Investigating Video Game Development Problems among Genres and Platforms

Fabian Fröding
University of Skövde
Skövde, Sweden
fabian.froding@gmail.com

***Abstract*—This paper presents a study on the relationship between video game development problems, genres and platforms. The aim of the study is to provide insight to potential development problems when developing games of a certain genre or on a certain platform. Using an explorative approach on a pre-existing dataset consisting of categorizations of different types of development problems, game genres and platforms, an analysis was conducted to investigate the distribution of the problems among genres and platforms, compensated with a statistical analysis for correlations and a comparison to related literature to compare other similar potential findings.**

The study yielded some useful indications of in which genres certain types of development problems are more likely to occur, and that some development problems differ slightly between mobile compared to PC and console, but did not show statistically significant results.

Keywords—video games, development problems, genres, platforms, software engineering.

I. Introduction

Video games are software applications in which players can control game objects and resources to achieve a goal [4], and contribute to a profitable entertainment industry [1], [2]. Because of the multidisciplinary fields involved in the development process of video games, the development process for video games differs from regular software application development and has made the industry known for its many management problems [1], [2], [3].

The fields involved in the game development process consists of sound, art, control systems, artificial intelligence (AI), and possibly human factors [2]. Game development also usually consists of many different types of teams and roles such as designers, programmers, composers and writers, to name a few, which also contributes to the difference between the development of games and traditional software. To further complicate things, the domain of computer games includes different types of game genres, platforms and modes.

Fatima *et al.* [3] attempts to integrate GSE (global software engineering) practices into the field of game development to mitigate the challenges of the game development processes by proposing a process called GDGSE (game development with global software engineering). The GDGSE is based on the three core game development phases; pre-production, production and post-production and looks at a case study to provide suggestions on how its proposed methodology can be used in the industry. Some aspects of the study presents suggestions related to areas such as design, documentation, prototyping, testing and marketing.

Politowski *et al.* [1] created a dataset of games with post-mortems categorized into three groupings and 20 sub-categorizations of different video game development problems: 1. *Production* (*Design*, *Documentation*, *Tools*, *Technical*, *Testing*, *Bugs*, *Prototyping*), 2. *Management* (*Unrealistic Scope*, *Feature Creep*, *Cuttings Features*, *Delays*, *Crunch-time*, *Communication*, *Team*, *Over Budget*, *Multiple Projects*, *Planning*, *Security*), and 3. *Business* (*Marketing*, *Monetization*).

The dataset also contains 12 different video game genres: *Action, Adventure, RPG, Simulation, Strategy, Puzzle, Sports, Platformer, Shooter, Racing, Rougelike,* and *Running,* and three types of platforms: *Console, PC,* and *Mobile*.

However, Politowski does not explore or analyze the types of development problems in relation to platform and genre. This paper aims to close that gap by visualizing and comparing the distribution of development problems by genre and platform.

II. Methodology

Politowski's dataset makes it possible to perform a data analysis to investigate the relationship between video game development problems, game genre and platform. To explore this topic, the following research questions were formulated:

- **RQ1:** What is the relationship between video game development problems and platforms?
- **RQ2:** What is the relationship between video game development problems and genres?

To answer the research questions, a visualization and analysis was conducted consisting of a comparison between the distribution of problems types, a statistical analysis and a comparison to previous similar work.

To visualize the number of problem-types within each genre and platform, software scripts were first developed that iterated the dataset to produce output yielding the number of occurrences for a given problem-type within a genre or platform-category. The source code for the used scripts are available on Github[1]. The raw data and cleaned data can also be found in the same Github repository[2]. The output of the scripts were then manually inserted into a Microsoft Office excel document.

Since the number of dataset-entries in different genres and platforms vary greatly within Politowski's dataset, comparing the distribution of the raw number of problems in different genres and platforms would yield skewed results. Instead, the number of problem types in each genre and platform was converted into a percentage value so that the distribution of problem types could be compared to each other on a normalized scale. This was done by creating a duplicate excel-sheet that referenced the raw-values from the original sheet and divided each value by the total number of problems in that genre or platform-category. The final values were then visualized by producing bar-charts in the same excel-document.

The steps involved in the data collection and data cleaning procedure can be seen in Figure 1.

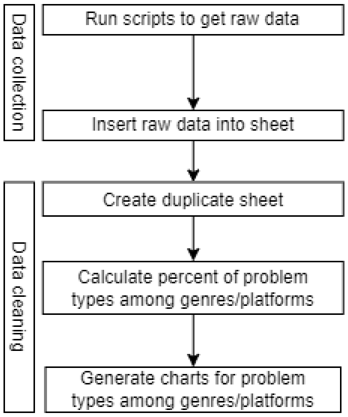
Data collection
Run scripts to get raw data
Insert raw data into sheet
Data cleaning
Create duplicate sheet
Calculate percent of problem types among genres/platforms
Generate charts for problem types among genres/platforms

Fig. 1: Steps performed to collect and clean the data.

The dataset only had 5 occurrences of problem-types within the *running*-genre category, 1 in each of the following problem-types; *tools, prototyping, delays, marketing, monetization*. The result for this genre would not yield a representative visualization of the distribution. If the *running*-genre was to be included, the visualization would show a 20% distribution in whichever chart it was present in. Because of this reason, the *running*-genre was excluded from the analysis. The *security*-problem type was only present in *simulation* and *shooter*-genres with a 1% occurrence rate and was excluded from the analysis since it would not yield any insightful results. Problem-types in genres that resulted in a 0% occurrence-rate were also excluded from the analysis.

Similarly to the occurrence rate of the *security*-problem type in genres, the same problem on *PC* and *Console* platforms also only had a 0.3% and 0.5% occurrence rates respectively, and was excluded from the analysis since it would not yield any insightful results.

All the results were placed on the same vertical scale in the produced charts to provide better reading consistency of the results. These charts were then compared to identify values that appeared more common than other values. The overall steps for the analysis process can be seen in Figure 2.

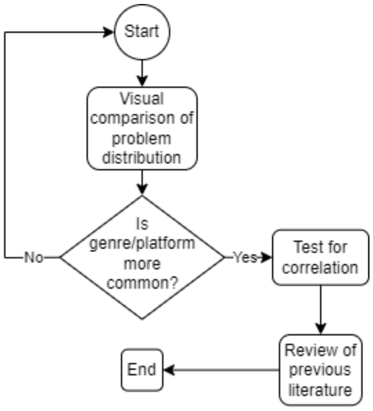
Start
Visual comparison of problem distribution
Is genre/platform more common?
No
Yes
Test for correlation
Review of previous literature
End

Fig. 2: Steps performed to analyze the data.

To gain an understanding of the association between the most noticeable occurrences of problem types and genres and platforms, a statistical correlation test was used on each of the more common values in the charts. To test the correlations, Cramer's V-test was used since all involved variables are categorical. To execute the test, each category of the chosen problems, genres and platform was converted into numerical values using scripts from the same Github repository mentioned earlier[3].

To compensate for potential insignificant results from the statistical analysis and to gain insight from related studies, a comparison to related literature was also used to look for findings that could be compared to the results from the visualization.

III. Results

1. *Results from the visualization*

Most of the produced charts show no drastic differences among the distribution of problem types among genres, but there are some noticeable differences.

Design problems seem to be most common in the *puzzle* genre (see Figure 3). *Documentation* problems seem more common in the *simulation* and *shooter* genres (see Figure 4). *Tool*-problems seem more common in the *roguelike* genre (see Figure 5). *Technical* problems seem more common in the *racing* genre (see Figure 6). *Testing* problems seem to be more common in the *strategy* genre (see Figure 7). *Bug* problems seem more common in the racing genre (see Figure 8). *Scope* problems seem more common in the *simulation* and *puzzle* genres (see Figure 10). *Feature creep* problems are more

[1] https://github.com/fabianfroding/gd-problems-analytics

[2] https://github.com/fabianfroding/gd-problems-analytics/tree/master/data

[3] https://github.com/fabianfroding/gd-problems-analytics

common in the *simulation* genre (see Figure 11). *Delay* problems seem more common in the *RPG* and *roguelike* genres (see Figure 13). *Communication* problems seem more common in the *shooter* genre (see Figure 15). *Multiple projects*-problems seem more common in the *adventure* genre (see Figure 18). *Marketing* problems seem more common in the *roguelike* genre (see Figure 20). *Monetization* problems seem more common in the *strategy* genre (see Figure 21).

As for the problem types among platforms, the problems seem more or less evenly distributed in all of the charts. However one of the few noticeable differences is that *documentation* problems seem more occurring in *PC* than on *console* and *mobile* platforms (see Figure 23). *Feature creep* and *budget* problems are less common on *mobile* (see Figure 30, 36). Monetization seems more common on mobile (see Figure 40). Overall, we can see that the types of development problems are mostly similar on *PC* and *console* platforms while problems on *mobile* are somewhat more unique.

2. Results from statistical analysis

The executed statistical tests yielded the following scores for Cramer's V: The association between the *puzzle* genre and *design* problems had a score of 0.11. The association between the *simulation* and *shooter* genres and *documentation* problems both had a score of 0.09. The association between the *roguelike* genre and *tool*-problems had a score of 0.09. The association between the *racing* genre and *technical* problems had a score of 0.11. The association between the *strategy* genre and *testing* problems had a score of 0.11. The association between the *racing* genre and *bug* problems had a score of 0.06. The association between the *simulation* genre and *scope* problems had a score of 0.09. The association between the *puzzle* genre and *scope* problems had a score of 0.07. The association between the *simulation* genre and *feature creep* problems had a score of 0.09. The association between the *RPG* genre and *delay* problems had a score of 0.08. The association between the *roguelike* genre and *delay* problems had a score of 0.04. The association between the *shooter* genre and *communication* problems had a score of 0.1. The association between the *adventure* genre and *multiple projects*-problems had a score of 0.07. The association between the *roguelike* genre and *marketing* problems had a score of 0.08. The association between the *strategy* genre and *monetization* problems had a score of 0.1. None of the correlation scores for the genres were significant ($p > 0.05$).

The association between *PC* and *documentation* problems had a score of 0.16. The association between *mobile* and *monetization* problems had a score of 0.0. None of the correlation scores for the platforms were significant ($p > 0.05$).

3. Findings in related literature.

Fatima *et al.* [3] had a case study on the performance of the Capcom game "Street Fighter V", which suffered from technical problems and bugs. According to the study, Capcom reported that the cause of these technical issues and bugs was a shortage of staff with expertise in technical experience and quality.

Petrillo *et al.* [5] mentions several reports from interviews with the developers of game companies that developed games suffering from different types of development problems. According to the reports, *Command and Conquer: Tiberian Sun [2000]*, *Vampire: The Masquerade [2000]*, and *The X-Files [1999]* suffered from *unrealistic scope* problems, and *Unreal Tournament [2000]* suffered from *feature creep* problems. The game *Wild 9* suffered from design problems. *Rangers: Lead the Way* suffered from delays. *Age of Empires II: The Age of Kings [2000]* suffered from technical issues.

IV. Discussion

Technical and bug problems are both most common in the racing genre. It is not surprising to see that these two problem types are most frequent in the same genre, since technical issues can be a cause for bugs. This could also indicate that racing games are more prone to technical issues and may therefore have more bugs. It is also important to mention that this was also the case for the Capcom game *"Street Fighter V"*, where both technical issues and bugs were the causes for its lacking launch-performance [3]. This further suggests that technical problems and bugs are related. Street Fighter would arguably be categorized as an action-game, which according to the data do indeed have a quite high occurrence of technical issues but very low occurrence of bugs.

According to the data in the visualization, the *platformer* genre is one of the genres least likely to have design problems, yet from interview-reports [5], platformer games such as *Wild 9* were especially prone to *design* problems.

Testing problems are most common in the *strategy* genre, which could suggest that strategy games require more robust testing or that they are more difficult to test than games in other genres. *Scope* problems are most frequent in the *simulation* and *puzzle* genres which could indicate that the development of games in these genres require clear goals of which features are required and which features are non-essential. *Scoping* problems refers to a game having too many features that are impossible to implement and feature creep refers to adding unplanned features [1]. It then makes sense that if simulation games are prone to scope problems they may also suffer from extra features.

Based on the report in Petrillo's work [5], the game *Command and Conquer: Tiberian Sun* suffered greatly from scope problems. This game is arguably a strategy game, which according to the visualization is the least likely genre to contain scope-problems (0.7%), (see Figure 10). Furthermore, the *Vampire: The Masquerade* and *X-Files* are both RPG games and also have high occurrences of scoping problems, but not more so than other genres according to the visualization (see Figure 10).

Subsequently, the *feature creep* problem is also most common in the *simulation* genre. Games such as the mentioned *Unreal Tournament* apparently suffered greatly from feature creep, and is an action game, which according to the visualization data is not more prone to this type of problem than other genres.

We can also see that the *feature creep* problem is least common on the *mobile* platform, and that *cutting features* is the most common on mobile. Cutting features refers to exclusion of planned features of the final product, which makes the occurrence of these problems on mobile very strange. A higher occurrence of feature creep would naturally yield a higher occurrence of cutting features, but for some reason the relationship seems to be polarized for mobile.

Yet even more inconsistencies are found when comparing the visualized data to individual interview-reports. For example, the *shooter* game *Rangers: Lead the Way* had high occurrences of delays, while according to the visualized data, shooter is the genre least likely to have delay problems (see Figure 13). Other reports from for example *Age of Empires II: Age of Kings*, a strategy game, suffered from technical issues but is also a genre that according to the visualized data should have few technical problems compared to other genres.

We can see that the findings for individual games in the review of related literature do not necessarily agree with the data in the visualization, which may or may not be a coincidence based on the quite limited amount of games found in the related literature.

Not surprisingly, all of the scores from the Cramer's V-tests yielded weak associations. This is most likely because of the size of the dataset. Even though the used dataset is quite big (927 entries of video game development problems) the representation of each of the 12 genres within the dataset were not big enough to provide statistically significant results. If each genre was evenly distributed in the dataset they would on average yield 77 entries per genre which apparently was not big enough to provide significant results. Therefore, no insights can be gained from the results of the correlation tests. With a bigger dataset, the tests might have produced different results.

Previous research on game development problems that takes genres and platforms into consideration is extremely scarce, perhaps even completely absent. Unfortunately, I did not manage to find any previous studies that investigated this area of research. However, research related to game development problems in general seem more common, and can be used to compare the findings in this paper even if it does not consider genres or platforms.

Both Petrillo *et al.* [5] and Holmer [10] found that the more common type of game development problems are related to scoping-issues and delays. According to Petrillo, the problem of delays is also shared with the traditional software development industry. Comparing these findings to the results in this study, we can see that delays are most common in the roguelike and RPG-genres (see Figure 13). But in general, delays do not seem to be more frequent than other types of development problems judging by their percentage values.

As a conclusion for the discussion, it is arguably more appropriate to rely on the findings in the data visualization than in the individual interview-reports since the reports are so few. With a more quantitative approach, the reports may have aligned more consistently with the problem types presented in the visualization.

V. Threats to validity

Politowski *et al.* [1] used 12 categories for video game genres. However, the report does not include where the genres originate from. Therefore, the validity of the results for the different genres may or may not be accurate depending on how "official" these categories for video game genres are. On the other hand, other research related to the categorization of video game genres use very similar categories to the ones Politowski presented [6], [7], [8], [9]. Regardless, having a unified categorization of game genres in research literature could benefit both this study and other studies that investigate genre-related areas.

The games mentioned in the findings from the related literature are not categorized by default. To find the correct genre for these games, I searched online to find which genre the game belonged to. The validity of the sources used to find this information may or may not be trustworthy, and is a threat to the validity of the discussion that involved these games.

VI. Related Work

Petrillo *et al.* [5] conducted a study that analyzed video game development problems from another perspective without focusing on platforms or genres. The study consisted of analyzing traditional software industry problems and comparing them to the game industry, analyzing game industry problems from specific literature, a qualitative post-mortem analysis from game industry professionals, and finally comparing the results obtained from the game industry to the traditional industry to find similarities and differences. The study found that both industries suffer mostly from unrealistic scope-related problems, where naive optimism is a contributing factor. Compared to the findings in this study, they do not seem to agree with Petrillo's findings, since scope-related problems are not more common than other problems based on the charts.

VII. Conclusion

This study investigated the relationship between video-game development problems, genres and platforms. The study found that some of the genres seem to be more prone to certain types of development problems, and that some development problems seem to be slightly different on mobile platforms compared to PC and console platforms.

Even though the correlation between development problems and genres and platforms showed weak associations from statistical tests, this is most likely because of the small size of the representation of the genres in the dataset, and does not significantly impact the conclusions based on the results in the produced charts and literature review.

The results are beneficial for video-game developers to become more aware of certain development problems when developing games of a certain genre or on a certain platform.

The methodology used in this paper did encounter any major obstacles and had few threats to its validity. Therefore, the methodology should be applicable to studies wanting to investigate similar issues.

Future work could apply the methodology proposed in this paper to investigate the relationship between video-game development problems and other factors, such as game modes (single-player, online etc). Such a study could yield interesting results showing what kind of problems are more prone to occur when developing games with different modes.

Furthermore, an extension of Politowski's dataset that contained more entries would yield more accurate statistical results, especially an extension containing more entries for the lacking *running*-genre.

References

[1] C. Politowski, F. Petrillo, G. C. Ullmann, J. A. Werly, Y-G. Guéhéneuc, "Dataset for Video Game Development Problems", 2020.

[2] S. Aleem, L. F. Capretz, F. Ahmed, "Game development software engineering process life cycle: a systematic review", Journal of Software Engineering Research 2016 4:6, 2016.

[3] A. Fatima, U. Qamar, "GDGSE: Game Development with Global Software Engineering", 2018 IEEE Games, Entertainment, Media Conference, August 2018.

[4] K. Salen, E. Zimmerman, "Rules of Play: Game design fundamentals", MIT Express, 2004.

[5] F. Petrillo, M. Pimenta, F. Trindade, C. Dietrich, "What went wrong? A survey of Problems in Game development", Computers in Entertainment, February 2009.

[6] J. H. Lee, N. Karlova, R. I. Clarke, K. Thornton, A Parti, "Facet Analysis of Video Game Genres", iConference 2014 Proceedings, 2014.

[7] S. M. Doherty, S. S. Davidson, E. M. Palmer, C. M. Frederick, "Recategorization of Video Game Genres", Proceedings of the Human Factors and Ergonomics Society 2018 Annual Meetings, September 2018.

[8] D. Johnson, L. E. Nacke, P. Wyeth, "All about that Base: Differing Player Experiences in Video Game Genres and the Unique Case of MOBA Games", Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015.

[9] T. Apperlay, "Genre and game studies: Toward a critical approach to video game genres", Simulation & Gaming, March 2006.

[10] E. Folmer, "Component based game development - A solution to escalating costs and expanding deadlines?", July 2007.

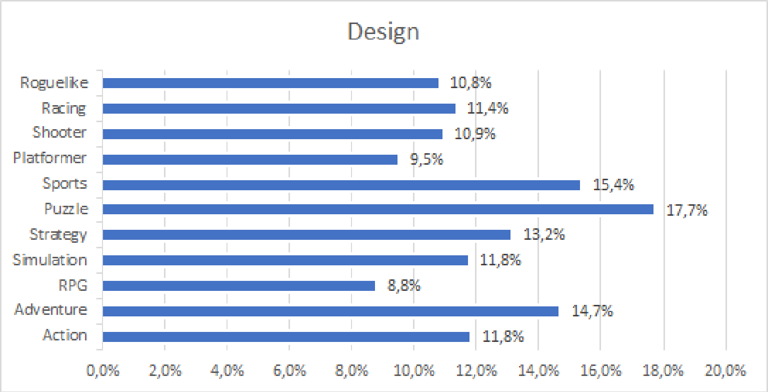

Fig. 3: Design-problems among genres.

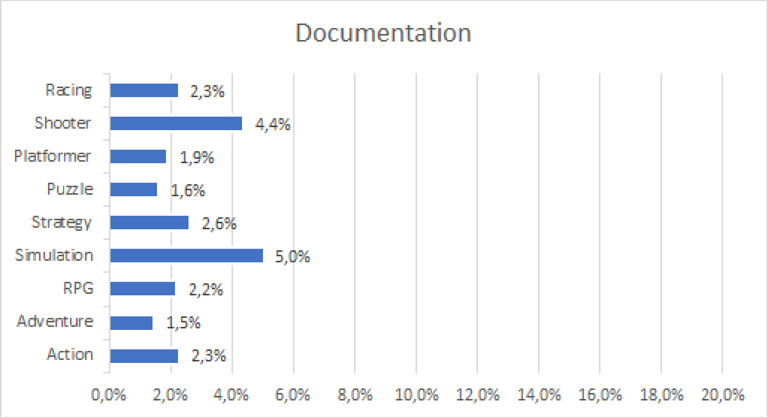

Fig. 4: Documentation-problems among genres.

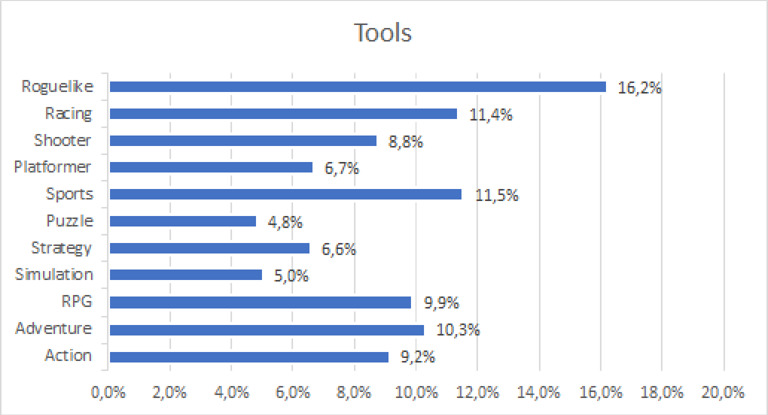

Fig. 5: Tool-problems among genres.

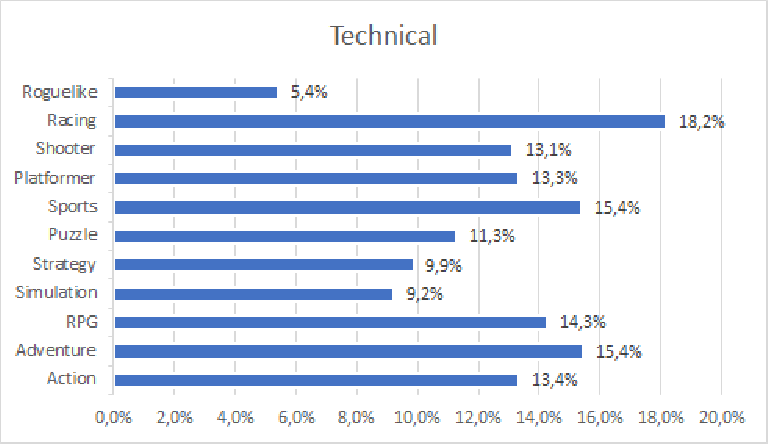

Fig. 6: Technical-problems among genres.

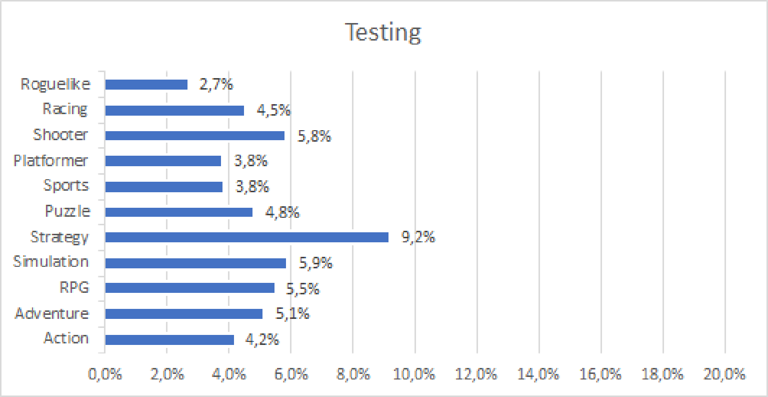

Fig. 7: Testing-problems among genres.

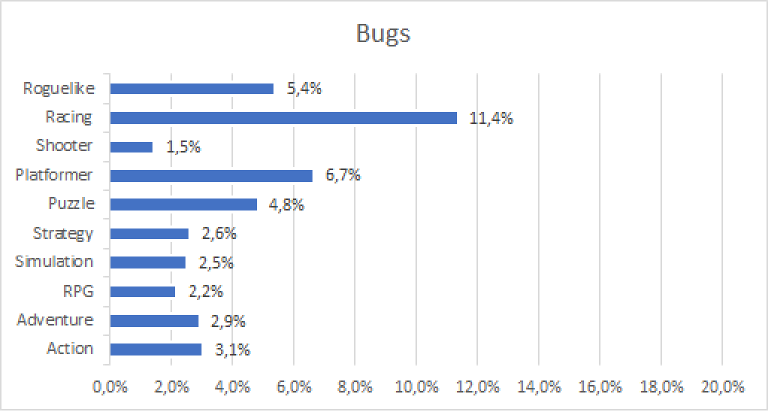

Fig. 8: Bug-problems among genres.

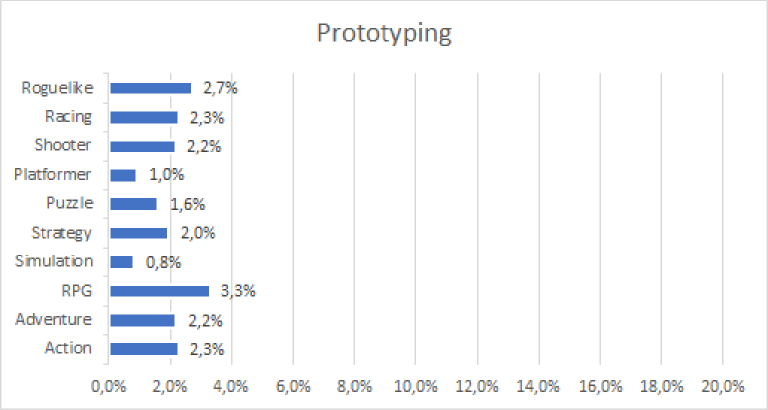

Fig. 9: Prototyping-problems among genres.

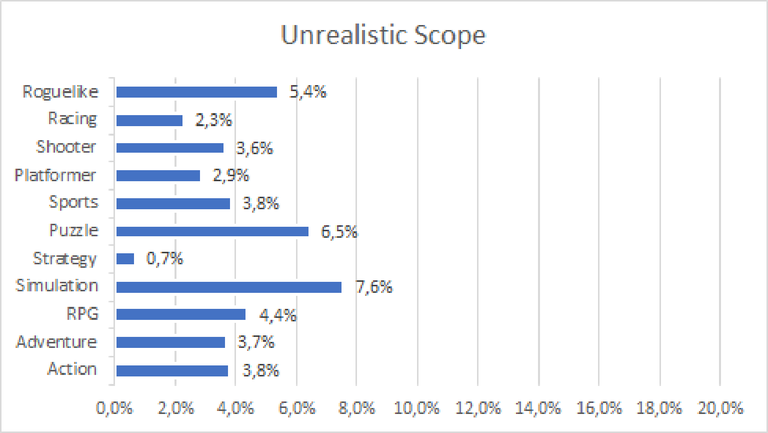

Fig. 10: Unrealistic Scope-problems among genres.

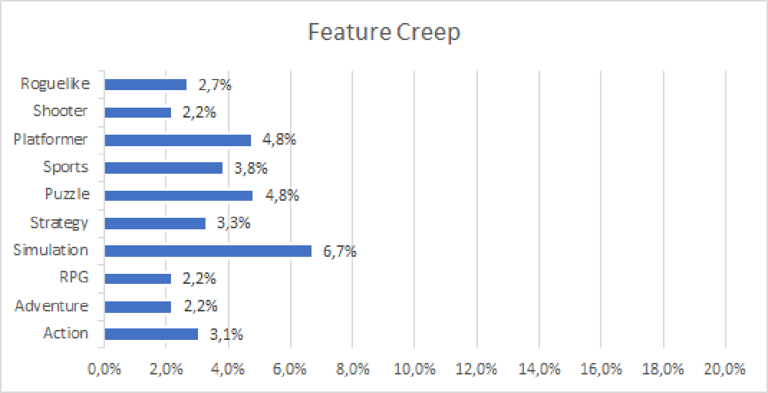

Fig. 11: Feature Creep-problems among genres.

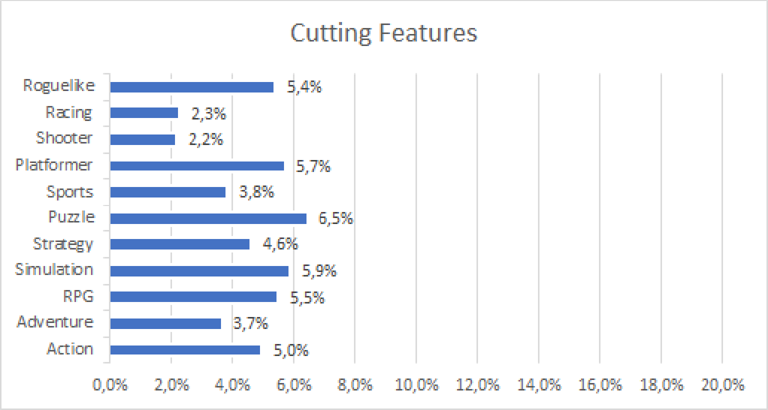

Fig. 12: Cutting Features-problems among genres.

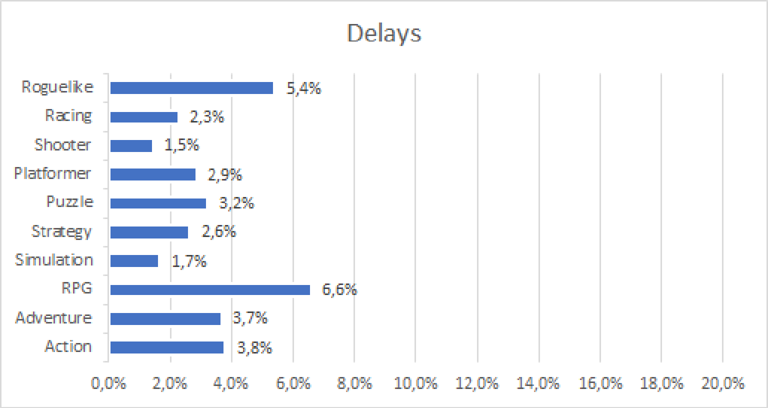

Fig. 13: Delay-problems among genres.

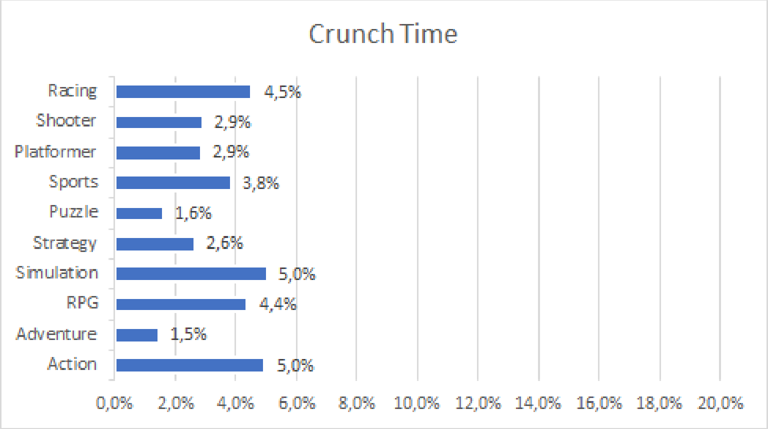

Fig. 14: Crunch Time-problems among genres.

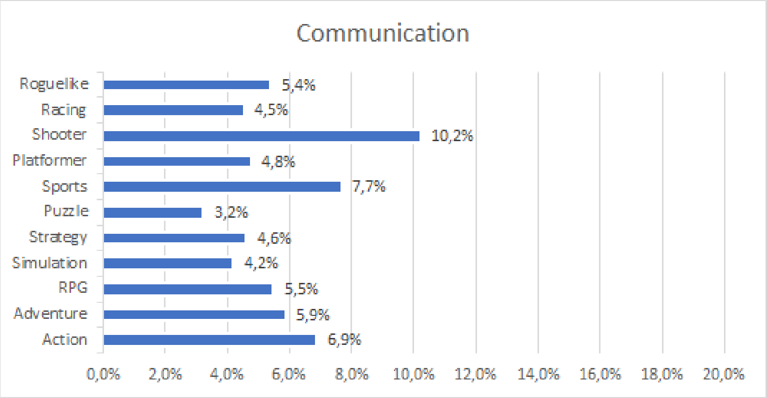

Fig. 15: Communication-problems among genres.

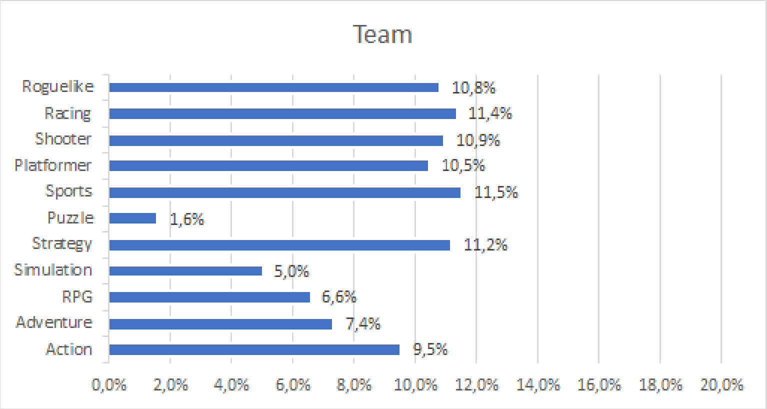

Fig. 16: Team-problems among genres.

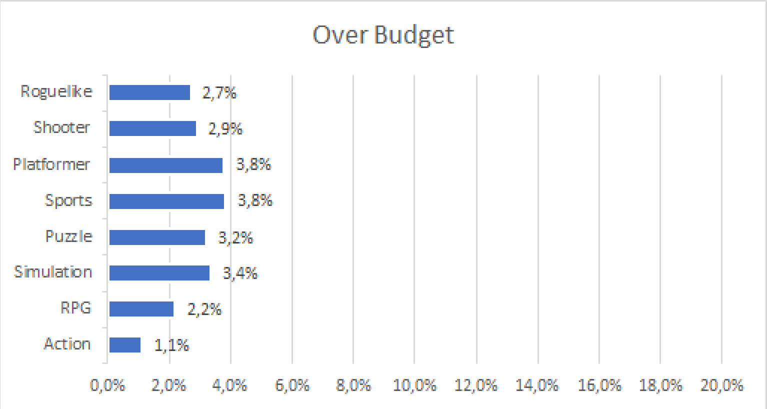

Fig. 17: Over Budget-problems among genres.

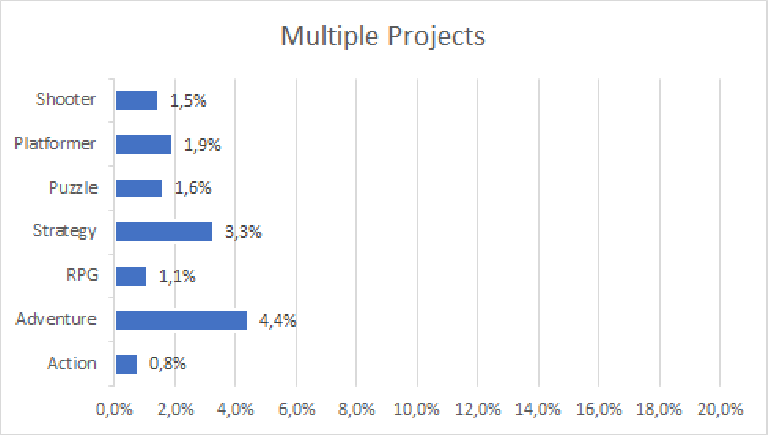

Fig. 18: Multiple Projects-problems among genres.

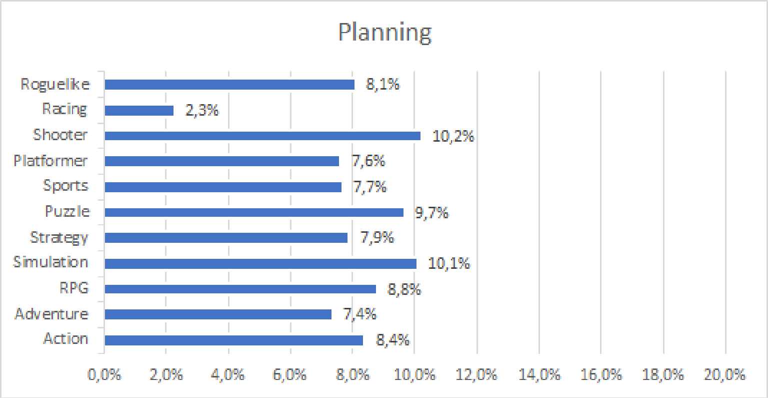

Fig. 19: Planning-problems among genres.

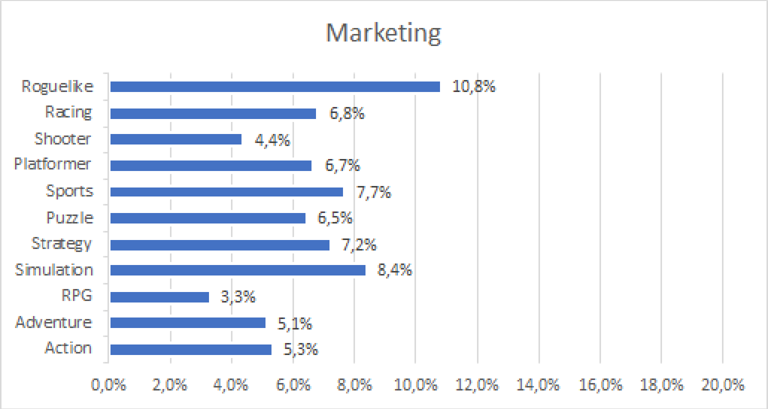

Fig. 20: Marketing-problems among genres.

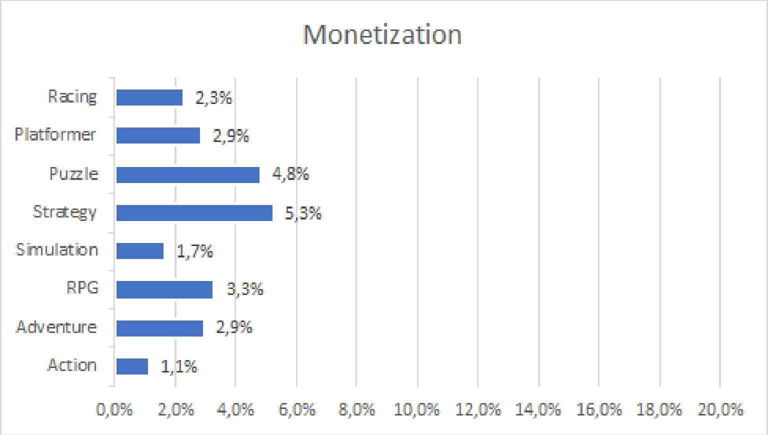

Fig. 21: Monetization-problems among genres.

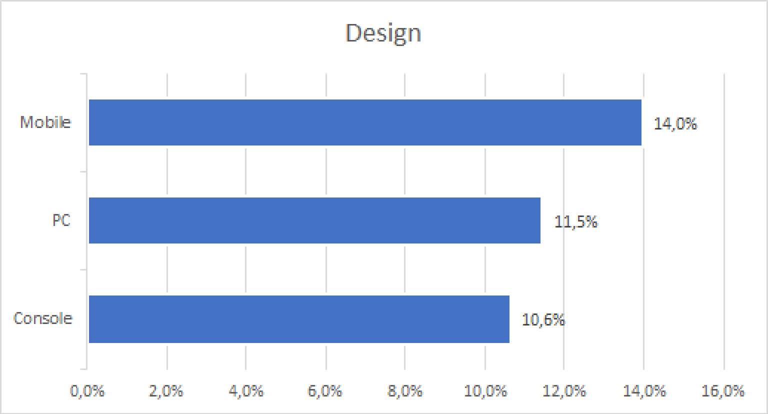

Fig. 22: Design-problems among platforms.

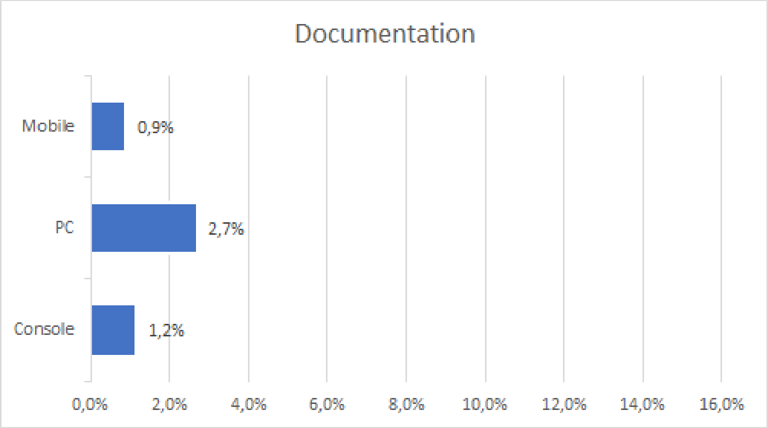

Fig. 23: Documentation-problems among platforms.

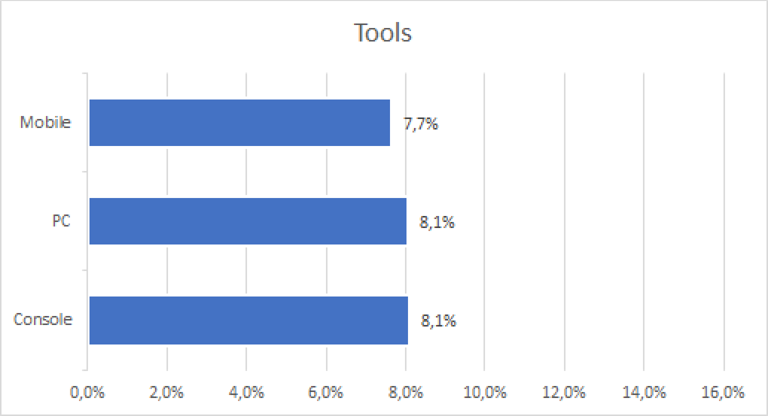

Fig. 24: Tool-problems among platforms.

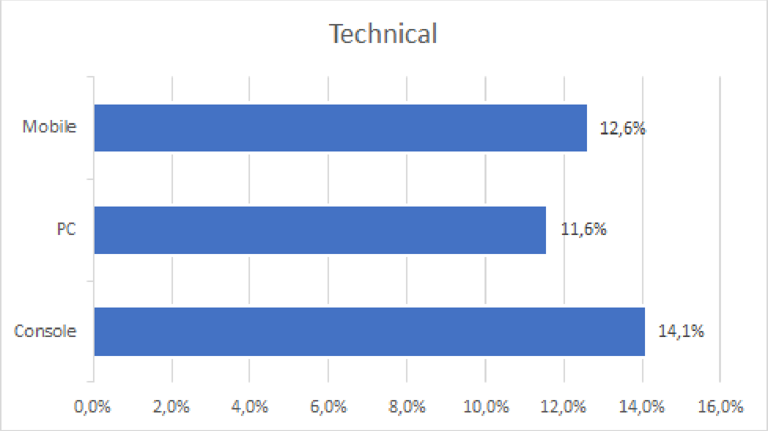

Fig. 25: Technical-problems among platforms.

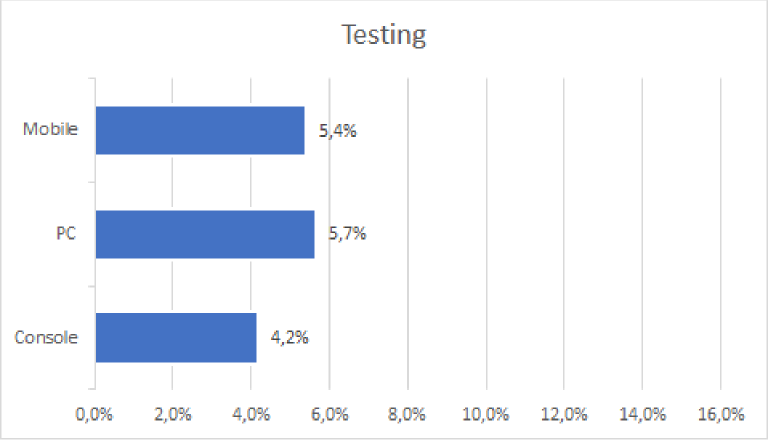

Fig. 26: Testing-problems among platforms.

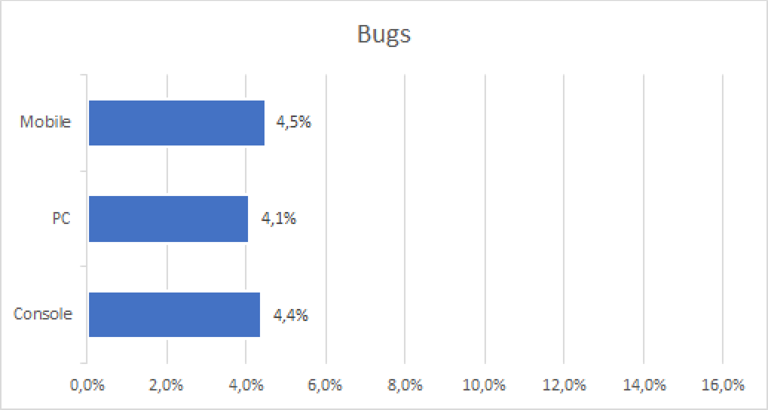

Fig. 27: Bug-problems among platforms.

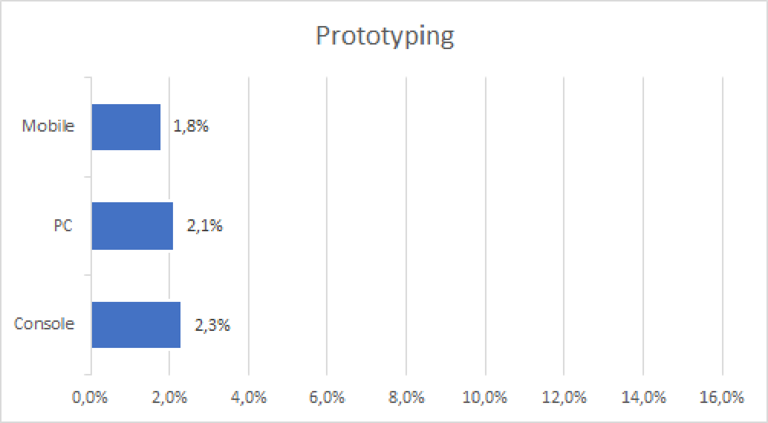

Fig. 28: Prototyping-problems among platforms.

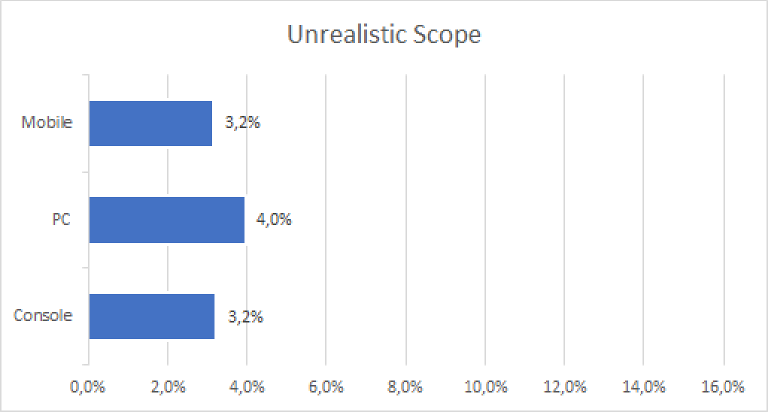

Fig. 29: Unrealistic Scope-problems among platforms.

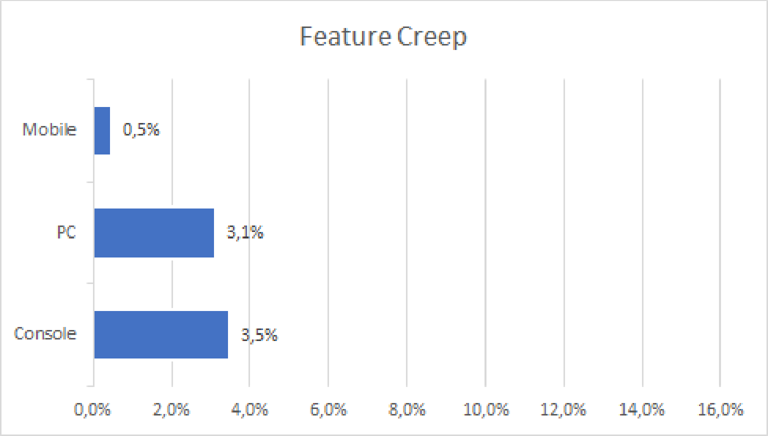

Fig. 30: Feature Creep-problems among platforms.

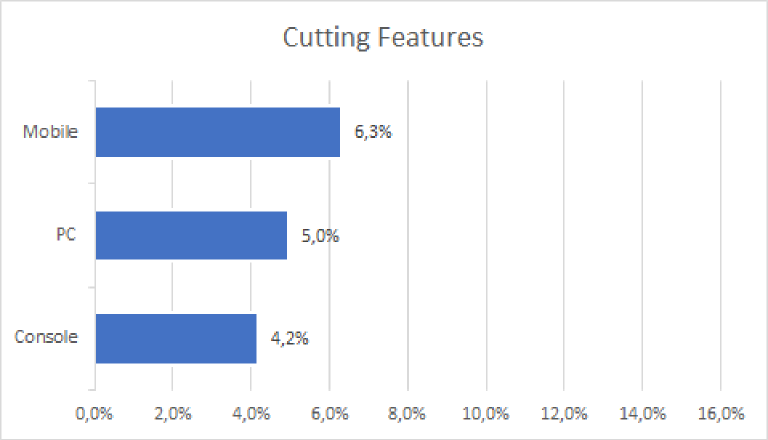

Fig. 31: Cutting Features-problems among platforms.

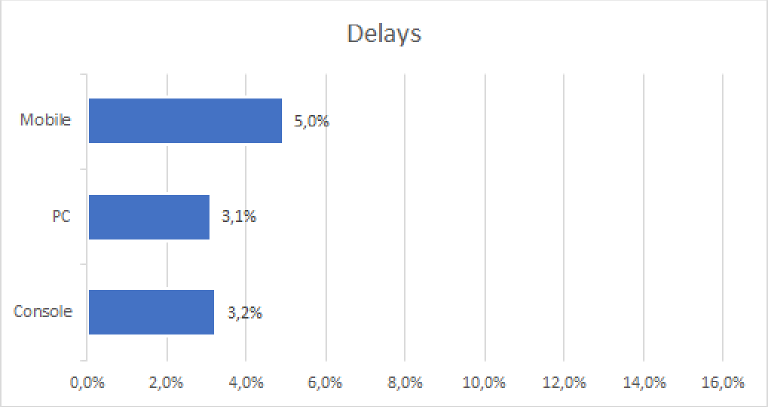

Fig. 32: Delay-problems among platforms.

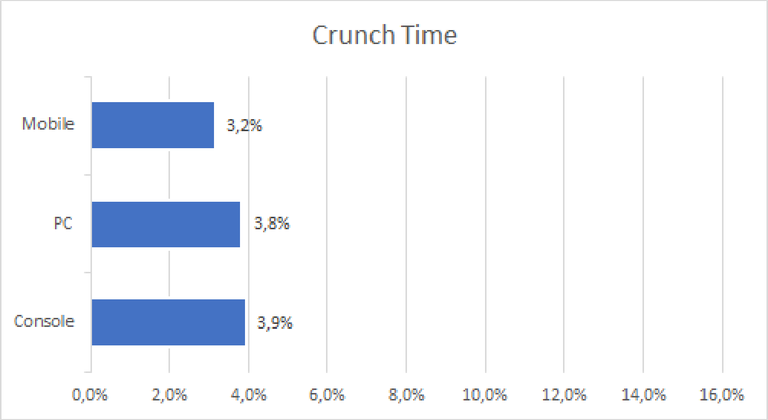

Fig. 33: Crunch Time-problems among platforms.

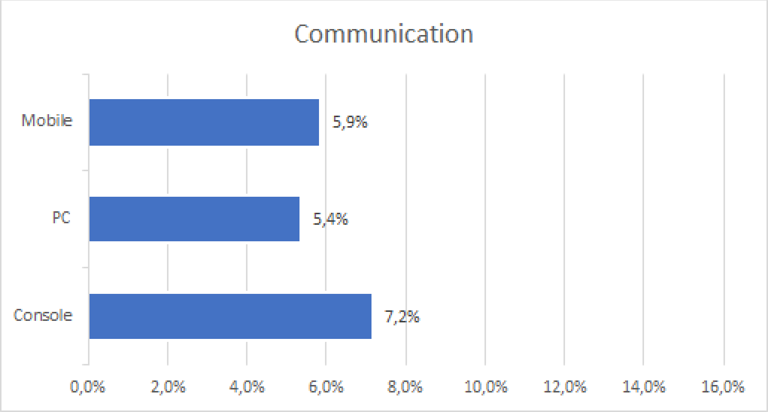

Fig. 34: Communication-problems among platforms.

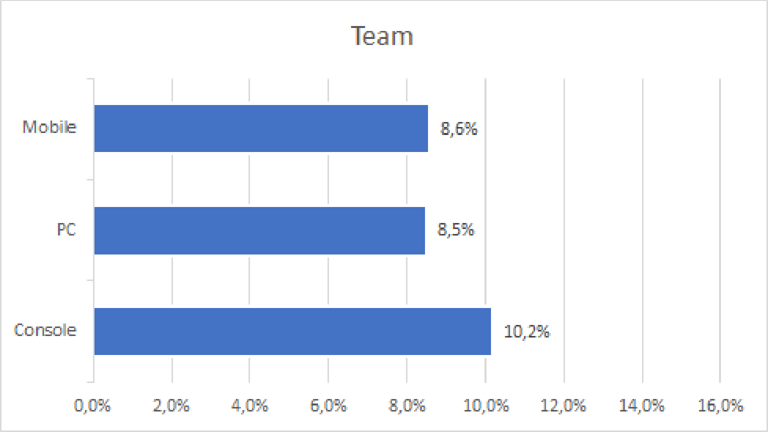

Fig. 35: Team-problems among platforms.

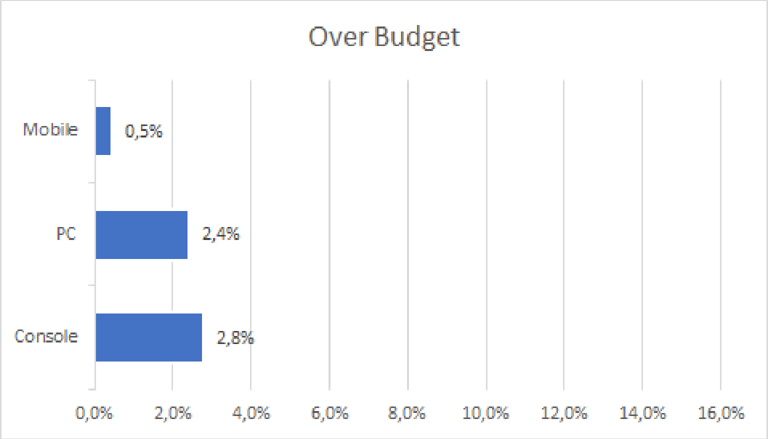

Fig. 36: Over Budget-problems among platforms.

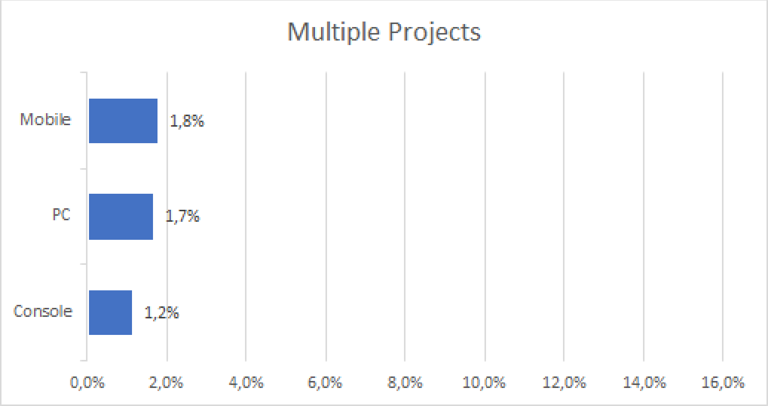

Fig. 37: Multiple Projects-problems among platforms.

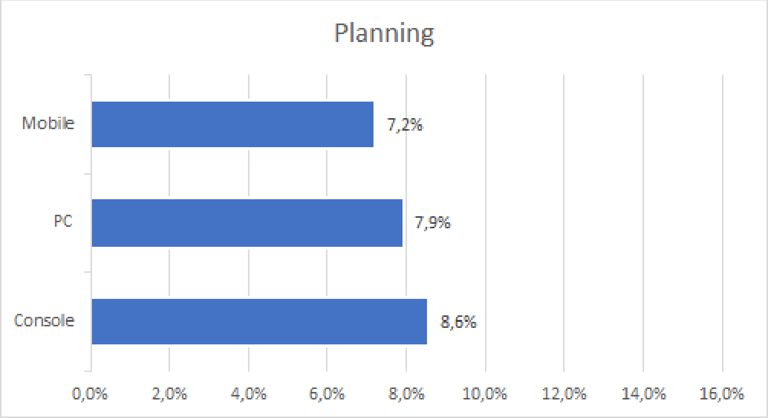

Fig. 38: Planning-problems among platforms.

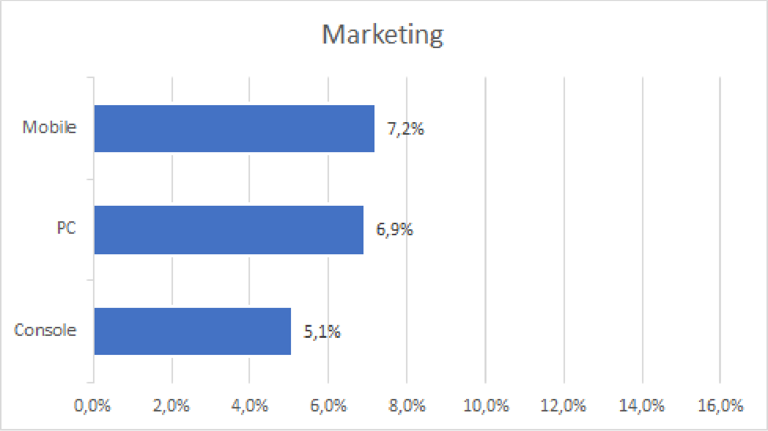

Fig. 39: Marketing-problems among platforms.

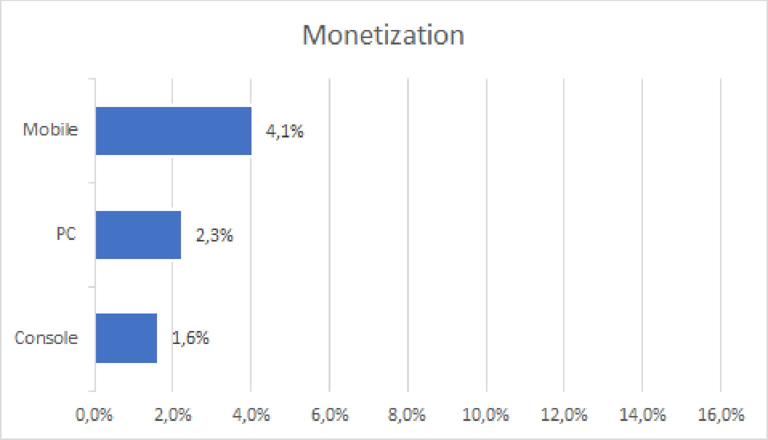

Fig. 40: Monetization-problems among platforms.