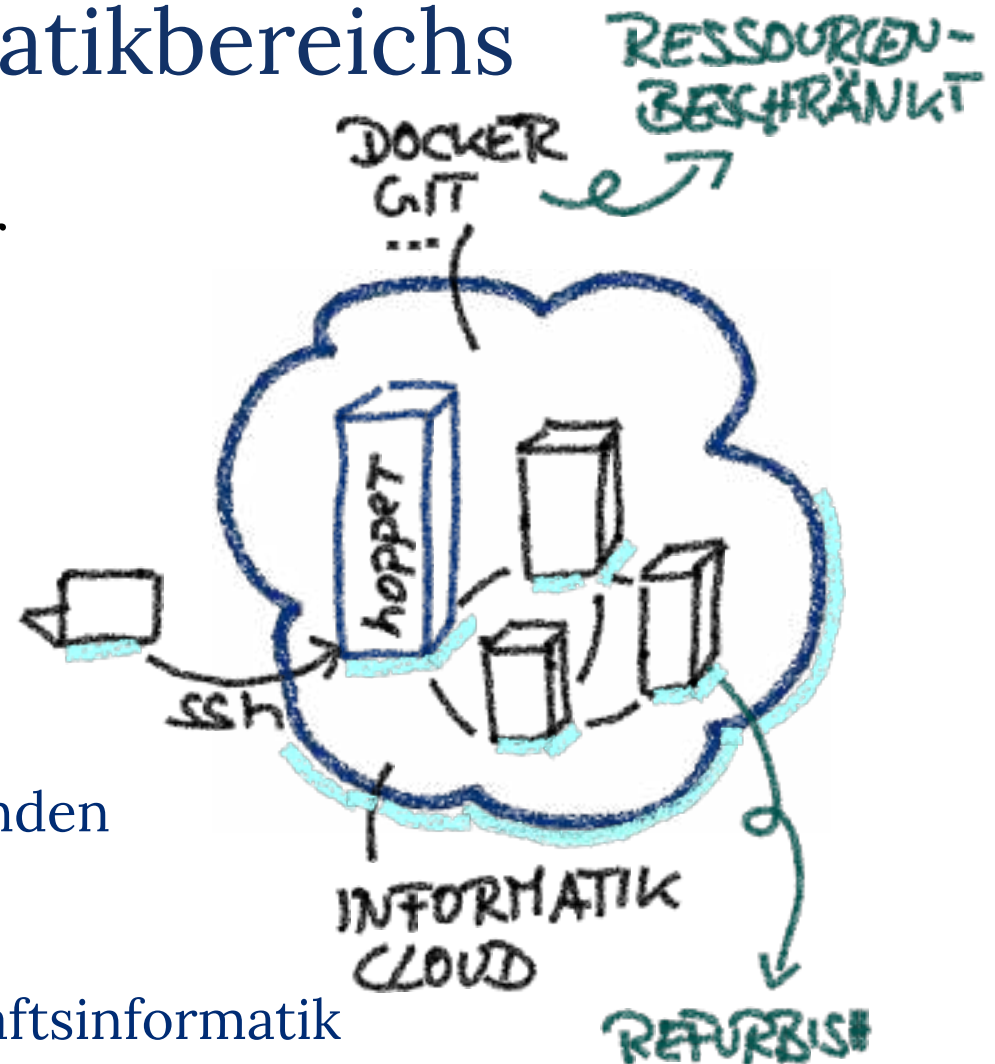
Nachhaltigkeit als Entwicklungskonzept

## IT-Infrastrukturgruppe

Außercurriculare Lerngemeinschaft von Lehrenden, Mitarbeitenden und Studierenden

## Module

Infrastrukturen bzw. Technik für Wirtschaftsinformatik



# Nachhaltigkeit in der Softwareentwicklung

## ■ Ressourcen-Sparsamkeit

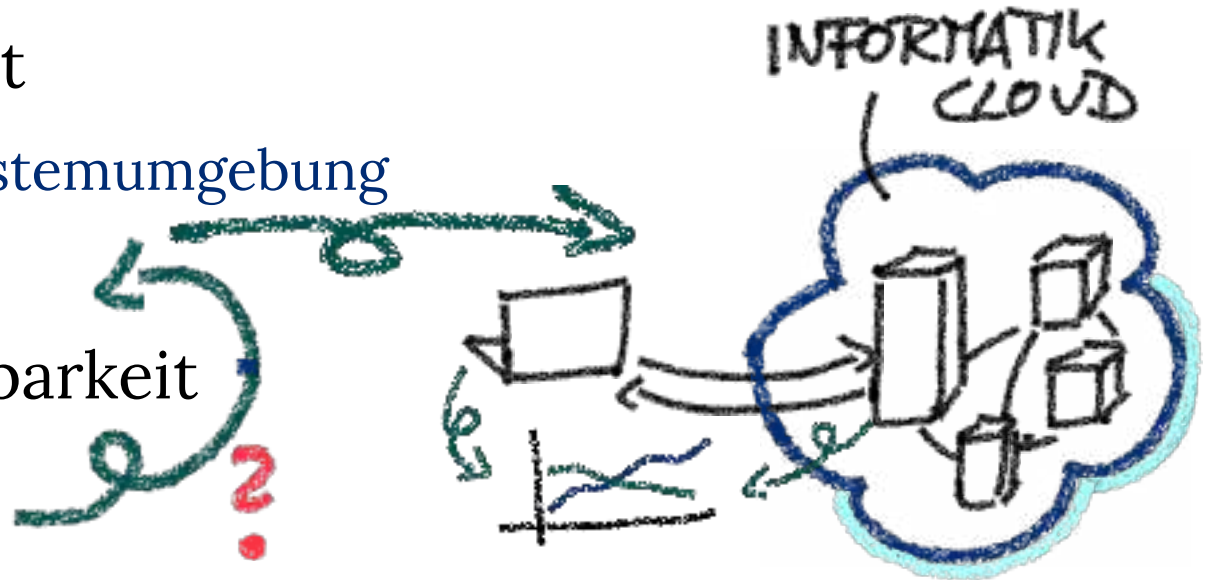
Ressourcen-beschränkte Systemumgebung

Monitoring

## ■ Langlebigkeit und Wartbarkeit

Softwarearchitekturen

Umsetzung in Java



# Grundlagen Qualitätsmanagement (WP)

## ■ Qualitätssicherung in Vorgehensmodellen

Teststufen und Testarten, QS in agilen Teams, DevOps

## ■ Testverfahren

Funktionale und nicht-funktionale Qualitätskriterien

## ■ Testautomatisierung

CI/CD-Prozess, Testwerkzeuge

## ■ Monitoring

Betrachtung von Gesamtsystemen



# Grundlagen Qualitätsmanagement (WP)

## Didaktisches Konzept

Wechselspiel zwischen Vermittlung von Konzepten und gemeinsame Umsetzung in einer Java-Umgebung

## Hands-on Workshops

gemeinsam kleine Beispiele coden

Ergebnisse der analytischen QS nutzen

Diskussionen über Matrix weiterführen

## Plakatausstellung

Studierende bereiten eine Fragestellung mit prototypischer Umsetzung



# Beispiel für Monitoring von Gesamtsystemen

## ■ Ziel des Beispiels

Qualität von Software nicht nur abhängig von der Funktionalität der neu- oder weiterentwickelten Software

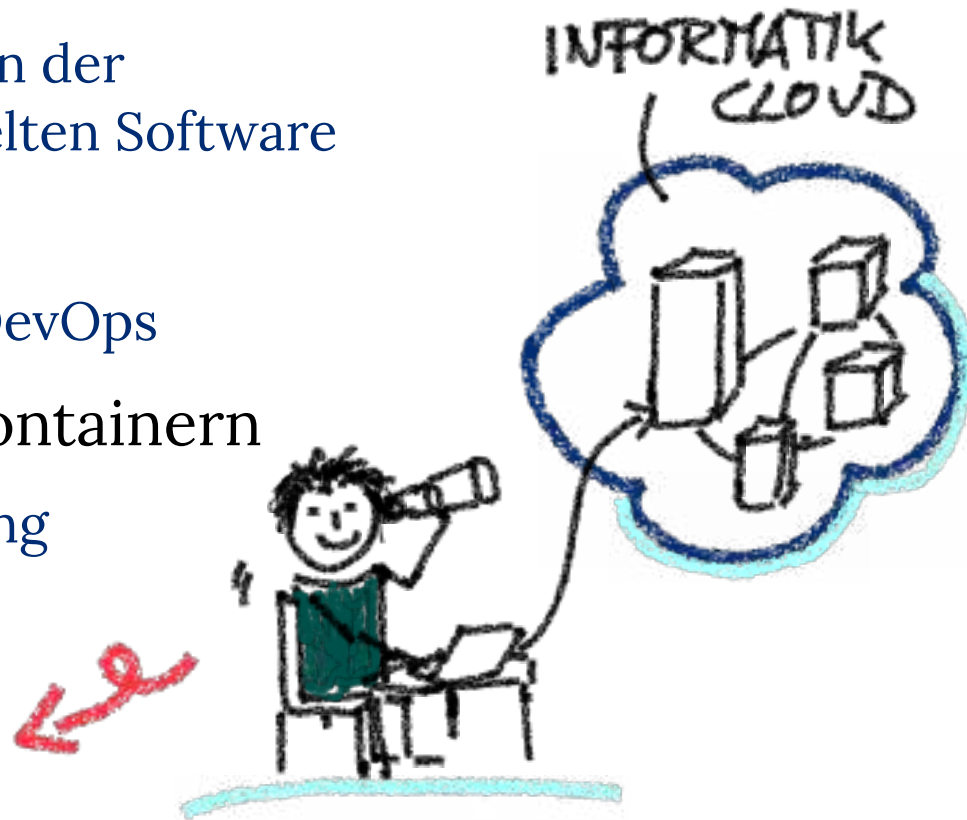
Sensibilisieren für Ressourcen-Verbrauch

Sensibilisieren für das Zusammenspiel von DevOps

## ■ Off-Heap Memory Leaks in Servlet-Containern

Problem beobachtet in der Docker-Umgebung der IT-Infrastruktur

Wie kann ein solches beobachtetes Problem systematisch untersucht werden?





# Off-Heap Memory Leaks in Servlet-Containern

## ■ Web-Anwendung

Docker-Container mit 4 virtuellen CPU und 2 GB virtueller Speicher

Websocket-Endpoint und ContextListener deployt im Tomcat

2 node-Prozesse mit jeweils 2500 geöffneten Websockets

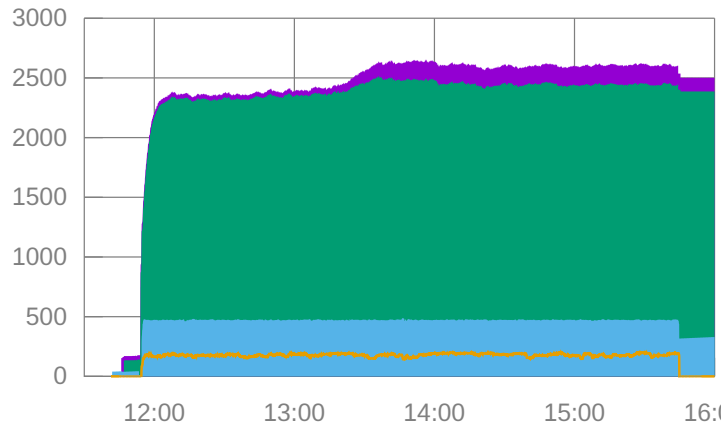
Nach 10 Minuten schlägt der OOM-Killer zu

## ■ Was ist die Ursache?

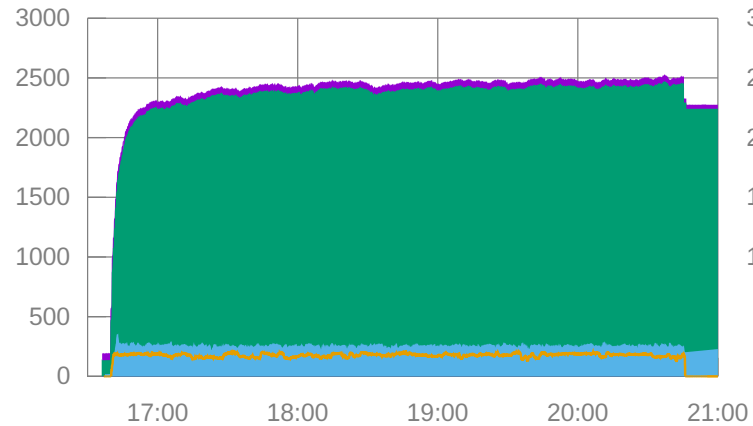
Problem im Tomcat, in der JVM oder liegt es an den beschränkten Ressourcen



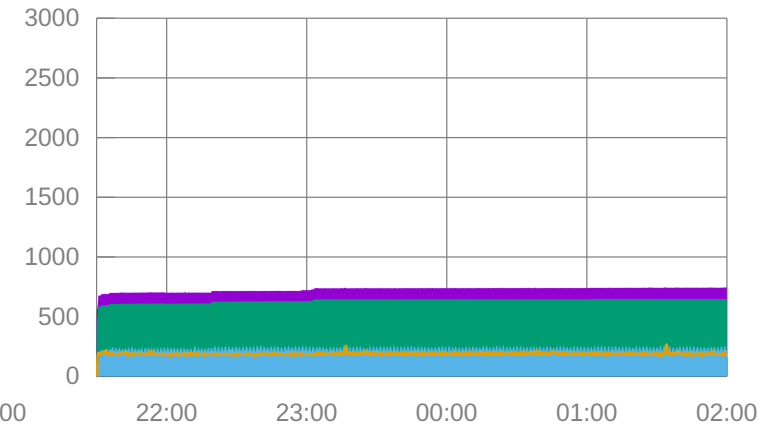
# Off-Heap Memory Leaks in Servlet-Containern



Tomcat



Jetty



Wildfly

## Legende:

- RSS MB Docker
- RSS MB Servlet-Container
- Java Heap
- Message-per-Second

# Off-Heap Memory Leaks in Servlet-Containern

## ■ Technisches Fazit

Tomcat mit deaktivierter *per-message-deflate* ist vergleichbar mit den Ergebnissen von *wildfly* - aber leistungsfähiger in *messages per seconds*

Weitere Experimente mit komplexeren Anwendungen sind notwendig

Gerade bei Last- und Performancetest und Ressourcenverbrauch spielt oft das Gesamtsystem eine große Rolle

## ■ Monitoring gehört zum soliden Handwerkszeug

Informationen aus dem Produktivsystem für die Weiterentwicklung





# Fazit

## ■ Theoretisches Wissen zu Handlungswissen

Studierende erhalten konkrete Ideen,  
zur Umsetzung von Konzepten der QS und  
den Mehrwert für Softwareentwicklung

Mischung zwischen Vorstellung von  
Konzepten und Umsetzung in Java wird  
von den Studierenden positiv bewertet

Große Herausforderung für Studierende  
tief in Java-Konzepte einzusteigen



