

Die Hirnpsychologischen und Verha ltenspsychologischen Hintergründe, was guten von schlechten Code unterscheidet [GERMAN POST]

Gastbeitrag von Rick Janda

Ich hab mit Rick das [2 Tage Clean Code Training](#) entwickelt das wir x-fach schon öffentlich und firmenintern gegeben haben. Bei jeder Begegnung mit Rick lerne ich immer was, so wie du gleich beim Lesen auch was Lernen wirst ?

Hier kommt Rick's Beitrag:

1. Begrenztes Kurzzeitgedächtnis

Wir Menschen haben ein sehr begrenztes Kurzzeitgedächtnis. Man geht heute davon aus, dass wir 7 (+/-2) von ähnlichen Dingen gleichzeitig im Kurzzeitgedächtnis behalten können.

Für Code bedeutet das:

Alles, was ich nicht im Langzeitgedächtnis habe und nicht auf dem Bildschirm sehen kann, muss ich durch Navigieren ins Bild holen. Daher ist eine kurze Navigationsweite so wichtig für Dinge, die ich nur zusammen verstehen kann. Sobald ein Sachverhalt zu viele Einheiten enthält und damit nicht mehr ins Kurzzeitgedächtnis passt, hat ein **Nicht-Experte**, der das nicht mit entsprechendem Wissen im Langzeitgedächtnis kompensieren kann, eigentlich kaum noch eine Chance, ein komplexen Sachverhalt im Code zu verstehen.

Alle möglichen Prinzipien wie: Geringe Kopplung, Hohe Kohäsion, Single Level of Abstraction, Single Responsibility, Separation of Concerns, DAMP over DRY für Tests und auch Smells wie Side-Effect, Hidden temporal Coupling, Lange Methode, Grosse Klasse, Zu viele Dependencies sind alles primär Konsequenzen aus dieser starken Begrenzung unseres Kurzzeitgedächtnisses.

2. Änderungen unter Angst

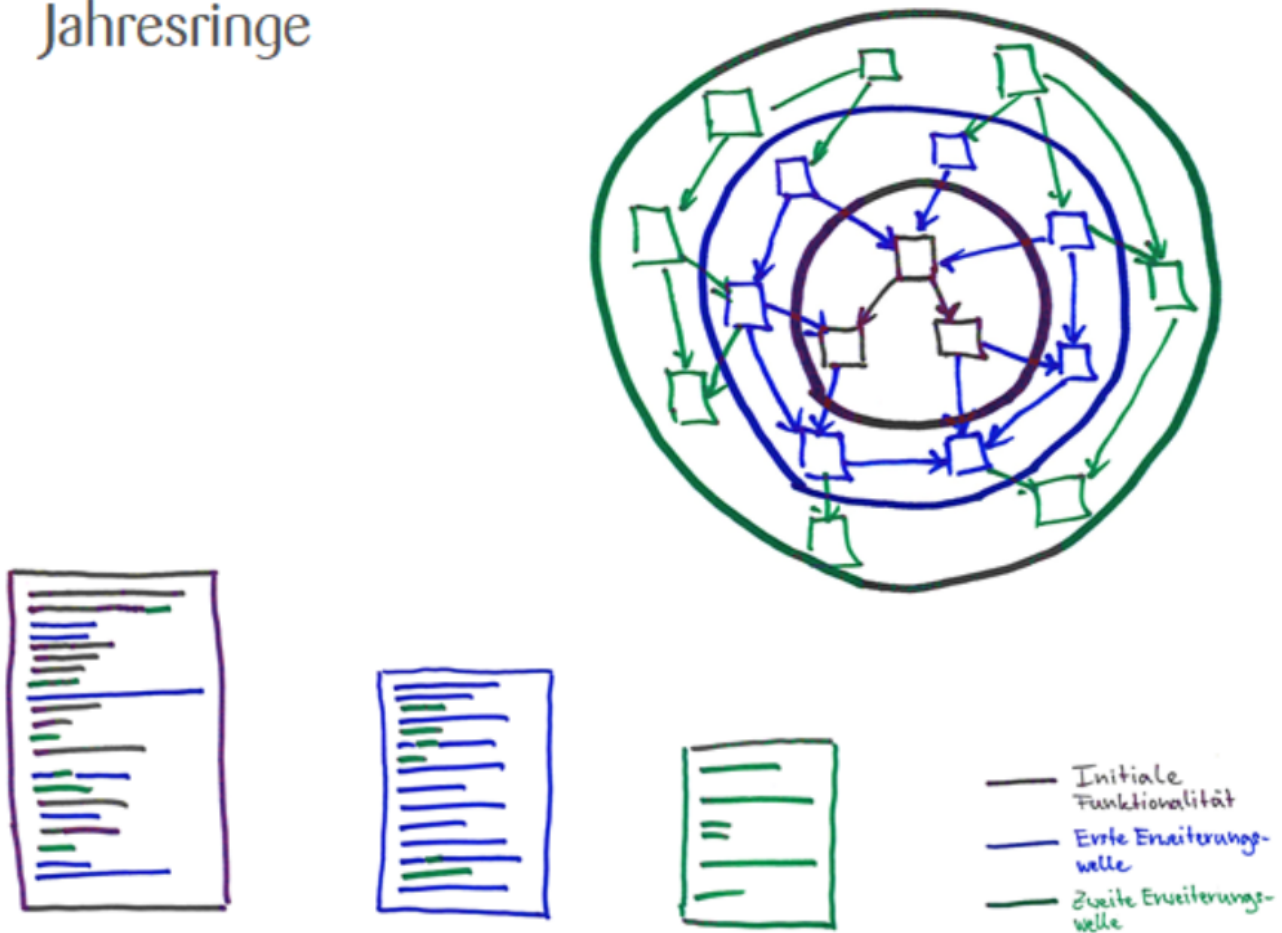
Wenn ich Code ändern muss, der so komplex ist, dass ich die Auswirkungen meiner Änderungen nicht schnell abschätzen kann (weil ich den Code nicht gut genug kenne und der Raum der möglichen Änderungen nicht ins Kurzzeitgedächtnis passt) dann führt die Angst vorm Fehler machen (ohne es überhaupt zu merken) zu folgenden Verhalten:

Die Hirnpsychologischen und Verhaltenspsych...

by Peter Gfader - <https://beyond-agility.com/die-hirnpsychologischen-und-verhaltenspsychologischen-hintergrunde-was-guten-von-schlechten-code-unterscheidet-german-post/>

- Analyse-Starre
- Minimalinvasive Code-Änderungen
 - nur Dranbauen, statt den neuen mit den alten Code in ein konsistenten Ganzes zu refaktorisieren
 - Das gibt dann die Jahresringe im Code

Jahresringe



Jetzt gibt es primär zwei Möglichkeiten auf Code-Ebene, diese Angst zu reduzieren:

1. Einfacherer Code, bei dem sich die Tragweite der Änderungen im Rahmen der Kapazität des Kurzzeitgedächtnisses einschätzen lässt (siehe 1.)
2. Testabdeckung mit automatisierten Tests die ich in kurzer Zeit so gut verstehen kann, dass ihnen vertraue (das Gegenteil von Angst). Deswegen ist u.A. DAMP over DRY und Single Level of Abstraction so extrem wichtig für Tests.

3. Mentales Mapping zwischen innen und Aussenwelt und begrenzte kognitive Kapazität.

Wir Menschen haben eine sehr begrenzte Kognitive Kapazität für Bewusstes Denken. Und alles, was ich nicht so gut automatisiert habe, dass es das Unterbewusstsein übernehmen kann (Unconscious Competence), raubt Denkkapazität. Ein Entwickler muss permanent zwischen verschiedenen Mentalen Modell übersetzen:
Innenwelt → Die Code-Welt

Drei Aussenwelten:

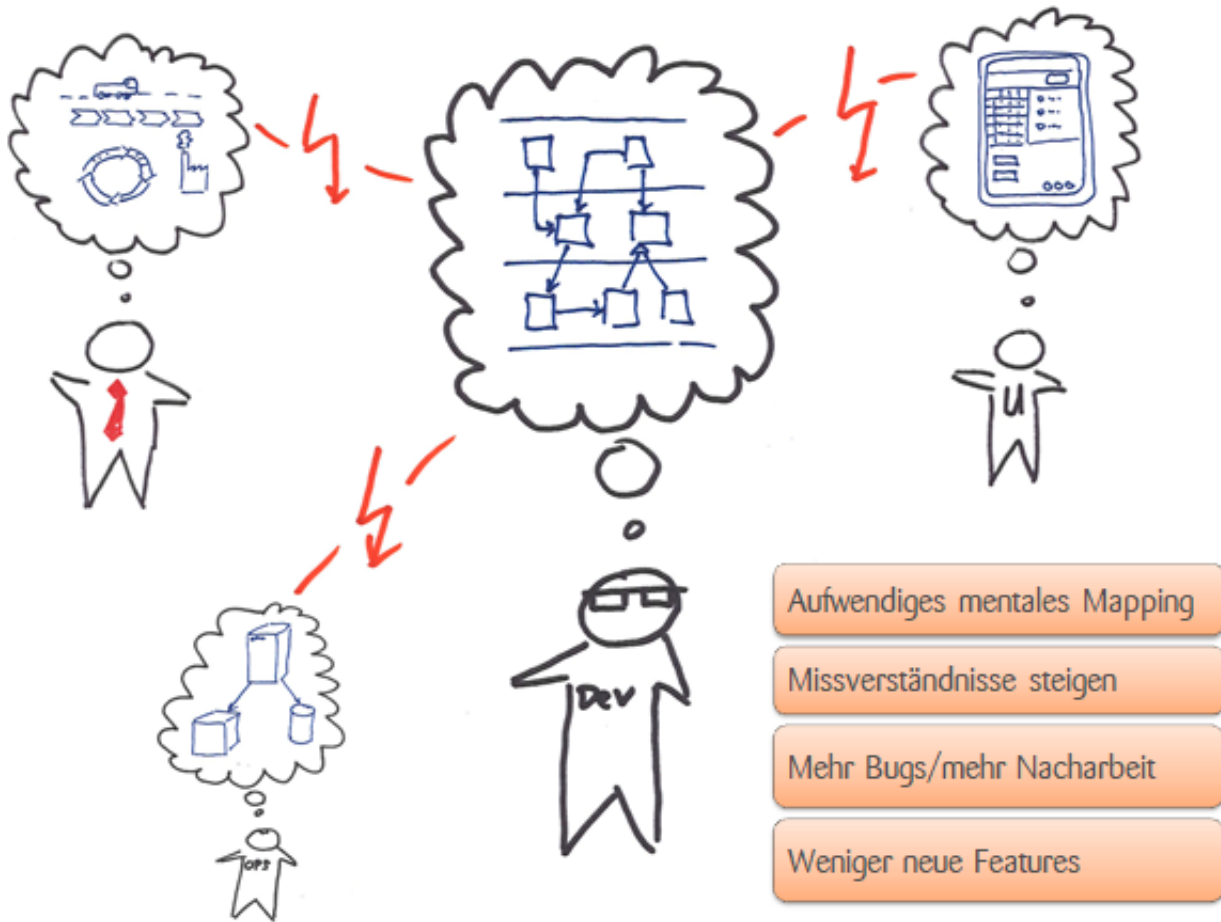
1. Fach-/Domainsicht
2. Nutzersicht
3. Technische Umgebung (Umsysteme, Configfiles,...)

Um so stärker sich die Innenwelt von der Aussenwelt unterscheidet, um so aufwändiger und fehleranfälliger ist das Mapping.

Irgenwann ist man beim permanenten Übersetzen zwischen Russisch, Indonesisch, Japansich und Deutsch. Das gibt den Mentalen Overkill.

Die Hirnpsychologischen und Verhaltenspsych...

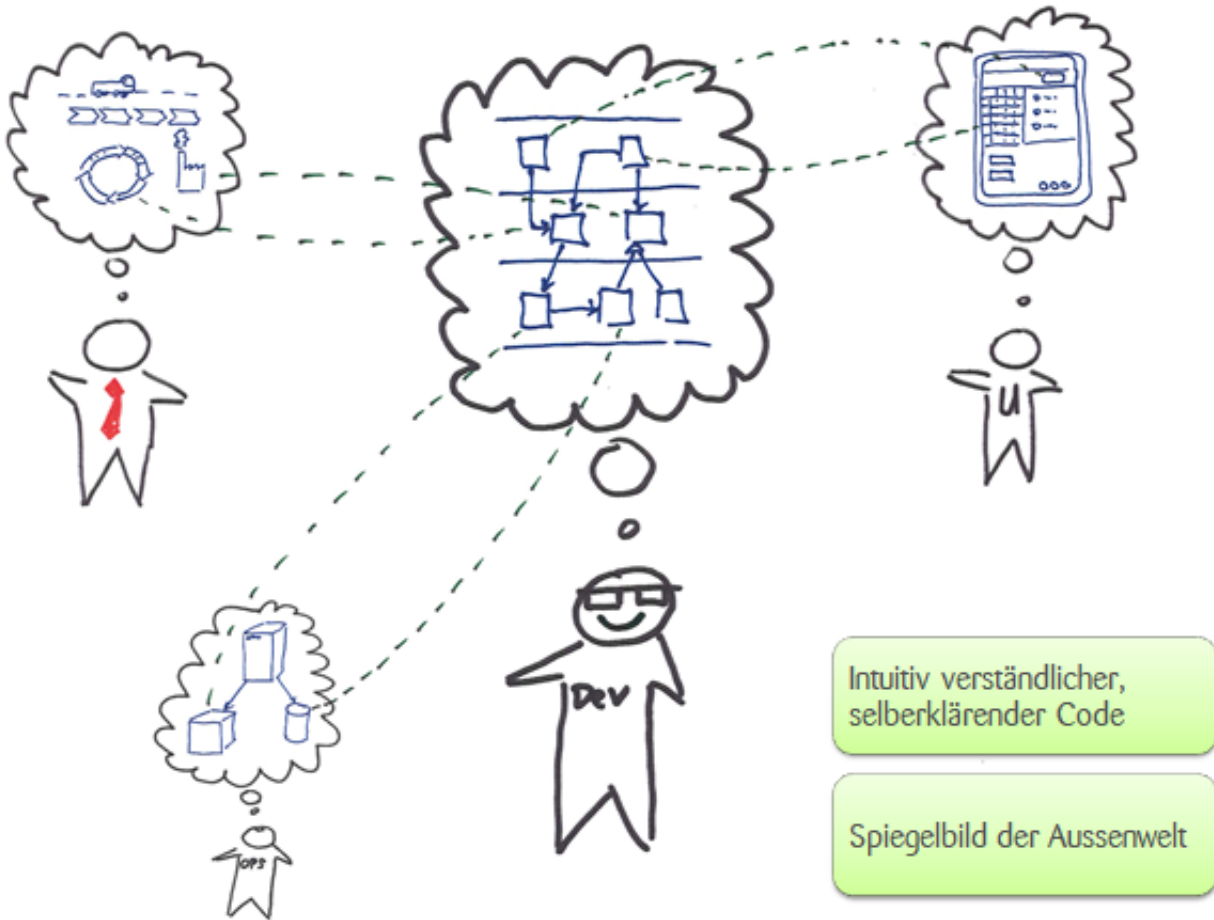
by Peter Gfader - <https://beyond-agility.com/die-hirnpsychologischen-und-verhaltenspsychologischen-hintergrunde-was-guten-von-schlechten-code-unterscheidet-german-post/>



DDD zielt primär darauf, das Mapping zwischen Fachwelt Code-Welt auf ein Minimum zu reduzieren. Aber es geht noch weiter. Ziel muss es sein, dass jedes Konzept aus den drei Aussenwelten durch eine adequate Struktur (Klasse, Aggregat, Komponente, etc.) im Code inkl. passendem Namen abstrahiert ist.

Die Hirnpsychologischen und Verhaltenspsych...

by Peter Gfader - <https://beyond-agility.com/die-hirnpsychologischen-und-verhaltenspsychologischen-hintergrunde-was-guten-von-schlechten-code-unterscheidet-german-post/>



Zusammengefasst kann man sagen, guter Code muss die kognitiven Beschränkungen von uns Menschen berücksichtigen und die Angst vor Fehlern durch Änderungen reduzieren resp. Vertrauen schaffen, dass man nichts wesentliches übersehen hat.

Mehr über Rick Janda auf der [Kegon webseite](#).

We Go Beyond Agile

Beyond Agility
Zürich - Switzerland