

Vorkurs in Informatik: Tag 2

Eine Einführung in die Systematische Software Entwicklung

Emanuel Giger, Giacomo Ghezzi, Michael Würsch, and
Harald Gall

University of Zurich, Switzerland



University of Zurich
Department of Informatics



Ablauf

1. Tag: Grundlagen
2. Tag: Software Engineering by Example
3. Tag: Einführung in die Programmierung

Ablauf: Tag 1

09:30 bis 12:00

- Was ist ein Computer?
- Wie ist ein Computer aufgebaut?
- Das Rechnen mit Wahrheitswerten
- Zahlensysteme
- Wie bringe ich den Computer dazu, für mich Probleme zu lösen?

13:00 bis 16:00

- Eine Einführung in die Programmierung mit Scratch

Ablauf: Tag 2

09:30 bis 12:00

- Eine Einführung in das systematische Entwickeln von Software (aka. Software Engineering)
- Beginn Gruppenarbeiten: Ein kleines eigenes Projekt mit Scratch

13:00 bis 16:00

- Fortsetzung vom Morgen

Ablauf: Tag 3

09:30 bis 12:00

- Kurzpräsentationen der Gruppenarbeiten vom Vortag
- Eine Einführung in die Programmierung mit Groovy

13:00 bis 16:00

- Fortsetzung vom Morgen

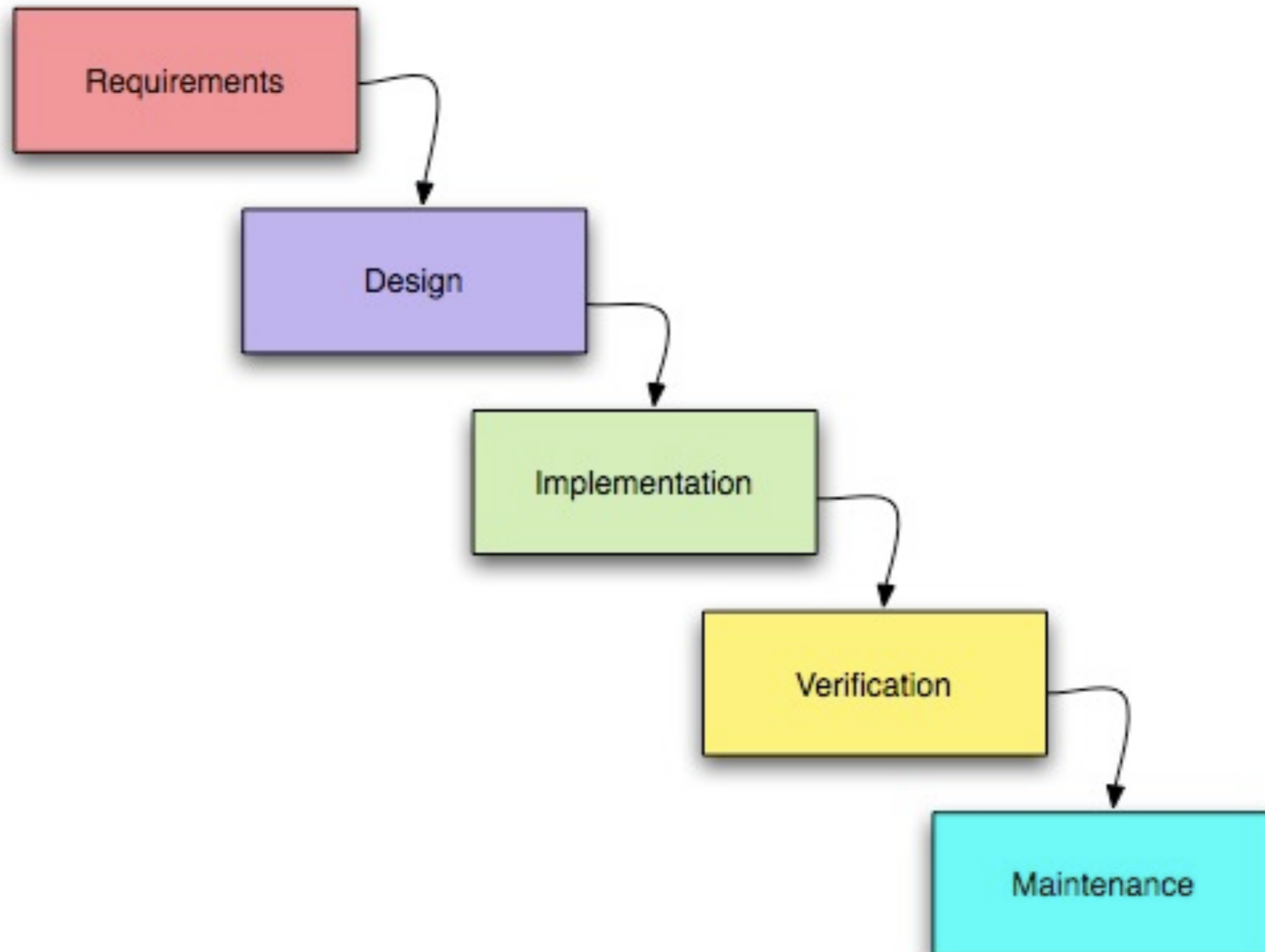
Zielsetzung

Einen Vorstellung davon erhalten, welche Schritte von der Anforderung bis hin zum fertigen Softwareprodukt nötig sind.

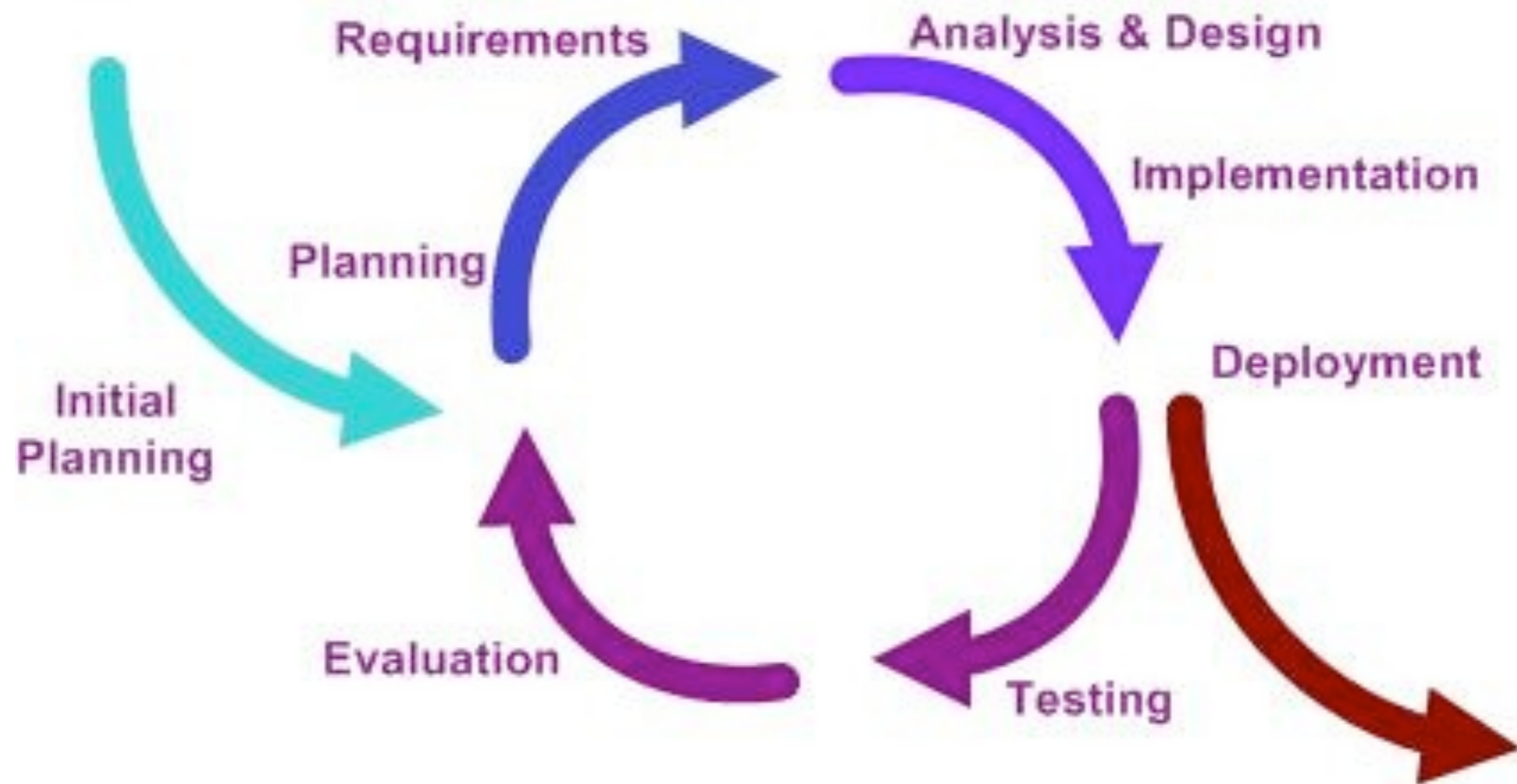
Ein erstes eigenes Produkt entwickeln.



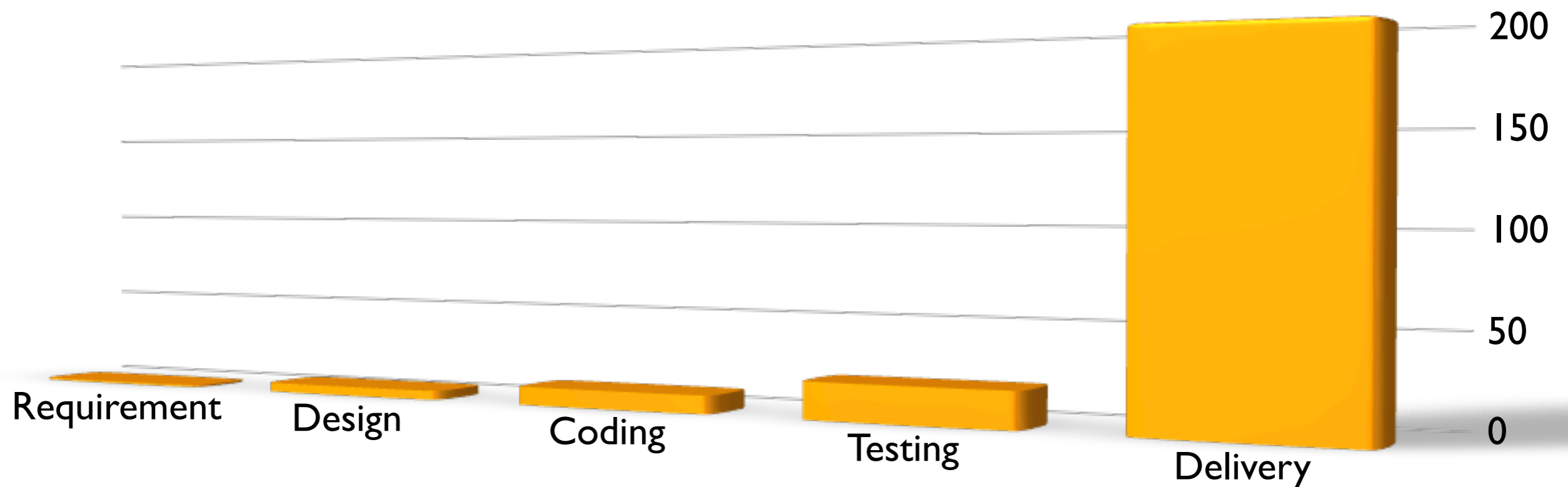
Der Software Lebenszyklus auf dem Papier



Der Software Lebenszyklus in der Realität



Die Kosten der Fehlerbehebung



Relative costs of fixing mistakes [Davi95]

Pareto-Prinzip* im Software Engineering

20% der Anforderungen bedingen 80% der Komplexität

80% des Systems sind in 20% der Zeit fertig gestellt

20% des Codes beinhalten 80% der Fehler

80% der Fehler werden in 20% der Zeit behoben



*Nach Vilfredo Pareto (1848-1923), Italienischer Ökonom und Gesellschaftstheoretiker.

Die Anforderungsanalyse

Perfektion ist erreicht, nicht, wenn sich nichts mehr hinzufügen lässt, sondern, wenn man nichts mehr wegnehmen kann.

▶ Antoine de St. Exupery, *Terre des Hommes*, 1939



Anforderungen?

Gut formulierte Anforderungen könnten wie folgt lauten:

- Eine Mitarbeiterakte kann lediglich von einer dazu speziell berechtigten Gruppe von Personen eingesehen werden.
- Der Texteditor soll Schlüsselwörter hervorheben, die abhängig vom gerade bearbeiteten Dateityp ausgewählt werden.

Wie erhebe ich Anforderungen?

Work with a User to Think Like a User

- ▶ Tip 52, Andrew Hunt and David Thomas in 'The Pragmatic Programmer'

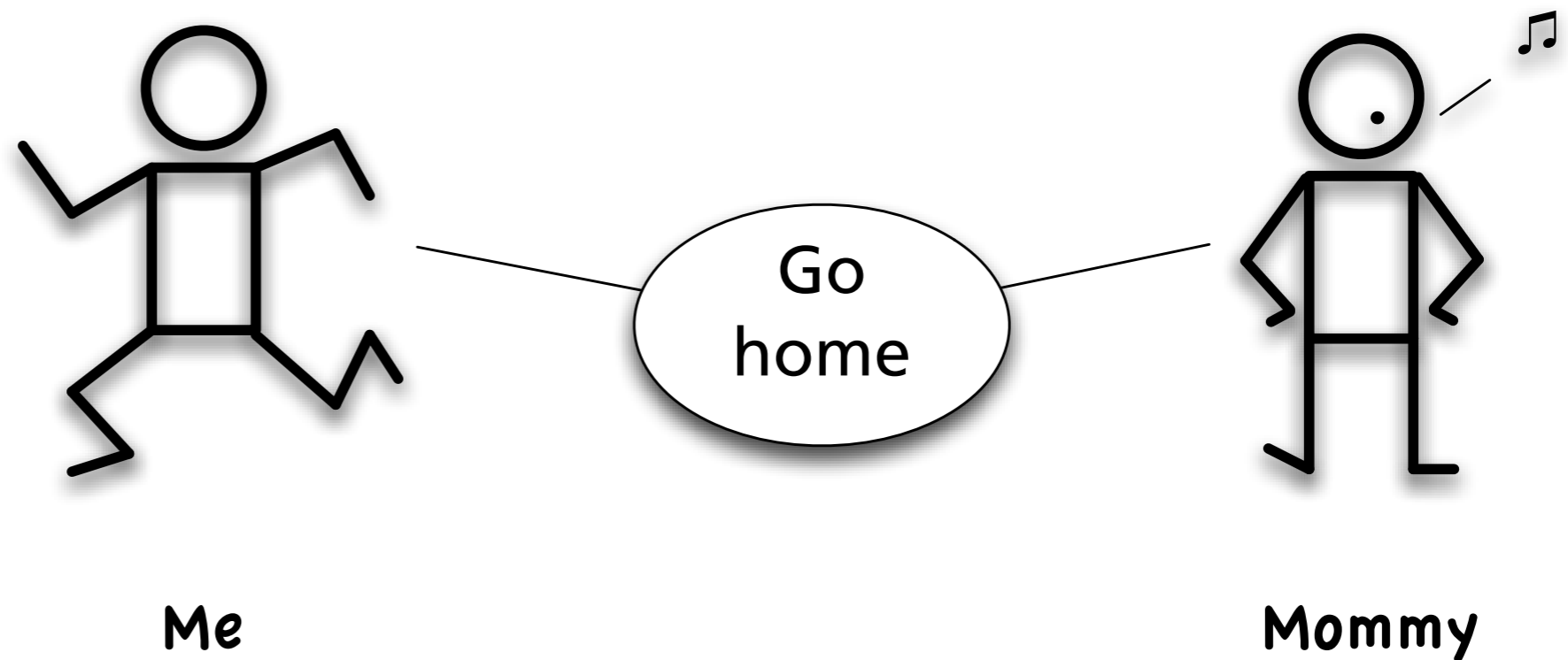


Für wen dokumentiere ich Anforderungen?



Wie dokumentiere ich Anforderungen?

Use Cases und Use Case Diagramme



Entwurf

Datenbank

- Daten laden
- Daten speichern
- ...

User Interface

- Bild darstellen
- Liste mit Namen anzeigen
- ...

Server

- Anfrage XY beantworten
- Passwortabfrage senden
- ...

Client

- Berechnung XY durchführen
- ...

Implementierung



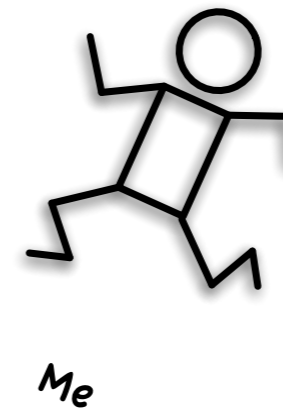
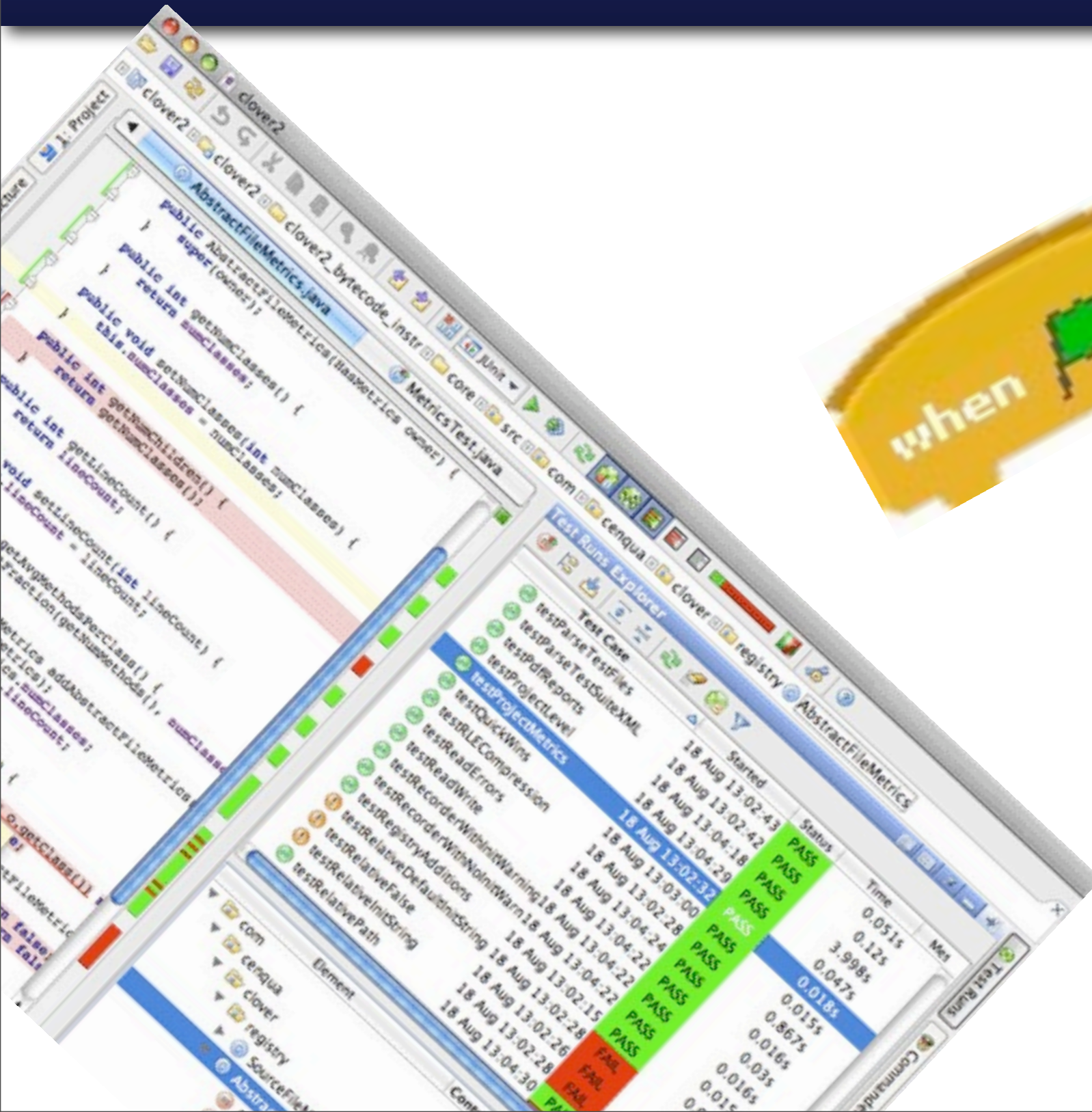
say Hello!

repeat 10

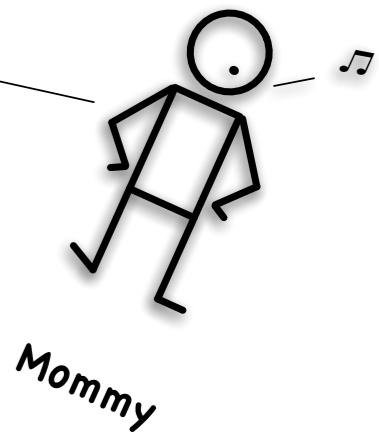
play sound pop

move 10 steps

Testen



Go home



Aufgabe: Ein erstes Projekt mit Scratch

Programmiert ein Spiel in Scratch!

1. Zweierteams bilden.
2. Zwei Teams erarbeiten gemeinsam Use Cases und ein grobes Design.
3. Implementierung im Team!
4. Kurzpräsentation morgen früh im Plenum.