Refactoring mit der Mikado-Methode

Falk Sippach

Leinfelden, 11.10.2024

0

Abstract

Refactoring mit der Mikado-Methode

Viele von uns haben tagtäglich mit Legacy-Code zu tun. Mal eben schnell etwas umzubauen, scheitert typischerweise an den fehlenden Tests, zudem ist der Quellcode oft überhaupt schlecht testbar.

In diesem Vortrag wird anhand von praktischen Codebeispielen gezeigt, wie man zunächst ein automatisiertes Sicherheitsnetz aufspannt. Anschließend werden komplexere Refactorings durchgeführt, ohne jedoch zu viele Baustellen gleichzeitig aufzureißen. Die Mikado-Methode hilft dabei, den Überblick zu behalten und in möglichst kleinen und nachvollziehbaren Schritten vorzugehen. Das Ziel ist das Aufbrechen stark gekoppelter Abhängigkeiten, um so neue Tests hinzufügen zu können. Zudem wird der Code besser lesbar sein und lässt sich so auch leichter warten und wiederverwenden.

1

Falk Sippach

- Softwarearchitekt, Berater, Trainer bei embarc
- früher bei Orientation in Objects (OIO), Trivadis

Schwerpunkte

- Architekturberatung und -bewertung
- Cloud- und Java-Technologien

embarc

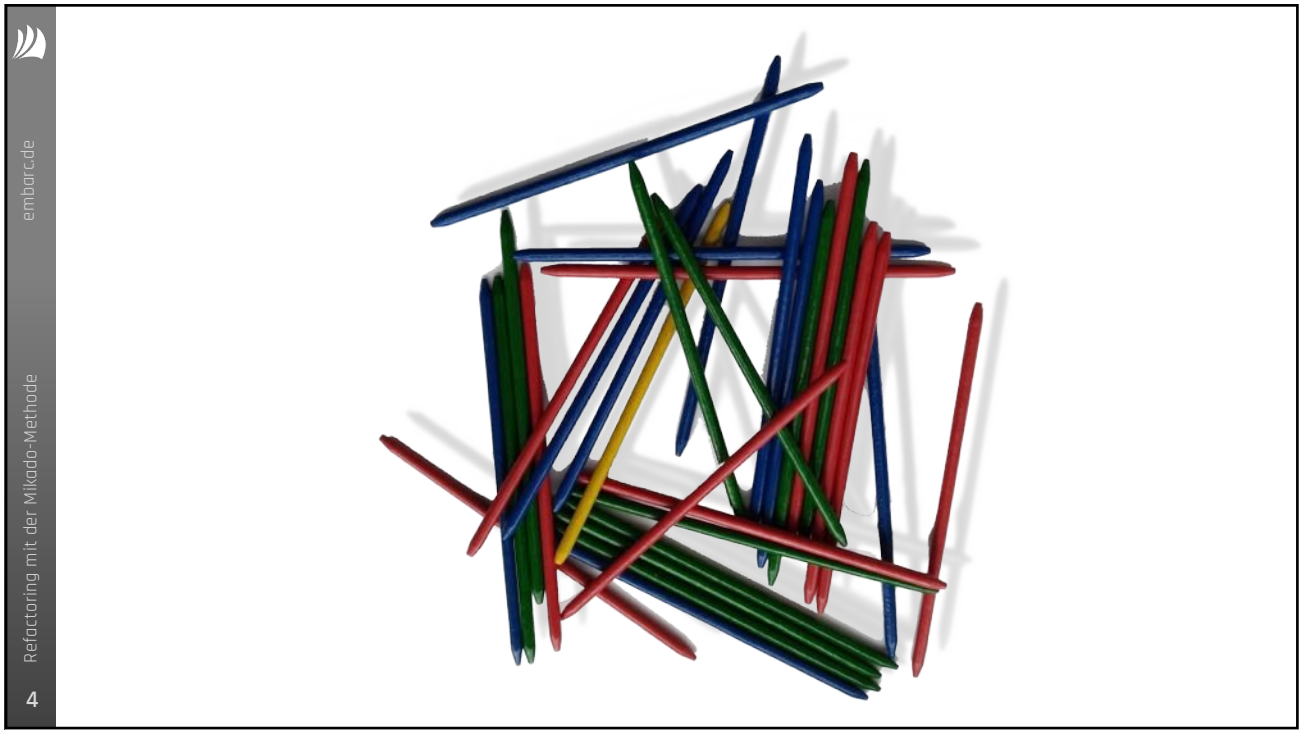
2

*The **Mikado Method** is a **structured way** to make **significant changes** to **complex code**.*

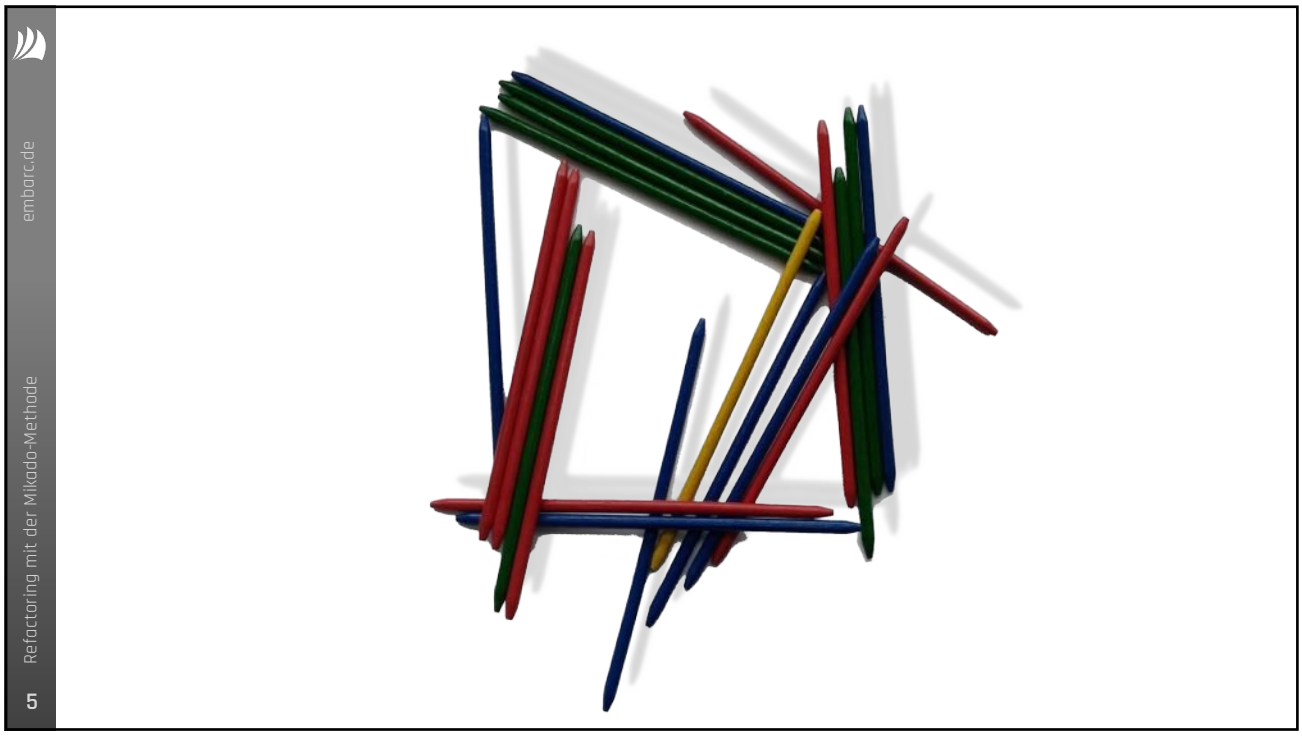
*... for **performing changes** to a **system that's too large for analyze-then-edit**, which means basically **any production system** in the world.*

(Ola Ellnestam, Daniel Brolund: "The Mikado Method")

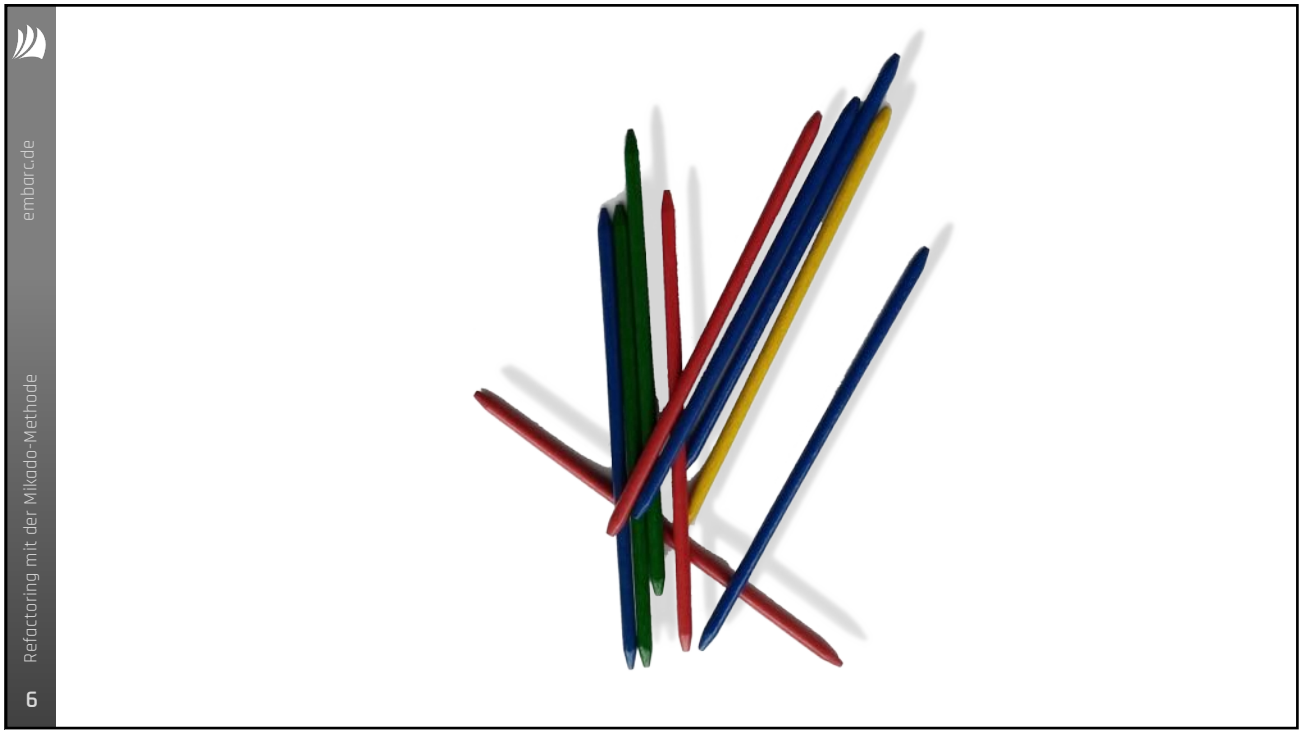
3


embarc.de
Refactoring mit der Mikado-Methode
4

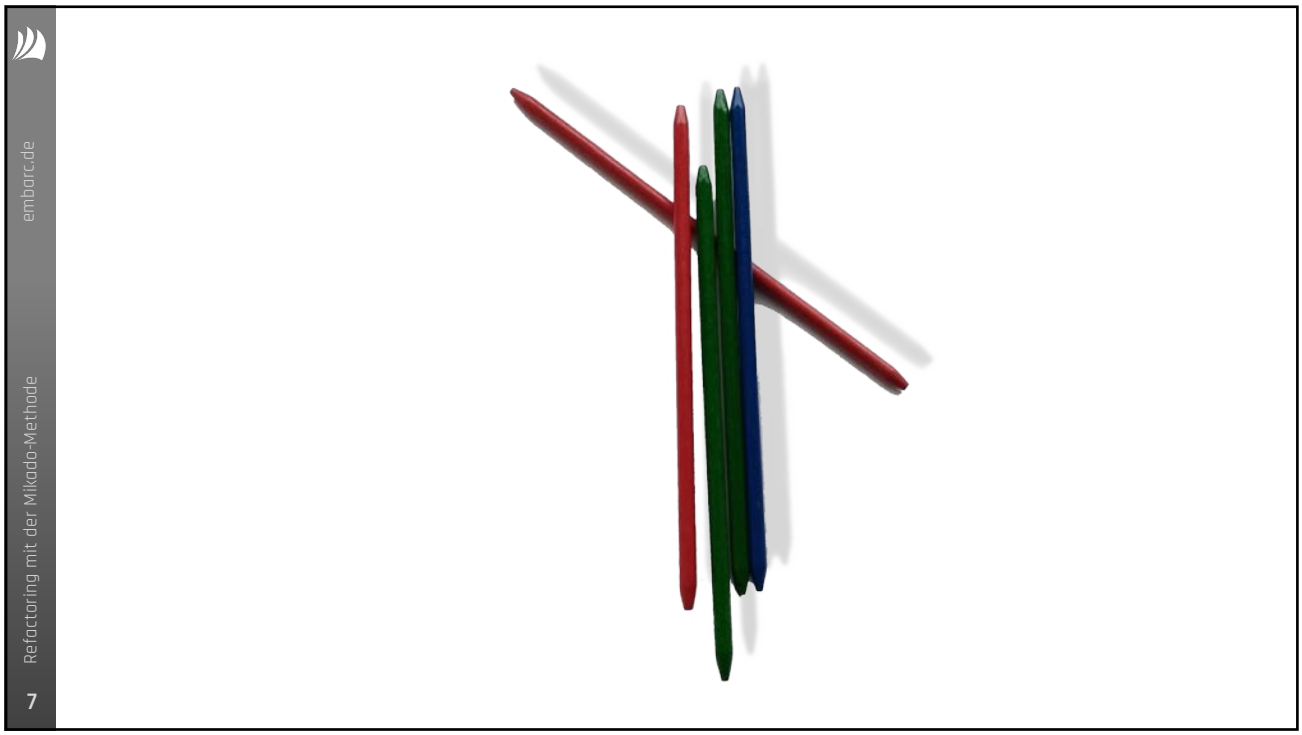
4


embarc.de
Refactoring mit der Mikado-Methode
5

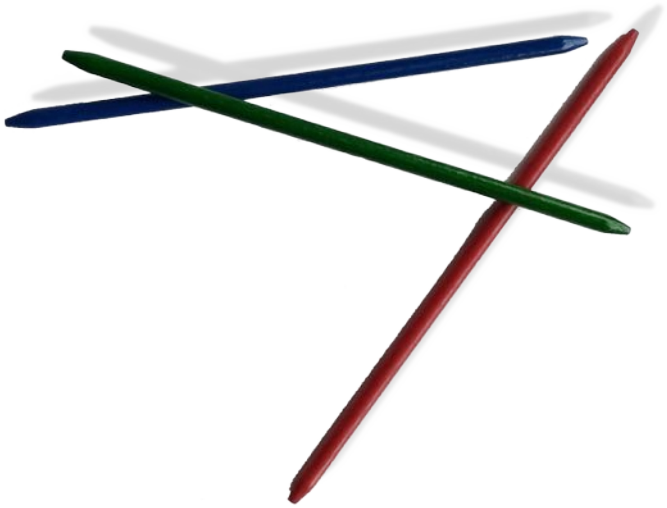
5


embarc.de
Refactoring mit der Mikado-Methode
6

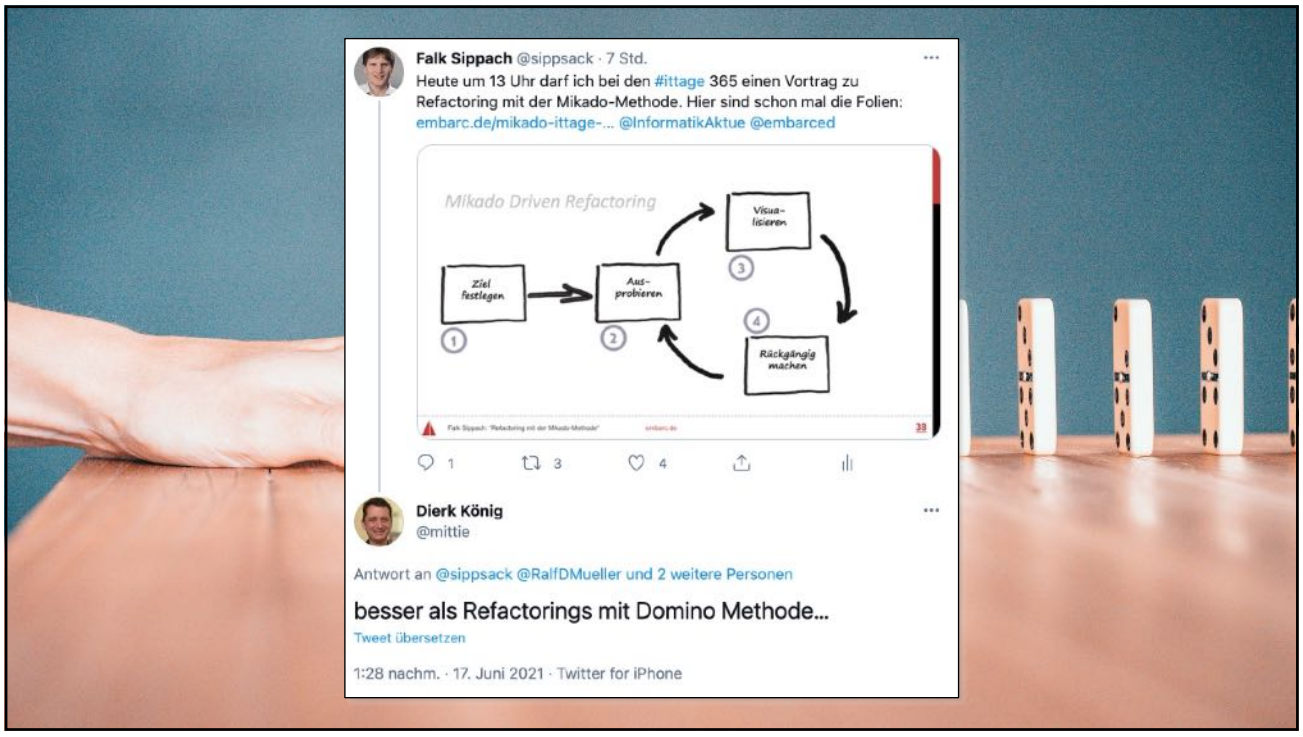
6


embarc.de
Refactoring mit der Mikado-Methode
7

7



8


Falk Sippach @sippsack · 7 Std.
Heute um 13 Uhr darf ich bei den #ittage 365 einen Vortrag zu Refactoring mit der Mikado-Methode. Hier sind schon mal die Folien: embarc.de/mikado-ittage-... @InformatikAktue @embarced
Mikado Driven Refactoring
Ziel festlegen
Aus-probieren
Visua-lisieren
Rückgängig machen
1
3
4
Dierk König
@mittie
Antwort an @sippsack @RalfDMueller und 2 weitere Personen
besser als Refactorings mit Domino Methode...
Tweet übersetzen
1:28 nachm. · 17. Juni 2021 · Twitter for iPhone

9

00.

Agenda

Was euch erwartet

01 Legacy Code

02 Mikado Methode

03 Reale Legacy Projekte

04 Fazit

10

01.

Legacy Code

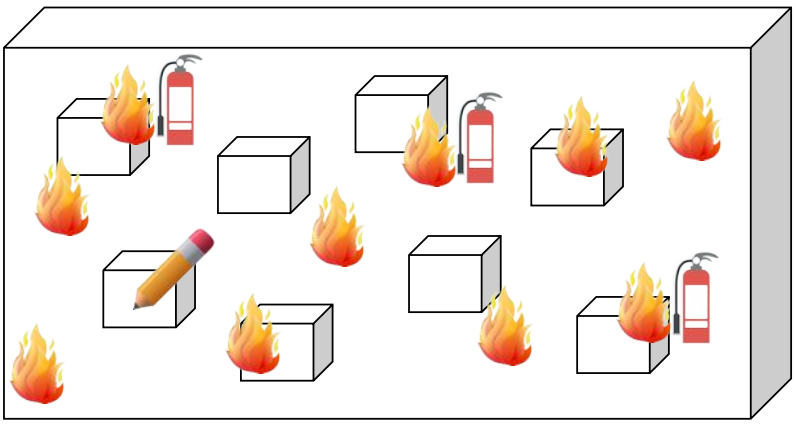
Was macht Legacy Code aus, was sind die Herausforderungen?

11

Code ändern? Nur wie?

***Brownfield**-Projekt, Code von **anderen**,*
***Fehlende Dokumentation**, kaum **Tests**,*
*Änderungen geraten **außer Kontrolle**.*

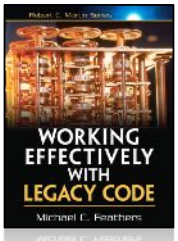
12



13

"Code **without tests** is **bad code**.

Michael Feathers



14

J. B. Rainsberger

"Legacy code is **valuable code** that we feel **afraid to change**.

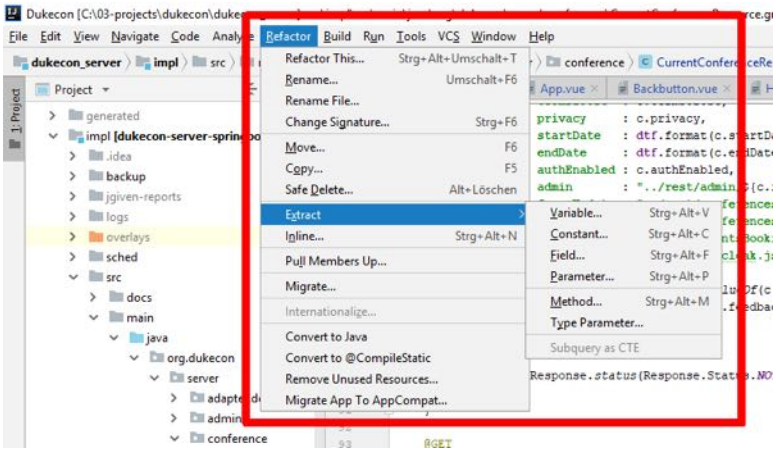
15

Gründe für Refactoring

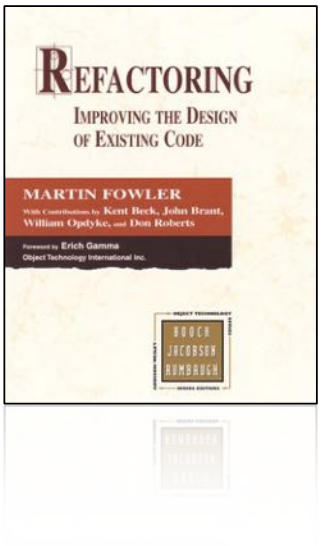
Verstehen
Fehler beheben
Neues Feature
Optimierung

16

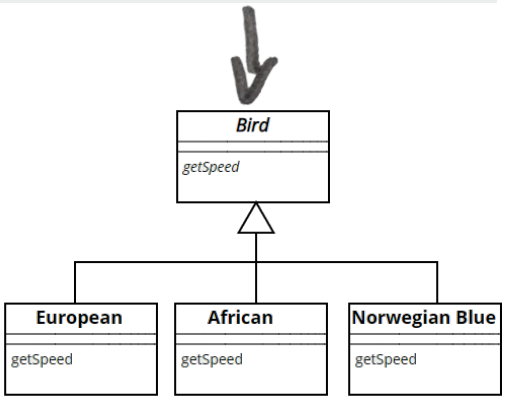
Toolunterstützung bei einfachen Refactorings



17




```
double getSpeed() {
  switch (_type) {
    case EUROPEAN:
      return getBaseSpeed();
    case AFRICAN:
      return getBaseSpeed() - getLoadFactor() * _numberOfCoconuts;
    case NORWEGIAN_BLUE:
      return (_isNailed) ? 0 : getBaseSpeed(_voltage);
  }
  throw new RuntimeException ("Should be unreachable");
}
```



18

02.

Mikado Methode

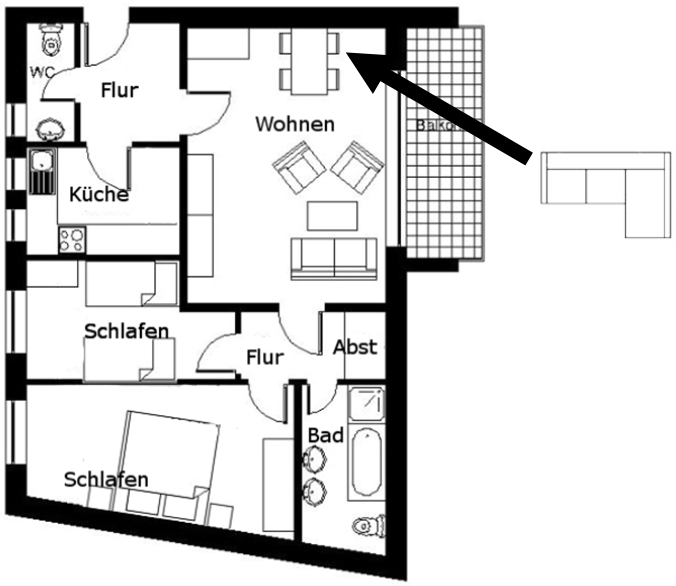
Wie funktioniert diese Methode denn nun genau?

19

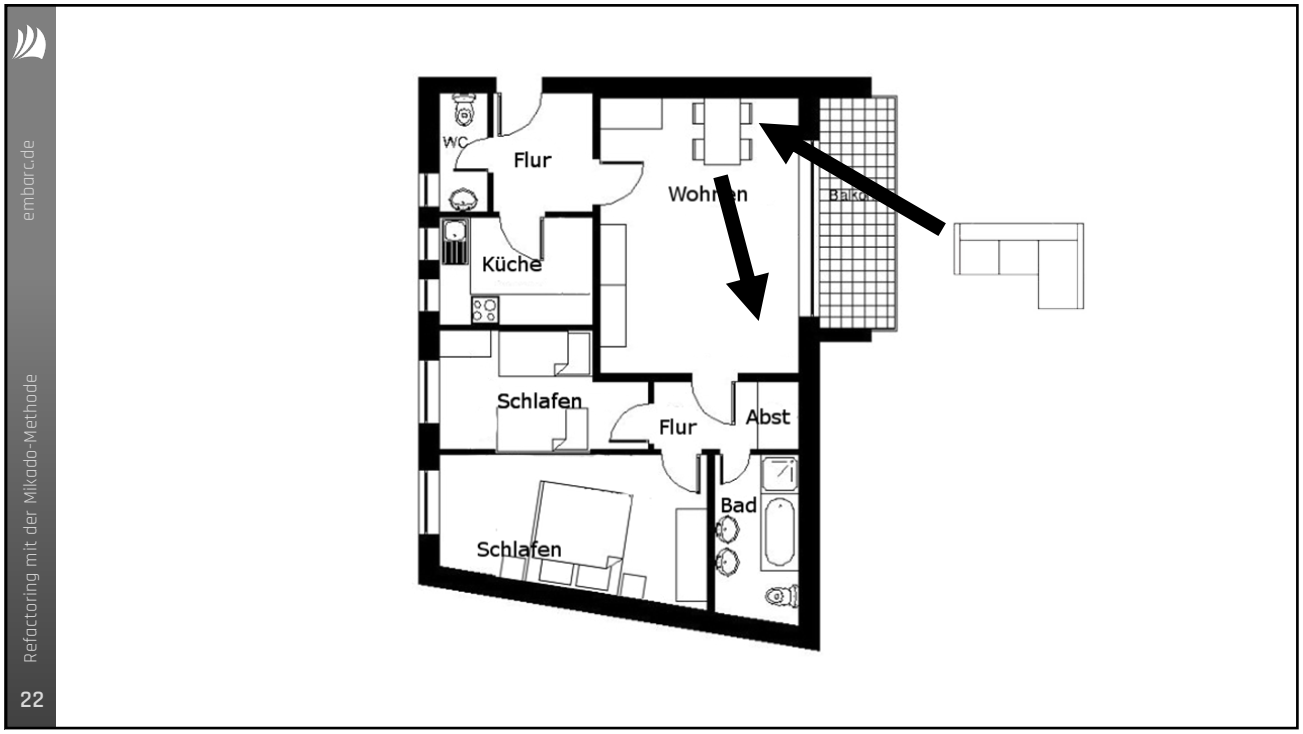
Mikado Methode für Dummies

Möbel rücken

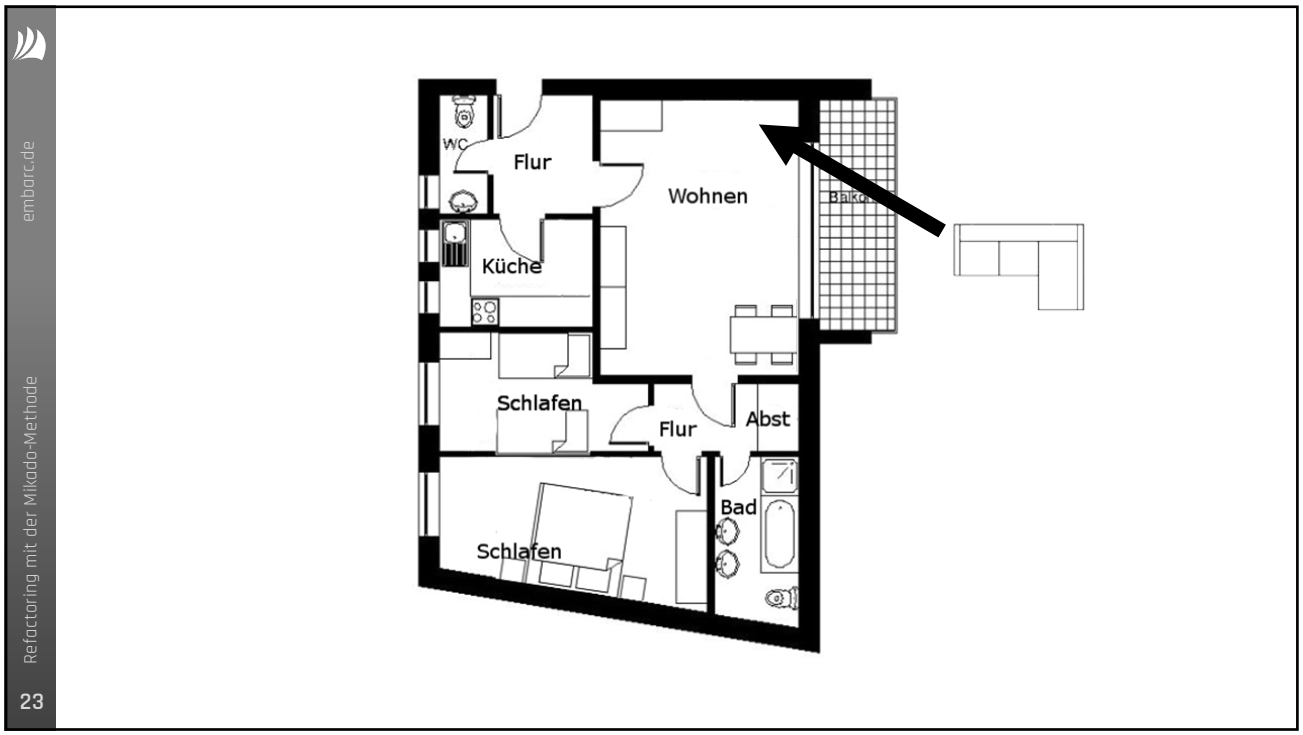
20



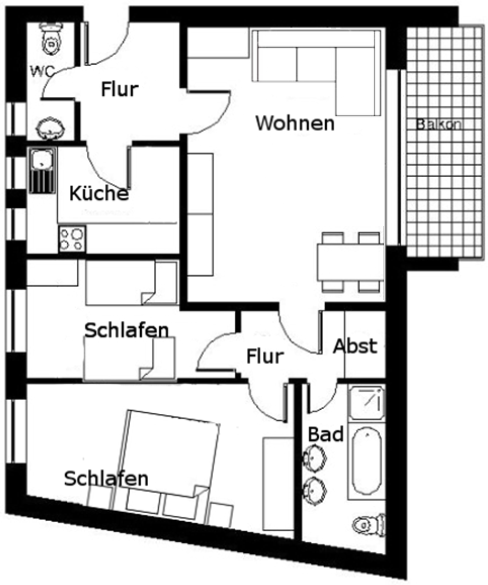
21


embarc.de
Refactoring mit der Mikado-Methode
22
WC
Flur
Wohnen
Balkon
Küche
Schlafen
Flur
Abst
Bad
Schlafen

22


embarc.de
Refactoring mit der Mikado-Methode
23
WC
Flur
Wohnen
Balkon
Küche
Schlafen
Flur
Abst
Bad
Schlafen

23



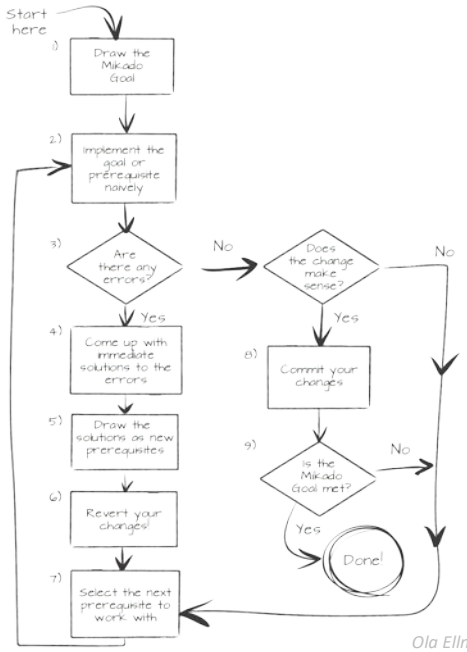
24

The Mikado Method can help you ***visualize, plan, and perform*** *business value–focused* ***improvements*** *over several iterations and increments of work,* ***without*** *ever having* ***a broken codebase*** *during the process.*

(Ola Ellnestam, Daniel Brolund: "The Mikado Method")

25

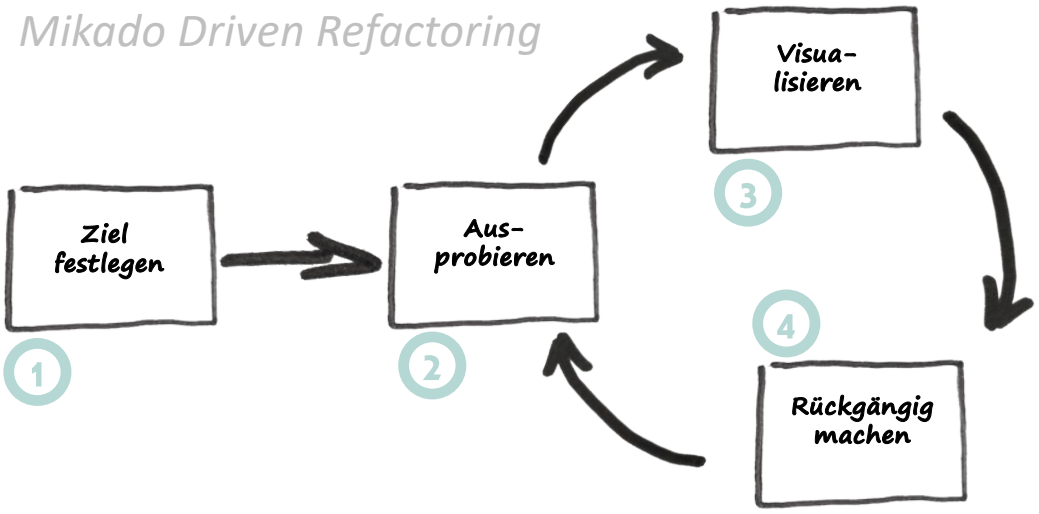
Ablauf im Großen:



Ola Ellnestam, Daniel Brolund: "The Mikado Method"

26

Mikado Driven Refactoring



36

1

Ziel festlegen

=

***Startpunkt** der Änderung.*

***Erfolgskriterium** für Ende.*

37

2

Ausprobieren

=

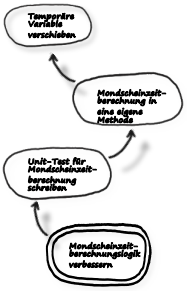
Experimentieren.

***Hypothesen** prüfen.*

*Sehen, was **kaputt geht.***

38

3



Visualisieren

=

***Ziel** und notwendige*
***Vorbedingungen** aufschreiben.*
***Mikado Graph** erstellen.*

39

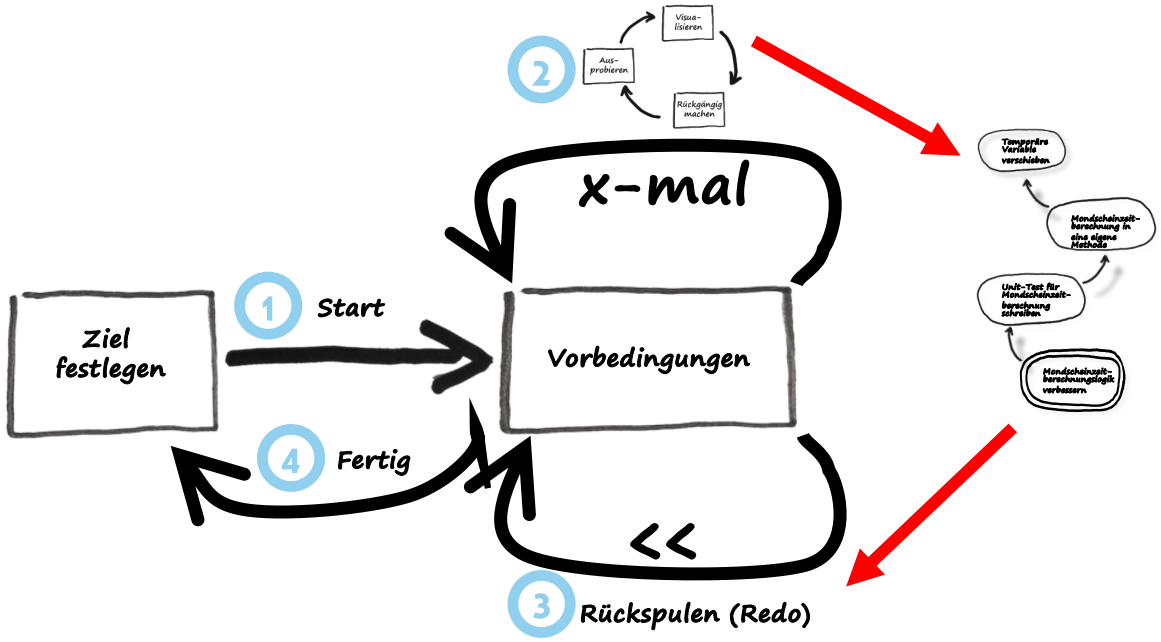
4

Rückgängig machen

=

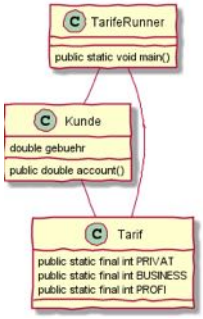
***Vorherigen** funktionierenden*
*Stand **wiederherstellen.***

40



41


```
package de.oio.refactoring.badtelefon;

public class Kunde {
  double gebuehr = 0.0;
  Tarif tarif;

  public Kunde(int tarifArt) {
    this.tarif = new Tarif(tarifArt);
  }

  public void account(int minuten, int stunde, int minute) {
    boolean mondschein = false;
    double preis = 0;

    // Mondscheinzeit ?
    if (stunde < 9 || stunde > 18)
      mondschein = true;

    // Gespraechspreis ermitteln
    switch (tarif.tarif) {
      case Tarif.PRIVAT:
      […]
    }
    gebuehr += preis;
  }

  public double getGebuehr() {
    return gebuehr;
  }
}
```

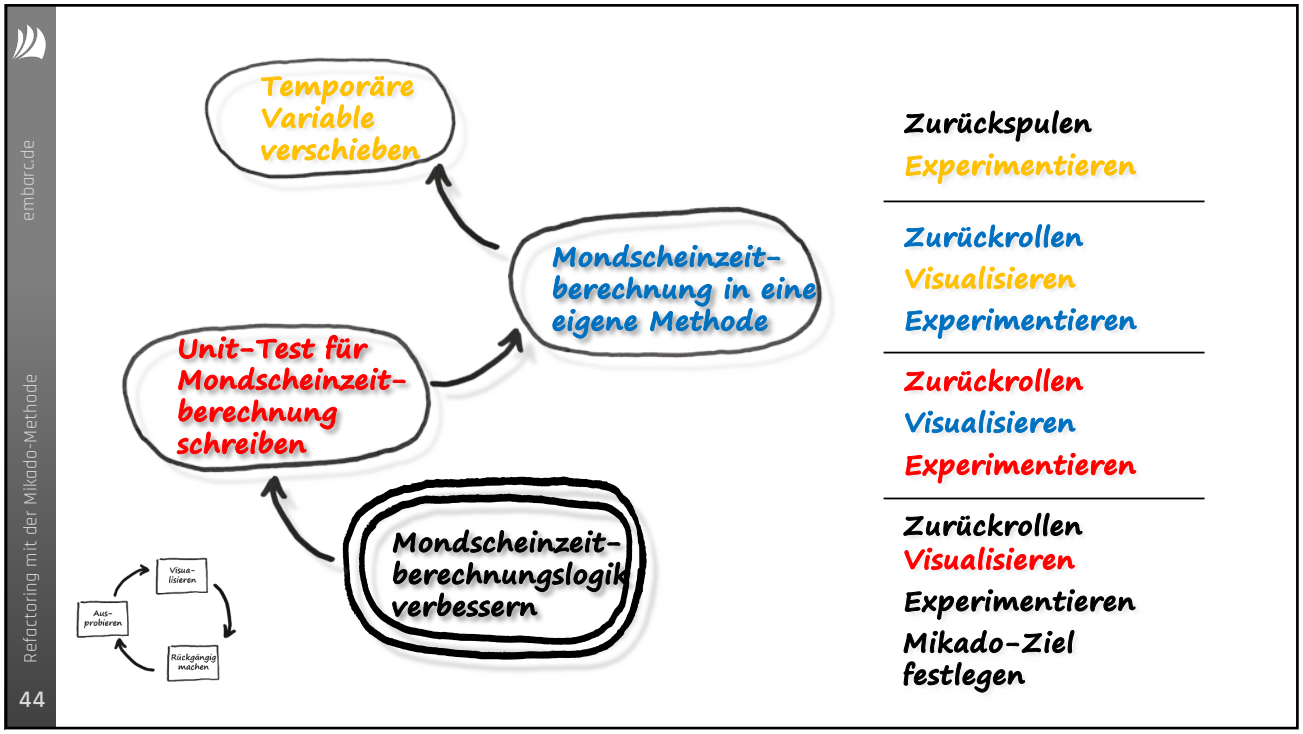



```
package de.oio.refactoring.badtelefon;

import java.util.Arrays;
import java.util.Random;

public class TarifeRunner {
 public static void main(String args[]) {
  Random random = new Random();
  for(Integer tarif : Arrays.asList(Tarif.PRIVAT, Tarif.BUSINESS, Tarif.PROFI)) {
   System.out.println(String.format("\nVerarbeitung von Tarif %s", tarif));
   Kunde k = new Kunde(tarif);
   [..]
 }
}
```

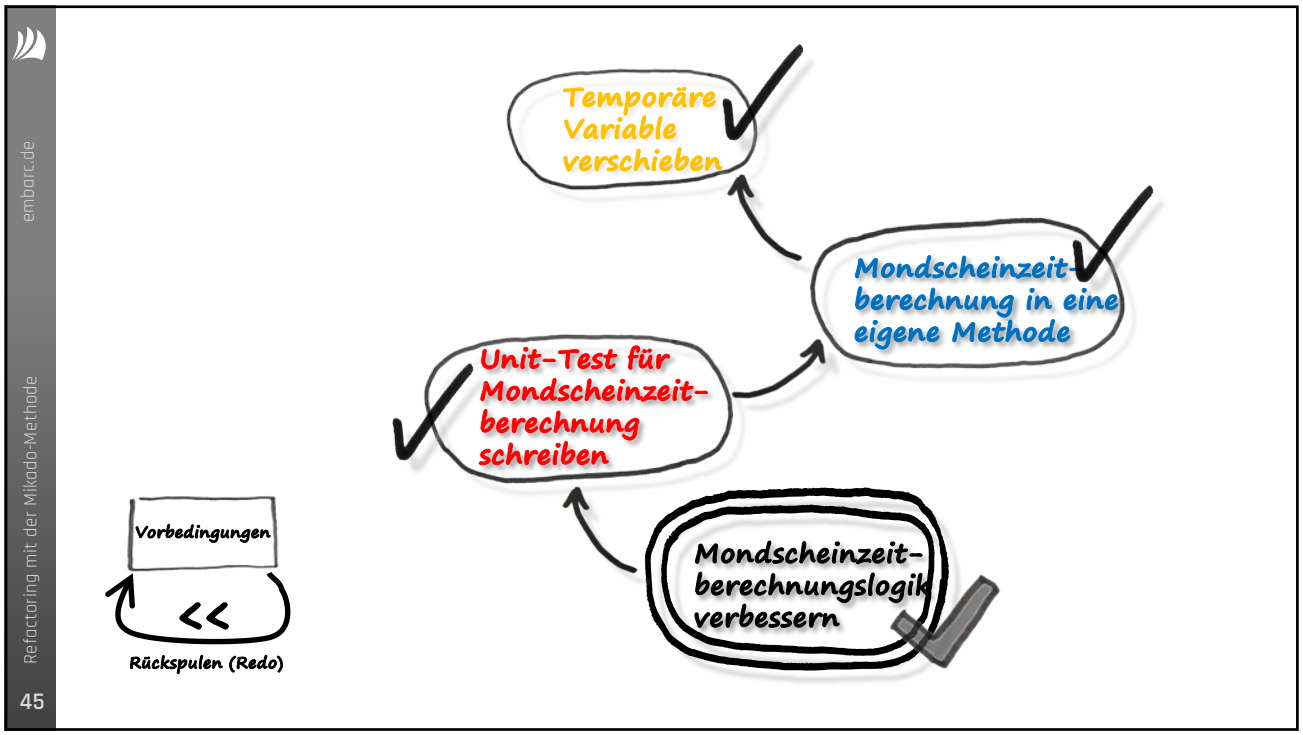
```
package de.oio.refactoring.badtelefon;

public class Tarif {
      public final static int PRIVAT = 0;
      public final static int BUSINESS = 1;
      public final static int PROFI = 2;

      int tarif = 0;

      public Tarif(int tarif) {
               this.tarif = tarif;
      }
}
```

https://github.com/sippsack/BadTelefon-Refactoring-Legacy-Code/blob/mikado-dev-initial/doc/mikado-live-coding.adoc

42


embarc.de
Refactoring mit der Mikado-Methode
44
Temporäre Variable verschieben
Mondscheinzeit-berechnung in eine eigene Methode
Unit-Test für Mondscheinzeit-berechnung schreiben
Mondscheinzeit-berechnungslogik verbessern
Visua-lisieren
Aus-probieren
Rückgängig machen
Zurückspulen
Experimentieren
Zurückrollen
Visualisieren
Experimentieren
Zurückrollen
Visualisieren
Experimentieren
Zurückrollen
Visualisieren
Experimentieren
Mikado-Ziel festlegen

44

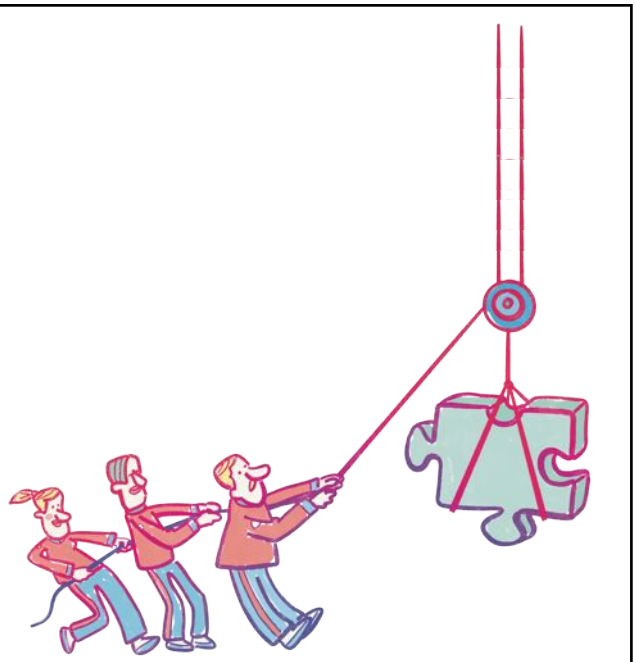

embarc.de
Refactoring mit der Mikado-Methode
45
Temporäre Variable verschieben
Mondscheinzeit-berechnung in eine eigene Methode
Unit-Test für Mondscheinzeit-berechnung schreiben
Mondscheinzeit-berechnungslogik verbessern
Vorbedingungen
<<
Rückspulen (Redo)

45

03.

Reale Legacy Projekte

Wie sieht das jetzt in realen (großen) Projekten aus, gerade wenn Tests fehlen?



46

Einfach loslegen?

*Es **fehlen** automatisierte **Tests!***

*Machen wir auch nichts **kaputt?***

47

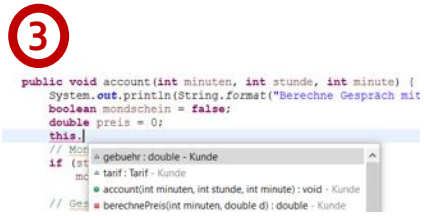
```
# suppose that our legacy code is this program called 'game'
$ game > GOLDEN_MASTER

# after some changes we can check to see if behaviour has changed
$ game > OUT-01
$ diff GOLDEN_MASTER OUT-01
# GOLDEN_MASTER and OUT-01 are the same

# after some other changes we check again and...
$ game > OUT-02
$ diff GOLDEN_MASTER OUT-02
# GOLDEN_MASTER and OUT-02 are different -> behaviour changed
```

Golden Master
(aka characterization tests)

48


```java
@Before
public void init() {
    originalSysOut = System.out;
    consoleStream = new ByteArrayOutputStream();
    PrintStream printStream = new PrintStream(consoleStream);
    System.setOut(printStream);
}

@Test
public void testSimpleOutput() {
    System.out.println("Hallo Publikum!");
    System.out.print("Hallo Falk!");
    assertEquals("Hallo Publikum!\r\nHallo Falk!", consoleStream.toString());
}

@After
public void teardown() {
    System.setOut(originalSysOut);
}
```

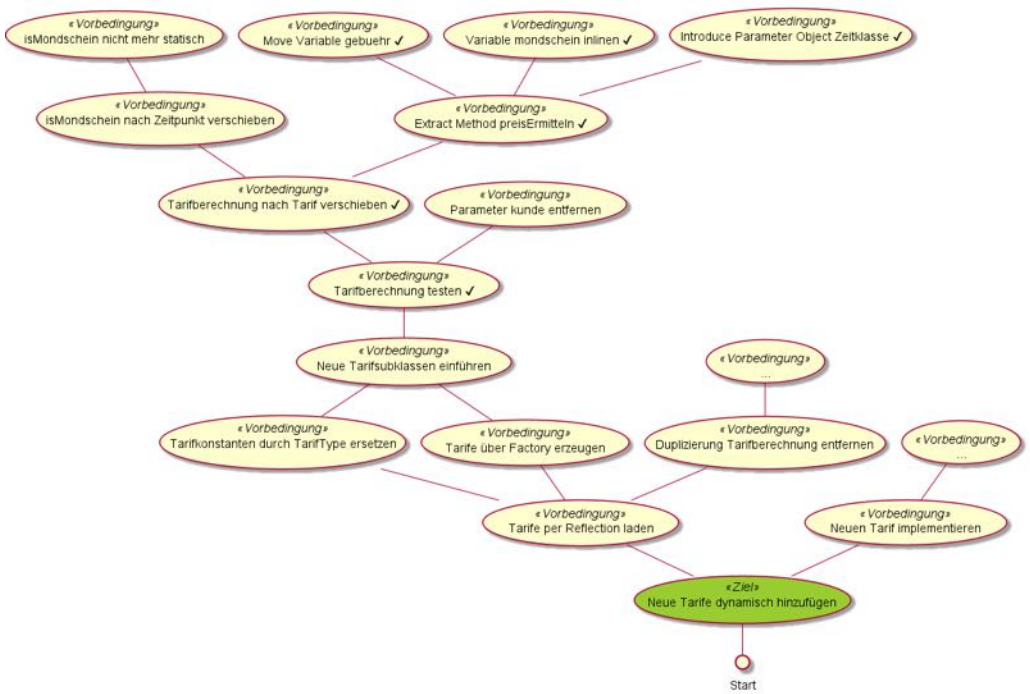

49




```
public static void main(String... args) throws Excep
  WebDriver driver = new FirefoxDriver();
  driver.get("http://www.retest.de");
  while (true) {
    List<WebElement> links =
driver.findElements(By.tagName("a"));
    links.get(random.nextInt(links.size())).click();
    Thread.sleep(500);
    List<WebElement> fields =
        driver.findElements(By.xpath("//input[@type='text']"));
    WebElement field = fields.get(random.nextInt(fields.size()));
    field.sendKeys(randomString());
    Thread.sleep(500);
  }
}
```

https://entwicklertag.de/frankfurt/2016/sites/entwicklertag.de.frankfurt.2016/files/slides/Bei%20uns%20testen%20lauter%20Affen_0.pdf

50



52

04.

Fazit und Ausblick

Zusammenfassung, Anwendungsbereiche und Werkzeuge

53

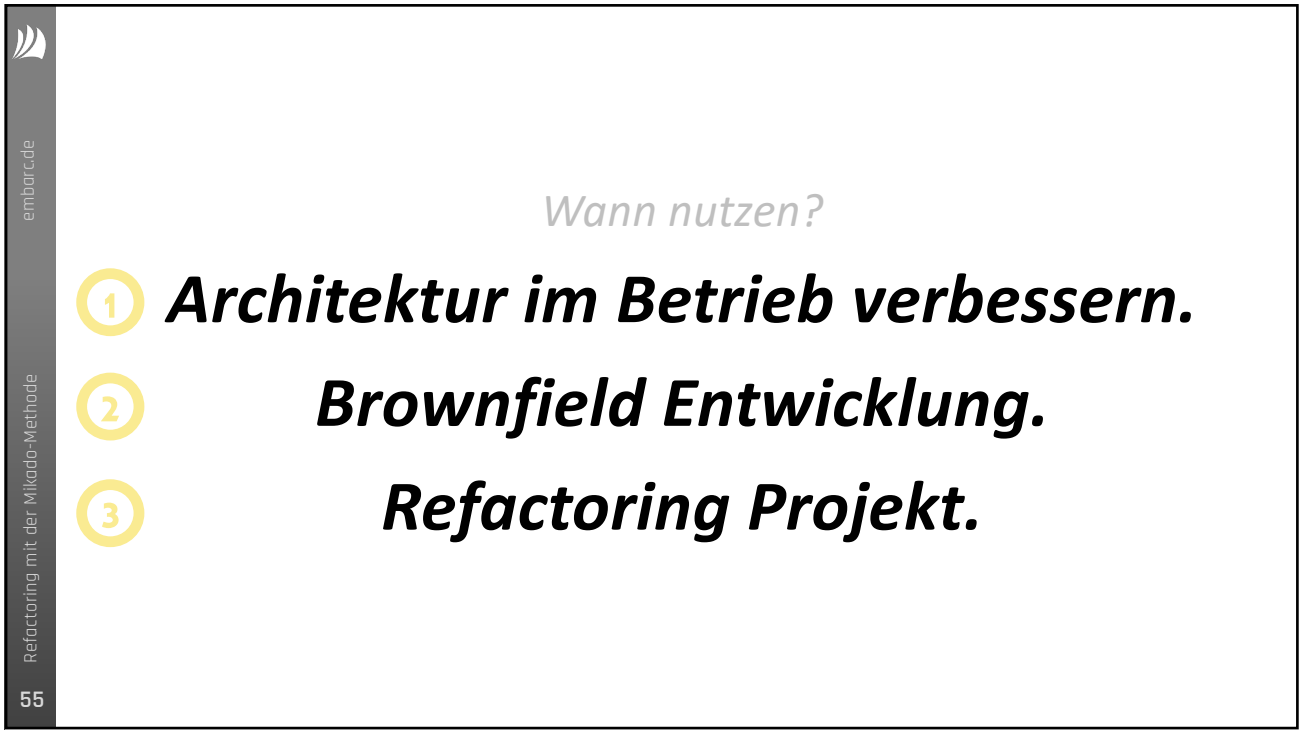
Vorteile

***Stabiles System** trotz Änderungen.*

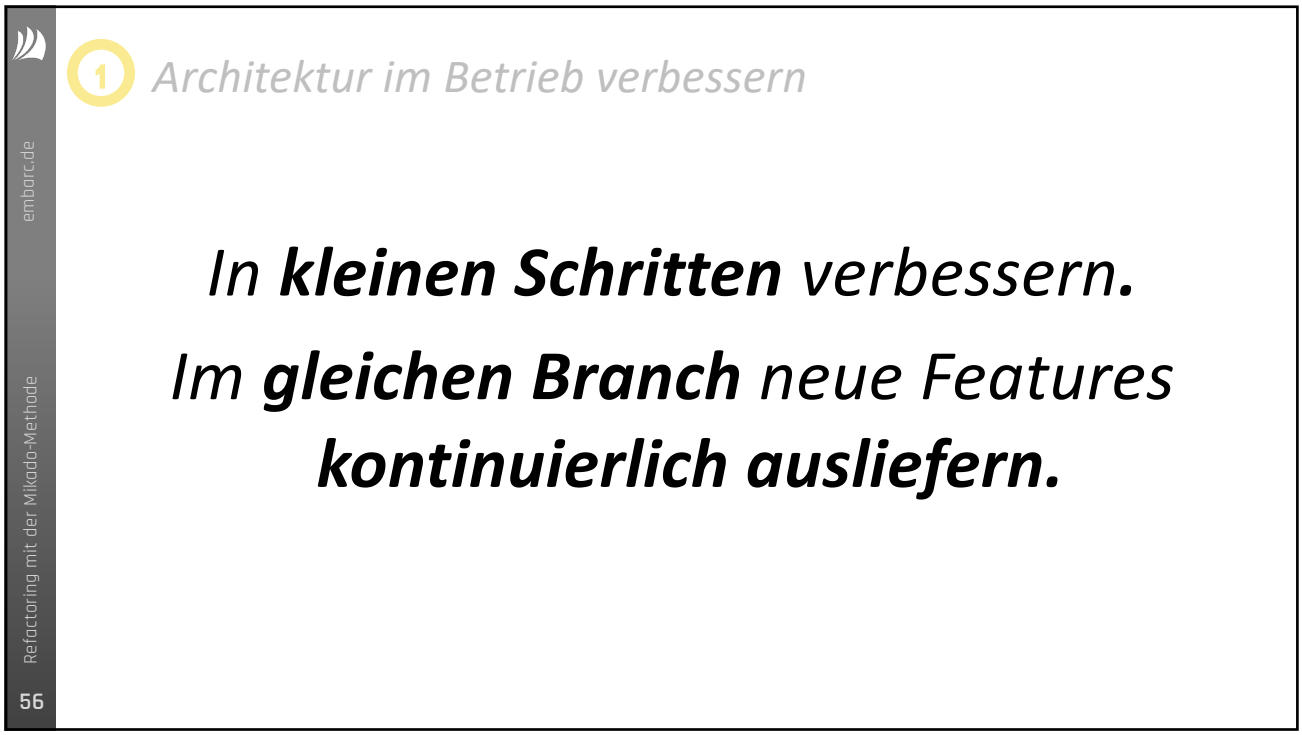
*Bessere **Kommunikation** und **Zusammenarbeit**.*

***Leichtgewichtig** und **fokussiert**.*

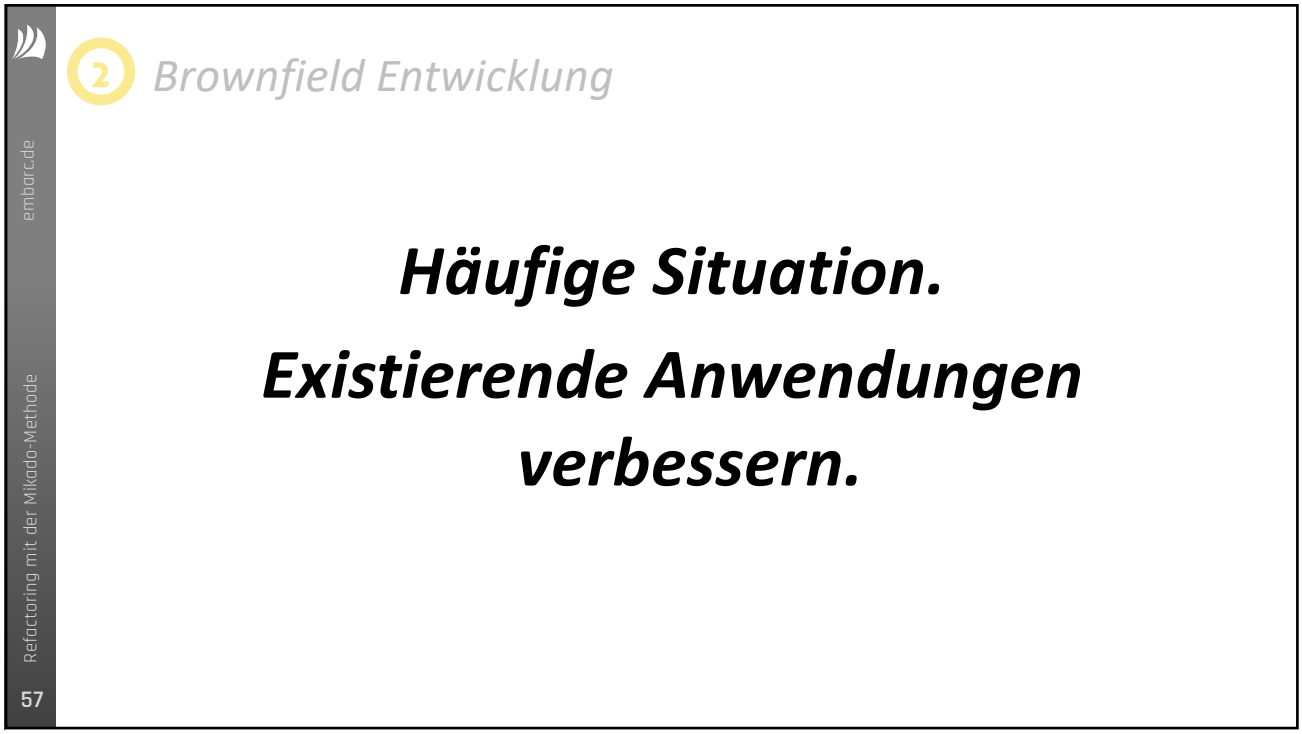
54


embarc.de
Refactoring mit der Mikado-Methode
55
Wann nutzen?
1 Architektur im Betrieb verbessern.
2 Brownfield Entwicklung.
3 Refactoring Projekt.

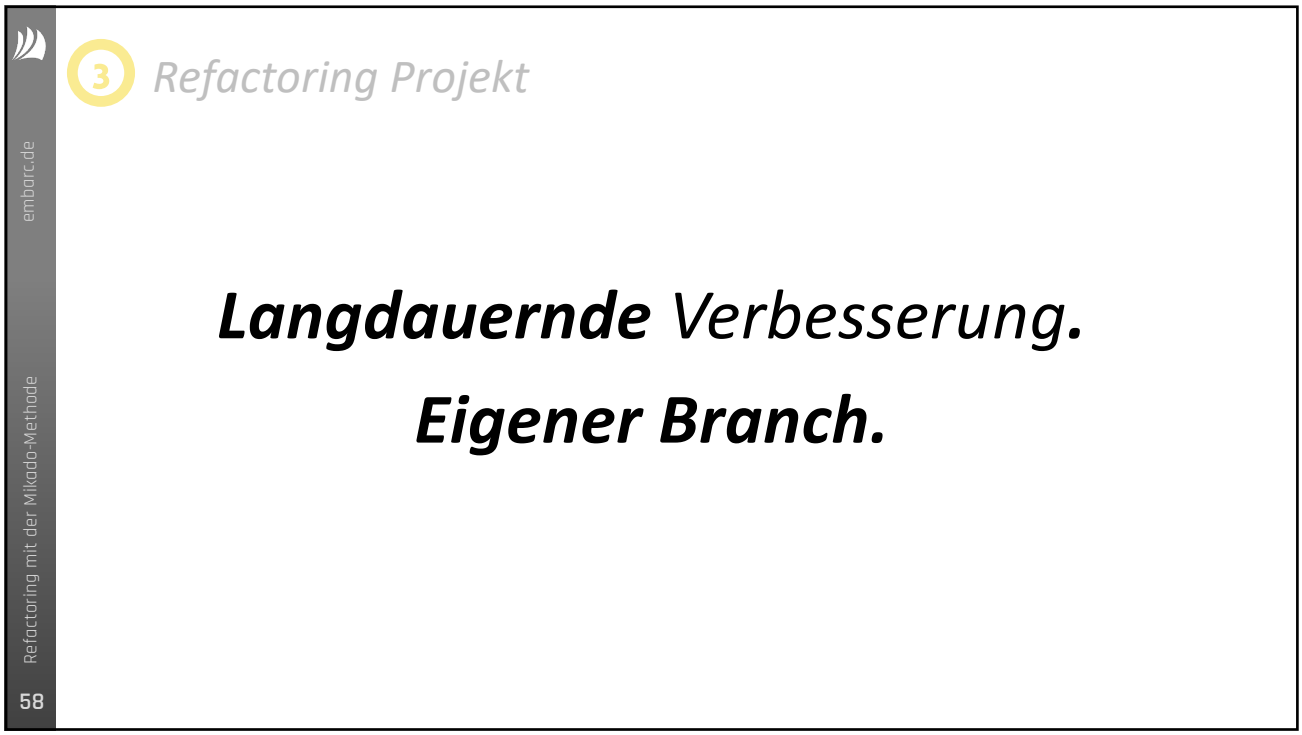
55


embarc.de
Refactoring mit der Mikado-Methode
56
1 Architektur im Betrieb verbessern
In kleinen Schritten verbessern.
Im gleichen Branch neue Features kontinuierlich ausliefern.

56


2 Brownfield Entwicklung
Häufige Situation.
Existierende Anwendungen verbessern.
embarc.de
Refactoring mit der Mikado-Methode
57

57


3 Refactoring Projekt
Langdauernde Verbesserung.
Eigener Branch.
embarc.de
Refactoring mit der Mikado-Methode
58

58

Angst vorm Revert:

Reverting** scheint wie **Wegwerfen.

*Aber Mikado-Graph **enthält Infos.***

***Wissensaufbau/-transfer** über*
System, Domäne, Technologie.

59

Angst vorm Revert: Änderungen doch aufheben

Stash (Git)

Patch-File (SVN, …)

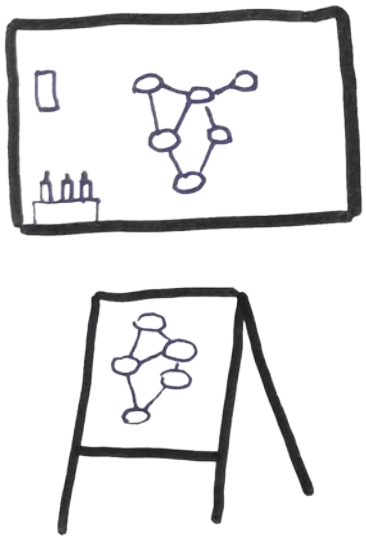
60

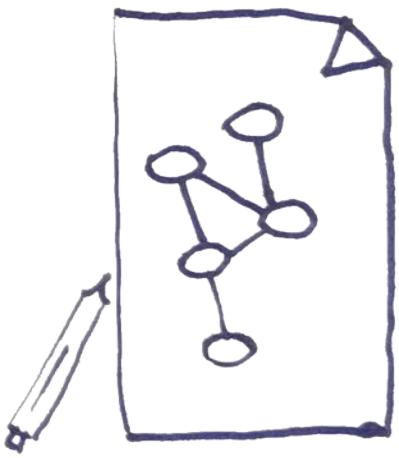
Arbeitsmittel:

1. ***Whiteboard/Flipchart/Papier***
2. ***Mindmap***
3. ***Visio, yEd***

61

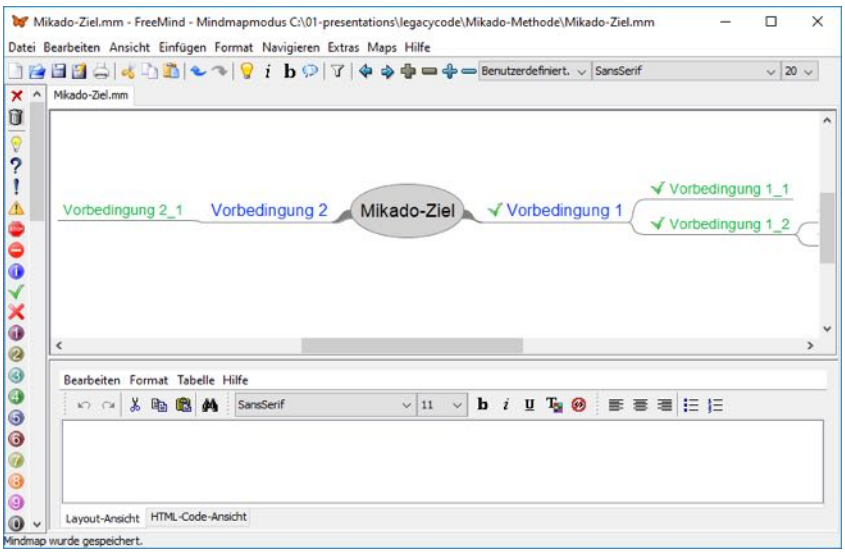
1





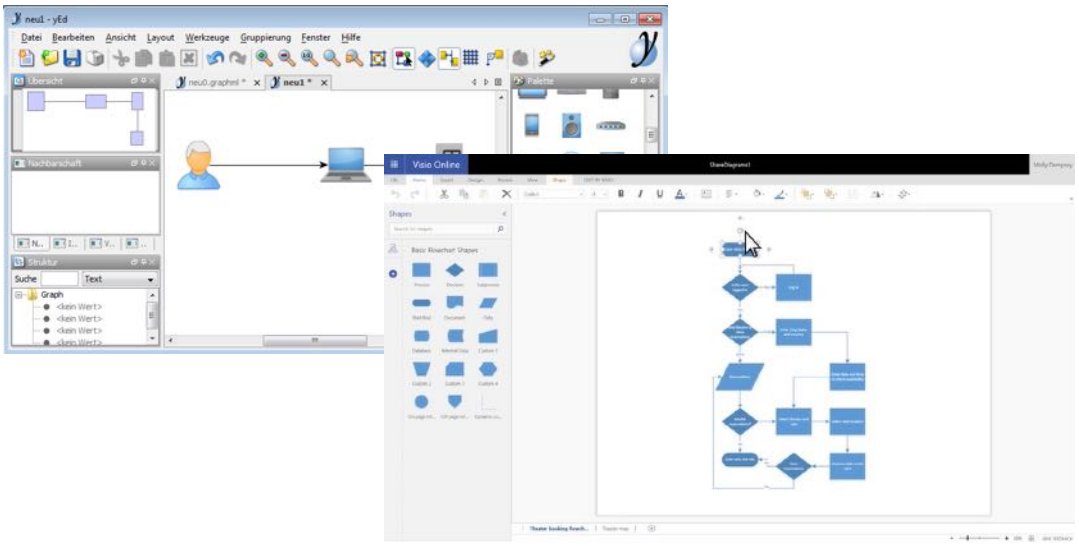
62

2

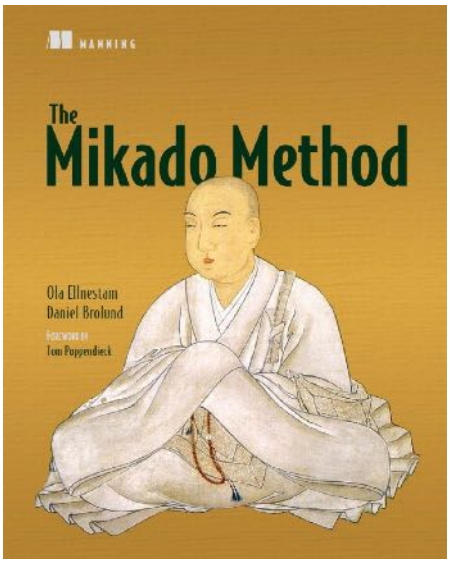

Mikado-Ziel.mm - FreeMind - Mindmapmodus C:\01-presentations\legacycode\Mikado-Methode\Mikado-Ziel.mm
Datei Bearbeiten Ansicht Einfügen Format Navigieren Extras Maps Hilfe
Mikado-Ziel.mm
Vorbedingung 2_1
Vorbedingung 2
Mikado-Ziel
Vorbedingung 1
Vorbedingung 1_1
Vorbedingung 1_2
Bearbeiten Format Tabelle Hilfe
Layout-Ansicht
HTML-Code-Ansicht
Mindmap wurde gespeichert.

63

3


neu1 - yEd
Datei Bearbeiten Ansicht Layout Werkzeuge Gruppierung Fenster Hilfe
Visio Online

64



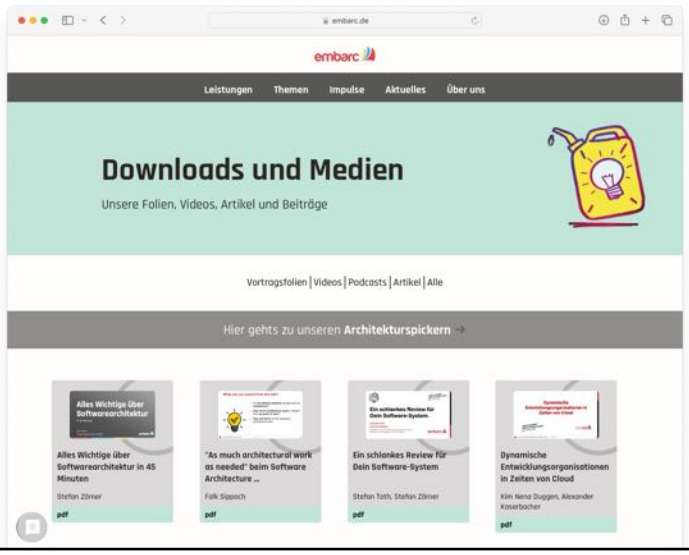
65

Blog-Beitrag zur Mikado-Methode

➔ embarc.de/mikado-methode/

66

Folien als PDF zum Download



➔ embarc.de/download/

67

Architektur-Spicker

„Mit unseren Architektur-Spickern beleuchten wir die konzeptionelle Seite der Softwareentwicklung."

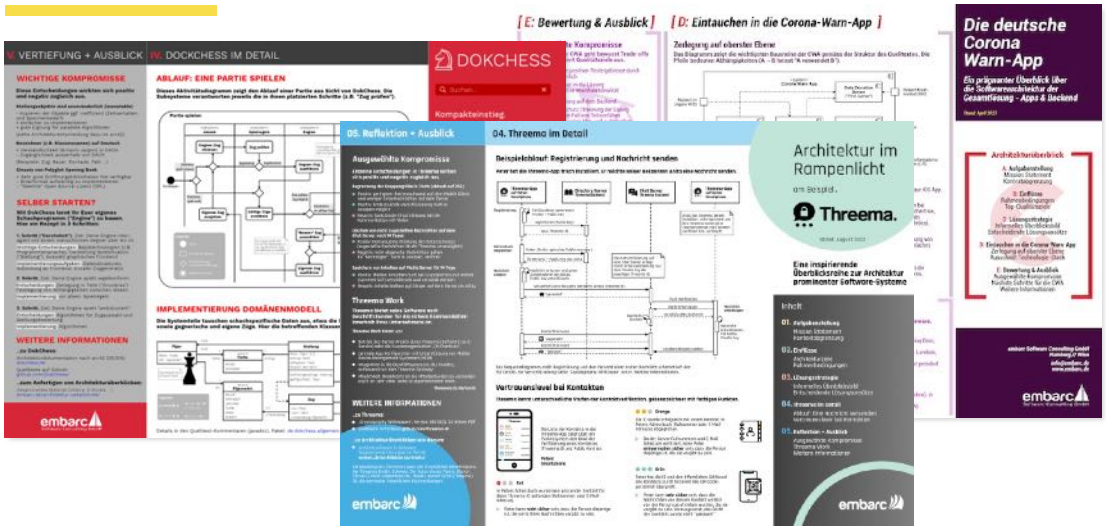
Architektur-Spicker #1
Der Architekturüberblick

PDF, 4 Seiten
Kostenloser Download.

➔ embarc.de/architektur-spicker/

68

Architekturüberblicke als Vorlage



➔ embarc.de/architektur-ueberblicke/

69

70

Feedback & Fragen?

Wir freuen uns auf Fragen, Diskussionen, Anregungen!

71

Vielen Dank.

Ich freue mich auf Eure Fragen!

falk.sippach@embarc.de

linkedin.com/in/falk-sippach

@sippsack

@sippsack@ijug.social

72