Analysis of Project - SWOT

Strengths

1. Specific target group
2. Versatile Platform
3. The majority of the multi project groups are using scrum
4. Version control (revision control)
5. Previous work experience within the group
6. Good communication with the customer
7. Agile development method

Weaknesses

1. Limited time
2. Inconsistent use of development method among groups
3. Lack of experience with development method and target platform
4. No project owner (Scrum)
5. Clutter source code
6. Undocumented source code

Opportunities

1. The target platform is growing in popularity
2. The platform is getting more widespread throughout the world, and might therefore get cheaper
3. Our target group is present in the entire world, we could therefore expand international

Threats

1. Project management
2. Miscommunication between the multi project groups
3. Testing
4. Multi project integration
5. Our customer change opinion and leave the project

Technique Matching

To eliminate the weaknesses we can use the following tools:

1. Planning to better use the time available.
2. Meeting with other groups to agree on development method.
3. Pair programming to achieve better understanding of the target platform.
4. Refactoring to prevent code clutter and make code more readable.
5. White box testing can point out code which is not documented well enough.

Technique Research

Pair Programming

“All production code is created by two programmers at one computer; they rotate using the input device periodically. Paris may change frequently, for different tasks. The observer is doing a Real-time code review, and perhaps thinking more broadly than the coding developer typically does, considering tests and so forth.

Certainly, team productivity is not simply a function of the number of hands typing - it is more nuanced. The XP claim is that the combination of cross learning, the peer pressure of more disciplined practice observance and more hours actually programming than procrastinating, defect reduction due to real-time code review, and the stamina and insight to carry on when one programmer is stuck, all add up to an overall team improvement.”

source: Craig Larman, Agile & Iterative Development - A Manager’s Guide, p. 149

Refactoring

“Refactoring is the continual effort to simplify the fine-grained code and larger design elements, while still ensuring all tests pass. That is, cleaning the code and design, without changing functionality.

The goal is minimal, simple, comprehensive code. It is achieved by small change steps, verifying tests after each, and ideally the use of refactoring tools, now available in some IDEs.”

source: Craig Larman, Agile and Iterative Development - A Manager’s Guide. p. 149