Continuous Experimentation in Mobile Game Development

Sezin Yaman, Tommi Mikkonen
Department of Computer Science
University of Helsinki, Helsinki, Finland
{sezin.yaman, tommi.mikkonen}@helsinki.fi

Riku Suomela
Next Games Ltd.
Helsinki, Finland
riku.suomela@nextgames.com

***Abstract*—Software companies need capabilities to evaluate the user value and the success of their products. This is especially crucial for highly competitive markets, such as the mobile game industry, where thousands of new games are introduced every month. Game companies often run continuous experiments as an integrated part of the overall development process. This paper presents a game company's journey on experimentation, and describes how the experiments are used at different stages of the development cycle to produce reliable, meaningful data for developers as well as how to balance between different data collection methods. Our study indicates that experiments are important in all stages of the development in different forms. Early stages in the development experiments can be run with proxy users due to lack of real users, whereas later in the development Key Performance Indicator (KPI) metrics play the most important role in experiments. Establishing concrete goals for the experiments, balancing between qualitative and quantitative data collection, experimentation throughout the development process with the guidance of an efficient leadership appears to be the key to success.**

Keywords: Continuous experimentation, experiment-driven software development, product management, customer development, customer involvement, organisational transition, agile software development.

I. Introduction

Continuous experimentation is a software development approach, where research and development activities are guided by iteratively conducting experiments and collecting user feedback [1], [2], [3]. Customers and users are involved in shaping the software by being subjects to experiments via interacting with experiment artefacts such as new products or updates. Therefore, product value is tested by directly evaluating user behaviour rather than relying on secondary sources or assumptions. This leads to a transformation from agile software development to continuous business experiments and business model evaluations [4], using data that is available at any point in development as the basis.

Many kinds of applications are being cultivated with experiments, and in particular those that do not involve installable software, such as Facebook [5]. In a context where an established, large enough set of users exists to produce data, and the system is provided as a service, running continuous experiments appears to be a common practise. However, for new applications that have no or limited user base, the situation is more complex. In addition, in particular in the context of mobile devices, apps are typically installed in user devices, and therefore running certain types of experiments becomes vastly more demanding.

In this paper, we consider experimentation in the context of mobile games, which is a very competitive domain [6]. A typical development life cycle of a mobile game is often more than a year long. The opening of the mobile application stores has made it easy for any developer to publish games and applications to global audiences. The number of mobile games being created is still growing and each month, more than 25,000 new games are to be released by other developers [7]. However, new statistics also reveal that an average mobile app loses 77% of its daily active users within the first 3 days of the install [8]. The focus of the paper is on how experimentation helps to validate product and feature assumptions, which are critical to the success of mobile games. Furthermore, experimentation is also a tool to ensure user satisfaction.

The rest of this paper is structured as follows. Section II presents the background and motivation to this study. Section III describes the case company's game development journey, including the context, and the data collection and analysis methods. Next, Section IV details experimentation practises at each stage of mobile game development at the company, while in Section V the findings of the paper are discussed, Section VI concludes the paper.

II. Motivation and Background

The mobile gaming market is expanding rapidly, by 2019 expected to take 45% of total worldwide gaming revenue amongst all legacy gaming forms such as console and PC games [9]. The mobile game market is truly competitive itself, with vastly more games published than is reasonable to review in a systematic competitor analysis or foresight, which are common techniques to position new products to existing markets. Therefore, companies need a development capability that maximises their chance to succeed especially when the game is out for the first time. This calls for a development approach that is optimized for getting reliable feedback early on, rather than trusting that the specified game concept is correct to begin with.

Petrillo et al. reveal in their survey study that game development problems do not usually come from engineering problems but from management and process [10]. Implementing agile methods in the development allow for faster game exploration through the use of techniques such as iterations and prototypes [11], [12], especially critical where innovation and speed to market are vital [6]. Aleem et al. in their systematic review emphasize that traditional game development approaches less frequently carry out empirical methods such as experiments [13]. They explain that this can be due to lack of experience in experimentation. Besides, even though game development might resemble general software development, it has its own differences and priorities [14], [15].

Operating a successful mobile game as a service requires multiple different entities to work together. In a typical game company analytics, technology, live operations, customer support, marketing and user acquisition teams support the game development. How these functions work together is especially important when the game enters the market for the first time. Operating a live product needs daily analysis on its performance.

In this section we aim to set the terminology and concepts by first providing an insight into characteristics of the mobile game development activities in general, and then discussing different forms of data that can be collected to support the development.

A. Typical Stages in Mobile Game Development

There are multiple life cycles propositions for game development, such as the sequence of phases – initiation, preproduction, production, testing, beta, and release [16]. Also adapted from McAllister and White's proposal [17], the following stages were identified in mobile game development that was subject to this study:

- **Concepting**: A stage when the developers are investigating different ideas and concepts for a game. This is largely a creative process, and the outcomes are the basis for starting systematic development for a game in prototyping. This stage is excluded from the study.
- **Ideation/Prototyping**: A stage when the developers are experimenting with different ideas for a game, and create prototypes that communicate the ideas to other stakeholders.
- **Preproduction**: A sequel to prototyping, the preproduction stage refines the prototype towards a go/no-go decision regarding the game. Experiments on product features, typically executed independently and in parallel, play a crucial role before going into the next stage, production.
- **Production**: Integration of separately tested features and the full implementation of the game, including also various other activities than just the development.
- **Market test**: The completed game is made available in a test area or made available for a test audience.
- **Live Service**: The game is in use and has active players. Often, also updates to the game are made to keep the players happy. It is very important to continuously measure how is the game engaging with the users.
- **End of life**: The game is no longer supported. This stage does not require any actions by the developers any more. Hence it has been excluded from the study.

While activities in different stages are partly overlapping – for instance, code is composed in all stages, at least to some extent – all stages are gated, with different requirements associated with each gate. Therefore, different documents, data, and other material are required to move from one stage to the next.

The development time of the entire product development could be anything from one year upwards in game development. Especially prototyping and preproduction are the key parts in defining the product and it is important to move fast in the early stages of development. Aleem et al. emphasize that once the game is fully implemented, it is very expensive to make a change, such as fixing a problem, and this will effect whole project schedule [13]. Starting from the times of traditional software engineering, research has been indicating that cost of change or fix significantly increases while the software product matures during the development [18].

While a typical live service can be counted in years for a successful game, the prototyping and preproduction stages last typically a few months, making it essential to move forward in the development at a rapid pace while ensuring that best game development strategy is followed. However, this is not done at the expense of data collection and analysis. In fact, since there is only a little room for changing the development direction completely, both qualitative and quantitative data is collected as a key part of the development process in every stage.

B. Classifying data

Both qualitative and quantitative data is valuable for a game development project. *Qualitative data* refers to data that cannot be described with numerical values. Typical sources of qualitative data in the case of the mobile game domain include both the internal sources (e.g., development team) and external sources (e.g., customers under non-disclosure agreements) on the game and its development. In contrast, *quantitative data* consists of data that can be directly represented in numbers. Unfortunately, when a development project is in its early stages, it is difficult to find relevant sources of quantitative data that would enable informed decision making.

At early stages of development it is possible to recruit potential end users as proxy users to experiment with the product, yet data gathered by the experiments, e.g., through prototypes, tells us quite little as the product is still evolving rather fast. That is why it is more important to focus on the desirability of the product, i.e. what the parts of the product that best resonate with potential customers are. This data can be achieved with a relatively small population and qualitative methods.

In particular, when dealing with qualitative data, cognitive biases form a major risk factor. However, qualitative data is useful to get potential users' thoughts and opinions on the things that they are feeling and perceiving, e.g., the design aesthetics. As for quantitative data, once the game is in active

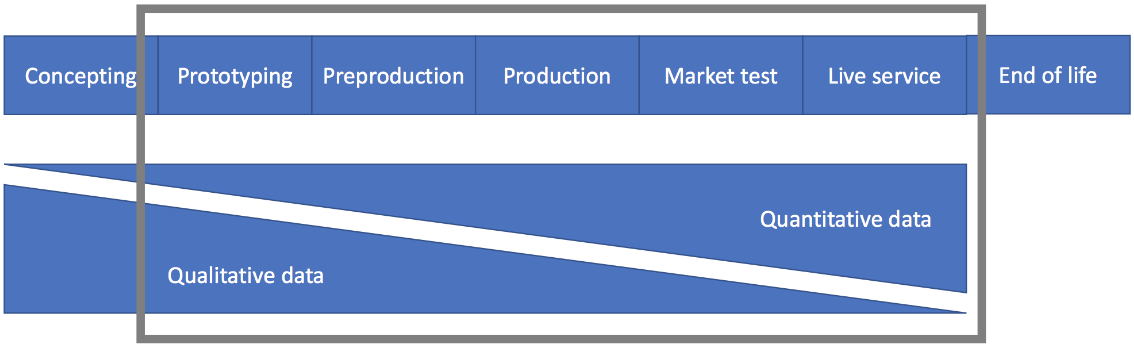

Fig. 1. Stages of mobile game development process along with typical form of data collection at Next Games.

use, the amount of data becomes so vast that it might be difficult to derive conclusions that would be meaningful for development especially if the metrics are not pre-defined.

In many cases qualitative and quantitative data is used in combination, depending on the stage of the game and its development. Balancing between methods that rely on data require different considerations for validity.

III. A Typical Development Journey at Next Games

In this section, we detail on a case study conducted with a game development company called Next Games. In particular, we describe Next Games' typical journey in mobile game development based on the series of interviews conducted with the company representatives. Due to confidentiality reasons, we are unable to disclose full details of the development and experimentation taking place, but rather snippets of information on the development life cycle. It should be also noted that the development life cycle alters for different games.

The case company is a game publisher and developer that works with license holders to build mobile games, covering all key functions needed for game development. Two games have been launched so far; *The Compass Point: West* [19] and *The Walking Dead: No Man's Land* [20].

An overview of the development process at the company is illustrated in Figure 1, with the stages of the process covered by this paper highlighted as well as type of data that is typically available to support the development. The typical team size is dependent on the development stage. In the very first stages there can be only a few people, but the further the game progresses the more people are involved. When the games go live, the team size with all supporting teams, e.g., analytics, is measured in tens of people. In the following subsections, we will discuss the goals and the key activities in each stage as well as status of experiments.

A. *Prototyping and Preproduction*

In prototyping and preproduction, the main goal in each iteration round is to validate one or more unknown aspects of the game. Aspects may be a technical issue, a function associated with users, or almost any other topic that is not understood yet. Key issues are listed in the following:

- **Focus**. To run an experiment, the first step is to define what to measure, and measure only that. Since the game is nowhere near complete, everything else should be minimalistic and hence unsatisfactory in many ways.
- **Keep the pace**. Since the desired development time of a prototype iteration is only 2-6 weeks, it is important to conduct the experiments rapidly. The goal is to learn as quickly as possible, and therefore figuring out whether an experiment validates or falsifies the assumptions are equally valuable. Validation means positive outcome that the development can continue, whereas falsification means negative outcome that something must be refined, reconsidered or be left out from the final product.
- **Consider the scope**. The line between prototyping and preproduction is partly a line drawn in sand, as the tools and techniques are largely similar. Probably the best differentiator is that iterations are partial in the prototyping stage, whereas the complete product is experimented in preproduction.
- **Run things in parallel**. Experiments that complement each other can be run in parallel. This helps in creating new iterations rapidly, including more experiments. It is important to determine whether the hypotheses being tested are dependent or independent of some other hypotheses in the pipeline, as this determines if they can be tested in parallel. For instance, if the output of an experiment is an input for another experiment, they might not be run at the same time.

B. *Production*

After preproduction, most of the final product features are implemented and tested separately. Unfortunately, once these features are integrated they more often do not work well together than they do. For this reason, it is difficult to determine if the final product is good, or if it only has good but separate features.

Defining and implementing a minimal set of features is an important enabler to get quantitative data out of the game.

Therefore, in the light of feedback and experiment results from previous stages, the best option is selected for implementation. The fact that still no real users exist remains to be accepted. On the other hand, alternative methods such as experimenting with proxy users are crucial to evaluate the progress.

Fast enough speed is important to keep at this stage. Once the implementation is complete, it is time for the next stage, market test to be finally evaluated with the real users. An important part of the implementation to instrument everything in the product for measurement.

C. Market test

The market test is the true first blood for a new game, as the game is shipped to the large amount of users for the first time. There is the main goal to determine – is the product going to be profitable or not?

There are several approaches to market testing. For instance, testing the market may be purely technical. Answering the question, *does the system work as intended?*, results in an extended quality assurance period. Tests can be open or closed depending on the purpose. In general, a closed beta test, i.e., invite-only users, allows more freedom, as the product is in beta status and not available for the general public yet. Open testing, i.e., open in a distribution channel, offers less freedom at this stage because everything should work out-of-the-box.

The question regarding the lack of quantitative data is no longer relevant at this stage, as there is so much data pouring. Rather, the problem is what to do with all that data and how to ensure that a statistically relevant analysis is conducted. One way to organize the data is to look at it through *key performance indicators* (KPIs). Some of the relevant KPI indicators are presented in Table I. The KPIs mentioned here relate to free to play game services that are initially free and some customers will pay for the services at a later stage. It should be noted that there might be many other KPIs that are relevant for a game product, the table only indicates the most relevant KPI from each category.

From these KPIs, the health of the product can be measured with a simple formula: if LTV >(eCPI + development and other costs) then the product makes a profit.

During the market test, several updates to the game are typically made to improve the KPIs. The development should stay in this phase until all KPIs reach the target zone. In the market test, typically the desirability is the first focus. This means that all the effort is put to maximizing the retention at different stages. Only when players like to play and stay in the game long enough, the development effort can be moved to optimise other values.

D. Live service

Once a game passes the market test, the profit-making stage, online services, comes in. Passing the market test means satisfying KPIs, which means that the players stay with the game long enough to bring sufficient revenue for profitable operations. Also, new user acquisition is cost efficient in relation to revenue generation, so it pays off to market the game. Furthermore, the estimated lifetime value of the game is healthy (and possibly growing) so that it makes sense to continue with the game.

At this point there is the major question: *What to do to keep the game profitable?* New features, more in-app purchases and other technical constructions can be introduced as the concrete mechanism, but usually the key question is how to keep the players coming back. Luckily, if there is a chance for true revenues, there is plenty of data and data collection opportunities to experiment with the quantitative data collected from real players.

The lifetime of a successful live service is counted in years, so this stage is where development heavily takes place. In contrast, the development prior to being in market is a short period, although it does involve the team's entire knowledge and requires capabilities to test product assumptions quickly. As going into live service is the first time where reliable data is available en masse, the goal is to enter the stage as soon as possible. Yet, sometimes it makes more sense to abandon a product and start building a new one, than work on a product for years if KPIs are not satisfying.

IV. Experiment Design at Different Stages

A company representative explains the role of experimentation in the company: *"Experimentation takes place throughout the organization (not only in one unit), and it is done at all the stages. We run experiments and decide whether to continue with the next sprint or not."* However, different stages of the development process require a different set of tools for the experiments. In this section we dwell on the experimentation design details studied at the case company along with examples.

The earlier in the development cycle the product is, the less quantitative data can be collected (see Section II-B) and the cycle of development can be quite fast. In general the most relevant metric to measure is *retention*, i.e., how long the players engage with the product. However, since the early product development stages mean there is no long-term content in the product, the retention should be measured in relation to how much of the product is developed. For instance, if a 15-minute set of the game is developed, it makes a lot of sense to test whether players like this 15-minute game or not before moving further. Retention at the end is ideally measured in months and years, but the road there starts from measuring the first minute. The focus will gradually move from retention to monetisation later on when there is a healthy population of active players. If retention is not good, there is no need to think about profitability when the customers are not engaging with the product.

A. Experiment design: Prototyping

Typically at the prototyping phase, there are various product assumptions that need to be validated. At this point the aim is to focus on the biggest unknowns that dictate the main characteristics of the game. Assumptions typically fall in two

TABLE I
KPI INDICATORS

KPI Name	Description	Notes
Retention D(x)	Indicator how many people are using the service on D(x). E.g., D7 means how many people who started the service are using it on exactly day 7.	This is the primary indication of the desirability of the product.
Conversion C	Indicator how many people convert to paying customers.	In non-free to play services the conversion is always 100 % as customers pay prior to acquiring the service.
Daily Active Users DAU	How many people engage with your product daily; the primary long term indicator of the overall performance of the product.	Monthly Active Users (MAU) can be used similarly.
Average Revenue per Daily Active User ARPDAU	For each user, how much they spend on average on a single day	This is the primary indication of the viability of the product.
Life Time Value LTV	How much each user is paying for the use of the service over the lifetime.	Often measured as LTV (Day). This measure is always an estimate and it is often relevant to consider Day x, e.g. LTV (D90).
Cost Per Install CPI	How much does it cost to acquire a single user	This is the primary marketing cost measure.
Effective Cost Per Install eCPI	Factoring in paid users and users who discovered the service on their own, how much did it cost to acquire a single user	This is the cost per new user across the entire new user population.

primary categories: *feasibility* (mainly technical merits) and *desirability* (how fun or attractive the product is):

- **Prototyping technology – feasibility.** The goal is to validate certain technology assumptions with given financial limitations and performance goals. All technology can be prototyped with quantitative data and can be experimented on, since technology performance can always be measured.
- **Prototyping product feature – desirability.** The goal is to validate whether a certain aspect of the product is desirable from the customer's perspective. At this point fast iteration is a must. As the company representative says: *"All learning is based on failures and it is important to learn and thus fail fast."* As there is no reliable way to get quantitative data, this is mostly a qualitative process. Therefore, user experiences with the product should be collected and analysed. Typical things to prototype are the features that are the most important to the game. They always depend on the game and can range from highly interactive areas (such as combat mechanics in many games) to slower interactions (such as asynchronous multi-player).

The main method of data collection is simply testing each single feature with as representative users as possible. Since this is an early stage, there may be confidentiality restrictions, and therefore the development teams are often inclined to go with internal team testing.

The number of experiments are high at this stage. A company representative states: *"When prototyping, there are so many options, then you experiment more often to make a decision on these options. In preproduction and production you still experiment, but less."* In addition to that he adds that team size is relatively small at this stage, in contrast to: *"[..] in the later stages of the development, we always interact and work with other teams."*

There is often very little useful quantitative data at these stages, but plenty of qualitative data from the experiments with proxy users, e.g., internally with the developer team. However then cognitive biases might pose a risk. For example, the confirmation bias can make the developer team interpret the results as if they validate the hypotheses and assumptions.

When all goals have been achieved and all the identified hypotheses confirmed, the development can move to the next phase: preproduction.

B. Experiment design: Preproduction

As already mentioned, the goal of the preproduction stage is to define the final product for fast execution. As the biggest assumptions have been validated at the prototyping phase, now the focus is on the complete product. The main question to ask here is how does the complete product work when it is ready? At this phase, it makes sense to collect some quantitative data already, for instance by running a short-term experiment with clear goals to determine how the users engage with the product.

- **Complete product – desirability.** The goal is to validate how each part of the product function together to create a coherent product. The best metric to measure at this stage would be related to *retention* - be it in seconds, minutes, hours or days. Developers should focus on whether the potential end users reach all desired states in the product and in the order that was desired. At this point it truly pays off to experiment often. A typical hypothesis to validate is to develop multiple parts of the game and test how they work together, for example: asynchronous battle, player discovery and player progression. After this stage, it is more difficult to change the major features of the game.
- **Validating technology – feasibility and viability.** At this phase the technology should be locked down. This means the scalability and viability (i.e. the costs) should be measured and projected to be on the right track.

At this stage, the main method of data collection is experimenting with a single product iteration with potential real customers. As there may still be confidentiality restrictions, teams are often reverting to internal team testing. However, at this stage it is also important to test outside the team to get unbiased results. However, there exists a limitation: *"The main problem in this phase is the lack of the final product. The experiments are focused on a non-final product, meaning it is still a subset of what is going to be the actual product."*

C. Experiment design: Production

The goal of the production phase is to implement the minimal final product for the market testing. The complete product is specified based on experiences from the earlier phases, and now it is time to prepare for the real tests in the market. The goal in this stage should be the speed of execution and quality, since the shorter time is spent here the quicker the product can be validated to be ready for the market.

- **Complete product – desirability.** The product is in full development and validating design decisions as often as possible with real users is a must. As new features mature, it makes sense to validate each feature with the potential users. Corrective measures can still be made with a penalty to schedules at this stage. If major flaws are found during production, the penalty to schedule is very long, since it strongly indicates that some hypotheses in preproduction were not true to begin with.
- **Complete product – first time user experience and speed.** As mentioned earlier, retention is the key feature in service development and it is important to focus on the first time user experience at this point. The product does not have a good *Day 1 retention* if users abandon the game in the first minute. The retention should be measured and studied gradually, starting from the first seconds, to minutes, to hours to days to months.

The main method of data collection is running experiments on a complete product iteration with real users. It is a good practice to have multiple milestones during production where each milestone build is tested with a certain set of potential users.

Besides reaching out to potential customers to experiment with, the other main problem in this stage is often the lack of speed. The more time that is spent here the longer it takes to actually validate the product in the market.

D. Experiment design: Market test

As the product is deemed ready implementation-wise, it is ready for the markets. In the previous stage the focus was on first time user experience and speed, now it is time to measure the performance of the product in market. The product is made available in a single or multiple markets globally via selected distribution channels, and then its actual performance is measured. In gaming, this stage is called soft launching. Now, the experiments are purely KPI-driven. Most important KPI metrics and their priorities when being used in decision making, can be seen in Table II.

Quantitative data regarding the quality of the product is available at this point and the main remaining problem in this phase is often the lack of performance of the product.

E. Experiment design: Live service

As the product has matured through the market test and achieved profitability status, it will hopefully live for several years to come. New features are still constantly added to the product during this stage but now the key is to keep or improve the performance, not make it worse. When adding a new feature to the product there is always a hypothesis on how the feature improves the overall performance. Therefore, it should be experimented. However, there are also cases where adding a single feature does improve a KPI (for example more players convert to paying customers), but it decreases the overall LTV (for example the players get exhausted by playing too much too early). Experiments are the mechanism to ensure these situations are avoided. The experimentation takes place in the following steps:

1) Establish your hypothesis.
2) Decide how to validate the hypothesis with measures (metrics).
3) Design the experiment, and select the correct methodology (for instance, user test with a limited set of people, live test with a subset of customers) with respect to the measurements.
4) Run the experiment to get the required data set.
5) Analyse your results. The hypothesis is either validated or falsified. Repeat the experiment if necessary.
6) Move forward to the next hypothesis in the process, or further iterate this hypothesis.

Company representative says: *"It is essential that every new feature that is added to the product needs to improve the product"*. This is a really important point, it is as easy to make the product worse with an update as it is to make it better. Due to this, all new features are A/B tested [21] at this stage. The effect on every KPI is measured and thus new features should not be released if they do not perform well. The same applies to removing a feature – such removal can also be A/B tested.

V. Lessons Learned and Discussion

It has become clear by now that the mobile gaming market is so competitive that it is almost impossible to create new, truly innovative games via mechanisms that would merely rely on planning. Instead, the key to success is to learn on the fly as quickly as possible. This calls for an approach where the focus is on learning, which can be realized by putting experiments in a central role in the development.

One of the key lessons we learned in this research is that experiments should be structured and they need a concrete goal, in the form of question, assumption or hypothesis, in the first place to get a solid outcome based on data; because generic or wide data collection that is used as basis for decision making is not applicable in the long run. Furthermore, different stages in the process have different needs regarding data, implying that also experiments need to be designed differently. For instance, A/B tests are a must in the line service stage to test whether new feature should be added

TABLE II
KPI PERFORMANCE IN MARKET TEST

KPI Name	Description	Priority
Retention D(x)	The primary measurement. D1, D7, etc. The product needs to be improved from the D1 first and then move to later. All focus should be here until this is at the desired level.	1
ARPDAU and **Conversion C**	As retention is working, the product efforts should move to optimising revenue generation.	2
Life Time Value LTV	Relates to the overall viability of the product. This takes a longer time to measure. Early estimates can be made at e.g. Day 7	3
CPI and eCPI	As the developer knows the LTV it can be estimated what should be the marketing costs for a single user to create a healthy product.	4

to the game. As the company representative expresses: *"We have more experimentations in the beginning of development cycle, the number goes down when the product matures from prototyping to production (but never goes to zero). Once we get in the market, it is a different story — experiments take place continuously."* In addition to that, we highlight that earlier in the development the experimentation is especially crucial since the cost of change once a product is fully implemented is much higher. In other words, invalidating wrong product assumptions as early as possible in the development is very important.

One of the main challenges is lack of real users to experiment with especially early in the development, that is a common case for new applications. Alternative methods such as testing with proxy users can aid the situation, yet resulting mostly in qualitative data. Later in the development, once the game is on the market, quantitative data is available en masse. Focused experiments with pre-defined metrics, e.g., KPIs, helps to validate product hypotheses and assumptions, therefore evaluating the success of the product. It should be noted that KPI metrics are measured continuously throughout the development process, but they mostly make sense with voluminous real user data, which only comes later in the development.

In general, both qualitative and quantitative data can – and usually should – be used in decision making, but finding a balance that leads to the best possible results is difficult (see Section II-B for classifying data). The fact that the majority of quantitative data is only available long after the initial decision to invest in a game complicates things during the early stages of the development. On the other hand, early stages in the development are so crucial to test the game idea and user value, as these stages define the product. Qualitative data is more likely to be collected to support the development in these stages, however, cognitive biases such as confirmation bias should be taken into account.

In addition to data directly related to experiments as such, additional data may also be gathered to foster observations on user behaviour. For instance, recommendations, reviews, gaming patterns, and other user actions can provide further insights to developers. This data can also be studied offline to find correlations and other statistical relations. For instance, an important practical insight is that updates that require several minutes to download is a turnoff for players in general, as for example the gaming session during a bus ride is ruined by it. Such observations will also help in the development of future hypotheses and associated experiments.

Furthermore, we were also interested in how experimentation culture has started in the case company and how the developers are motivated to run continuous experimentation. Company representative explains: *"In gaming, continuous experimentation is not about motivation. Experimentation is a must, everybody agrees to that. It is all about scheduling and managing the process."*. He further emphasizes the role of leadership in experimentation practices in order to schedule the process, yet also adds: *"There is an established experimentation culture in our company and one of the key points is that the development teams are self-organized to run experiments."*

In this study, due to ongoing development of the games and confidentially, rather than detailing on the experiments we focused on the experiences gained during the case company's game development journey. We believe that the learnings presented in this paper will shed light on general game development and data-driven software development.

VI. Conclusion

In this paper, we presented a game company's product development journey, with special emphasis on experimentation that takes place continuously. The goal is to ensure that the game meets user expectations in such a competitive market, where users are quick to choose another product if one fails to satisfy their needs.

We observed that experimentation takes many forms, and that the different stages in the development calls for different experimentation strategies. A key differentiator in the nature of experiments is the available target audience – in the early stages, usually only a few gamers produce data, which is mainly collected via interviews, whereas later on, statistically meaningful data can be collected as the product enters the full production phase. Therefore, challenges such as biases might occur early in the development as qualitative data overweighs. On the other hand, in the later stage the focus shifts to choosing the right metrics and experimenting with large amounts of data collected from real users. Data collection methods should be balanced based on the development stage and development goals of the game.

Furthermore we learned that experimentation is a learning process and it is valuable to fail fast and eliminate wrong product assumptions, which otherwise would turn into undesirable product implementations that are expensive to fix. Continuous experimentation in early product development is especially crucial since early development stages have bigger influence on determining the final product and its success. As well as having self-organized development teams on planning and running structured experiments, guidance provided by the leadership is important to manage the process and allocate the time and resources efficiently.

Although experiments are carefully crafted and serve predefined goals, there still is room for ad-hoc analysis of the produced data. This can reveal patterns and insights that help us understand how the users actually use the product, which in turn helps us create better experiments in the future. This feedback loop needs to be included in the development process as a part of the retrospective analysis of the development.

References

[1] F. Fagerholm, A. S. Guinea, H. Mäenpää, and J. Münch, "Building Blocks for Continuous Experimentation," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, ser. RCoSE 2014. New York, NY, USA: ACM, 2014, pp. 26–35.

[2] S. G. Yaman, M. Munezero, J. Münch, F. Fagerholm, O. Syd, M. Aaltola, C. Palmu, and T. Männistö, "Introducing continuous experimentation in large software-intensive product and service organisations," *Journal of Systems and Software*, vol. 133, pp. 195–211, 2017.

[3] O. Rissanen and J. Münch, "Continuous Experimentation in the B2B Domain: A Case Study," in *Proceedings of the Second International Workshop on Rapid Continuous Software Engineering*, ser. RCoSE '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 12–18.

[4] J. Järvinen, T. Huomo, T. Mikkonen, and P. Tyrväinen, "From agile software development to mercury business," in *International Conference of Software Business*. Springer, 2014, pp. 58–71.

[5] D. G. Feitelson, E. Frachtenberg, and K. L. Beck, "Development and deployment at Facebook," *IEEE Internet Computing*, vol. 17, no. 4, pp. 8–17, 2013.

[6] A. O. O'Hagan, G. Coleman, and R. V. O'Connor, "Software development processes for games: a systematic literature review," in *European Conference on Software Process Improvement*. Springer, 2014, pp. 182–193.

[7] Statista, "Number of new apps/games submitted to the iTunes store per month 2012-2016," https://www.statista.com/statistics/258160/number-of-new-apps-submitted-to-the-itunes-store-per-month, 2018, accessed: 2018-02-02.

[8] A. Chen, "New data shows losing 80% of mobile users is normal, and why the best apps do better," http://andrewchen.co/new-data-shows-why-losing-80-of-\ \your-mobile-users-is-normal-and-that\-the-best-apps-do-much-better/, accessed: 2018-02-02.

[9] A. Meola, "Mobile gaming is about to become the undisputed king of the jungle," http://www.businessinsider.com/mobile-gaming-will-surpass-legacy-gaming-in-2016-2016-4?r=US&IR=T&IR=T, 2016, accessed: 2018-02-02.

[10] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "What went wrong? a survey of problems in game development," *Computers in Entertainment (CIE)*, vol. 7, no. 1, p. 13, 2009.

[11] C. M. Kanode and H. M. Haddad, "Software engineering challenges in game development," in *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. IEEE, 2009, pp. 260–265.

[12] M. Washburn Jr, P. Sathiyanarayanan, M. Nagappan, T. Zimmermann, and C. Bird, "What went right and what went wrong: an analysis of 155 postmortems from game development," in *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, 2016, pp. 280–289.

[13] S. Aleem, L. F. Capretz, and F. Ahmed, "Game development software engineering process life cycle: a systematic review," *Journal of Software Engineering Research and Development*, vol. 4, no. 1, p. 6, 2016.

[14] M. M. McGill, "Defining the expectation gap: a comparison of industry needs and existing game development curriculum," in *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 2009, pp. 129–136.

[15] J. Kasurinen and K. Smolander, "What do game developers test in their products?" in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2014, p. 1.

[16] R. Ramadan and Y. Widyani, "Game development life cycle guidelines," in *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*. IEEE, 2013, pp. 95–100.

[17] G. McAllister and G. R. White, "Video game development and user experience," in *Game User Experience Evaluation*. Springer, 2015, pp. 11–35.

[18] B. W. Boehm *et al.*, *Software engineering economics*. Prentice-hall Englewood Cliffs (NJ), 1981, vol. 197.

[19] NextGames, "The compass point: West," [Software] http://www.compasspointwest.com/, 2015.

[20] ——, "The walking dead: No man's land," [Software] http://www.thewalkingdeadnomansland.com/, 2015.

[21] R. Kohavi and R. Longbotham, "Online controlled experiments and a/b tests," *Encyclopedia of machine learning and data mining*, pp. 1–11, 2015.