Identifying Development-Metrics for Use in a Gamified Mobile Web Application to Support Software Development

Tim Wenzel, Thomas Franke, Rainer Wasinger, and Michael Körner

Presenter: Rainer Wasinger (rainer.wasinger@fh-zwickau.de)
Faculty of Physical Engineering/Computer Sciences
University of Applied Sciences, Zwickau 08056 Germany
Rainer works as a Professor for Mobile Informatics at the Westsächsische Hochschule Zwickau (WHZ). His interests are in the disciplines of HCI, Software Engineering, User-centred design, and mobile computing.

During his career, he has designed, developed, and tested many user-centred systems; led the development of such systems; and taught in the fields of software development and interaction design at multiple Universities across the globe.

Much of his work is focused on mobile and ubiquitous computing (including the Internet of Things).

He is strongly committed to life-long learning and is passionate in applying his skills in teaching and technology to create lasting benefit to the community.

Gamification in Software Development

1. Motivation
2. Gamification
   o General Information
   o An Example in Software Development
3. The Process of Designing the Mobile Web App
   o Development-Metrics
4. The Prototype
   o ... and it’s functions
   o ... in action
Success and failure in software projects: (1)

~ 19% Failure
~ 52% Partial Success
~ 29% Success

~ 71% of software projects do not lead to their planned result.

Motivation

Reasons for failure in software projects: (2)

• Lack of the end user’s input
• Incomplete specifications
• Not enough resources
• Insufficient planning
• Unrealistic expectations
• Psychological reasons
  – Mental pressure
  – Developer motivation

Gamification

Gamification: General Information

The use of game elements in non-game contexts

- Increases the motivation
- Improves the quality
  ... of the process
  ... of the result
- Improves the final result
Gamification: An Example in Software Development

*Stack Overflow*

- **Nice Question**: Question score of 10 or more
- **Scholar**: Ask a question and accept an answer
- **Civic Duty**: Vote 300 or more times
- **Refiner**: Edit and answer 50 questions (both actions within 12 hours)
- **Famous Question**: Question with 10,000 views
- **Fanatic**: Visit the site each day for 100 consecutive days
The Process of Designing the Mobile Web App

- Sort metrics
- Create concept
- Prototype

- Ionic
- Web
- GitLab
- REST
- SonarQube

APP

IonCE Web GTIab REST SnorQube
The Process: Development-Metrics

Relevancy of Software Metrics:
- Objective verifiability
- Automated processing
- Direct obtainment from version control systems
- No restrictions for actual work process

<table>
<thead>
<tr>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commits</td>
<td>Merges without complaints</td>
</tr>
<tr>
<td>Issues / Stories</td>
<td>Solved bugs</td>
</tr>
<tr>
<td>Pull Requests</td>
<td>Solved security issues</td>
</tr>
<tr>
<td>Milestones</td>
<td>Code smells</td>
</tr>
<tr>
<td>Lines of Code</td>
<td>Code duplications</td>
</tr>
<tr>
<td>Test Coverage</td>
<td>Complexity</td>
</tr>
</tbody>
</table>
The Prototype: Functions

Dashboard:
• User specific settings
  (Tokens, Challenges)
• Items
  (Amount, Icon, Name, Value)

Progress view:
• Overview of metrics with explanations
  (Sorted by the previously determined categories)
• Detailed statistics for each metric
  (Progress bar, rewards)

Competition view:
• List of users / colleagues
  (Amount of their items, Ready to compete)
• Competitions only available after confirmation
The Prototype: in Action

Video: https://wwwstud.fh-zwickau.de/raw17hkw/Video-GamificationInSoftwareDevelopment.m4v
The Prototype: in Action

Quantitative metrics
- Commits
- Issues
- Merge Requests
- Merges
- Milestones
- Lines of Code

Qualitative metrics
- Bugs
- Security
- Code Smells
- Code Duplications
- Cyclomatic Complexity
- Cognitive Complexity

247 of 1 Commits reached
247 of 30 Commits reached
247 of 50 Commits reached
247 of 100 Commits reached
247 of 150 Commits reached
247 of 200 Commits reached
247 of 250 Commits reached
247 of 300 Commits reached

26 of 1 Security Issues solved
26 of 3 Security Issues solved
26 of 5 Security Issues solved
26 of 10 Security Issues solved
26 of 15 Security Issues solved
26 of 20 Security Issues solved
26 of 25 Security Issues solved
26 of 30 Security Issues solved
Calculation of Competition Winner

Sum of a player’s rewards (item values):

\[
SumPlayer_X = \sum_{i=0}^{n} ItemValue(i)
\]

Overall sum of item values:

\[
SumRewards = SumPlayer_1 + SumPlayer_2
\]

Chance to win in percent:

\[
ChancePlayer_X = \frac{SumPlayer_X}{SumRewards}
\]
Calculation of Competition Winner (Code)

```
// Basic winning probability for player 1 (50%)
let chancePlayer1 = 0.5;

// Summation of item values for player 1
player1.items.forEach(item => chancePlayer1 += item.value);

// Basic winning probability for player 2 (50%)
let chancePlayer2 = 0.5;

// Summation of item values for player 2
player2.items.forEach(item => chancePlayer2 += item.value);

// Calculation of overall reward sum (player 1 and player 2)
const sum = chancePlayer1 + chancePlayer2;

// Calculation for chance of winning for each player
chancePlayer1 = chancePlayer1 / sum;
chancePlayer2 = chancePlayer2 / sum;

// Round chances
this.chance1 = Math.round(chancePlayer1 * 1000) / 10;
this.chance2 = Math.round(chancePlayer2 * 1000) / 10;

// Determination of winning player by random value and calculated chances
return Math.random() < chancePlayer1 ? this.user1 : this.user2;
```
Thank you for your attention!