

**Käpsele.de** Deine Lernplattform



Käpsele.de vermittelt Wissen spielerisch, indem es die Nutzer durch Herausforderungen zum Lernen motiviert.



## Der Gamification-Ansatz

### Spielerisches Lernen

- In **Quests** können Nutzer durch Multiple Choice-Fragen ihr Wissen selbst überprüfen
- In **Teamcombats** treten zwei Lerngruppen gegeneinander an und stellen sich dabei gegenseitig Fragen
- Den Abschluss jeder Lerngruppe krönt der **Bossfight**, der eine alte Klausur enthält
- Nutzer können sich untereinander **befreunden** und ihre Leistungen durch **Ränge** vergleichen.

### Wissensaufbau

### Motivation

Freundschaften

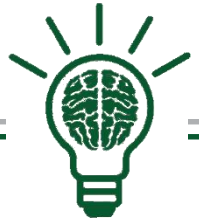
Ränge

Bossfights

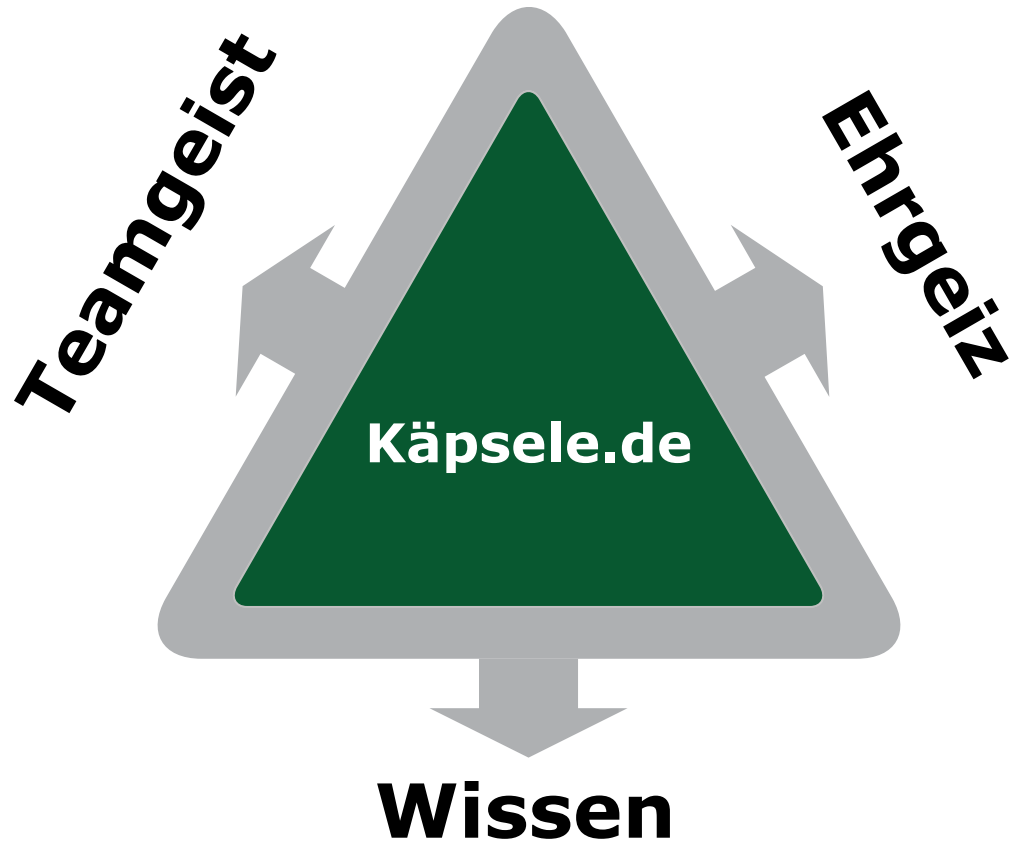
Teamcombats

Quests

Durch die eingebauten Gamifications werden im Benutzer intuitiv Ehrgeiz und Teamgeist geweckt, was die Wissensaufnahme stärkt.



## Der Gamification-Ansatz



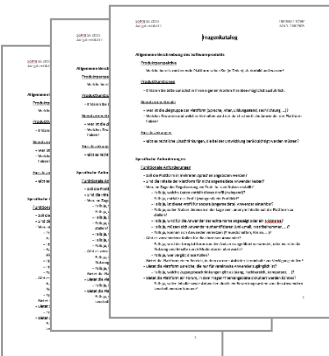
Während der Entstehung hat käpsele.de sämtliche Schritte des Entwicklungsprozesses einer Software durchlaufen.



## Der Entwicklungsprozess

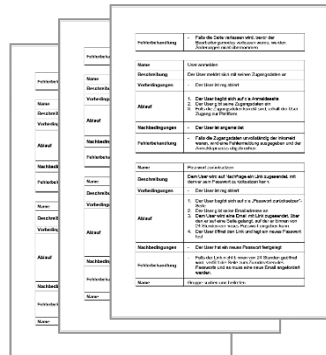
### Anforderungs- erhebung

- Erhebung der Anforderungen des Kunden mithilfe eines Fragenkatalogs



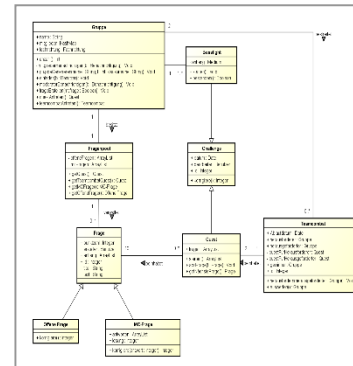
### Spezifikation

- Dokumentation sämtlicher Anforderungen
- Dokumentation sämtlicher Abläufe



### Entwurf

- Konzeption der Klassen und Methoden
- Konzeption des Datenmodells



### Software

- Programmierung der Software
- Test der Software
- Auslieferung

```
public Aufgabe(int typ, Object sender, Object adressat, Object anhang) {  
    ...  
    switch (typ) {  
        case 1:  
            senderGruppe = (Gruppe) sender;  
            empfangenderBenutzer = (Benutzer) adressat;  
            anhangGruppe = (Gruppe) anhang;  
            inhalt = "Du wurdest in eine Gruppe eingeladen";  
            inhalt = "Du wurdest in die Gruppe "+  
                + ((Gruppe) sender).getname() + " eingeladen";  
            break;  
        case 2:  
            senderBenutzer = (Benutzer) sender;  
            empfangenderBenutzer = (Benutzer) adressat;  
            anhangBossfight = (Bossfight) anhang;  
            inhalt = ((Benutzer) sender).getname() +  
                + " hat einen Bossfight bearbeitet. (bitte korrigiere und"  
                + " bewerte diesen Bossfight eintrah, damit "  
                + ((Benutzer) sender).getname() + " sein Ergebnis erhält.";  
            break;  
        case 4:  
            senderGruppe = (Gruppe) sender;  
            empfangenderBenutzer = (Benutzer) adressat;  
            anhangTeamcombat = (Teamcombat) anhang;  
            inhalt = "Herausforderung zum Teamcombat";  
            inhalt = "Die Gruppe "+  
                + ((Teamcombat) anhang).herausforderer.getname() + " hat "+" hat "+"  
                + " "+" zum Teamcombat herausgefordert. Du hast 3 Tage Zeit,"  
                + " um dein Quest zu bearbeiten";  
            break;  
    }  
    ...  
}
```

Um eine einfache Wartung und hohe Portabilität gewährleisten zu können, ist die Software modular nach dem Model-View-Controller Schema aufgebaut.



## Modulare Konzeption

View

Der Benutzer sieht und bedient ausschließlich die grafische Benutzeroberfläche der Plattform. Hier werden dem Benutzer interaktiv alle Inhalte und Aktionen dargeboten und intern an den Controller weitergeleitet.

Controller

Der Controller verarbeitet die gespeicherten Daten und sorgt für den korrekten Ablauf der verschiedenen Aktionen und dafür, dass dem Benutzer stets die richtige Seite mit den richtigen Inhalten angezeigt wird.

Model

Eine SQL-Datenbank stellt mit dem Datenmodell die unterste Schicht von käpsele.de dar. In ihr werden sämtliche Benutzerdaten sicher gespeichert und dem Controller zur Verfügung gestellt.

Danke für die Aufmerksamkeit!



**Käpsele.de**