

Technische Schulden in Softwareunternehmen

Auswirkungen auf Motivation, Zusammenarbeit & Teamdynamik

Fabrizio Monaco, 12.02.2025

Gliederung

01

Definition & Entstehung

02

Auswirkungen auf Software

03

Auswirkungen auf Teams

04

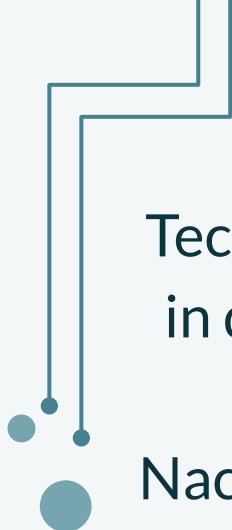
Fazit & Ausblick

01

Technische Schulden

Definition





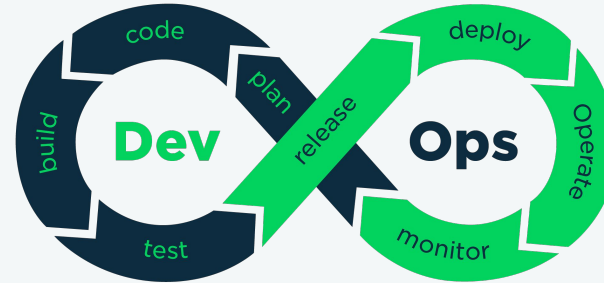
Technische Schulden bezeichnen
in der Softwareentwicklung die
zusätzlichen Kosten für
Nachbesserungen, die entstehen,
wenn zugunsten einer schnelleren
Umsetzung auf sauberen und
effizienten Code verzichtet wird.

—SonarSource



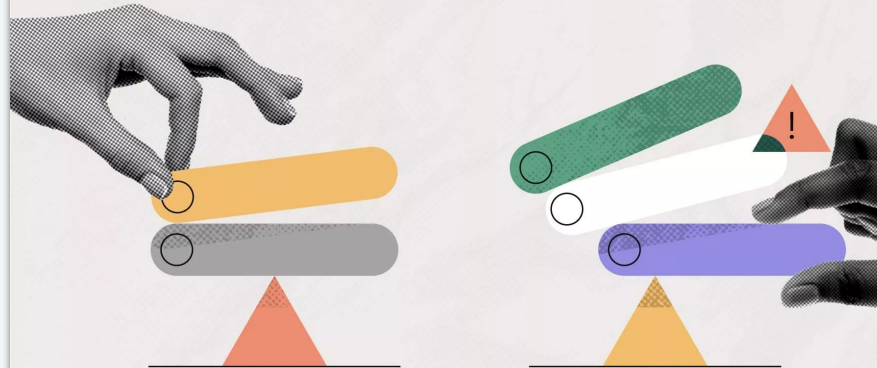
Definition

- Releasezyklen beschleunigen
- Niedrigere Priorisierung von Code-Qualität
- Erhöhte Wartungskosten im Nachhinein



Klassifizierung

Arten von technischen Schulden



Absichtliche Schulden

treten auf, wenn ein Unternehmen die bewusste Entscheidung trifft, die für den Augenblick praktischste Vorgehensweise zu verfolgen und die Entscheidung nicht mit Weitblick für die Zukunft zu treffen.

Unabsichtliche Schulden

treten auf, wenn ein Unternehmen einen unvermeidbaren Fehler macht.

Klassifizierung

Die vier Quadranten zur Einteilung technischer Schulden



Technische Schulden

Entstehung



Entstehung



Zeitdruck

Begrenzte Ressourcen
oder enge Deadlines

Entstehung



Zeitdruck

Begrenzte Ressourcen
oder enge Deadlines



Mangelnde Erfahrung

Verminderte Code
Qualität

Entstehung



Zeitdruck

Begrenzte Ressourcen
oder enge Deadlines



Mangelnde Erfahrung

Verminderte Code
Qualität



Unklare Anforderungen

Schnelle Workarounds
durch wechselnde
Anforderungen

02

Auswirkungen auf Software



Auswirkungen auf Software



Verlangsamte Entwicklung

Hoher Zeitaufwand durch
Erlangung von Code
Verständnis

Auswirkungen auf Software



Verlangsamte Entwicklung

Hoher Zeitaufwand durch
Erlangung von Code
Verständnis



Geringere Qualität

Fehleranfälliger Code fördert
Instabilität

Auswirkungen auf Software



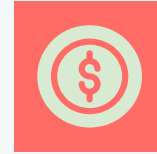
Verlangsamte Entwicklung

Hoher Zeitaufwand durch
Erlangung von Code
Verständnis



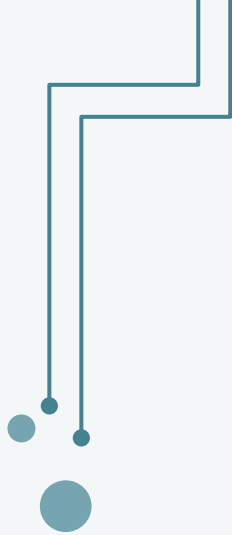
Geringere Qualität

Fehleranfälliger Code fördert
Instabilität



Erhöhte Wartungskosten

Budgetverlagerung und
Verlust von
Weiterentwicklungspotenzial



“Technische Schulden zu
begleichen kostet 3,61 \$ pro
Zeile Code.”

Y. Guo, R. O. Spínola, und C. Seaman



03

Auswirkungen auf Teams

Motivation & Arbeitszufriedenheit





Menschen machen Software

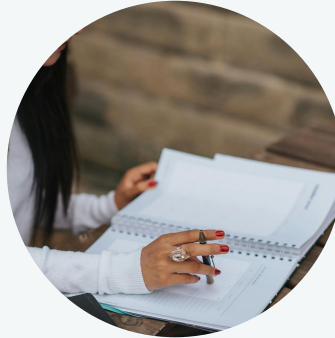
Empirische Studie aus 2017:
Motivationsfaktoren für Mitarbeiter in der
globalen Softwareentwicklung

Auswirkungen auf Teams



Autonomie

Treffen eigenständiger
Entscheidungen



Kompetenz

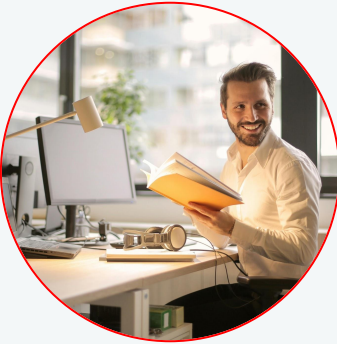
Fachliche
Weiterentwicklung



Soziale Eingebundenheit

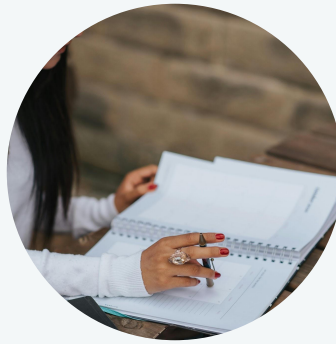
Zugehörigkeit und
Anerkennung im Team

Auswirkungen auf Teams



Autonomie

Treffen eigenständiger
Entscheidungen



Kompetenz

Fachliche
Weiterentwicklung



Soziale Eingebundenheit

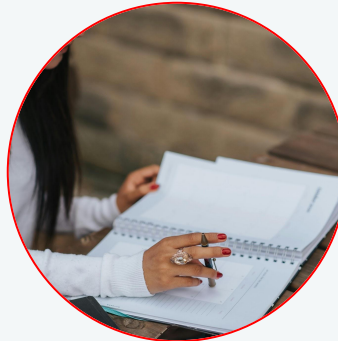
Zugehörigkeit und
Anerkennung im Team

Auswirkungen auf Teams



Autonomie

Treffen eigenständiger
Entscheidungen



Kompetenz

Fachliche
Weiterentwicklung



Soziale Eingebundenheit

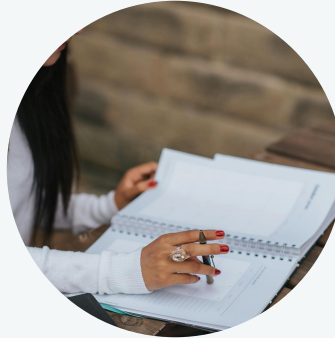
Zugehörigkeit und
Anerkennung im Team

Auswirkungen auf Teams



Autonomie

Treffen eigenständiger
Entscheidungen



Kompetenz

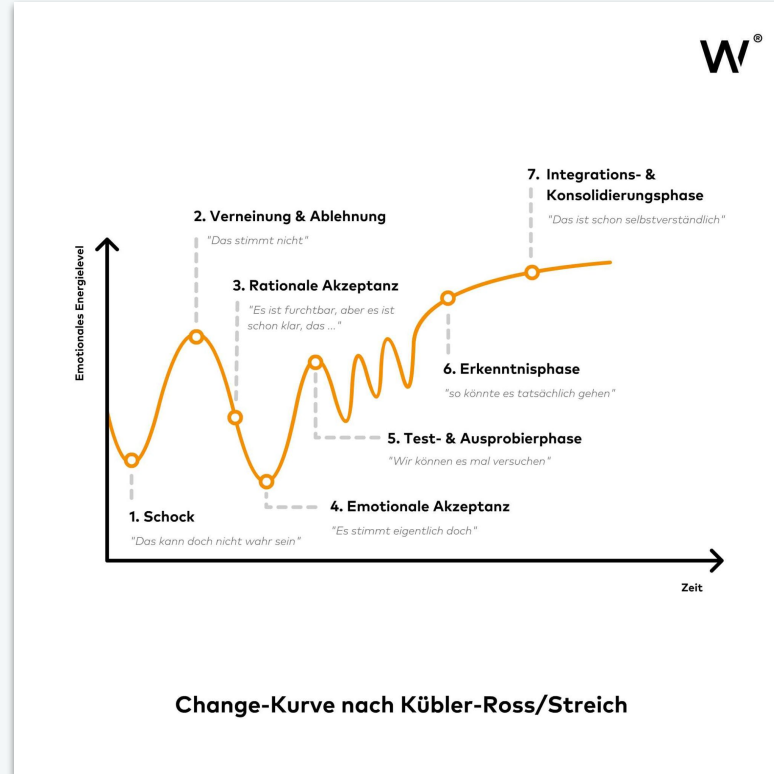
Fachliche
Weiterentwicklung



Soziale Eingebundenheit

Zugehörigkeit und
Anerkennung im Team

Reaktion auf Veränderung



Auswirkungen auf Teams



Widerstand und Frustration

Ohne Unterstützung sinken
Motivation und Produktivität

Auswirkungen auf Teams



Widerstand und Frustration

Ohne Unterstützung sinken
Motivation und Produktivität



Konflikte und Spannungen

Unterschiedliche Prioritäten
führen zu Teamkonflikten

Auswirkungen auf Teams



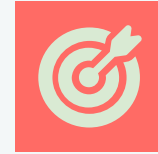
Widerstand und Frustration

Ohne Unterstützung sinken
Motivation und Produktivität



Konflikte und Spannungen

Unterschiedliche Prioritäten
führen zu Teamkonflikten



Beeinträchtigte Unternehmenskultur

Misstrauen und
Qualitätsverlust durch
schlechte Schuldenstrategie

04

Fazit & Ausblick



Fazit

- Langfristige Auswirkungen
- Teamdynamik und Motivation
- Strategien werden notwendig



Ausblick

Clean Code	Die Anwendung von Prinzipien des Clean Code sorgt für lesbaren, wartbaren und testbaren Code, der weniger fehleranfällig ist
Refactoring	Regelmäßiges Refactoring verbessert die Codequalität und hilft, technische Schulden abzubauen.
Test-Driven Development	Höhere Testabdeckung erreicht und Senkung der Fehlerquote.

Vielen Dank!

