

# Stud.IP sichern

Wie kann die Finanzierung von Basis- und Pflegearbeiten den Anforderungen der Digitalisierung angepasst werden?

*Was ist Stud.IP?*



*Stud.IP ist  
kritische Infrastruktur.*

# Betriebssystem für Hochschulen

- ⊙ Stud.IP unterstützt eine Vielzahl ein Prozessen, ohne die die Hochschule handlungsunfähig werden würde.
- ⊙ Stud.IP bietet Dank seiner flexiblen Struktur kostengünstige Erweiterungsmöglichkeiten.
- ⊙ Der Funktionsumfang wächst kontinuierlich.

# Betriebssystem für Hochschulen

Kommunikation

Integration

Lehrunterstützung



Modulverwaltung

Ressourcen

Dokumenten-  
management

Courseware

*„Es läuft doch alles?!“*

*„Ähm, nein ...“*

# Stud.IP *sichern*?

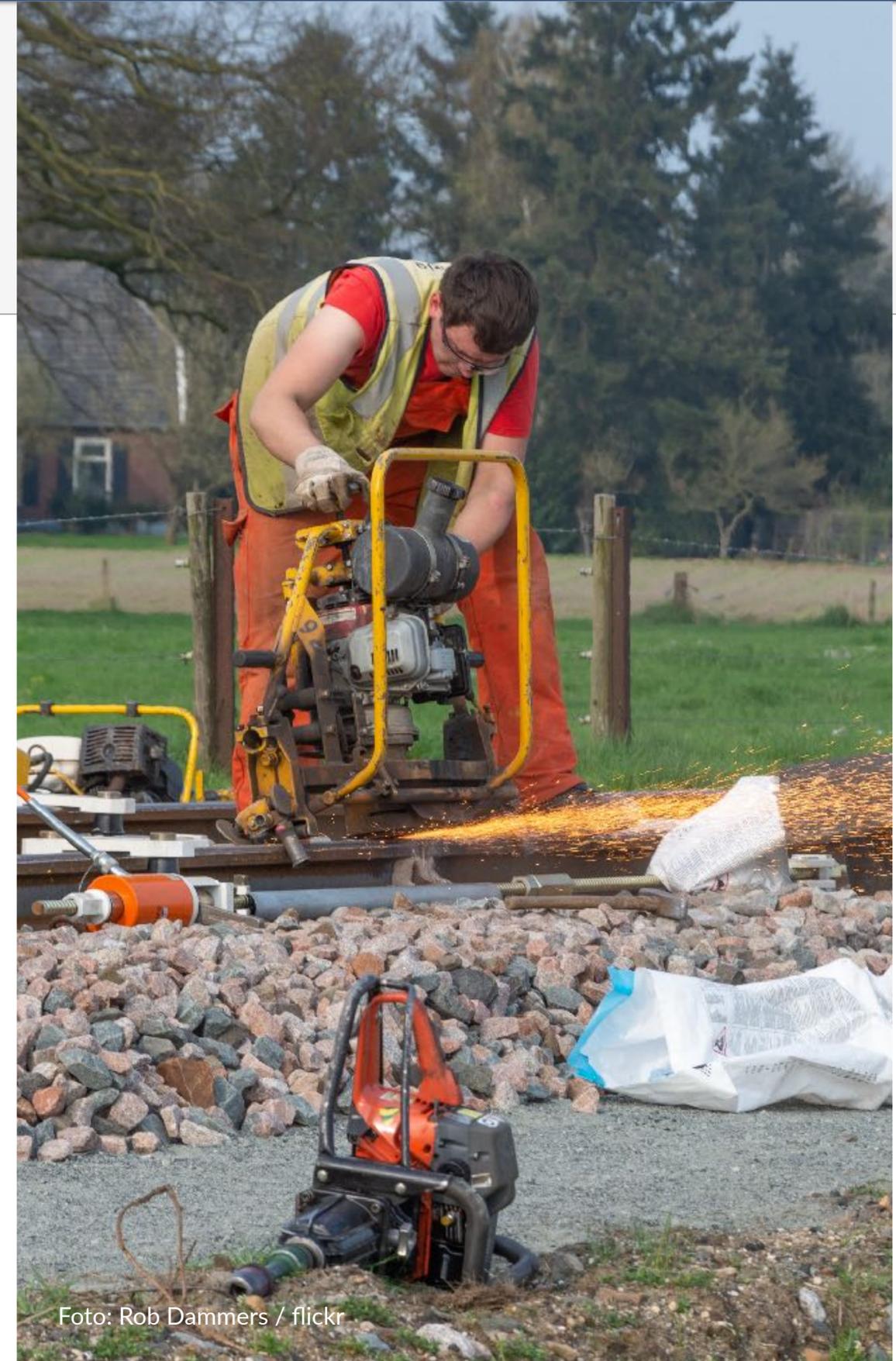


Foto: jseliger2 / flickr

- © Es rollt eine fast unglaubliche Menge an Themen und Chancen auf Stud.IP zu – sie nennen es Digitalisierung

# Stud.IP *sichern*?

- © Der Funktionsumfang wächst, aber die Zahl der Entwickelnden nimmt nur sehr langsam zu



# Stud.IP *sichern*?



Foto: Joegoauk Goa / flickr

- ⦿ Ressourcen gehen allzu oft in Projekte, die nur bestimmte Funktionalitäten *hinzufügen*

# Stud.IP *sichern*?

- ⊙ Kernsanierung funktioniert gut, aber nicht unbedingt im nötigen Tempo



# Stud.IP *sichern*?

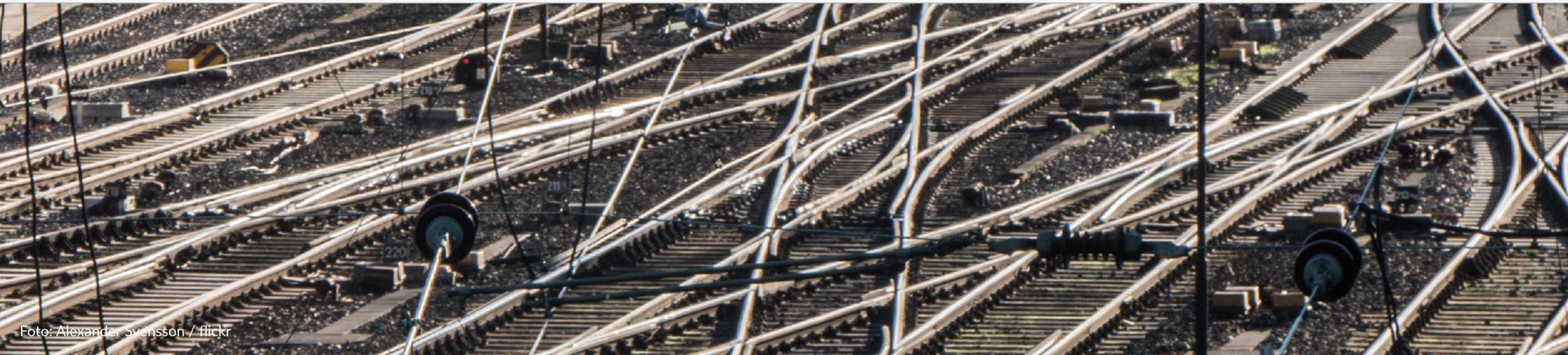


Foto: Alexander Svensson / flickr

- ⦿ Es warten sehr viele gute, nützliche und nötige Ideen in den Schubladen und Köpfen auf ihre Umsetzung

*Warum ist das so?*

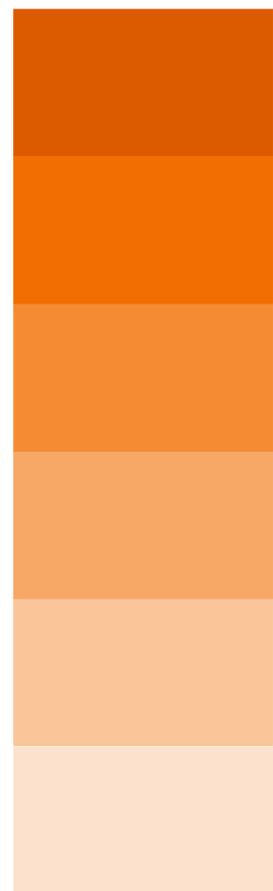
# Kommerziell vs. Open-Source



# Kommerziell vs. Open-Source



## Kommerzielle Software



Funktionserweiterung  
Standortsupport  
Firmengewinne  
Pflege und neue Technologien  
Forschung und Tests  
Marketing



## Open Source Software



Entwicklungsaufträge (neue Features)  
Standortsupport  
Eigenentwicklungen der Hochschule  
**keine Pflege und neue Technologien**  
**keine Forschung und Tests**  
*kein Marketing*

*Stud.IP ist Open-Source.*

*Aus guten Gründen.*

# Vorteile Open-Source

- ⊙ Insgesamt sehr wirtschaftlich
- ⊙ Große Mitbestimmung möglich
- ⊙ Maximale Transparenz
- ⊙ Keine Abhängigkeit vom Hersteller und dessen Strategie
- ⊙ Geteilter Aufwand, Erfolg kommt allen zugute
- ⊙ Hohe Entwicklungsgeschwindigkeit (wenn Ressourcen vorhanden)

# Nachteile Open-Source

- ⊙ Planbare Weiterentwicklung ist schwierig
- ⊙ Was nicht zwingend nötig ist, geschieht oft langsam
- ⊙ Jedes Entwicklungsprojekt muss einzeln definiert werden
- ⊙ Keine automatische Pflege – auch wenn das gern gedacht wird
- ⊙ Support an Standorten ist keine Systempflege
- ⊙ Budgets werden gedrückt, um mehr zu sparen als möglich

*„Man müsste mal ...“*



# Kernentwickler wissen was wichtig ist

- ⦿ Hilfeseiten
- ⦿ Neue Technologien
- ⦿ Responsive Design
- ⦿ Dokumentation
- ⦿ Schnittstellen
- ⦿ Zukunftsfähigkeit/Sicherheit
- ⦿ Vorarbeiten für kommende Entwicklungen
- ⦿ Konsistenz/UI
- ⦿ Apps

Wenn jede Hochschule  
5.000 € pro Jahr gäbe ...

... ohne dass der Zweck  
schon genau festgelegt wäre ...

... mit dem Ziel einer höheren  
Qualität und Geschwindigkeit:

# Was hätten wir alle davon?

- ⊙ Entwicklung wird wirtschaftlicher
- ⊙ Schnellere Entwicklungszyklen
- ⊙ Ausbau des Standortvorteils Open-Source
- ⊙ Reduzierten Nutzersupport
- ⊙ Weniger Standortaufwand
- ⊙ Einfachere Wartung
- ⊙ Zukunftssicherheit
- ⊙ Modernisierung
- ⊙ Interoperabilität mit modernster Softwarelandschaft



Foto: Rob Dammers / flickr

*Thank you for travelling  
with Stud.IP.*