

Macoun

Vorbereitung:

1. Demos, Files im Finder anordnen.
2. Alle Programme die man für Demos benötigt vorher(!) starten.
Bpsw: Xcode, Terminal, Refelection.
3. Präsentation starten und auf Zeichen oder Signal warten.

TDD Code Retreat

Christian Tietze @ctietze
Oliver Böhm @oboehm

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss



WTF

Whow

WTF

WTF

WTF

WTF

Einleitung – **Format**

- 5 Sessions à 45min Coding + 15min Pause & Reflexion
- Pair-Programming
- Clean Code & Test-Driven Development
- Partnertausch nach jeder Session
- Code nach jeder Session sofort löschen

Was ist Pair-Programming?



2 Entwickler an 1 Rechner

- gleichberechtigt
- arbeiten gemeinsam an einer Aufgabe
- alle 5 Minuten werden Rollen / Tastatur gewechselt



Rollen

Pilot - steuert Tastatur, schreibt Code

Navigator - überprüft Code, denkt über Verbesserungen des Designs nach

bitte
häufig
Rollen
wechseln



Einleitung – **Code Retreat**

Es geht um die Erfahrung.

- Jede Session ist eine Fingerübung: Versucht nicht, fertig zu werden
- Stoppt, wenn die Zeit rum ist, und löscht den Code
- Einigt euch auf eine Programmiersprache/-umgebung

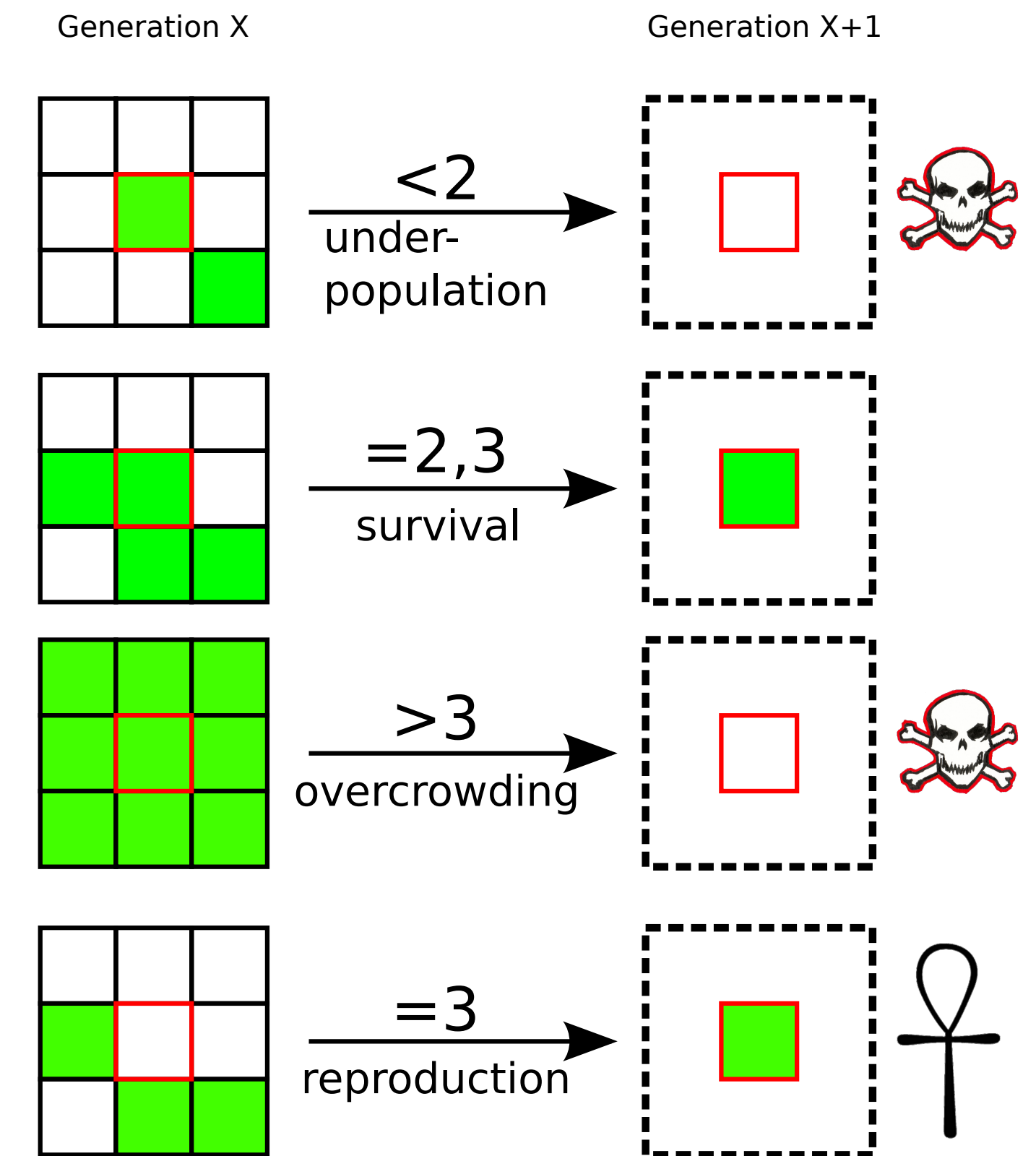
Einleitung – **Session-Fokus**

- Jede Session wird einen Fokus haben
- Für jede Session gelten die XP "Simple Design" Regeln
 - ✓ Expresses every idea that we need to express
 - ✓ Says everything Once and Only Once (DRY)
 - ✓ Has no superfluous parts
 - ✓ Passes all the tests

Einleitung – Game of Life

Die Welt besteht aus einem (unendlichen) orthogonalen 2D Gitter von Zellen.

- **Zustand** jeder Zelle: *lebend* oder *tot*
- **Regeln:** Jede Zelle interagiert mit ihren 8 direkten Nachbarn (siehe Grafik)
- **"Seed":** Initialkonfiguration der Welt
- **"Tick":** Anwendung der Regeln auf die Welt als reine Abbildung

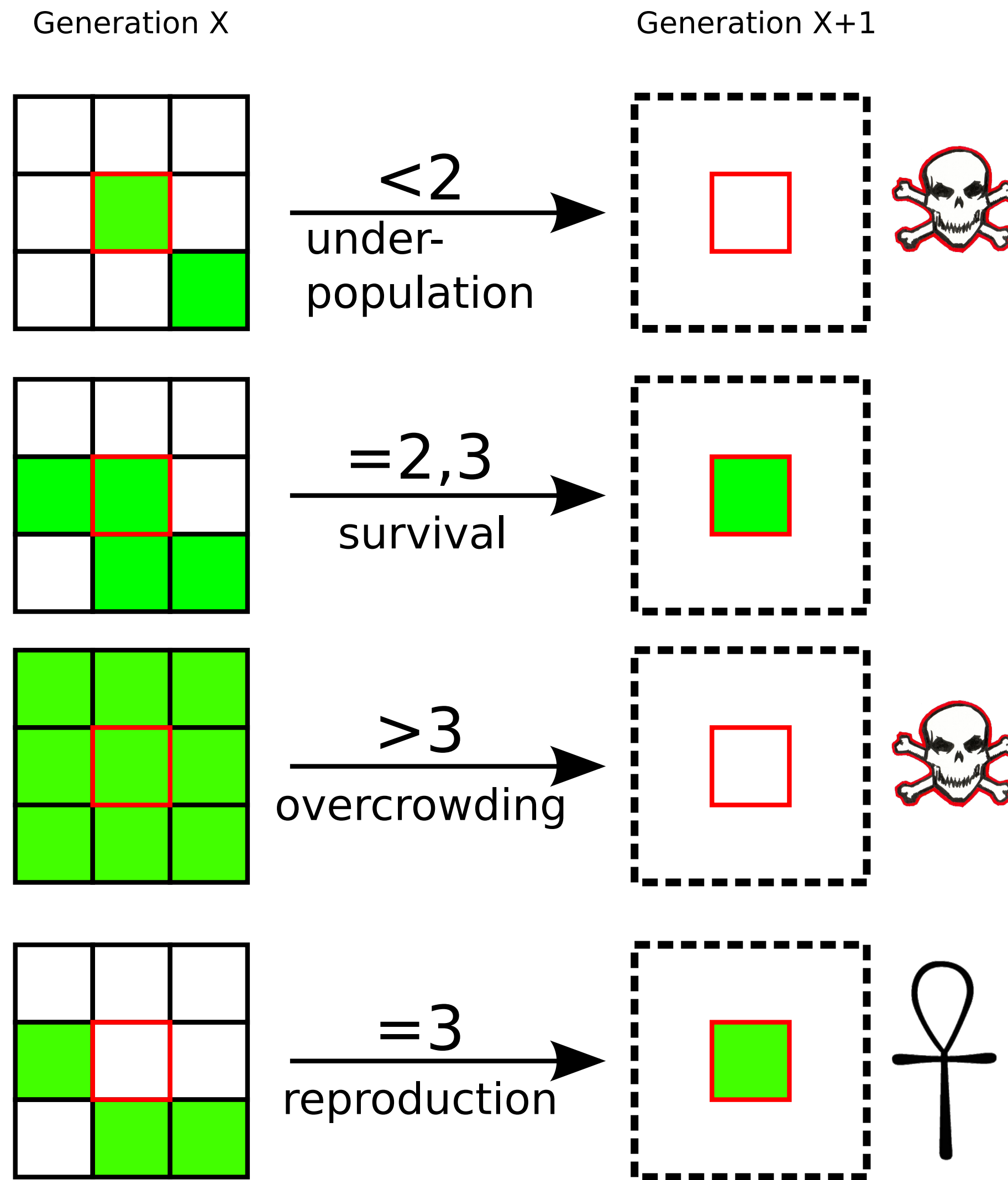


#1 Aufwärmen

Extreme Programming: "Simple Design":

- ✓ Expresses every idea that we need to express
- ✓ Says everything Once and Only Once (DRY)
- ✓ Has no superfluous parts
- ✓ Passes all the tests

Game of Life Rules



@markoemrich
webmasters.de

Creative Commons
Attribution 3.0 Unported,
<http://creativecommons.org/licenses/by-sa/3.0/>

Retrospektive Session #1

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss

#2 Kurze Methoden

Einschränkung:

- Maximal 4 Zeilen Code pro Methode

Extreme Programming: "Simple Design":

- ✓ Expresses every idea that we need to express
- ✓ Says everything Once and Only Once (DRY)
- ✓ Has no superfluous parts
- ✓ Passes all the tests

Retrospektive Session #2

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss

#3 Schleifen verboten

(XP "Simple Design" gilt weiterhin!)

- Keine Schleifenkonstrukte (`for/while/repeat...while`)

Retrospektive Session #3

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss

Einleitung – Refaktorisierung

- Implementierung duplizieren
- Duplikat mit Tests abdecken ("learning tests")
- Originalaufrufe zum neuen Code umleiten
- Alte Implementierung löschen
- Tests prüfen

#4 "TDD as if you Meant It"

- Schreib einen fehlschlagenden Test
- Schreibe eine Implementation **in die Testfunktion**

Wiederhole mit weiteren Tests. Extrahiere sich wiederholdenen Code:

1. in ein neues Feld (*property*) in der Testklasse
2. in eine neue Funktion in der Testklasse
3. in eine bestehende Funktion in der Testklasse

Extrahiere zusammengehörige Funktionen in eine neue Klasse.

Retrospektive Session #4

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss

#5 TDD: Keine "Primitives"

"

- Verwende nur eigene Datentypen (`String`, `Int`, `Array`, ... einkapseln)
- Schreib einen fehlschlagenden Test
- Schreibe eine Implementation in die Testfunktion

Wiederhole mit weiteren Tests. Extrahiere sich wiederholdenen Code:

1. in ein neues Feld (*property*) in der Testklasse
2. in eine neue Funktion in der Testklasse
3. in eine bestehende Funktion in der Testklasse

Extrahiere zusammengehörige Funktionen in eine neue Klasse.

Retrospektive Session #5

Ablauf

10:30 Ankommen & Einleitung

11:00 Session #1

11:45 Retrospektive & Pause

12:00 Session #2

12:45 Retrospektive & Pause

13:00 Session #3

13:45 Retrospektive & Pause

15:00 Einleitung Nachmittag

15:15 Session #4

16:00 Retrospektive & Pause

16:15 Session #5

17:00 Retrospektive & Pause

17:15 Session #6

18:00 Abschluss

#6 Bedingungsloses TDD

- Keine Bedingungen (`if/else/guard/switch`)
- Schreib einen fehlschlagenden Test
- Schreibe eine Implementation in die Testfunktion

Wiederhole mit weiteren Tests. Extrahiere sich wiederholdenen Code:

1. in ein neues Feld (*property*) in der Testklasse
2. in eine neue Funktion in der Testklasse
3. in eine bestehende Funktion in der Testklasse

Extrahiere zusammengehörige Funktionen in eine neue Klasse.

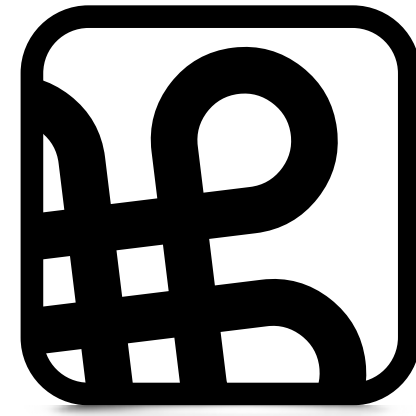
Retrospektive Session #6

und Feedback

#7 Refaktorisierung

- Implementierung duplizieren
- Duplikat mit Tests abdecken ("learning tests")
- Originalaufrufe zum neuen Code umleiten
- Alte Implementierung löschen
- Tests prüfen

Vielen Dank



Macoun