Consulting
Development
Solutions

generic//de
the clean code company

Lessons Learned?

A retrospective view of the Clean Code Days 2016

30.06.2016

Michael Puder

generic//de

Ralf Schoch

Consulting
Development
Solutions


AGILE


clean code
developer


Sprint
Backlog
Team
ScrumMaster
Standup
Scrum
User-Stories
Sprint-Planning
Time-to-Market
Anforderungen
Software-Entwicklung
Retrospektive
ProductOwner
Review

Herausforderungen

Wer macht mit?	Was sind die Basics?	Was ist mit Legacy?
Kommunikation?	Reviews?	Nur Alles oder Nichts „clean“?

Lessons Learned?

Nach zwei Tagen Clean Code Days:

Was ist jetzt eigentlich Clean Code Development?

...und wozu ?

Am Ende

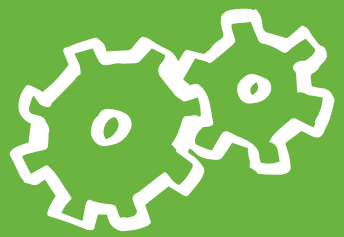
...geht es nur darum:

Softwarequalität

Softwarequalität

Funktionalität

Zuverlässigkeit



Benutzbarkeit

Effizienz

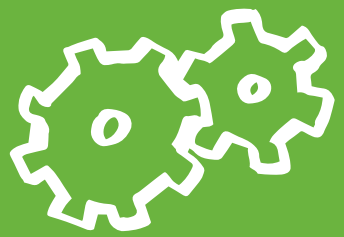
Sicherheit

Skalierbarkeit

Äußere Softwarequalität

Funktionalität

Zuverlässigkeit



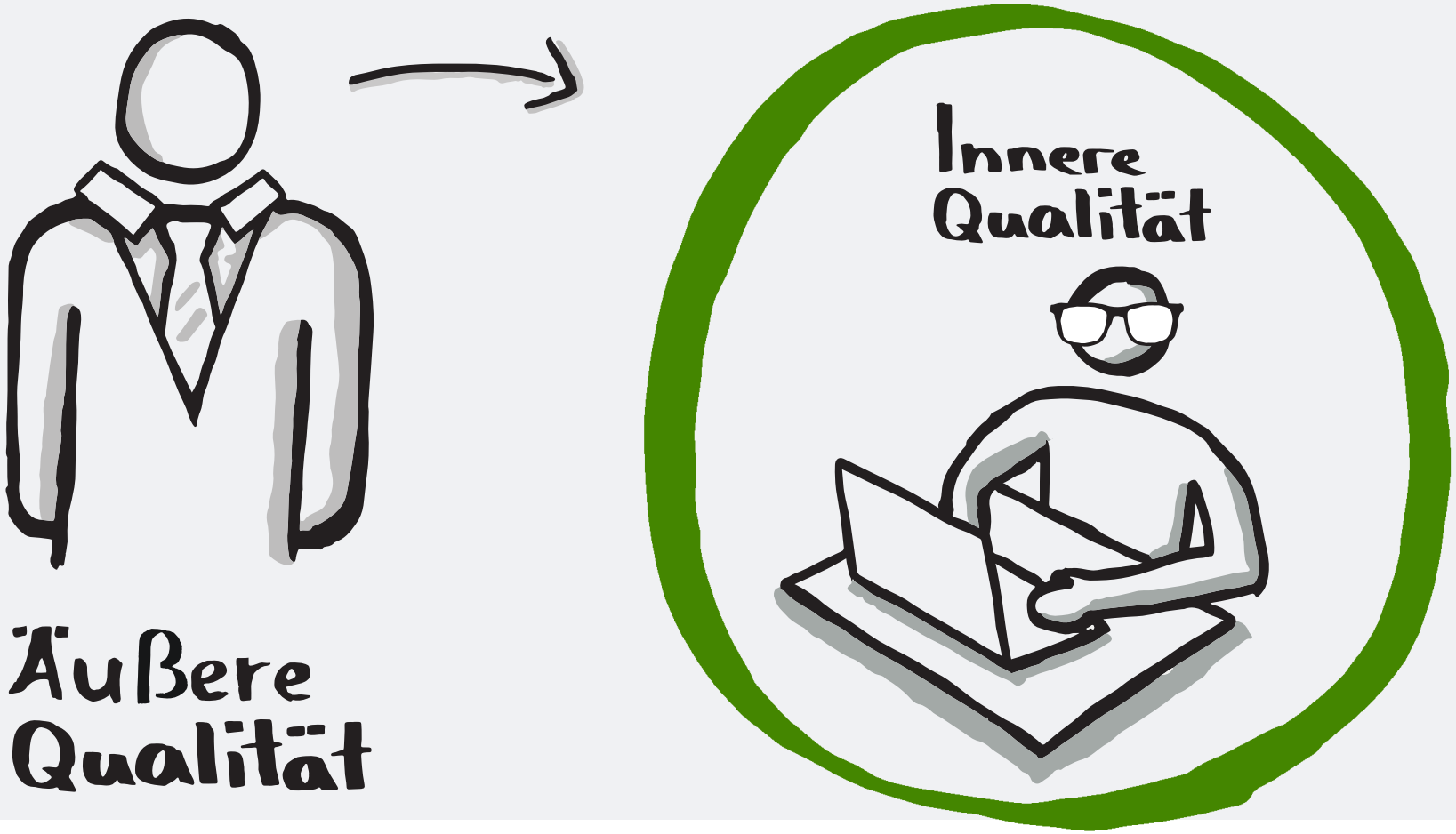
Benutzbarkeit

Effizienz

Sicherheit

Skalierbarkeit

Innere Softwarequalität



Innere Softwarequalität

Lesbarkeit

Nachvollziehbarkeit

Testbarkeit

Wiederverwend-barkeit

Evolvierbarkeit

Warum ist innere Softwarequalität wichtig ?

80%

Code lesen
und verstehen

20%

Code schreiben

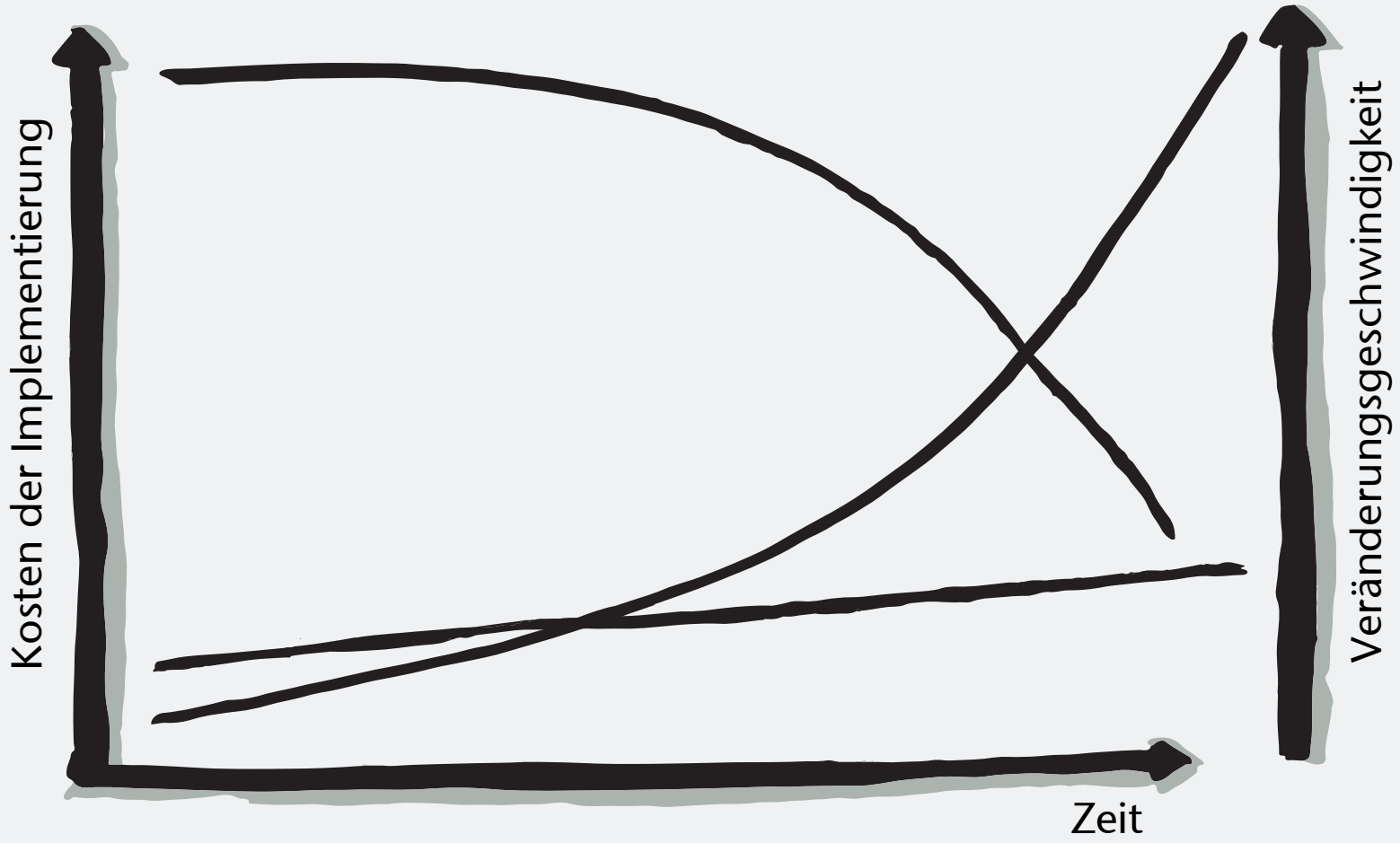
Innere Softwarequalität
ist das Fundament für den
nachhaltigen Erfolg
der Softwareentwicklung

Werden Zeit und Umfang von äußeren Einflüssen festgelegt, dann sinkt die Innere Qualität

Technische Schuld ist
eine aufgeschobene
Investition
in Innere Qualität

Die Kosten für neue Features
z.B. steigen. Dramatisch!

Kosten zur Implementierung eines neuen Features



Der Super-GAU der Softwareentwicklung

Also

Clean Code Development

Aber: wie erhöhe ich damit die Softwarequalität ?

Ziel:

Verantwortungsvolle und professionelle Entwickler

Qualität ist nicht verhandelbar !

Clean Code Developer

Werte
Tugenden
Prinzipien & Praktiken

Clean Code Developer

Werte

Tugenden

Prinzipien & Praktiken

Werte


Evolvier-
barkeit
Korrektheit
Produktions-
effizienz
Kont. Ver-
besserung

Clean Code Developer

Werte

Tugenden

Prinzipien & Praktiken

Tugenden (Auszug)

Tue nur das Nötigste	Isoliere Aspekte	Minimiere Abhängigkeiten
Wertschätze Qualität	Halte Ordnung	Bleib am Ball

Beispiel: Isoliere Aspekte

Don‘t Repeat Yourself (DRY)

Separation of Concerns (SoC)

Single Level of Abstraction (SLA)

Single Responsibility Principle (SRP)

Interface Segregation Principle (ISP)

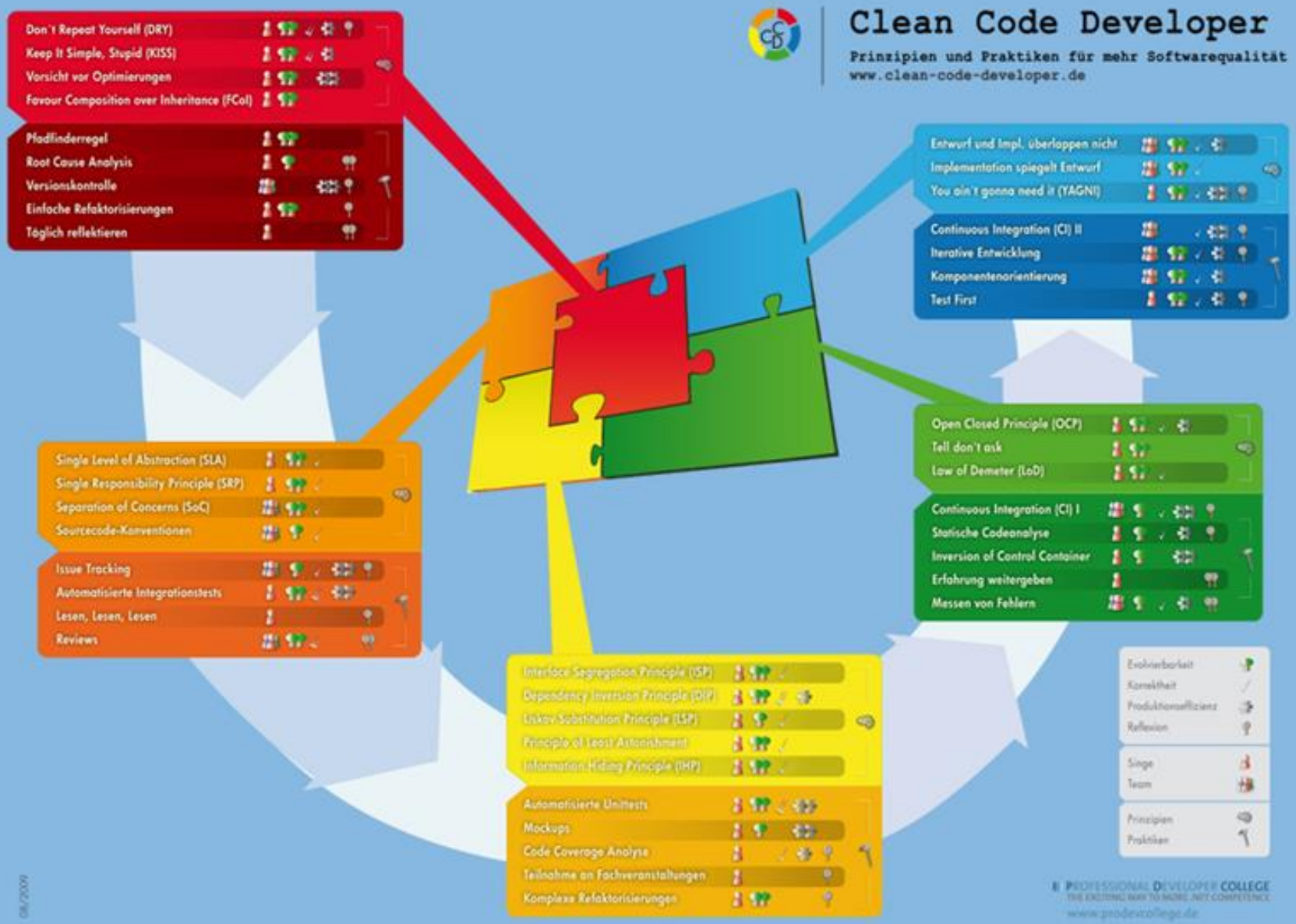
Entwurf und Implementation überlappen nicht

Clean Code Developer

Werte

Tugenden

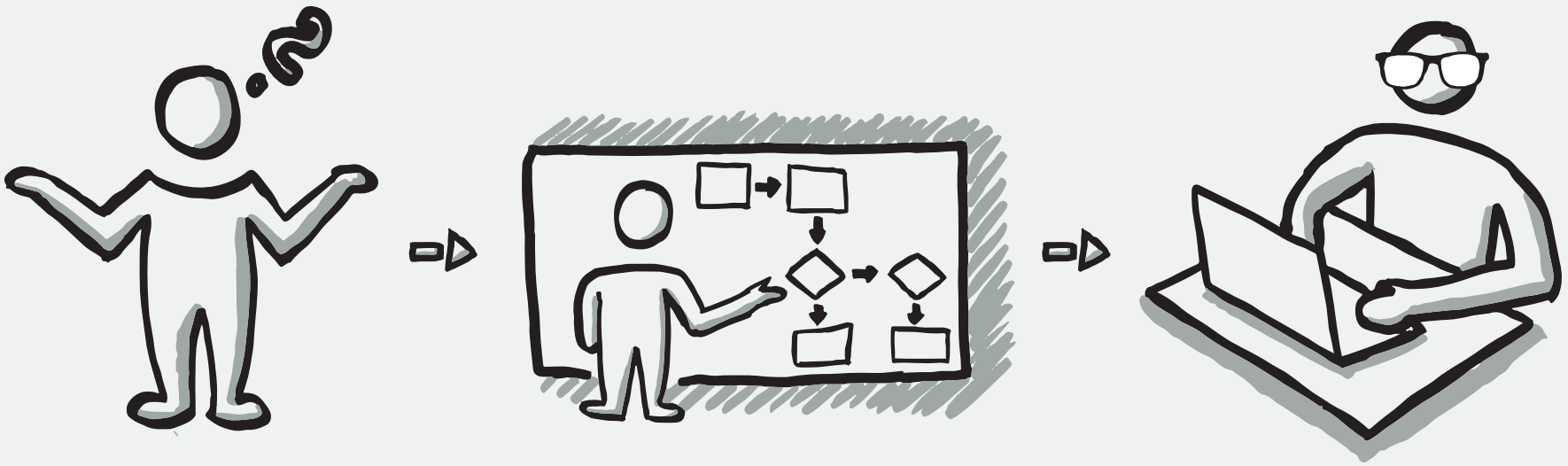
Prinzipien & Praktiken


Clean Code Developer
Prinzipien und Praktiken für mehr Softwarequalität
www.clean-code-developer.de
Don't Repeat Yourself (DRY)
Keep It Simple, Stupid (KISS)
Vorsicht vor Optimierungen
Favour Composition over Inheritance (FCoI)
Pfadfinderregel
Root Cause Analysis
Versionskontrolle
Einfache Refaktorisierungen
Täglich reflektieren
Entwurf und Impl. überlappen nicht
Implementation spiegelt Entwurf
You ain't gonna need it (YAGNI)
Continuous Integration (CI) II
Iterative Entwicklung
Komponentenorientierung
Test First
Single Level of Abstraction (SLA)
Single Responsibility Principle (SRP)
Separation of Concerns (SoC)
Sourcecode-Konventionen
Issue Tracking
Automatisierte Integrationstests
Lesen, Lesen, Lesen
Reviews
Open Closed Principle (OCP)
Tell don't ask
Law of Demeter (LoD)
Continuous Integration (CI) I
Statische Codeanalyse
Inversion of Control Container
Erfahrung weitergeben
Messen von Fehlern
Interface Segregation Principle (ISP)
Dependency Inversion Principle (DIP)
Liskov Substitution Principle (LSP)
Principle of Least Astonishment
Information Hiding Principle (IHP)
Automatisierte Unittests
Mockups
Code Coverage Analyse
Teilnahme an Fachveranstaltungen
Komplexe Refaktorisierungen
Evolvierbarkeit
Korrektheit
Produktionseffizienz
Reflexion
Singe
Team
Prinzipien
Praktiken
PROFESSIONAL DEVELOPER COLLEGE
THE EXCITING WAY TO MORE .NET COMPETENCE
www.prodevcollege.de

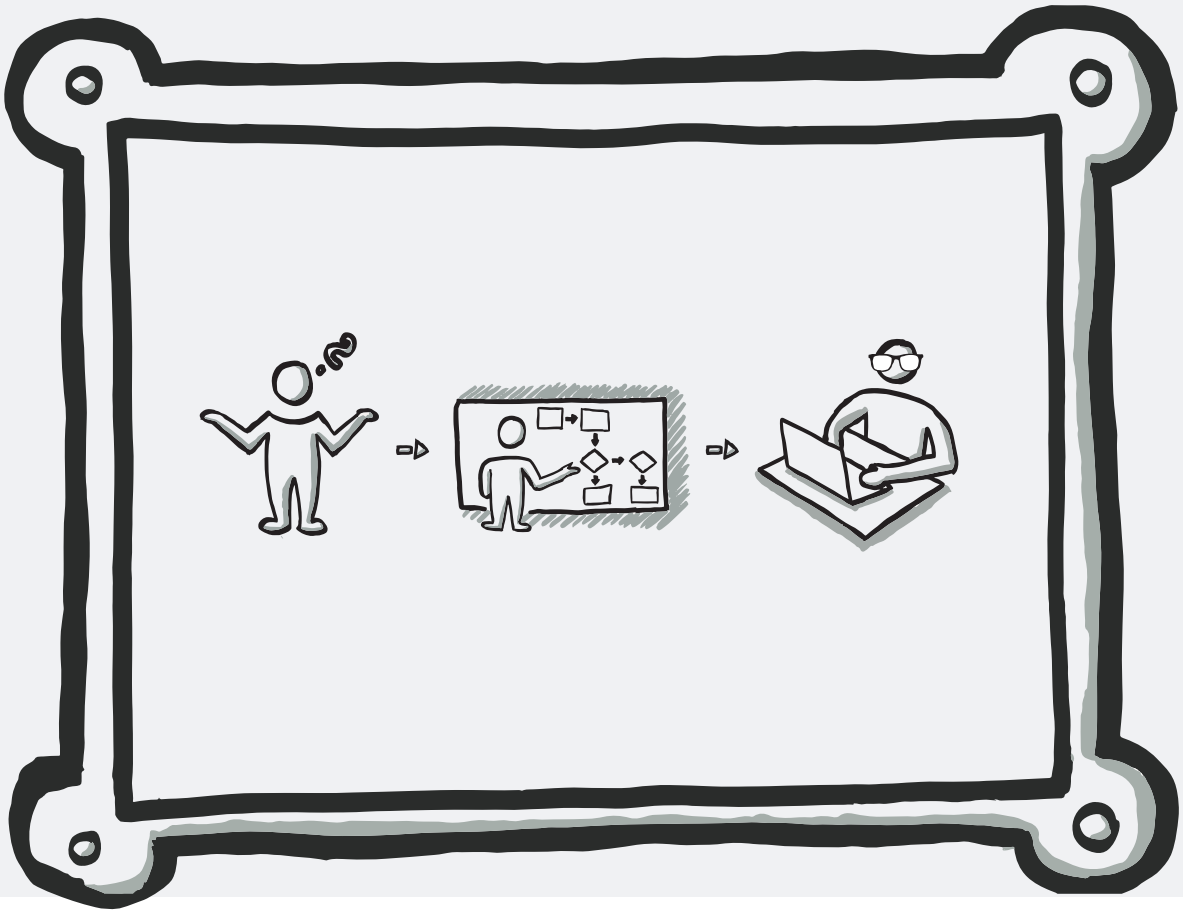
..und damit sind alle Probleme gelöst ?

Clean Code bezieht sich auf den handwerklichen Anteil der Arbeit eines Entwicklers...

... jedoch ist der gesamte Entwicklungsprozess zu betrachten...



… und es benötigt Bewusstsein in den übergeordneten Entscheidungsebenen.



... was nehmen Sie also mit?

Herausforderungen

Wer macht mit?

Was sind die Basics?

Was ist mit Legacy?

Kommunikation?

Reviews?

Nur Alles oder Nichts „clean“?

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)

Kunde
Management
Co-Worker

Argumente?

Metriken

Qualität als Invest

„Developer Scalability“

Niedrigere Kosten pro Bug

Kundenzufriedenheit

„Sloppiness Short Term Boost“

vs.

„Clean Code Long Term Benefit“

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)

SOLID – Grundlagen verstehen

Robert C. Martin:
Clean Code

Single Responsibility

Open Closed

Liskov Substitution

Interface Segregation

Dependency Inversion

Single Responsibility

Open Closed

Liskov Substitution

Interface Segregation

Dependency Inversion

Westphal/Lieser:
clean-code-developer.de

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)

SOLID – Grundlagen verstehen

Legacy – Clean Cobol! Clean C++!

Auch Legacy Sprachen oder Code kann clean werden.

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)

SOLID – Grundlagen verstehen

Legacy – Clean Cobol! Clean C++!

Clean Communication

Guter Code bedingt gute Kommunikation

Code schreiben IST Kommunikation

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)

SOLID – Grundlagen verstehen

Legacy – Clean Cobol! Clean C++!

Clean Communication

Reviews, aber gewaltfrei und respektvoll

Gut vorbereitet sein

Immer den Teamgedanken haben

Berufsehre

Antworten

„The Lone Stranger“ – alleine geht es nicht (Keynote)	SOLID – Grundlagen verstehen	Legacy – Clean Cobol! Clean C++!
Clean Communication	Reviews, aber gewaltfrei und respektvoll	Schrittweise etablierbar

Es
MUSS NICHT ALLES
clean sein.

Ein letztes Fazit...

Üben, üben, üben ...


Sitzt
Ihnen ihr
BAD CODE
im Nacken,
dann kontaktieren Sie uns...
+49 172 539 85 0
michael.puder@generic.de
ralf.schoch@codeso.de
www.generic.de
www.codeso.de