

# Rollenspielmanager

## für „Die Legenden von Cysteron“

Vortrag 1. Iteration

Projektgruppe Angewandte Softwaretechnologie  
Sommersemester 2015



# Inhalt

- |                      |               |
|----------------------|---------------|
| 1. Idee des Projekts | 4. Demo       |
| 2. Übersicht         | 5. Vorschau   |
| 1. Rollenverteilung  | 6. Reflektion |
| 2. Stories           | 7. Fazit      |
| 3. Fortschritt       |               |
| 1. Nicht-Inhaltlich  |               |
| 2. Inhaltlich        |               |

# Idee des Projekts

- Unterstützung des Spielleiters beim Pen&Paper Rollenspiel DLVC
- Insb. Hilfe bei würfel- und rechenlastigen Kampfsituationen
- Desktop-Applikation
- Entwicklung in Java

# Rollenverteilung

Teamleiter:

Britta Heymann

Kunde:

Hans Klein

Entwickler:

Andreas Kofer,  
Boris Prochnau,  
Britta Heymann,  
Nooshin Naghavi

# Stories



verschoben



später hinzugekommen

Nr	Titel	Priorität	Bausteine (1 = 2 Paar-Stunden)	abhängig von
1	Coding Conventions	1	6,5	-
2	Datenspeicherung	1	4	-
3	Datenstruktur	1	2	-
4	Architektur festlegen	1	6	(3)
5	Hauptmenü	1	2,5	1,4
6	Spielgruppenmanager	1		1,2,3,4,(5)
7	Kampfsimulator	1	12	1,2,3,4,(5)
8	Würfelsimulator	2	2	1,(5)
9	Vortrag	1	1,5	(-)

# Fortschritt: Nicht-inhaltlich

- Einigungen auf Coding Conventions
  - Stil
  - Dokumentation
  - Tests
- Wahl von Technologien, Installation, Tutorials

# Verwendete Technologien / Frameworks

- *JavaFX*
  - Java-Spezifikation von Oracle, Konkurrenz zu Adobe Flash
  - Beschreibung von Oberflächen durch XML
  - Unterstützung von CSS
- *Scene Builder*
  - Layout Tool

# Verwendete Technologien / Frameworks

- *Hibernate*
  - Persistenz-Framework für Java
  - Objektrelationale Abbildung
- *H2 Database Engine*
  - Relationales Datenbankmanagementsystem



# Fortschritt: Inhaltlich

- Gespräch mit dem Kunden
- Einlesen in Material zu DLVC
- Erstellung des Datenbankmodells

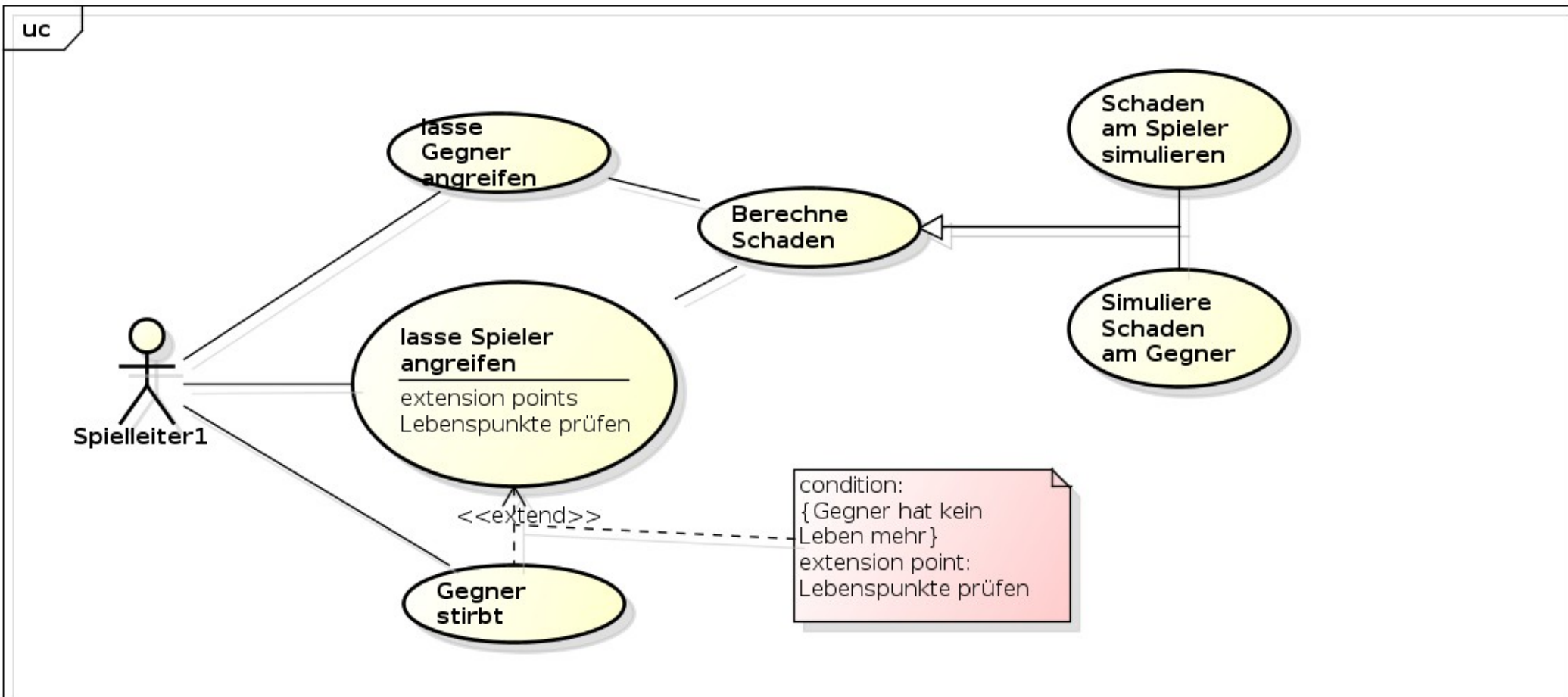
# Fortschritt: Inhaltlich

- Gespräch mit dem Kunden
- Einlesen in Material zu DLVC
- Erstellung des Datenbankmodells
- Erste Szenarien und Use Cases

# Use Case: Charakter speichern

Kurzbeschreibung	Der Charakter eines Spielers wird gespeichert.
Akteure	Spielleiter, Spieler
Vorbedingung	
Ereignisfluss	<ol style="list-style-type: none"><li>1) Spielleiter nimmt die Daten der Spieler auf:<ul style="list-style-type: none"><li>- Name, Sozialer Rang (optional)</li><li>- Rasse, Disziplin</li><li>- DefR, DefH, DefS,</li><li>- Schaden, Sonstige Kampfrelevantes</li></ul></li><li>2) Spielleiter bestätigt die Eingabe.</li><li>3) Charakter wird aktualisiert.</li></ol>
Nachbedingung	Charakter ist mit eingegebenen Daten in der Datenbank gespeichert.
Alternativablauf	3) falls der Charakter in der Datenbank noch nicht vorhanden war, wird er aktualisiert.

# Use Case Diagramm: Kampf



# Fortschritt: Inhaltlich

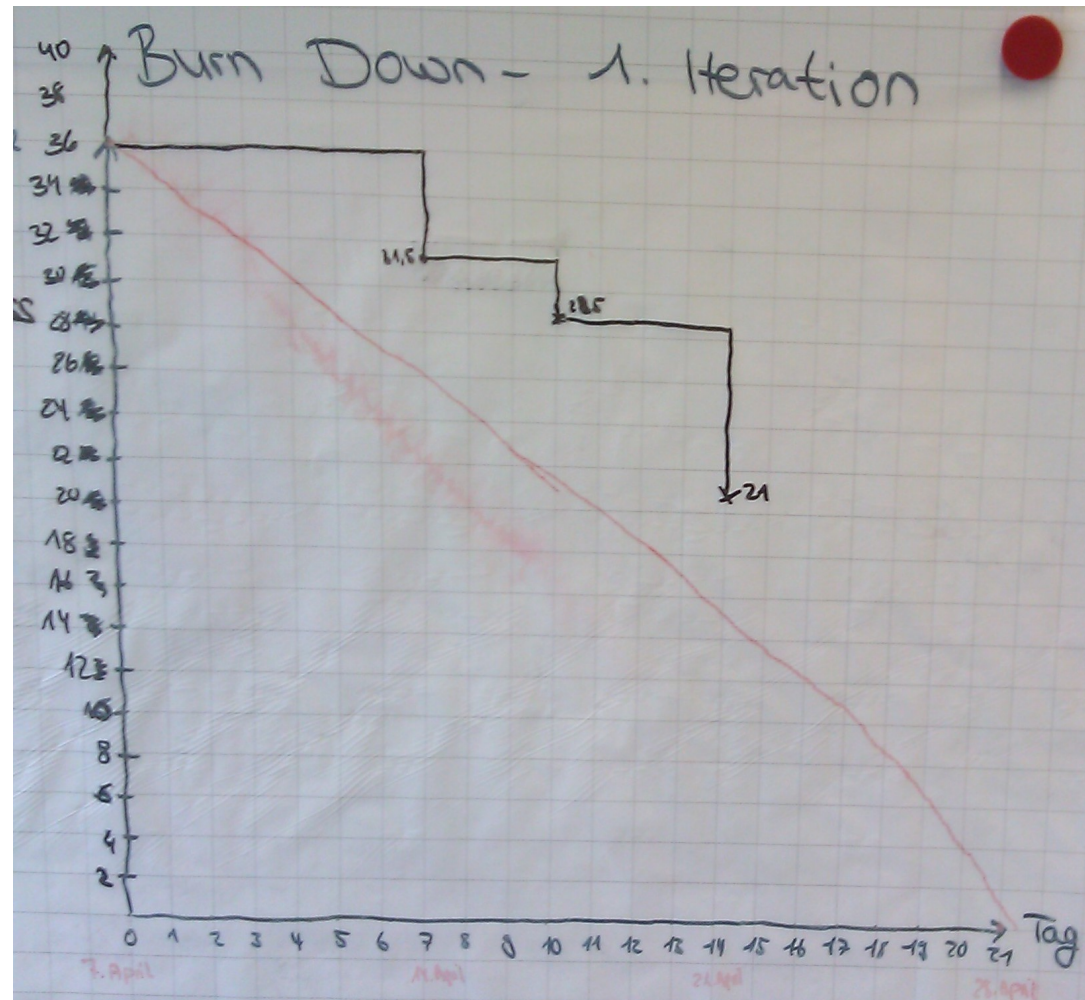
- Gespräch mit dem Kunden
- Einlesen in Material zu DLVC
- Erstellung des Datenbankmodells
- Erste Szenarien und Use Cases
- Hauptmenü
- Würfelsimulator
- Begonnen: Spielgruppenmanager

# Demo

# Iteration 2



- Fertigstellung: Charaktermanagement
- Kampfsimulator

# Burn-Down-Chart





# Erinnerung: Stories

 verschoben  
 später hinzugekommen

Nr	Titel	Priorität	Bausteine (1 = 2 Paar-Stunden)	abhängig von
1	Coding Conventions	1	6,5	-
2	Datenspeicherung	1	4	-
3	Datenstruktur	1	2	-
4	Architektur festlegen	1	6	(3)
5	Hauptmenü	1	2,5	1,4
6	Spielgruppenmanager	1		1,2,3,4,(5)
7	Kampfsimulator	1	12	1,2,3,4,(5)
8	Würfelsimulator	2	2	1,(5)
9	Vortrag	1	1,5	

# Reflektion der Planung

- Nicht alle Arbeitsschritte beachtet
  - Gespräch mit Kunden
  - Schätzen der Stories
  - Stories vergessen (Hauptmenü, Vortrag)
- Einige Stories dauerten länger als gedacht
  - Datenspeicherung: Probleme bei Installation
  - Coding Conventions: Recherche, Diskussion
  - Erweiterung der Story „Charaktermanager“

# Reflektion der Planung

- Nicht alle Arbeitsschritte beachtet
- Einige Stories dauerten länger als gedacht
- Die Beschreibung der Stories u. Tasks war ausreichend
  - bei als risikoreich eingeschätzten Stories auch mal den höheren Wert nehmen
  - prüfen, ob alle Arbeitsschritte abgedeckt

# Reflektion über Teamprozesse

- Kommunikation erlernen: nicht aneinander vorbeireden → nachfragen, zusammenfassen
- Abwägen, wann ein Thema das gesamte Team angeht
- Verschiedene Anwesenheiten → wechselnde Teams, Skype
- Alle waren engagiert :)

# Fazit

- Gewöhnung an Methoden der agilen Softwareentwicklung
  - Entstehung eines Teams
  - Komponenten ausgesucht, erste Funktionalität
- alles bereit für intensive Phase der Entwicklung des ersten Prototyps