

Christian Gebauer, Sebastian Große, Benjamin  
Pfeiffer, Nico Smeenk, Jonathan Wiens

# ANTS ON CRACK

Der Ameisenalgorithmus als Lösung für das Problem des  
Handlungsreisenden

BBC



# Motivation



# Überblick

## Aufgabenstellung

Das Problem des Handlungsreisenden

Der Ameisenalgorithmus

## Projektmanagement

Agile Softwareentwicklung

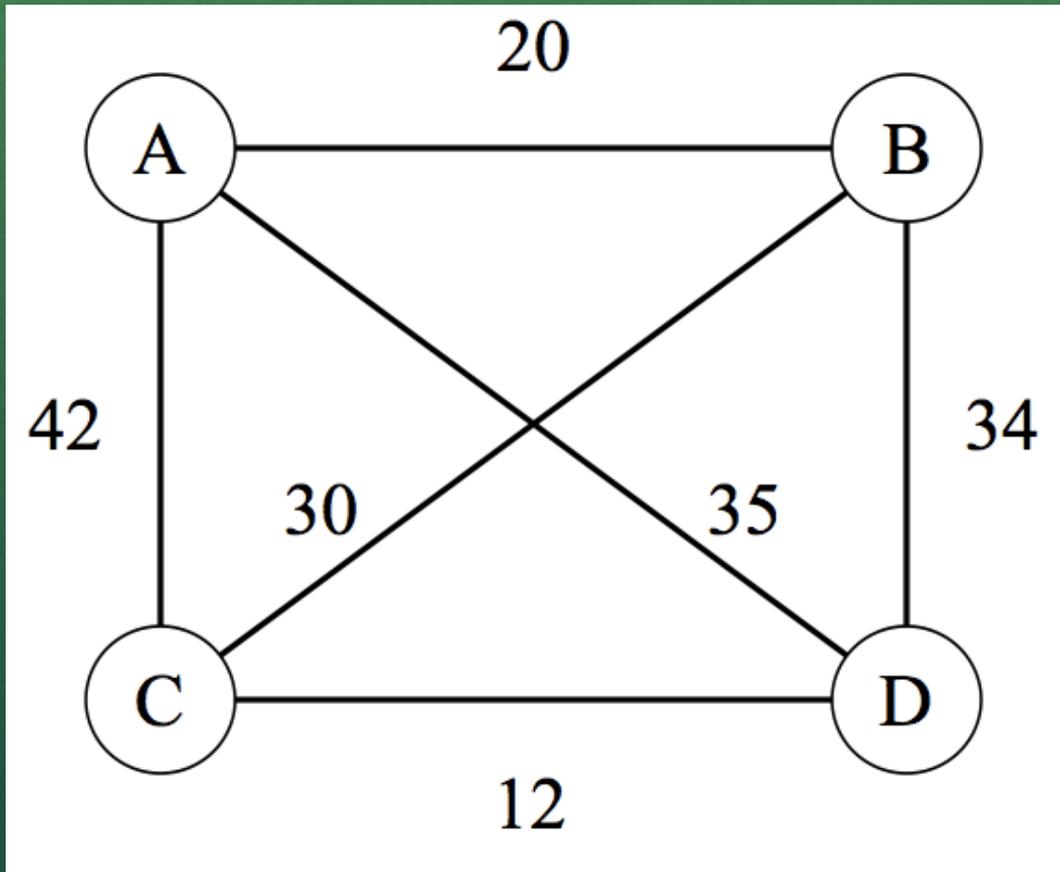
Implementierung

## Fazit

Bewertung

Ausblick

# Das Problem des Handlungsreisenden



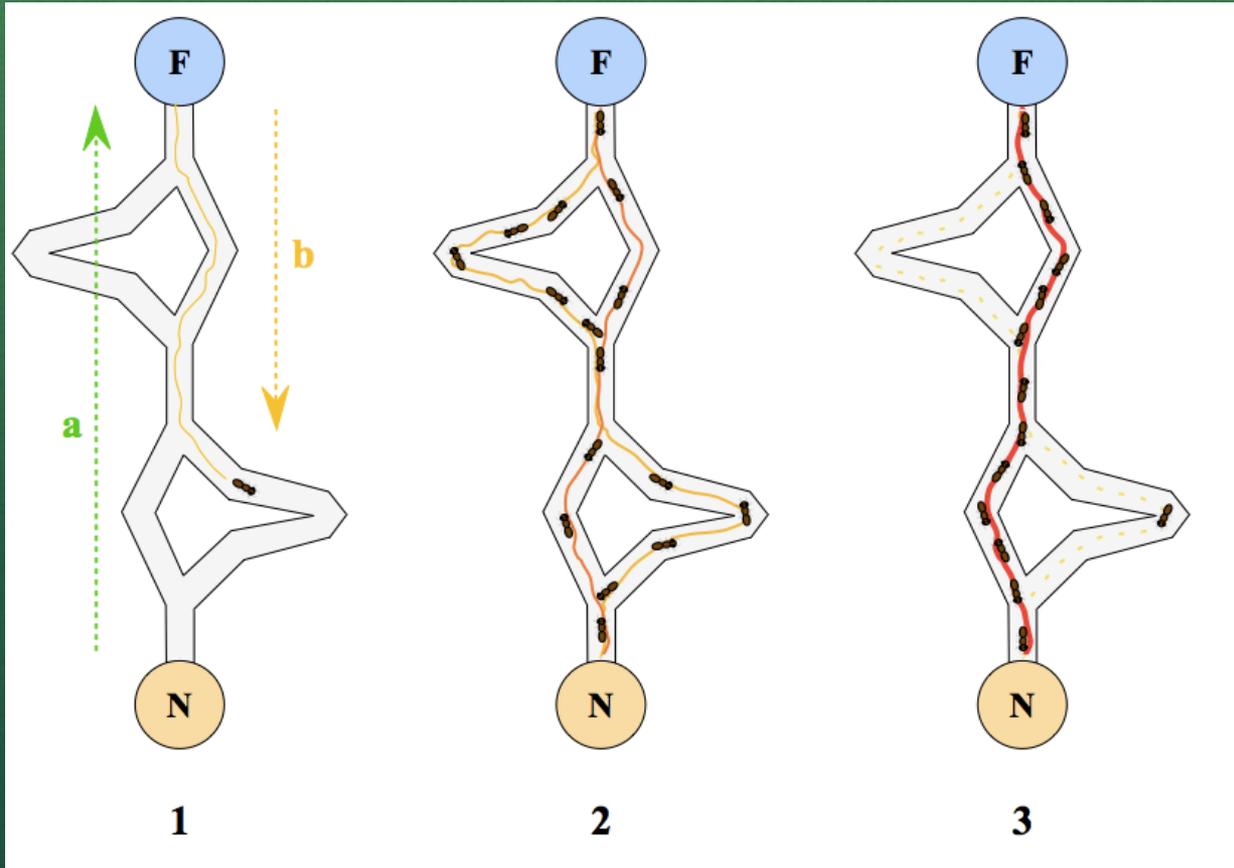
Quelle: wikipedia.org

# Inspiration Ameise



Quelle: [printactivities.com](http://printactivities.com)

# Ameisenalgorithmus dargestellt



Quelle: wikipedia.org

# Ameisenalgorithmus als Formel

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}$$

$p_{ij}^k(t)$  = Wahrscheinlichkeit einer Transition  $t$  für Ameise  $k$  von Stadt  $i$  nach Stadt  $j$

$\tau_{ij}(t)$  = Pheromonwert von Transition  $t$  über Stadt  $i$  nach Stadt  $j$

$\alpha$  = Wichtigkeit der Pheromonspur

$\eta_{ij}$  = lokale Information über Weg von Stadt  $i$  nach Stadt  $j$

$\beta$  = Wichtigkeit der lokalen Information

$\sum_{l \in J_i^k} [\dots]$  = Summe aller noch möglichen Wege mit Pheromon- und Heuristikbewertung

# Die Teammitglieder



Jonathan;  
Projektleiter



Nico;  
Architekt



Benjamin;  
Programmierer,  
Dokumentation



Sebastian;  
Programmierer,  
Dokumentation,  
Serveradministrator



Christian;  
Programmierer,  
Dokumentation,  
Qualitätssicherung

# Projektmanagement

## Agile Softwareentwicklung

Stand-Up  
Meetings

Story Cards

XP

Media Wiki

Coding  
Habits

# Projektmanagement

## Stand-Up Meetings



# Projektmanagement

## Story Cards

### Story-Cards in Warteschleife

Titel: #5 Java Datenbank-Zugriff: Gruppen in der Datenbank verwalten	
Datum:	28.09.2011
Autor:	Benjamin
Beschreibung:	Zur vorhandenen DAO-Klasse sollen verschiedene Methoden zur Gruppenverwaltung angelegt werden: Es sollen dann folgende Methoden implementiert werden: <ul style="list-style-type: none"> <li>• Methode um eine Gruppe in die Datenbank zu schreiben (Gruppe erstellen)</li> <li>• Methode um Informationen einer Gruppe zu verändern (Gruppe soll bearbeitet werden)</li> <li>• Methode um eine Gruppe zu löschen</li> <li>• Methode um eine Gruppe anhand einer Id-Nummer aus der Datenbank abzufragen</li> </ul> Anmerkung: Es muss vorher eine Daten-Schnittstelle geschaffen werden. Dazu soll es eine Java-Bean geben "Group", die alle nötigen Attribute enthält um Gruppeninformationen zu transportieren.
Art:	technische Anforderung
Geschätzter Aufwand:	8h
Tatsächlicher Aufwand:	
Datum der Erledigung:	
Entwickler:	
Priorität:	1

### Story-Cards in Arbeit

Titel: #5 Java Datenbank-Zugriff: Benutzer in der Datenbank verwalten	
Datum:	28.09.2011
Autor:	Benjamin
Beschreibung:	Es soll zunächst eine DAO-Klasse geschaffen werden, die Methoden enthält, die die gesamte Logik enthält Daten aus der Datenbank zu holen und auch wieder welche hineinzuschreiben.  Es sollen dann folgende Methoden implementiert werden: <ul style="list-style-type: none"> <li>• Methode um einen Benutzer in die Datenbank zu schreiben (registrieren)</li> <li>• Methode um Informationen eines Benutzer zu verändern (User möchte sein Profil bearbeiten)</li> <li>• Methode um einen Benutzer zu löschen</li> <li>• Methode um einen Benutzer anhand einer Id-Nummer aus der Datenbank abzufragen</li> </ul> Anmerkung: Es muss vorher eine Daten-Schnittstelle geschaffen werden. Dazu soll es eine Java-Bean geben "User", die alle nötigen Attribute enthält um Benutzerinformationen zu transportieren.
Art:	technische Anforderung
Geschätzter Aufwand:	8h
Tatsächlicher Aufwand:	
Datum der Erledigung:	
Entwickler:	
Priorität:	3

### Story-Cards bearbeitet, ggf. Test

Titel: #1 Config-Manager: Einstellungen lesen	
Datum:	25.09.2011
Autor:	Benji
Beschreibung:	Es soll eine Java Klasse geben, die Funktionen bereitstellt um eine Konfigurations-datei zu lesen. Es kann unterschieden werden in Konfigurationsdateien für Server oder Clienten. Wird keine Datei im System gefunden, wird eine Standard-Config erstellt. Die ausgelesenen Einstellungen werden in Form einer Hash-Map übergeben, wobei der erste Wert der Name der Einstellung ist, der zweite der Parameter-Wert. Alle Konfigurationen müssen in der Form: "ParameterName = Wert" geschrieben werden. Alles andere wird ignoriert.
Art:	technische Anforderung
Geschätzter Aufwand:	6h
Tatsächlicher Aufwand:	7h
Datum der Erledigung:	28.9.2011
Entwickler:	Benjamin
Priorität:	1

Titel: #3 Datenbank modellieren und aufbauen	
Datum:	26.09.2011
Autor:	Benji
Beschreibung:	Es soll für den Server eine Datenbank aufgebaut werden. (Siehe HWR-Chat-Datenbank konzept). Das sql-Skript zum erstellen dieser Tabellen soll angelegt werden
Art:	technische Anforderung / fachliche Anforderung
Geschätzter Aufwand:	5h
Tatsächlicher Aufwand:	4h
Datum der Erledigung:	27.09.2011
Entwickler:	Benji
Priorität:	3

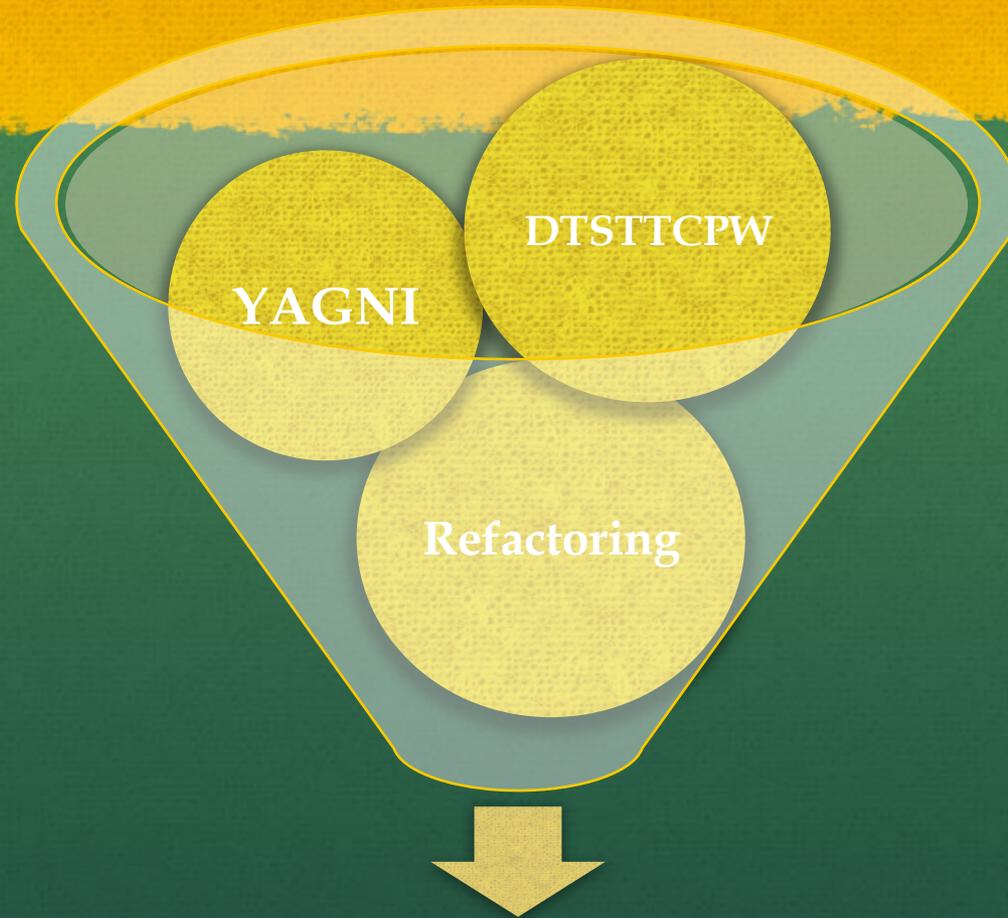
Titel: #4 Datenbank-Schnittstelle schaffen	
Datum:	26.09.2011
Autor:	Benji
Beschreibung:	Für den Datenbank zugriff muss eine Schnittstelle programmiert werden, die es erlaubt: <ul style="list-style-type: none"> <li>• sich mit der Datenbank zu verbinden (auch Standard möglich, also z.B. User-admin)</li> <li>• Daten in die Datenbank zu schreiben und auch wieder auszulesen</li> <li>• Daten in der Datenbank abzufragen</li> <li>• Status der Verbindung abzufragen</li> </ul> Schnittstelle: Daten an die Datenbank: SQL-Skript. Daten aus der Datenbank: Java-Resultat
Art:	technische Anforderung
Geschätzter Aufwand:	6h
Tatsächlicher Aufwand:	6h
Datum der Erledigung:	27.09.2011
Entwickler:	Nico
Priorität:	3

# Projektmanagement

## Story Cards

Titel:	#10 GUI - Handbuch
Datum:	08.09.2011
Autor:	Sebastian
Beschreibung:	<ul style="list-style-type: none"><li>■ kurze Anleitung des Programmes</li></ul>
Art:	fachliche Anforderung
Geschätzter Aufwand:	4 h
Tatsächlicher Aufwand:	2 h
Datum der Erledigung:	15.10.2011
Entwickler:	Benjamin, Jonathan
Priorität:	1

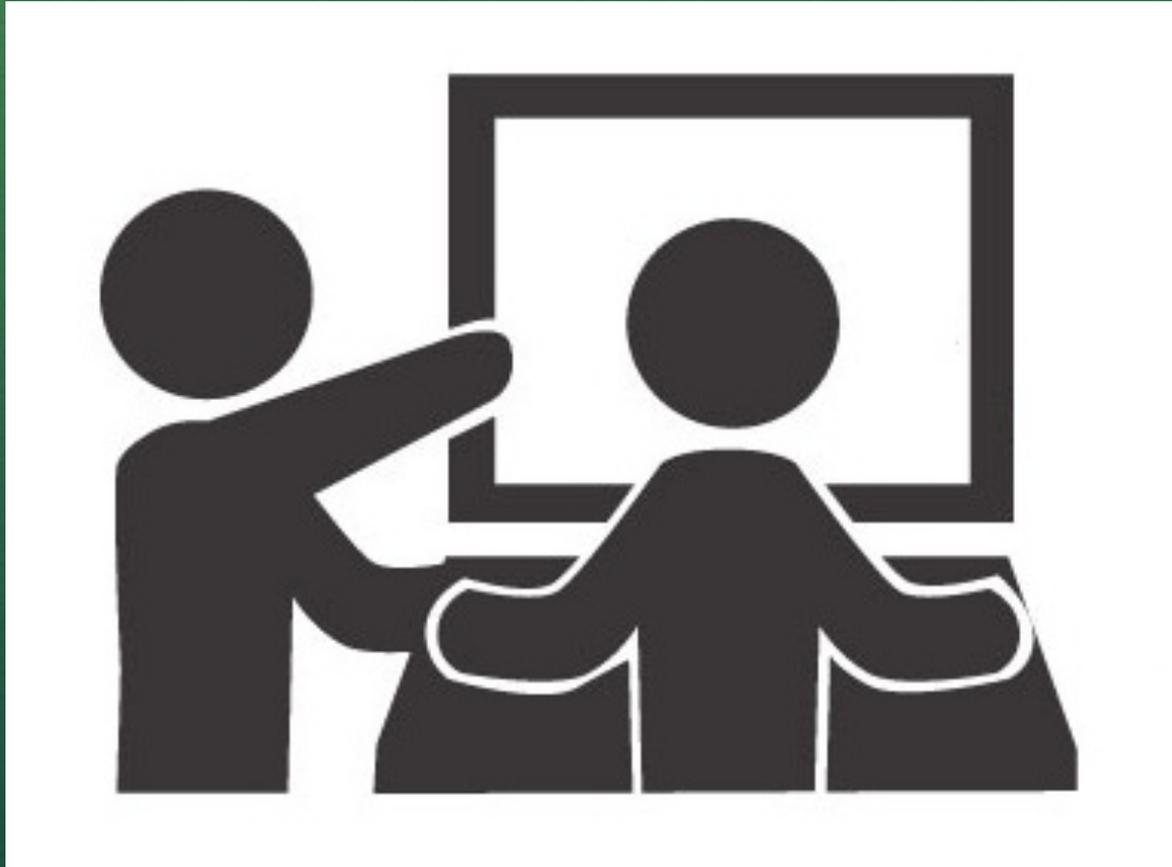
# Projektmanagement



Coding Habits

# Projektmanagement

## Pair Programming



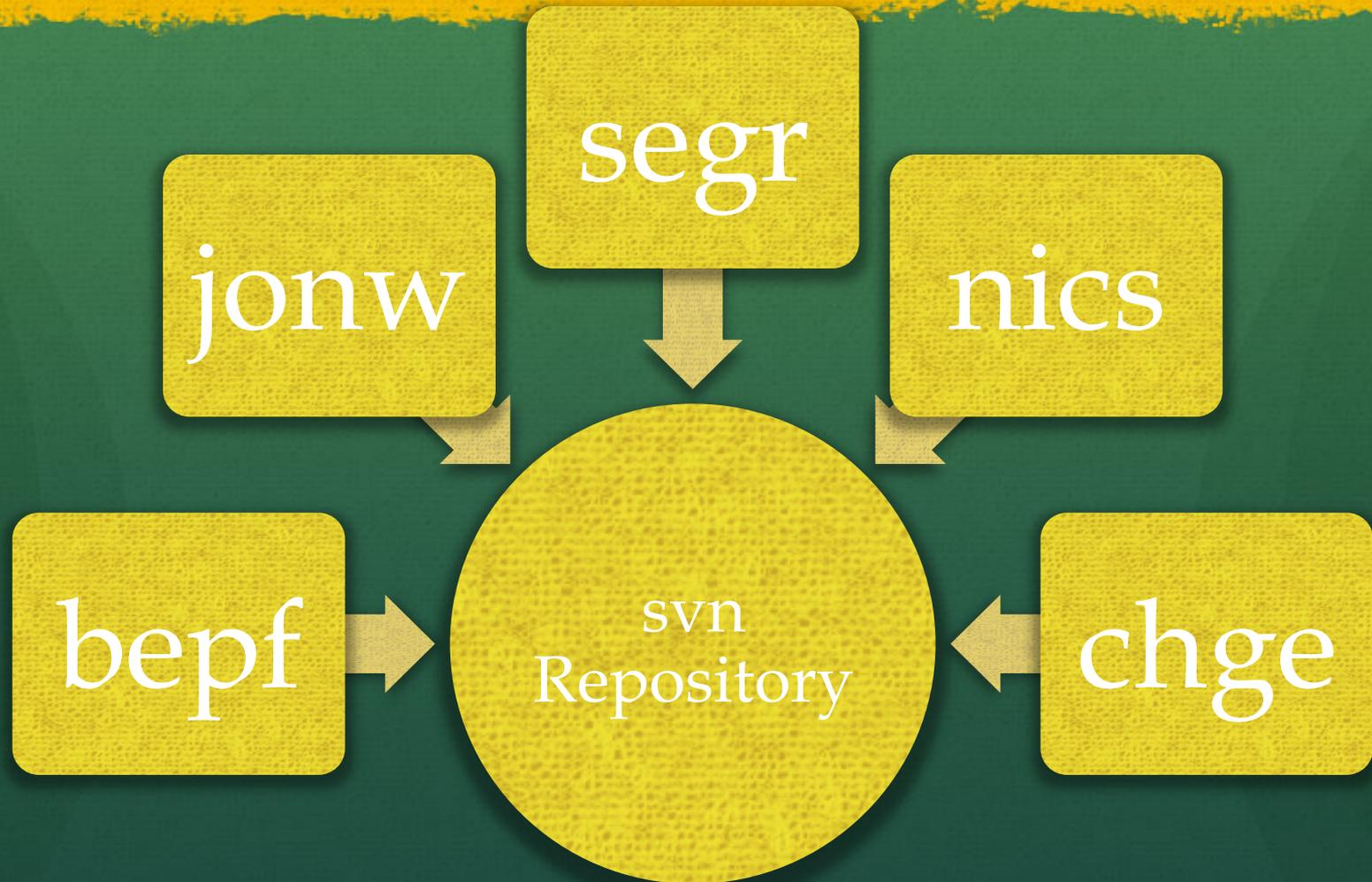
# Projektmanagement

## Media Wiki



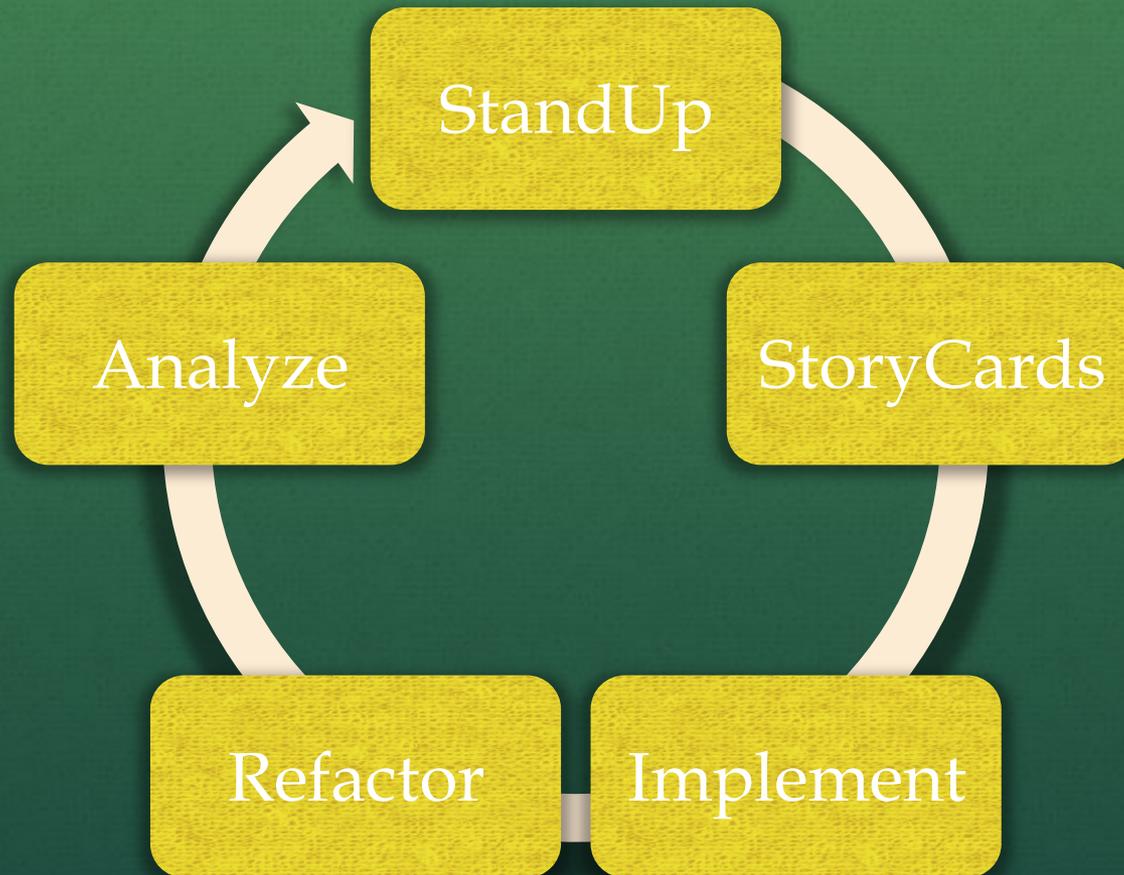
- wikipedia.org-Klon
- GPL-Lizenzierung
- “Because ideas want to be free”

# Projektmanagement Organisation



# Projektmanagement

## Iteration Progress



# Implementierung

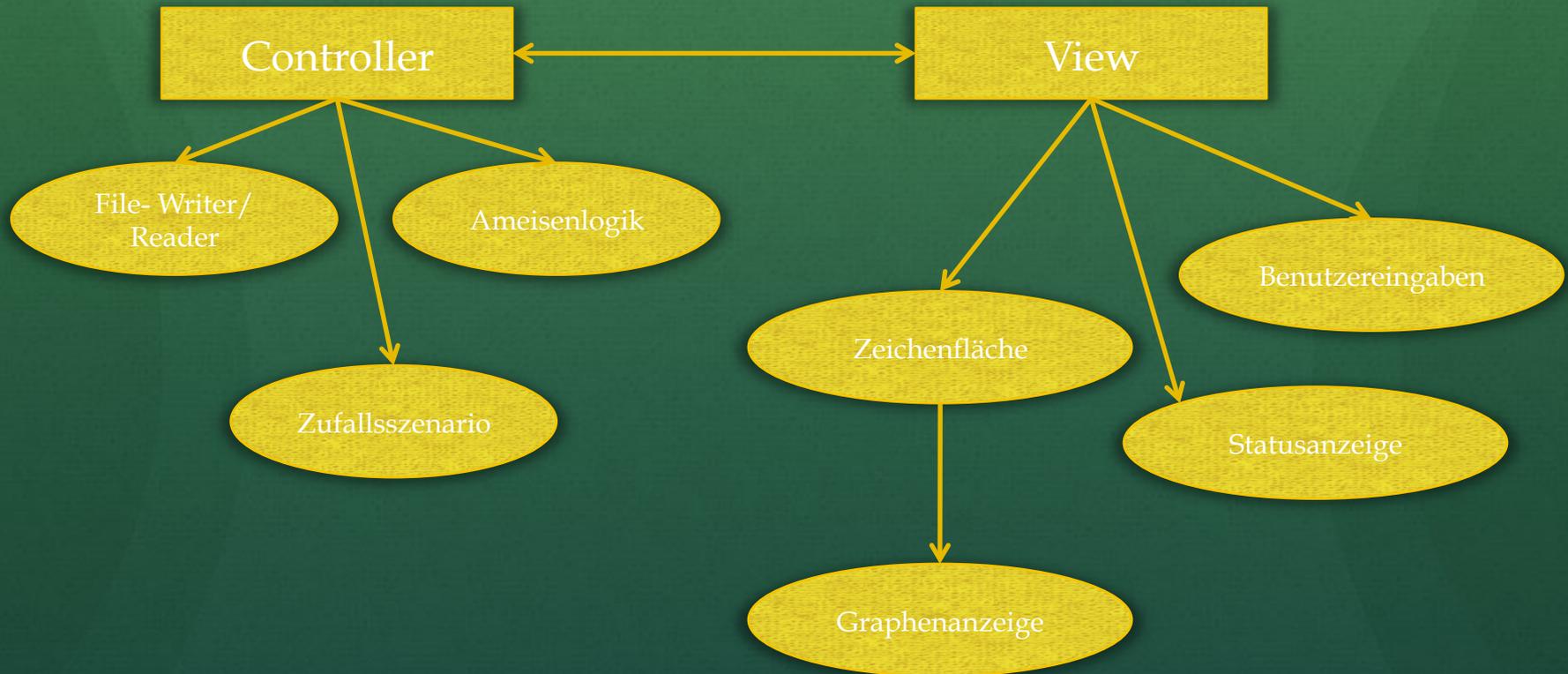
## Architektur

- Programmiersprache: Java 1.6
  - Plattformunabhängig
  - Teammitgliedern bekannt
  
- Framework: Standard Widget Toolkit
  - Passt sich Betriebssystem an
  - Einfach zu erlernen



# Implementierung Architektur

- Modularer Aufbau:



# Implementierung Architektur

- Simulation der Ameisenwege durch Java Maps  
→ Weg zugeordnet zu Weginformationen
- Ameisen “laufen” diese Wege ab  
→ Entscheidungsroutine, Pheromonupdate,  
Stoppkriterium

# Implementierung Architektur

- Zeichenoberfläche in eigenem Thread
- Zeichnen der Wege während der Algorithmus arbeitet
- Kein echtes MVC-Prinzip
- Testen durch verschiedene Testklassen

# Implementierung

## Teamarbeit

- Team spaltet sich in Expertengruppen

View/SWT

Logik/Controller

- Zeitweise “Schulungen” um Architektur zu erklären
- Gemeinsame Schnittstellen definieren

# Bewertung

- Umsetzung des Algorithmus hätte besser geplant werden können
- Architektur / Arbeitsaufteilung erwies sich als sehr gut
- Unterstützung durch SVN brachte Konsistenz

# Bewertung

- SWT: Schwierigkeiten beim Ausliefern
  - Jedes Betriebssystem muss anders berücksichtigt werden
- Portsperren in der Hochschule erschweren den Zugang zu unseren Entwicklungstools
- Extreme-Programming-Techniken wurden fast immer eingehalten

# Fazit

- Trotz relativ wenig Zeit ist ein gutes Programm entstanden
  - Arbeit war nicht immer einfach
  - Zukünftiges: Algorithmus optimieren
  - Jeder hat neue Erfahrungen gesammelt
- Persönlich / Technisch

# Fazit

- Erfahrungen können in die Firmen getragen werden

*Jeder hatte Spaß am Projekt*