

# Agile Methoden im Softwareprojekt

Janet Siegmund  
Thomas Thüm  
Sandro Schulze  
Elmar Jürgens

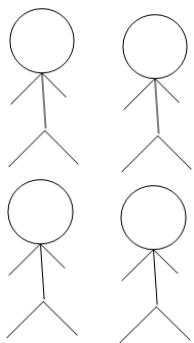
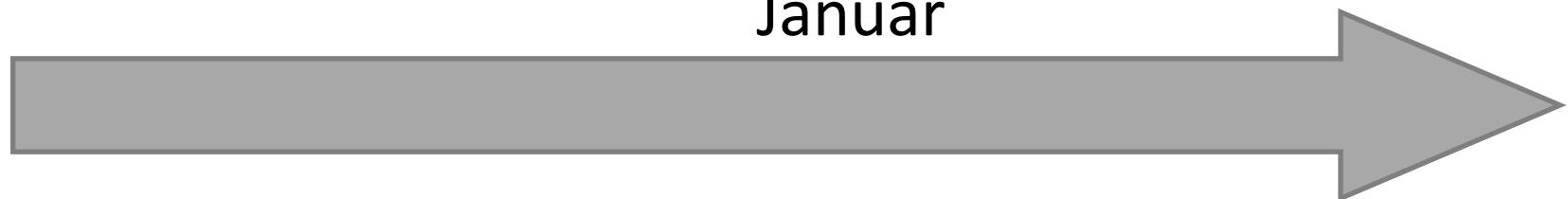


# Eigene Erfahrung

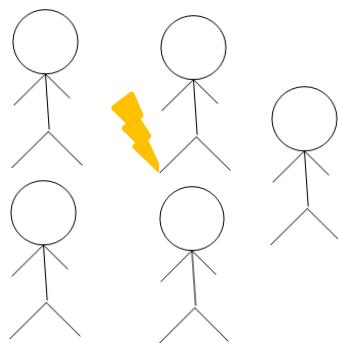
September

Dezember/  
Januar

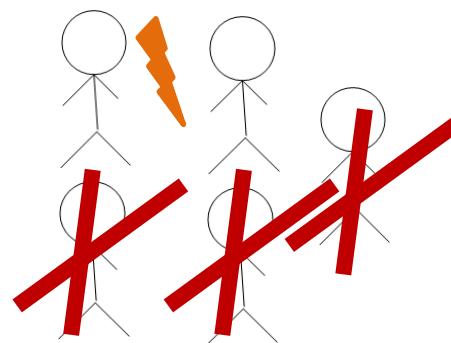
Februar



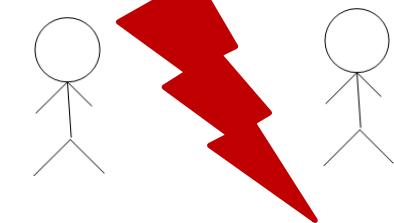
Aufgaben  
festgelegt  
und verteilt



Neues  
Team-  
mitglied



Aussteigen von  
Teammitgliedern



Nur Konflikte

# Gute Erfahrung?

- „Ich würde dieses Projekt jedem Studenten empfehlen“
- „Es ist wirklich ein Projekt was die Arbeit und Zeit wert ist.“
- „Insgesamt hat es mir viel gebracht.“

# 3 Probleme

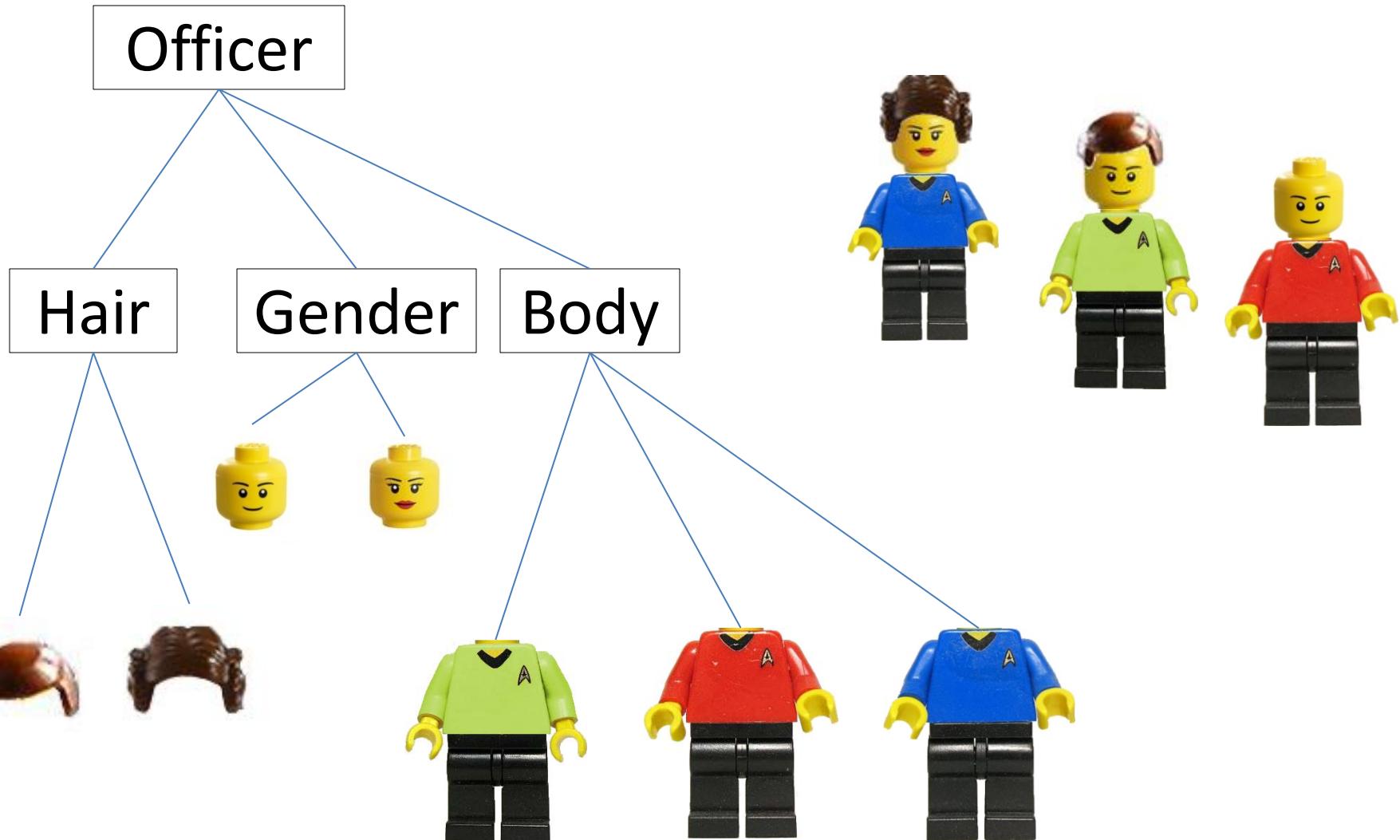
- Verlauf: Arbeitslast im Laufe des Semesters nimmt zu, so dass keine Zeit mehr für Projekt ohne feste Termine bleibt
- Zeitpunkt: Inhalte von Software Engineering sollen geübt werden, werden aber gerade erst gelehrt
- Themen: Unterschiedliche Themen mit verschiedener Komplexität, oft nach Wasserfallmodell

- Lösungen

# Verlauf und Zeitpunkt

- Softwareprojekt als Block nach den Prüfungen
- 3 Wochen, 8 Stunden pro Tag
- Anwesenheitspflicht

# Produktlinien



# Variable Software



```
sched_rotate=NULL,
sched_forget=NULL,
sched_declare=NULL,
sched_pick=>nsched_idle_pick,
sched_setparam=>nsched_idle_setparam,
sched_getparam=>nsched_idle_getparam,
sched_trackprio=>nsched_idle_trackpriority,
.weight=XNSCHED_CLASS_WEIGHT(0),
.name="idle"
};

#endif
```

```
#include <nucleus/pod.h>
#include <nucleus/heap.h>
#include <nucleus/registry.h>
#include <nucleus/thread.h>
#include <nucleus/assert.h>

#ifndef
CONFIG_XENO_OPT_DEBUG_REGISTR
Y
#define
```

```
1 public class Class5 {
2     int number1, number2;
3     public static void main(String[] args) {
4         int temp;
5         do {
6             if (number1 < number2) {
7                 temp = number1;
8                 number1 = number2;
9                 number2 = temp;
10            }
11            temp = number1 % number2;
12            if (temp != 0) {
13                number1 = number2;
14                number2 = temp;
15            }
16        } while (temp != 0);
17        System.out.println("result: " +
number2);
18    }
19    public void setNumber1(int num){
20        number1 = num;
21    }
22    public void setNumber2(int num){
23        number2 = num;
24    }
25 }
```

```
1 public class Class5 {
2     int number1, number2;
3     public static void main(String[] args) {
4         int temp;
5         do {
6             if (number1 < number2) {
7                 temp = number1;
8                 number1 = number2;
9                 number2 = temp;
10            }
11            temp = number1 % number2;
12            if (temp != 0) {
13                number1 = number2;
14                number2 = temp;
15            }
16        } while (temp != 0);
17        System.out.println("result: " +
number2);
18    }
19    public void setNumber1(int num){
20        number1 = num;
21    }
22    public void setNumber2(int num){
23        number2 = num;
24    }
25 }
```

# FeatureIDE

FeatureIDE - GPL-FH-Java/model.xml - Eclipse Platform

File Edit Navigate Search Project Run Window Help

Quick Access FeatureIDE

GPL-FH-Java Model

GraphProductLine

Algorithm

Type

Search

Shortest

Connected

Cycle

Direction

Wight

BFS

DFS

Legend:

- Mandatory
- Optional
- Or
- Alternative
- Abstract
- Concrete
- False optional
- False optional constraint

Constraints

- Connected  $\Rightarrow$  Undirected  $\wedge$  Search
- Cycle  $\Rightarrow$  Direction  $\wedge$  DFS
- Shortest  $\Rightarrow$  Directed  $\wedge$  Weighted

Context menu (right-clicked on Unweighted):

- Create Feature Above
- Create Feature Below (Ins)
- Create Constraint
- Rename (F2)
- Delete (Del)
- Delete including subfeatures
- Mandatory (Double Click)
- Abstract
- Hidden
- Set Layout
- Set Calculations
- Reverse Feature Order

Feature Diagram | Feature Order | Source

The screenshot shows the FeatureIDE interface within the Eclipse Platform. The main window displays a feature hierarchy diagram under the title 'GPL-FH-Java Model'. The root node is 'GraphProductLine', which branches into 'Algorithm', 'Type', and 'Search'. 'Algorithm' further branches into 'Shortest', 'Connected', 'Cycle', and 'Direction'. 'Direction' branches into 'Directed' and 'Undirected'. 'Type' branches into 'Wight' and 'BFS/DFS'. A legend on the right defines symbols for different feature types and constraints. A context menu is open over the 'Unweighted' node under 'Direction', listing options like 'Create Feature Above', 'Delete', and 'Mandatory (Double Click)'. The bottom navigation bar includes tabs for 'Feature Diagram', 'Feature Order', and 'Source'.

# Ablauf

- 2-3 Tage:
  - Einführung in Eclipse und FeatureIDE
  - Einführung in Versionsverwaltung
  - Einführung in agile Methoden
- Danach:
  - Implementierung in Paaren
  - Tägliche Stand-up Meetings
  - Abnahmen erledigter Aufgaben

# Aufgaben

- Innerhalb weniger Tage lösbar
- Aufgabe war nur als gelöst markiert, wenn vom Betreuer abgenommen
- Beispiele:
  - Hintergrundfarben, um Features hervorzuheben in verschiedene Views integrieren
  - Jeder View als Teilaufgabe



# Implementierung

- Einteilung in Teams je nach Anzahl Studierender
- Ein Team bekommt Aufgabe
- Innerhalb der Teams Pairprogramming mit regelmäßigem Rollenwechsel

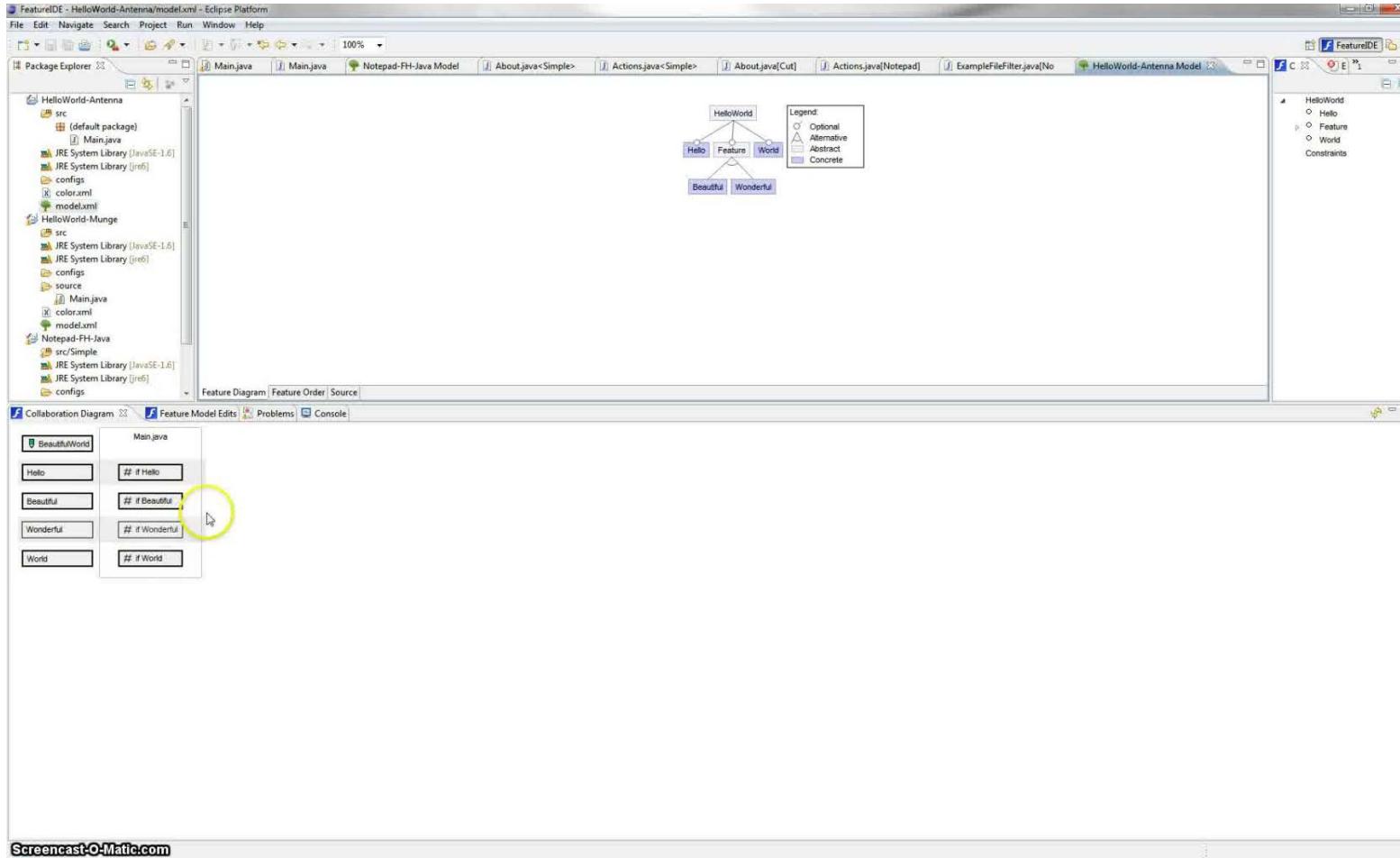
# Stand-up Meetings

- Täglich, 12 Uhr
- Paarweise Fortschrittsbericht

# Betreuer

- Zu Beginn Einführung in Grundlagen
- Mindestens einer immer griffbereit

# Ergebnisse



Screencast-O-Matic.com

# Feedback von Studierenden

- „Ich würde dieses Projekt jedem Studenten empfehlen“
- [...] Sehr motivierend fand ich ebenfalls die täglichen Teammeetings und das Story board, da man dort immer die weiteren Fortschritte besprechen bzw. sehen konnte. **Ich würde dieses Projekt jedem Studenten, mit genügend Java Kenntnissen und Willen zum Vertiefen dieser Skills, empfehlen.**

# Feedback von Studierenden

- „Es ist wirklich ein Projekt was die Arbeit und Zeit wert ist.“
- “Man wird sehr schnell lernen wie man produktiv arbeitet, welche Probleme man zu bewältigen hat und ob seine Arbeit wirklich gut ist. Und das beste ist, dass man etwas handfestes erstellt! Das Projekt hat Zukunft und auch viele Interessenten  
- **es ist wirklich ein Projekt was die Arbeit und zeit wert ist.**“

# Feedback von Studierenden

- „Insgesamt hat es mir viel gebracht.“
- “Zuerst hatte ich einige Bedenken, aber das das Projekt zu festen Zeiten in der Uni statt fand war wirklich gut. Dadurch war der Ablauf sehr gut planbar und man konnte jeden Tag seinen ‘Feierabend’ genießen. [...] **Insgesamt, hat es mir viel gebracht** und kann es jedem weiterempfehlen. [...] Man arbeitet an einem großen, bestehendem Projekt, dass auch tatsächlich eingesetzt wird. Man lernt etwas darüber sich in fremden Code zurecht zu finden.”