UNIVERSITY OF TURKU

ABSTRACT

	Bachelor's thesis
X	Master's thesis
	Licentiate's thesis
	Doctoral dissertation

Subject	Information System Science	Date	29.04.2020
Author	Henrik Byman	Number of pages	113 + appendices
Title	Player participation in video game development		
Supervisor	Ph.D Jani Koskinen		

Abstract

The trend of involving players in early game development has increased considerably during recent years. As an alternative for the traditional game creation methods, in player participation, players get to play the game and influence its development direction before the game is finished and published. Similar development methods have been employed already for long in the information systems development context, but in the early game development context, the participation of players is still a relatively unexplored area. This research attempts to fill that gap and provide a more comprehensive understanding of player participation as well as investigate its usefulness in game development.

A qualitative case study approach was adopted in this study. Four Finnish game studios were invited for semi-structured interviews to share their knowledge of involving players in their projects, and 21 players responded to a questionnaire about their experiences with game development projects. Interviews were transcribed and coded along with players' responses.

Player participation is best described as a collaborative game development method characterised by voluntariness, and emphasizing communication and interaction between the player community and the game studio for creating a gameplay experience according to the emerging discourse between game studio's development vision and players' desires. In it, a game studio and its player community engage in various community management, communication and testing activities to create a mutually satisfying product. The participation process itself is characterised by voluntariness of participation as well as varying degrees of participation thresholds and levels of commitment towards the project among players. The benefits of player participation are found in the availability of additional resources and more cost-efficient development, validating game design choices with and targeting the gameplay experience for the player com-

tential for better re-

ce the challenges in managing the increased workload, potential changes in various work practices, and creating and maintaining the player community, which consists of diverse individuals with each of their varying interests and desires towards the game development project.

Keywords	video game development, user participation, player participation

UNIVERSITY OF TURKU

TIIVISTELMÄ

	Kandidaatintutkielma
X	Pro gradu -tutkielma
	Lisensiaatintutkielma
	Väitöskirja

Oppiaine	Tietojärjestelmätiede	Päivämäärä	29.04.2020
Tekijä	Henrik Byman	Sivumäärä	113 + liitteet
Otsikko	Pelaajien osallistaminen videopelienkehityksessä		
Ohjaaja	FT Jani Koskinen		

Tiivistelmä

Pelaajien osallistaminen aikaisessa pelinkehityksessä on lisääntynyt viime vuosina huomattavasti. Vaihtoehtona perinteisille pelinkehitysmenetelmille pelaajien osallistamisessa pelaajat pääsevät pelaamaan ja vaikuttamaan pelin kehityssuuntaan ennen pelin valmistumista ja julkaisua. Vastaavan kaltaisia kehitysmenetelmiä on jo pitkään käytetty tietojärjestelmien kehityksessä, mutta aikaisessa pelinkehityskontekstissa tämä on vielä varsin tutkimatonta aluetta. Tämä tutkimus pyrkii täyttämään tuota tyhjyyttä kirjallisuudessa ja tarjoamaan kokonaisvaltaisempaa ymmärrystä pelaajien osallistamisesta sekä tutkimaan sen hyödyllisyyttä pelinkehityskontekstissa.

Tämä tutkimus suoritettiin laadullisena tapaustutkimuksena. Neljä suomalaista pelistudiota kutsuttiin puolistrukturoituihin haastatteluihin jakamaan heidän tietämyksensä pelaajien osallistamisesta heidän projekteissaan, ja 21 pelaajaa vastasi kyselyyn omista kokemuksistaan pelinkehitysprojekteissa. Haastattelut litteroitiin ja koodattiin pelaajien vastausten ohella.

Pelaajien osallistamista voi parhaiten kuvata yhteistyökeskeisenä pelienkehitysmetodina, jossa painottuvat vapaaehtoisuus sekä kommunikointi ja vuorovaikutus pelaajayhteisön ja pelistudion välillä, ja jossa pelielämys rakentuu pelistudion vision ja pelaajayhteisön toiveiden välisen vuoropuhelun mukaisesti. Siinä pelistudio ja pelaajayhteisö osallistuvat erilaisiin yhteisönhallinta-, kommunikointi- ja testausaktiviteetteihin luodakseen molempia osapuolia tyydyttävän tuotteen. Osallistumisprosessia luonnehtivat vapaaehtoisuus, sekä pelaajien vaihtelevat osallistumiskynnyksen ja sitoutumisen tasot peliprojektia kohtaan. Pelaajien osallistamisen hyödyt tulevat lisääntyneiden resurssien saatavuudesta ja kustannustehokkaammasta kehityksestä, pelisuunnitteluun liittyvien päätösten hyväksyttämisestä pelaajien kanssa sekä pelikokemuksen kohdentamisesta pelaajille, ja teknisesti toimivamman pelin kehittämisestä sekä mahdollisuudesta parempaan vastaanottoon pelin valmistuttua. Kuitenkin pelistudion täytyy myös kohdata pelaajien osallistamisesta juontavat haasteet, kuten lisääntyneen työmäärän hallitseminen, mahdolliset muutokset eri työkäytännöissä, ja pelaajayhteisön, joka koostuu erilaisista yksilöistä, joista jokaisella on omat eriävät intressit ja toiveet peliprojektia kohtaan, rakentaminen sekä ylläpito.

Avainsanat	videopelienkehitys, käyttäjien osallistaminen, pelaajien osallistaminen

UNIVERSITY OF TURKU
Turku School of Economics

PLAYER PARTICIPATION IN VIDEO GAME DEVELOPMENT

Master's Thesis
in Information System Science

Author:
Henrik Byman

Supervisor:
Ph.D Jani Koskinen

29.04.2020
Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Research gap

In the video game development industry, it has long been a prevalent practice to maintain a shroud of secrecy around upcoming game titles and gradually drop hints to build up interest among consumers. Players are usually left to wait and those who get a chance to participate in game development, such as in closed beta-testing, are in most cases required to sign a non-disclosure agreement. (Zimmerman, 2014) Players who participate in the (closed) beta-testing receive access to the game and through playing can assist the development team. However, as the game is usually close to the final release version at this point, the contribution value of players is mainly limited to helping developers to root out bugs and optimize gameplay. Open beta-tests function more as a stress test whereas in closed beta-tests there is little bit more margin to influence the development (Gandolfi, 2018).

Nevertheless, in the 2010s a phenomenon in which players are engaged already during the early stages in the game development process and are able to influence the direction of the project has gained popularity. For example, in 2013 digital game distribution platform Steam by Valve Corporation launched its Early Access platform, which enables game developers to sell their incomplete games to interested consumers and to gather feedback and suggestions to enhance further development (Early Access FAQ, 2019). Since then over 3600 games with sales data have been released on the Early Access platform (Steam Spy, 2017) and other digital game distributors have followed in Steam's footsteps (Alex Wawro, 2016).

Despite the apparent popularity of this phenomenon, little research has been conducted on the topic so far and many issues are yet to be studied and uncovered. Characteristics of the Early Access games have been studied (Lin et al. 2018) in addition to main activities in the development process where players are engaged (Thominet, 2018) as well as quantitative analysis has been done on participants attitude towards player engagement (Gandolfi, 2018) but this phenomenon is still lacking research in many frontiers such as in theories and methods (Thominet, 2018). The closest alternative research area could be considered to be the studies of user involvement and participation in information system development.

1.2 Theoretical background

User involvement and user participation are frequently employed methods in custom system development and have been extensively studied in the information system literature for almost half a century. Its goal is to enhance the outcome and success rates of information system development. User participation refers to the behavioural side of user engagement and consist mainly of the activities users perform in system development whereas user involvement refers to the attitudinal side of user engagement, which means the psychological stance users hold towards system development and its outcome (Barki and Hartwick, 1989; Kappelman and McLean, 1991). User engagement is used as an umbrella term for both concepts (Kappelman and McLean, 1991). Since the 1960s, user participation has been assumed to lead frequent system usage and user satisfaction through high-quality systems (Hwang and Thorn, 1999). When information systems fall short of delivering expected benefits, it is cited that insufficient involvement of users in the design process is one of the major reasons. When users are not involved in planning and designing systems, organizations postpone problems into the future, where problems become more difficult to trace and more expensive to fix. (Damodaran, 1996) According to Kujala (2003), the general understanding appears to be that involving users is generally beneficial to system development. The benefits of involving users are for example in acquiring more accurate user requirements, avoiding unnecessary system features, and improving system acceptance (Damodaran, 1996; Kujala, 2003).

However, as user participation is mostly employed in business-to-business context and custom software development, it cannot be directly applied into game development context, that resembles more the characteristics of packaged software development in business-to-consumer environment. The main differences between these two approaches are that in packaged software development there is no single well-defined user but potentially multiple different user types with diverse preferences, users are usually unknown until the product has been sold complicating user participation before product release, and companies need to compete against each other in an uncertain and competitive marketplace where there is no insurance of product sales before product release contrary to custom software development where contracts are signed between clients and system providers before system development. (Carmel and Becker, 1995; Karlsson et al. 2002) Despite that, user participation literature in information system development research provides a good foundation, due to its long and extensive research history, to start examining

the video game player participation phenomenon, given that the insights of user participation research are critically evaluated in terms of the differences between custom and packaged software development as well as the differences between software engineering and game development. These topics will be briefly covered in the following literature review section alongside the literature concerning user participation.

1.3 Research setting

There appears to be no established concept yet to address this new phenomenon where video game players engage in video game production with the video game creators and co-produce the final product. Concepts such as open development (Gandolfi, 2018; Thominet, 2018), participative development, extended prototyping (Gandolfi, 2018), participatory design and perpetual beta (Jacobs and Sihvonen, 2011); have been used in the literature but all roughly signify the same phenomenon where players engage in the video game development with developers. For example, Thominet (2018) defines open development as "a user-centred design practice where a developer publicly distributes an incomplete game and iterate on it while gathering feedback from the player community". Jacobs and Sihvonen (2011) discuss of perpetual beta as a state in game development where the game lives in a constant state of fluctuation of new content with updates being introduced regularly. However, these constructs sound like that there is a technically centred aspect (software or its development) embedded in the core of these concepts, and for that reason, none of them will be used in this research to address this phenomenon. Instead, a new concept – player participation – is formulated for several reasons and will be used throughout this research:

- To stay consistent with the prior user involvement and user participation research in information system literature which functions as a major part of the academic foundation for this research.
- To shift the aspect of observation from the technical perspective more to the side that emphasizes the interaction between individuals and its process and results.

Therefore, the goal of this research is to construct a better understanding of the 'player participation' trend by asking:

What is the meaning of player participation in video game development?

This main research question is divided into four smaller research question each addressing a specific part of the bigger picture:

- Who are involved in player participation in game development?
- How is the nature of player participation in game development?
- What are the relevant activities in player participation in game development?
- What kind of outcomes are there to player participation in game development?

The focus will be on those commercial projects and platforms that enable player participation starting from the early stages of the development process, on those game studios or publishers that utilize player participation, and on those individuals who either participate or are involved in these settings. In the following section literature on user participation research in information system development, game and packaged software development research, and player participation research will be reviewed.

2 LITERATURE REVIEW

2.1 Video game development

2.1.1 The prevalent practices in game development industry

2.1.1.1 The business practices

The structure of the video game industry resembles other creative industries consisting of producers (game developers), publishers, distributors and retailers. Game developers cannot usually finance the development by themselves which leads them to seek funding from game publishers. Game publishers accept the financial risk for the development project, but in turn, want to ensure that the game is finished and sales well. (Zackariasson and Wilson, 2010) This leads to the prevalent issue of tension between the two opposing forces in the game development industry: the push for creativity by game developers and rationalization of production processes by business people (Tschang, 2007).

Rationalization is defined as "the predominant focus on business interests or productivity-oriented production processes, usually at the expense of creativity". It is driven by many factors such as the emergence of large publishers in the game development industry, who in the face of ever-increasing development costs and complexity within creative industries, rely on combining interesting intellectual property with well-tried gameplay designs to reduce the risks. Rationalization may also stem from factors of publishers originating from outside game development industry knowing little about games or about their development, publishers' obligations towards investors and financers, from the limited shelf space of retailers forcing publishers to focus on popular titles, or from increasing sizes and complexities of game projects leading to the rationalization of development teams and work procedures in order to manage potential risks. Eventually, this rationalization leads to incremental innovation instead of radical innovation producing less diverse gaming experiences to consumers, who in future expect less from game development industry thus reinforcing further rationalization. However, some consumers seek for completely new experiences and developers who want to push their innovative ideas in game development, leading to tensions between the forces in favour of rationalization and the forces in favour of radical innovation. (Tschang, 2007)

Another prevalent practice in the game development industry is to maintain a shroud of secrecy around upcoming game titles. The purpose of secrecy is to generate interest

and desire to know more about the upcoming titles among players and is hoped to translate later into stronger sales. Small hints are dropped gradually in a form of teasers with a specific amount of time between them to allow discussions and speculations emerge among the consumer base. Teasers usually carry the message that the game development project exists and that more is around the corner. (Zimmerman, 2014)

Cox (2014) identified that factors such as a game being released by a major publisher, released on a popular home platform, or a game being of higher quality than others, are associated with a higher probability of a game title becoming a blockbuster in terms of sales. Assessing game quality is a subjective topic but game reviews and review ratings are usually employed as an approximation (Zackariasson and Wilson, 2010; Cox, 2014). Game quality and gameplay experience are perceived as more important in terms of sales than securing a game license or a franchise (Cox, 2014). Aleem et al. (2016b) investigated the effects various business factors have on the performance of games in a market and concluded that "customer satisfaction, time to market, monetization strategy, market orientation, and brand name strategy are positively associated with the performance of a digital game organization". However, no strong association between innovation or relationship management and game performance was found. The importance of third-party tools and middleware in game development has also increased (Wang and Nordmark, 2015) alongside with monetization during the recent years (Murphy-Hill et al. 2014). The technical aspects of game development have become easier also but game development itself is still as difficult due to higher player expectations and game complexities (Wang and Nordmark, 2015).

2.1.1.2 The game development life cycle

Videogames share similarities with other entertainment and interactive products (Tschang, 2005). A game is usually built a little section at a time (Amaya et al. 2008) through flexible and iterative development processes (Musil et al. 2010) which are the preferred way in game development industry (Kanode and Haddad, 2009; Lewis and Whitehead, 2011; Murphy-Hill et al., 2014). After a game has reached playable state, it is fleshed out and refined, contrary to productivity applications which are usually shipped out after they are stable and include the required functionality (Amaya et al. 2008). Agile development approaches are suitable when innovation and time to market are at stake (O'Hagan et al. 2014) and are sometimes even instinctively adopted by game studios (Petrillo and Pimenta, 2010). Hybrid approaches are also utilized, which are a mixture of

agile and traditional software development elements, mainly differ in terms of the role of iteration, and are more suitable to games that have a longer lifespan and more stable requirements (O'Hagan et al. 2014). Waterfall processes are also still in use but declining in popularity (Politowski et al. 2016).

Game development is characterized by fixed deadlines (Murphy-Hill et al., 2014), tight financial and technical constraints (Lewis and Whitehead, 2011). It has become customary to view game development life cycle consisting of several phases. Concept phase, prototyping, pre-production, production phase that also includes testing, and release phase construct the development life cycle (McAllister and White, 2010) whereas production life cycle includes publishing, distribution, and retail phases in addition to the development life cycle (Kerr, 2006). In the absence of formal risk management procedures in the game development industry, agile practices, prototyping, and pre-production phase have become informal practices to mitigate development-related risks (Schmalz et al. 2014).

The concept phase is either initiated by a game development studio or by a game publisher. Initial game design document (GDD) and concept art are produced alongside with a rough budget, and a development plan that includes development milestones. (McAllister and White, 2010) A GDD is a plain-language narrative of the game that attempts to describe its story, mechanics, look and feel, and capture the creative vision of developers and designers (Callele et al. 2005). There is no specific standard for it but it mostly determines the elements that create an enjoyable game experience and describe how those are delivered (Sansone, 2014). Through requirements engineering, GDD is transformed into game specifications for the production phase (Callele et al. 2005).

A game is made of its functional requirements that concern the software implementation of a game as well as the game rules within the game, and of non-functional requirements that determine what kind of emotions and experiences the game has to generate (Callele et al. 2006). According to Callele et al. (2008) players care about the functional requirements to the point it affects gameplay experience. Otherwise, they are just considered to be the minimum requirements that must be met. Emotional requirements on the other hand, are the most important requirements in terms of a successful game, but because of their nature, emotional requirements are highly subjective, difficult to quantify (Callele et al. 2006), and may require significant contextual information and localization effort (Callele et al. 2008). Emotional requirements consist of the target emotional state and of the means to induce it (Callele et al. 2006; 2008). The target emotion itself may be

universal but there are multiple ways to deliver it (Callele et al. 2008). Besides emotional requirements, Callele et al. (2010) propose the concept of experience requirements that, in addition to emotional experiences, consists of gameplay experiences (cognitive and mechanical experiences) and sensory experiences (visual, auditory, and haptic experiences).

However, requirements engineering practices are yet to be systematically adopted in game development industry (Kasurinen et al. 2014) even though they are among the most studied topics in software engineering research for computer games (Ampatzoglou and Stamelos, 2010). Requirements engineering practices do not contradict with creativity and are very much needed as games may evolve substantially between the prototypes and final versions based on the feedback received from the market and customers (Kasurinen et al. 2014). Practices and ideas could be adopted from software engineering research as "computer games are a fertile domain for applying software engineering technologies" (Ampatzoglou and Stamelos, 2010). Some form of requirements engineering and requirements management may exist as implicit adoptions such as functional requirements being outsourced by acquiring ready-made third-party tools like game engines that already handle the required functionality or non-functional requirements such as fun in gameplay and associated risks being managed through practices of prototyping and user testing, but are mainly limited to individual practices. (Kasurinen et al. 2014) Code and component reuse are situational in game development due to the emphasis on project-specific innovation and performance and is mostly present in subsequent game releases in the forms of game engines as well as in-house developed tools (Murphy-Hill et al., 2014). The value and utility of GDDs have also been questioned as to whether they are too static, restrict developers' creativity and inhibit the fluid and evolving nature of game design, and whether they are nowadays only a relic of the time when games were mainly developed using waterfall-like methods (Sansone, 2014).

Sketches and plans designed thus far function as a basis for the prototyping stage where examples of game features are constructed, tested and evaluated together with the game concept. If the game concept appears promising and prototypes functional, the development moves on to the pre-production stage, where core game features are constructed. (McAllister and White, 2010) Potential licensing, financing, publishing, and distribution deals are also negotiated at this point (Kerr, 2006). The game development studio may sign a contract with a publisher to develop a game, the game studio may itself be owned by the publisher and assigned to develop a game, or the game studio develops a

game independently without any dependencies to publishers and distributes the game digitally through Internet (Kerr, 2006; Zackariasson and Wilson, 2010). Schmalz et al. (2014) found out that when a publisher is involved, developers feel a greater need to pay attention to development budget and schedule while trying to preserve the minimum feature set whereas, in self-funded situation, gameplay and fun were emphasized at the expense of development schedule and budget.

In the production phase, gameplay features and components are fleshed out, while expert players with a good understanding of video game mechanics may be recruited to test the game to uncover playability, usability, and quality issues (McAllister and White, 2010). Those who are given the privilege to participate at this point are usually required to sign a non-disclosure agreement (NDA) to protect the game studio of content theft or leak, and in order not to reveal public the incomplete, potentially broken status of the game, which could affect the game sales after the release (Zimmerman, 2014).

Various milestones can be laid out for the development project but few are widely acknowledged phases or states in the game development industry. A game can be said to have reached the Alpha state after all necessary content has been implemented in the game but not necessarily in final form, the Beta state when all content and features are implemented and fleshed out but may still require additional polishing, and the Gold state when the game is officially finished and handed over to manufacturing and distribution. Before the Gold state, if a game is released to consoles it must go through the format holders' quality testing to ensure it matches the quality standards of the target platform. (McAllister and White, 2010)

A beta test is usually conducted during the Beta phase to polish the game of hidden bugs and other software and hardware related compatibility issues. In public (open) beta testing, access to the game is given to players for playtesting purposes. This enables the game to be tested on a much wider repertoire of hardware and software combinations by players than what would be possible to simulate in an in-house (closed) laboratory environment. However, public beta testing has its conditions and restrictions such as a game has to be in stable and functional condition since it is usually its first public exposure, and since the playtesting typically occurs at players home, developers are unable to control the context, gather similar feedback or give similar instructions than what would be possible if the testing was conducted in a controlled environment. (Amaya et al. 2008) Therefore, a public beta test functions more as a stress test whereas in closed beta testing there is a little bit more margin to influence the development (Gandolfi, 2018).

Public beta testing is fairly common among productivity applications and PC games but relatively new among consoles games and in improving the core gameplay experience. (Amaya et al. 2008) Game studios usually perform their game tests too late in the development life cycle, making it more difficult to incorporate gathered feedback into a final game (McAllister and White, 2010). To enable beta testing as a tool to gather feedback from players and incorporate it into game design, beta testing must be brought forward in game development life cycle so that gathered feedback can be integrated into a game and verified by players through iterative approaches. Special attention must be paid to selecting participants so that the gathered feedback truly represents the target population. (Amaya et al. 2008) However, arguments exist that conducting tests more towards the end of development life cycle would yield more representative and valid test results of user experience as all the gameplay components would be in place (McAllister and White, 2010) and that distributing early build versions of a game as a beta version may produce bad impressions due to more unfinished state and may affect future sales negatively (Amaya et al. 2008).

2.1.2 Challenges in game development and design

There are multiple challenges and problems that plaque the game development industry. All the main problems that are found in the software development industry are also present in the game development industry, such as unrealistic scope, exaggerated optimism, scheduling problems (Petrillo et al. 2009; Washburn et al. 2016), problems in requirements analysis, and problems in budgeting, quality and management practices (Petrillo et al. 2009). Unrealistic scope refers to overly ambitious development goals that are unrealistic to acquire and combined with overly optimistic evaluations of required effort, leads to the birth of the most problems within the two industries. (Petrillo et al. 2009) Insufficient requirements engineering is usually held accountable for project failures when transitioning from the pre-production phase to the production phase. In the case of games, this usually concerns transforming abstract game concepts such as fun into programmable goals. (Callele et al. 2005) Furthermore, people in leading positions, such as senior executives or game publishers that possess a Machiavellian personality, cause trouble and suffering in both industries (Petrillo et al. 2009) In a study by Petrillo and Pimenta (2010) only in a quarter of the projects good management practices had been adopted.

Problems that are mostly exclusive to game development industry are feature creep (Tschang, 2005), cutting off features during development, various problems in the design

phase, technological problems such as integrating third-party application programming interfaces of hardware, crunch time, lack of documentation, communication problems in multidisciplinary teams, and elaboration of subjective and abstract game requirements (Petrillo et al. 2009). Blow (2004) suggests that problems emerge either due to project size and complexity or due to highly domain-specific requirements.

The multidisciplinary teams of game development, consisting amongst other things of artists, musicians, software engineers, and scriptwriters, is a fertile ground for communication problems and misunderstandings to emerge between different groups of people due to lack of common language (Petrillo et al. 2009). Besides the potential communication problems, coordination and integration of concurrent work efforts by different specialist groups (e.g. artists and developers) so that no group needs to wait for input from other teams, is a substantial challenge among the multidisciplinary teams (Tschang, 2005).

Feature creep is the process of adding new functionalities or modules into a game during its development phase without planning first (Petrillo et al. 2009). It can cause delays to the development process or it can alter the original vision of a game (Tschang, 2005). It is partly a consequence of the evolutionary and constructivist nature of video games and their development, in which it is fairly straightforward to include additional content and features into a game (Tschang, 2005). More specifically, feature creep may result from new requirements emerging after the scope has been defined, from integration work required to incorporate external code libraries to save development time, or from building algorithms from scratch despite having a pre-existing code library that already provides similar functionality. However, through feature creep it is possible to discover new functionality that can make a game more successful. (Petrillo et al. 2009)

The opposite problem of feature creep is cutting off features during development, which results either from an attempt to correct initial unrealistic development scope or to match development with available financial resources or catch up with development schedule (Petrillo et al. 2009). Problems with schedule may stem from underestimating required resources for development, but also from fixed release dates, set by expectations of the target market or by major holiday seasons such as Christmas (Tschang, 2005).

Crunch time is another method of attempting to catch up with the development schedule by forcing developers to work overtime with reduced amounts of days off. It appears to have become so common that it is almost considered a standard in game development industry while the periods of crunch time have extended from a couple of weeks prior the

release of a game to span over multiple months. Despite of its intentions to produce quality content in the promised schedule, it has serious implications on developers' family lives and wellbeing causing exhaustion and lack of sleep leading to inability to produce quality work and make rational decisions and eventually to high staff turnover rate. (Petrillo et al. 2009) Furthermore, it appears that there exists a correlation between crunch time and feature creep, which are more likely to be symptoms of workflow problems and process integration issues, rather than problems by itself. Seasonal variation in the game industry sales could also explain crunch time when game studios rush to release their titles just before Christmas sales. (Musil et al. 2010)

Besides the development-related challenges and problems, a set of issues arise when the point of focus is shifted to game design. Games are supposed to entertain and part of that entertainment and enjoyment is to overcome challenges. However, being able to tell apart a good challenge in gameplay from frustrating usability problems is a challenge itself besides to the gradually increasing challenge level to keep players engaged in the game. Besides challenges, players' different skill levels must be addressed: not every player is equal in their gaming experience or talent. Too easy challenge level becomes boring quickly and too hard challenge level is frustrating to players. (Pagulayan et al. 2012)

The other side of the gameplay challenge is rewarding players. Since playing video games is a voluntary action, players can drop games anytime if they do not feel enjoyable. (Pagulayan et al. 2012) Particular attention should be given to the development of the first-hour game experience since it determines whether players continue to immerse themselves in the game world or whether they give up and stop playing (Cheung et al. 2014). The challenge in game design is how to attract players' attention and motivate them to come back over and over again i.e. how to reward a player for playing the game. Finding and finetuning the balance of multiplayer games is also a relevant game design challenge. (Pagulayan et al. 2012)

To increase the chances of success in game development, developers ought to come up with a well-defined game concept before development, plan and design the project upfront and utilize iterative development methods and prototypes. Implementation should be kept in mind while designing a game to prevent overly ambitious design choices. One should also invest in risk management, project leadership, and talented and motivated people to carry out the development project. (Washburn Jr. et al. 2016)

2.1.3 Comparison of games and productivity applications

2.1.3.1 Comparison of the products

The purpose of productivity applications is to fill a need, provide functionality (Kasurinen and Laine, 2014; Kasurinen, 2016), enhance users' productivity by allowing users to produce higher quality work and to use less time to execute specific tasks (Pagulayan et al. 2012). They are aimed to minimize the time used to reach a specific outcome (Kasurinen and Laine, 2014; Kasurinen, 2016) and to be easy to use and adopt yet still provide enough powerful tools for professionals to conduct their work (Pagulayan et al. 2012). They mostly consist of technical (code) components and possibly of graphical components (Kasurinen and Laine, 2014) such as graphical user interface (GUI) (Kasurinen, 2016). Testing whether productivity applications fulfil their purpose is fairly straightforward (Pagulayan et al. 2012) and their quality is based on how well they fulfil the defined software requirements (Kasurinen and Laine, 2014; Kasurinen, 2016).

Games, on the other hand, are built to generate fun, enjoyment and entertainment in their users (Kanode and Haddad, 2009; Lewis and Whitehead, 2011; Pagulayan et al. 2012; Kasurinen and Laine, 2014; Murphy-Hill et al. 2014) while attempting to maximize the time spent with them (Kasurinen and Laine, 2014; Murphy-Hill et al., 2014; Kasurinen, 2016). They consist of technical, creative, narrative, graphical and audio components (Kasurinen and Laine, 2014) such as code, story, art, music (Kanode and Haddad, 2009; Kasurinen, 2016) and artificial intelligence (Amaya et al. 2008) and focus on innovating and experimenting with novel things (Davis et al. 2005). Another way to view the structure of video games is that they consist of the game setting, sensory stimuli, and the associated game rules (Zackariasson and Wilson, 2010). Draper (1999) suggests that the main goal of games is to produce enjoyment, and that fun is only one form of enjoyment. Games are supposed to include a healthy amount of challenge whereas in productivity applications challenge is an unnecessary, interfering factor sought to be minimized (Davis et al. 2005; Amaya et al. 2008; Pagulayan et al. 2012). Testing whether games fulfil their purpose, i.e. are enjoyable, is much more complicated since fun and enjoyment are subjective concepts (Pagulayan et al. 2012) and the quality of a game is determined by its users through user experience and non-functional requirements (Kasurinen and Laine, 2014; Kasurinen, 2016). As a form of entertainment, games also compete against other forms of leisure and free-time activities and thus, the competition within and against the game industry is more intense than among productivity applications (Pagulayan et al.

2012). In addition, games represent a source of interface innovation in forms of new input devices and user interface presentation whereas in productivity applications these kinds of changes and innovations are considered as a learning cost and a potential burden or productivity risk unless their benefits exceed adoption costs (Amaya et al. 2008; Pagulayan et al. 2012) leading to productivity applications attempting to remain consistent with slight changes between versions (Davis et al. 2005).

2.1.3.2 Comparison of the development practices

In software engineering, requirements are functional and based on real-world concepts (Kasurinen and Laine, 2014; Murphy-Hill et al., 2014; Kasurinen, 2016). They are absolute in the sense that failing to implement a requirement correctly could cause a system to misbehave or render completely inoperative (Murphy-Hill et al., 2014). Testing in software engineering is usually automated (Murphy-Hill et al. 2014). In the software market, few producers have acquired a strong position in their software product segment, whereas in gaming industry many publishers and suppliers exist (Amaya et al. 2008), and it is highly competitive and driven by consumer trends (Kanode and Haddad, 2009).

Requirements in game development are subjective and based on abstract concepts (Kasurinen and Laine, 2014; Murphy-Hill et al., 2014; Kasurinen, 2016) like the aforementioned fun and enjoyment. There are no real functional requirements, instead, the focus is on the performance, quality attributes and usability of games (Wang and Nordmark, 2015) These subjective requirements lead to planning being refrained from in game design in contrast to software engineering due to the chance of wasting effort on planning requirements that do not turn out to be fun. Consequences of failing to implement a requirement are not as severe as in software engineering since players are usually quick to move on to next gameplay experience after an incomplete one. (Murphy-Hill et al., 2014) Moreover, game development is more inclined to have constant changes in design even late in development (Kasurinen and Laine, 2014). Testing in game development is in most cases conducted manually due to the complexity and multistate nature games possess, which renders automated testing unfeasible (Murphy-Hill et al. 2014; Lewis and Whitehead, 2011). Testing is more focused on soft aspects of games such as game mechanics and balancing issues instead of technical issues which are more or less addressed through utilizing third-party tools such as licensed game engines (Kasurinen and Smolander, 2014). These third-party tools usually require specific expertise to be utilized properly (Kasurinen and Laine, 2014). Due to games consisting of different components, a wide

variety of skills and specialists are also required to build the final product (Murphy-Hill et al., 2014). After a game has been completed it may receive for a short period fixes for bugs that went unnoticed during development or it may receive completely new additional content that is usually available for a fee whereas productivity applications receive maintenance, bug fixes, and support for a prolonged period (Kasurinen and Laine, 2014; Kasurinen, 2016).

2.1.4 Player participation thus far

2.1.4.1 In theory

Player participation, especially during the early stages in game development, is a fairly new trend in video game development. Many different constructs have emerged to describe it such as open development, community development, transparent development, performative development, community-informed development, crowd sharing (crowdsourcing and crowdfunding) (Thominet, 2018), participative development, extended prototyping (Gandolfi, 2018), participatory design, perpetual beta (Jacobs and Sihvonen, 2011), but all concepts address more or less the same phenomenon of players participating in a video game development process together with a game studio. There exists no established theory nor methods yet to guide open development (Thominet, 2018) and the amount of literature that has been so far accumulated is relatively small compared to more established research areas such as user participation in information system development.

The recent increase in the popularity of player participation can be partially attributed to increase in the number of independent game developers (indie developers or indies) due to democratization of game development tools and processes (Ruffino, 2013) as well as to digital distribution that enables circumventing the traditional publisher—developer relationship and distributing content that could not previously be sold in offline retail (Jöckel et al. 2008). Also, player participation could be viewed as a consequence of game developers getting frustrated of the current hegemony of game publishers, who prefer to fund only well-tried game genres and franchises to minimize risks and maximize profits and view any divergent, innovative attempts as risky investments. Through crowdfunding, developers are released of publishers' financial control and are free to pursue their innovative ideas, but simultaneously it changes players' role from the passive buyer into active prosumer investor role, who determine with their wallets which projects are

funded. (Planells, 2017) Players can facilitate development either through testing, where the role of players is to test closed or public game builds ranging from alpha to beta versions, or provide feedback and funding for the game development projects (Gandolfi, 2018). Projects that appear to have higher levels of quality or those game developers who possess wide social networks are more likely to get funded, but as the size of the project or received funding increases, the likelihood of project delay increases as well most likely due to increased project complexity and expectations from backers (Mollick, 2013).

There are potentially multiple distribution and crowdfunding platforms that enable game studios and players to come together but the digital game distributor Steam's Early Access platform and the crowdfunding platform Kickstarter are mentioned here because of their common appearance in the literature. Platforms may share some similarities and vary in other aspects. For example, Kickstarter and Early Access share the dimension of crowdfunding, but there may not exists early playable game version at the beginning of crowdfunding campaign in Kickstarter, whereas in the Steam's Early Access platform a playable version of the unfinished game must be made available for backers prior the launch of the campaign (Lin et al. 2018). In a similar vein, Steam's Early Access appears to attract mostly individual developers and small studios (Lin et al. 2018) and emphasizes more players' and developers' co-production by encouraging players' ability to influence the core features of a game whereas, in Kickstarter, players' contribution is more economical and involvement is variable (Gandolfi, 2018). However, what is common across these and other instances of crowdfunding is that projects can be abandoned and paying customers have to acknowledge the existence of a risk of never receiving the final product (Lin et al. 2018; Arafat, 2019).

Even though in the face of the risk of losing all invested time, money, and effort, some players are still willing to support the production of these games. Factors such as being able to test games before purchasing. playing for free, being involved in the development and able to improve games, the originality of game idea or its genre, or developers' credentials or presence of friends attract players and motivate to participate in the game development process. Usually, these players are committed individuals who share a passion towards certain games or genres, are willing to back up projects they enjoy, are more experienced regarding games and understand their underlying structure and mechanics, and may possess an analytical approach, which helps them to discover and trace bugs and be of more help for developers. (Gandolfi, 2018) Players are more likely to engage in constructive discussion in game-forums rather than leave reviews of the

incomplete game, they appear to be more forgiving of the game quality while it is still in development, and the length of early access period nor the update frequency of a game does not appear to correlate with the game approval rates suggesting that developers do not need to rush away from early access stage (Lin et al. 2018).

2.1.4.2 In practice

Holmström (2001) notes several fields where virtual player communities can benefit the development of a video game. A virtual community can function as a platform for product testing, where players share bug reports and feedback with other players and developers. The community itself enables multidirectional communication where players can provide input to other players' input and thus enhance the discussion. A properly nurtured player community may live long and engage players for a prolonged time making it possible to accumulate vast amounts of useful tacit knowledge. In addition, players and virtual communities can facilitate product diffusion, help in designing the product through feedback and suggestions, and evaluate the final product. The voluntary and anonymous nature of virtual community encourages players to provide honest and independent feedback to developers. Besides those issues, Aleem et al. (2016b) suggest that user involvement would contribute to user and customer retention and that online communities would function as an important source of idea generation and innovation.

In his attempt of clarifying the practical nature of open development, Thominet (2018) built a theoretical model of it that consists of three distinct main activities: distributing access, developing transparency, and collecting feedback. Distributing access mainly concerns the issue of providing access to the early game builds to players, which is accomplished through established digital distribution platforms, through developers' websites, or by utilizing both. In addition to the unfinished games, access can be granted to game modification or development tools enabling players to create in-game content by themselves, or even to the game source code itself. The chosen distribution methods in addition to other factors such as sliding pricing scale can be utilized as an access control method in planning and adjusting future community size. (Thominet, 2018) For example, by lowering the price of a game, developers can try to attract more players in hopes of gathering more feedback, and by increasing the price, developers can aim to utilize the platform as a financing model (Lin et al. 2018) or aim to restrict the size of a community (Thominet, 2018).

The idea behind developing transparency is either related to being more transparent of the current game state, game design choices, or game development work. Transparent game state is about informing players of the current game development progress as a favour in return for players being willing to back up the development of an incomplete game. Being transparent about the game design choices can function as a method of bringing "players' expectations for the game in line with developers' design goals" and includes describing and explaining design goals, their reasons, and limits of the scope of game design. Being transparent about the game development work can be used as a way to educate players about game development as well as transforming the development work itself into consumable content for players. (Thominet, 2018)

Developers' responsibilities in the activity of collecting feedback are mostly associated with establishing the channels for communication, engaging in conversation with players, and gathering and analysing feedback (Thominet, 2018). The bidirectional communication between players and developers is of most importance (Amaya et al. 2008; Lin et al. 2018) Players are likely to get emotionally involved in the development and build a sense of ownership of a game through their participation, to which developers ought to respond by being honest and transparent about the development, of its upsides and downsides, as well as involving players in the decision-making process (Lin et al. 2018). Also, responding to players feedback is vital as it makes players feel that their contribution is valued and reinforces the participation cycle (Amaya et al. 2008).

The types of feedback the players provide can vary between direct, communicative feedback, where both sides engage actively in a discussion through a specific medium such as forums, and non-direct, passive feedback, where developers gather the feedback of players' gaming habits through telemetry (Thominet, 2018). The former can be also understood as direct participatory design and the latter as a silent participatory design (Jacobs and Sihvonen, 2011). More innovative ways of collecting feedback should also be considered such as embedding a feedback widget within the game itself, which can be activated by a press of a button and can send feedback to developers without exiting the game (Thominet, 2017). In addition, there exists a possibility for players to engage in participatory feedback in a form of creating in-game content. However, this requires developers to provide players with necessary tools and create a system for uploading, reviewing, and approving created in-game content. (Thominet, 2018)

There are few issues that have so far emerged in the literature as important factors perceived by players or developers in determining the successfulness of player participation:

- Innovative game ideas and concepts (Arafat et al. 2019)
- Frequent updates and content (Arafat et al. 2019)
- Transparent and organized development plan (Arafat et al. 2019)
- Bidirectional communication between developers and player community (Amaya et al. 2008; Arafat et al. 2019; Gandolfi, 2018; Lin et al. 2018)
- Consistency between the initial project promises and final development (Gandolfi, 2018)
- Developers' track record (Arafat et al. 2019)
- Ability to make a difference as a player in development (Gandolfi, 2018)

However, there are also potential downsides to player participation. Jacobs and Sihvonen (2011) noted that in the case of freemium games, whose monetization methods are amongst other things built around selling players ways to overcome the built-in game mechanic of scarce game resources, players tend to engage in the game design from player's perspective rather than from game designer's, requesting things which are tedious to acquire in-game, thus eventually leading to unbalanced, less engaging and less challenging gameplay. Player community can function either as a positive or a negative factor attracting more players and enhancing development process or chasing away potential players due to toxic atmosphere (Gandolfi, 2018). Solely relying on players to provide all the required funding for the development is also a dangerous assumption (Lin et al. 2018). Finally, Taylor (2006) raises few concerns regarding participatory design in game development. One is that only a small portion of players' concerns are addressed, mostly those that correspond and contribute to developers' design goals, and participatory design becomes only a method to build gaming community and market the game prior to release. Second is that those who handle the communication with the community, such as community managers, lack the power to genuinely influence development and bring players' concerns to the table. Thirdly, participatory design is only utilized in form of user testing minimizing the effect to merely having players approving ready-made design choices. Finally, Taylor (2006) stresses the challenge of acquiring genuinely representative sample of the player population and tapping into those hidden players.

2.1.5 Summary of video game development

Video game development is about building and creating engaging and enjoyable products that are a combination of various diverse assets and created by multidisciplinary teams. The industry resembles other creative industries with its producers, publishers, distributors, retailers. It is characterized by the shroud of secrecy around upcoming games and by the tension between creativity and rationalization, which is a result from the conflict between the underlying business interests and interests for creating new and innovative gameplay experiences. Games are usually built in an iterative manner starting with concept and prototyping phase and eventually transitioning to production and testing phase before release. However, game development is not without its challenges from which some are shared with the software development industry. The common challenges are summarized in Table 1.

Table 1 Challenges in the game development industry

Challenges mostly exclusive to the game development industry	***Challenges in the game and software development industry***
Feature creep	Unrealistic scope
Cutting off features	Exaggerated optimism
Game design-related challenges	Scheduling challenges
Technological integration challenges	Requirements analysis challenges
Crunch time	Budgeting, quality, and management
Lack of documentation	
Communication problems in multidisciplinary teams	
Elaboration of subjective and abstract game requirements	
Player retention	

Recently, the democratization of game development tools and processes as well as the shift towards digital distribution has, at least partially, given a boost towards the player participation trend. The role of publishers is substituted by players, giving game studios more freedom to pursue innovative game concepts and thus departing from the rationalization, but it simultaneously pushes the financial risk of the game never being released onto players' shoulders. Little research has been done of player participation and it lacks still in theories and methods. However, activities such as distributing access to the game,

developing transparency of the development, and collecting feedback from players, have been identified.

Although video games are software of its kind, they are completely different to non-game software such as productivity applications. Use of games is voluntary and the primary purpose of games is to be played for enjoyment, whereas productivity applications are utilized to conduct work more efficiently and their use is more or less mandatory. The differences between games and productivity applications as well as the differences between their development characteristics are shown in Table 2 and Table 3.

Table 2 Differences between video games and productivity applications

Attribute	*Games*	*Productivity applications*
Purpose	Provide enjoyment	Enhance user's productivity
Time orientation	Maximize time spent on playing	Reduce the required time to reach a specific outcome
Composure	Technical, creative, narrative, graphical, and audio components	Technical and potentially graphical components
Quality	User experience and fulfilment of non-functional requirements	Fulfilment of functional, software requirements
Challenge orientation	Healthy amount required for enjoyment	Redundant, interfering factor
Competition	Other games and forms of enjoyment	Other similar productivity applications
Innovation orientation	Embraced, source of interface innovation	Potential risk and learning cost for users, well-tried solutions preferred

Table 3 Differences between the development processes of video games and productivity applications

Attribute	*Game development*	*Software development*
Requirements	Subjective and non-functional, based on abstract concepts, non-final	Functional, based on real-life concepts, absolute
Testing	Manual, focuses on soft aspects	Automated, focuses on the requirements
Planning	Less formal	Organized
Changes during development	Common, even during later stages	
Team composure	Various team members, different skills required	
After support	Short period of patches and bug fixes, and potentially additional downloadable content (DLC)	Updates, bug fixes and support for a prolonged period

2.2 User participation in information system development

User involvement and participation have been consistently studied in the information system development literature for over five decades. The first paper on the topic was published as early as 1959 (Abelein and Paech, 2015) and since the 1960s user involvement has been considered to be a critical factor in the development process of information systems (Barki and Hartwick, 1994a). The research activity of user involvement and participation has increased since the beginning and stretched out to cover more aspects and. But despite the long research history, no one has yet to come up with a clear solution how to implement user involvement and participation in practice (Abelein and Paech, 2015), indicating the complexity of the topic and its implementation challenges.

2.2.1 What is user participation?

2.2.1.1 Definition

In the beginning of user involvement and user participation research, there was no distinct definition for the constructs, which were used interchangeably in the literature. Lack of solid conceptual foundation for user participation and user involvement plagued studies and undermined understanding and implications of prior research (Ives and Olson, 1984). Definitions of the time were drawn and adapted from other disciplines such as from organizational behaviour research and its participative decision-making and planned organizational change theories, and were concise and pragmatic: user involvement is “participation in the system development process by representatives of the target user group” (Ives and Olson, 1984).

In 1989, Barki and Hartwick (1989) proposed a separate definition for user involvement and user participation constructs, which since appears to have become the most common way to distinguish them from each other in information system development field. The definition of user participation was based on concrete actions and defined as “a set of behaviours or activities performed by users in the system development process”, whereas the definition of user involvement was built to be consistent with the involvement constructs of psychology, marketing, and organizational behaviour and defined as “a subjective psychological state reflecting the importance and personal relevance of a system to the user”. Hence, “a user is involved when he or she considers a system to be both important and personally relevant” (Barki and Hartwick, 1989).

Similarly, Kappelman and McLean (1991), as well as Hwang and Thorn (1999), defined user involvement as a "need-based mental or psychological state of system users … towards the development process and its product" and user participation as "observable behaviour of system users in the information system development process". The construct 'user engagement' is used when referring to both users' participative (behavioural) and involving (attitudinal) relationships towards the system development and its end-product. (Kappelman and McLean, 1991; Hwang and Thorn, 1999) Therefore, we can also talk about the users' behavioural engagement and attitudinal engagement in the information system development (Kappelman and McLean, 1991).

Besides to user involvement and participation, the concepts 'user' and 'information system development' (ISD) require definition. Users are those who either utilize the outputs generated by an information system (primary users) or those who provide system input or maintain a system (secondary users) (Ives and Olson, 1984). Users can be categorized in terms of various attributes such as stakeholder group membership, knowledge or skill related to IT, managerial ranking etc. (Markus and Mao, 2004). Contribution from each kind of user may be required at different stages in ISD (Cavaye, 1995). Markus and Mao (2004) differentiate between the concepts of stakeholders and participants. "Stakeholders are those who are likely to be affected by a solution" (i.e. system) "and who are therefore logical candidates for participating in solution development and implementation" whereas participants are a selected group of stakeholders allowed to participate in the development and implementation activities. Various change agents are also present in the user participation processes facilitating participation opportunities for stakeholders. Traditionally this task has been assumed to be part of IS professionals' territory, but management, HR professionals, or external consultants may also be involved in enabling change. (Markus and Mao, 2004)

ISD is defined per Cavaye (1995) to describe the whole development process from the initial problem definition to system analysis, design, development and finally implementation. Even though ISD process definitions may vary in development stages and naming conventions, it essentially boils down to the same process (Cavaye, 1995). Potential maintenance development is also taken into account to complement ISD process even though it is not present in Cavaye's (1995) definition but it is still an essential part of the development life cycle especially in present service models and agile development approaches.

2.2.1.2 Attributes

User participation is not merely a static method to be employed in ISD, but its several attributes can help categorizing and differentiating various participations from each other. Cavaye (1995) briefly presents six attributes: type, degree, content, extent, formality, and influence of participation. Type of participation refers to either direct participation, in which all groups affected by the system are involved, or indirect participation, in which user representation is utilized (Ives and Olson, 1984; Cavaye, 1995). Degree of participation denotes that users may have different levels of responsibility in the development at, such as acting in advisory roles, possessing sign-off responsibilities, being more deeply involved in designing the system or even being responsible for the whole system development (Ives and Olson, 1984; Cavaye, 1995). Another approach to degrees of participation is to consider it as a depth of involvement. Damodaran (1996) describes these three forms of user involvement as: informative, where "users provide or receive information", consultative, where "users comment on a predefined service or range of facilities", and participative, where "users influence decisions relating to a whole system".

Content of participation distinguishes that there are other aspects besides the technical one in information system design, such as social or human aspect, and that users may be involved in any of them. Extent of participation refers to different ISD project phases users can be part of. Formality of participation refers to the degree of formality in organizing the user participation process and influence of participation indicates the actual effect users can have on the ISD project through their participation. (Cavaye, 1995) Users' influencing ability may vary between having no ability to influence the ISD process to being in complete control over the direction and outcomes of the development process. (Ives and Olson, 1984; Cavaye, 1995).

2.2.1.3 Approaches

Whereas attributes describe the elements of user participation, approaches determine how user participation is viewed and implemented in practice. Cavaye (1995) differentiates between two philosophies of opposing ends: functionalist and neo-humanist approaches. In the functionalist philosophy user participation is considered a tool to facilitate developing functionally correct systems by eliciting better requirements, overcoming user resistance, and validating design choices. In cases where user participation is deemed to provide no benefits, it may not be utilized at all. Contrary in the neo-humanist philosophy,

the focus of ISD shifts from system to people and potential users. User participation aims to improve participants understanding of the system and empower users by giving them a chance to affect their working life and environment. (Cavaye, 1995)

On the other hand, Kujala (2008) takes different yet a bit similar approach, and summarizes three motives for user involvement. In addition to practical development-oriented motives that focus on delivering functionally working, correct and efficient system; and democratic motives that seek to enable workers to influence their work and accumulate their expertise; Kujala (2008) identifies a third category – organizational motives – that aim at facilitating diffusion, deployment, and utilization of the new information system.

Kujala (2003) suggests user-centred design, participatory design, contextual design, and ethnography to be the four main approaches to user involvement which all explain why and how to involve users. User-centred design focuses on usable and useful products through task analysis and prototyping, participatory design is of Scandinavian origin and focuses on democratic participation through workshops and prototyping, ethnography emphasizes social aspects of work settings through observation and video-analysis, and contextual design focuses on users' work and work context through observation and inquiry on the spot. The main differences between these approaches relate to the activity of users' roles, users' decision influencing ability, and participation extent in development work (Kujala, 2008).

2.2.1.4 Models

To better understand how user participation and involvement can be viewed in the ISD context, few models presented on the topic are reviewed here. Ives and Olson (1984) built a descriptive model of user involvement based on other descriptive models presented before and enhanced it with research on participative decision-making and planned organizational change. In their model, illustrated in Figure 1, user involvement is composed of two factors: involvement roles of the participants and development characteristics of the underlying system. Primary and secondary users combined with users' willingness or tendency to become involved, composed of available incentives and users' competence, dictate involvement roles of the participants, whereas the type of the system and its related probability to benefit from user involvement as well as the stage of the development process define the development characteristics of the system. User involvement is then considered to contribute directly to system quality and system acceptance and indirectly through users' cognitive and motivational factors. (Ives and Olson, 1984)

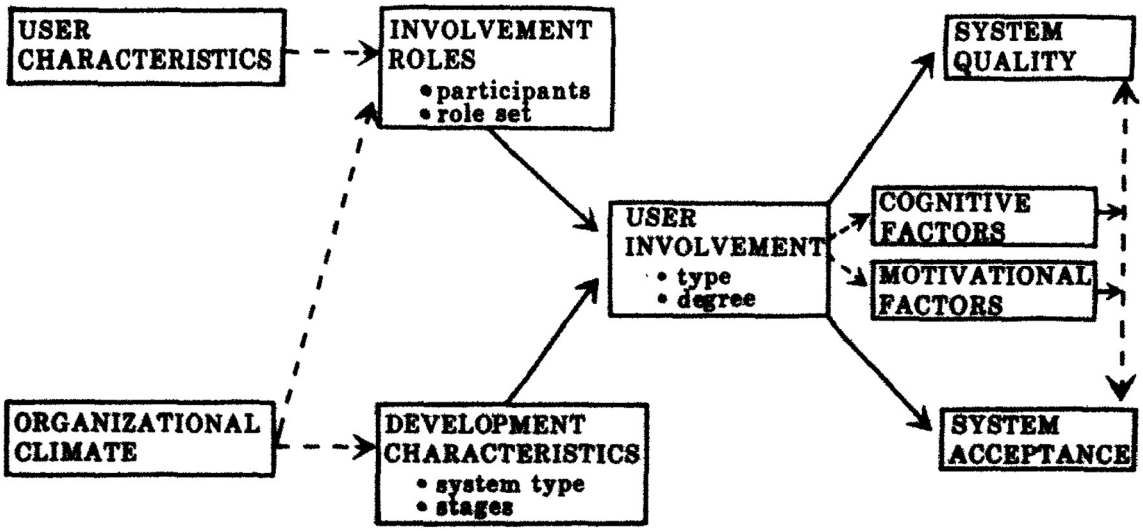

Figure 1 A model of user involvement (Ives and Olson, 1984)

Cavaye (1995) presents a descriptive model of user participation—system success link which recognizes that various contingencies from participation context may inhibit or benefit user participation. Intervening variables may affect participation—success link, may cancel it, and user participation process in itself is a dynamic process that can vary in quality and effectiveness. The model is illustrated in Figure 2.

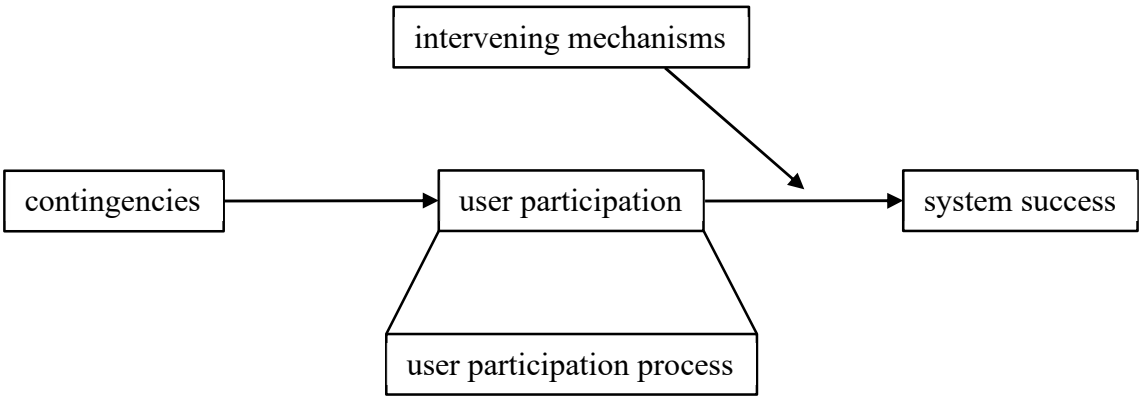

Figure 2 A model of user participation (Cavaye, 1995)

The contingencies are categorized into organizational variables, project-related factors, and user-related factors and mostly relate to available resources and top management commitment (organizational), project and product complexity as well as the magnitude of change expected to be caused by the system (project-related), and users' competence and mental readiness to participate (user-related). The intervening variables refer to users' perceived control over the development, desired versus actual level of participation,

perceived importance and relevance of the system, and potential prior involvement in similar settings. Any deviation from expected levels of these variables may cause divergence in the expected system success. For example, increasing perceived control may lead to increased system acceptance. (Cavaye, 1995) The user participation process in itself is according to Cavaye (1995) considered to be an interactive relationship between users and developers where people's different backgrounds and perspectives affect the interaction process, power relationships determine ability to influence the process and outcomes, and communication is used to facilitate understanding and build agreements. The relationship itself between users and developers may fluctuate over time, which leads to the dynamic nature of the process and renders the process outcome unpredictable.

Markus and Mao (2004) built a theoretical framework to address logical gaps found in the IS participation literature and gaps that have emerged through the evolvement of IS practices since the foundation of IS participation theory. The framework, illustrated in the Figure 3, depicts a structure of the relevant actors present in the participation process, the participation activities, in which the actors engage with, and the relation these two factors have with the system creation outcomes.

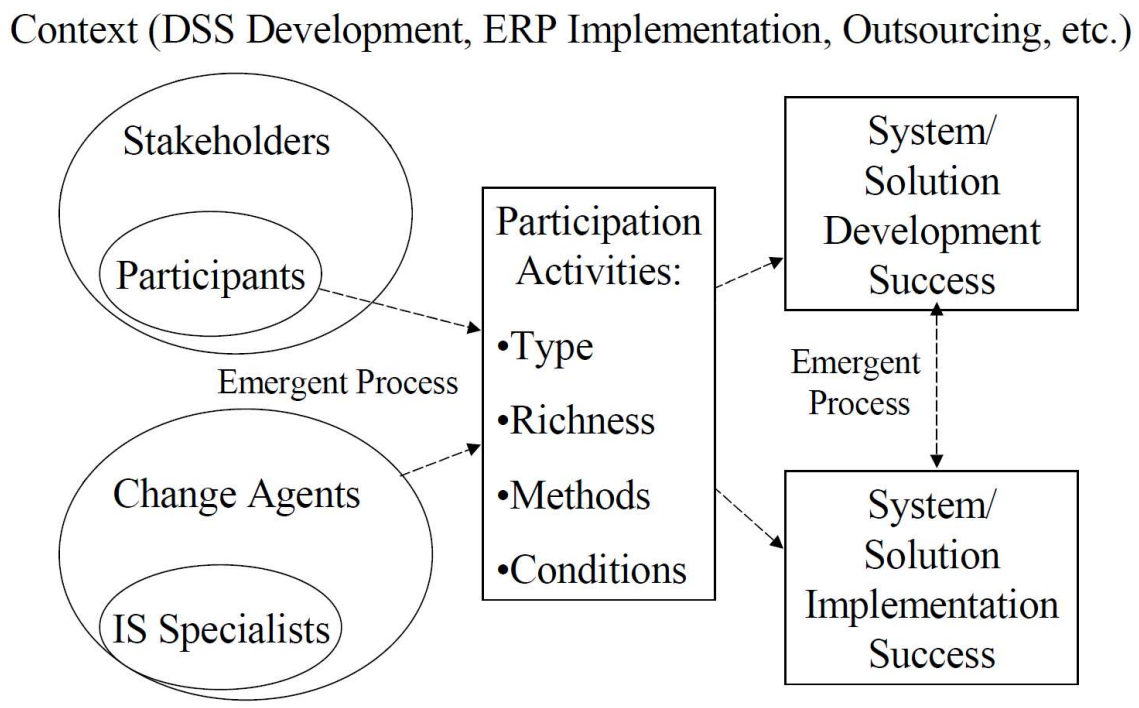

Figure 3 Elements of user participation (Markus and Mao, 2004)

Contrary to the traditional causal IS participation theory where participation is considered to be a necessary and a sufficient factor in achieving system success yet moderated by contingent factors, Markus and Mao (2004) adopt an emergent approach where the causal processes between actors, activities, and outcomes are considered to be enabling and constraining rather than necessary or sufficient in achieving system creation outcomes. In the theoretical framework, actors (participants, change agents, IS specialists) engage in participation activities, that can be characterized in terms of type, richness, methods, and conditions, which eventually contribute either to system development success or system implementation success.

2.2.2 Why user engagement?

In this section, the benefits of employing user involvement and participation in information system development are reviewed. There have been many attempts and studies in information system literature seeking to uncover the benefits of involving users and to establish a connection between user involvement and participation and system success. Results have been varying and occasionally contradicting, which has led to several meta-studies and systematic literature reviews aiming to uncover and summarize the true nature behind user involvement and participation. Therefore, instead of covering potentially hundreds of individual studies, the focus in this section will be mainly the released meta-studies and systematic literature reviews. Also, the potential challenges and downsides of involving users are covered in this section.

2.2.2.1 User engagement and system success

Ives and Olson (1984) conducted a research review of 22 studies examining the connection between user involvement and indicators of system success published between 1959 and 1981. However, they could not reliably demonstrate a positive link between user involvement and various measures of system success. Eight studies (36 %) presented positive results, seven studies (32 %) presented mixed results and the final seven (32 %) were either negative or insignificant. Instead, it was revealed that “much of the existing research is poorly grounded in theory and methodologically flawed” and more rigorous conceptual foundation, measurement, and methodology are needed for future research.

Pettingell et al. (1988) conducted a meta-analysis on 15 studies on user involvement released between 1974 and 1986 and concluded that user involvement is strongly related

with system success in terms of attitudinal and behavioural measures. They also provide evidence that involving users in the definition and design phases of a project is much more strongly linked with system success than involvement in feasibility or implementation phases.

Cavaye (1995) reviewed 19 empirical studies between 1982 and 1992 concluding that "findings are still just as inconclusive as those of earlier studies" mainly referring to the study of Ives and Olson (1984). Seven studies (37 %) showed positive results between user participation and system success, nine studies (47 %) provided inconclusive results, and the final three studies (16 %) presented negative results. Extensive use of quantitative methods that limit exploring the context and finding new connections between variables, inconsistent and mixed-use of constructs, difficultness of measuring system success, diversity in research instruments, and varying significance of participation in development method are potential reasons to explain the inconclusiveness of the results.

McKeen et al. (1994) confirmed the existence of a relationship between user participation and user satisfaction which is moderated by four contingency factors: task complexity, system complexity, user influence, and user-developer communication. Task and system complexity were shown to be mediating the relationship between user participation and user satisfaction: the higher system or task complexity, the higher user satisfaction would result from user participation. However, the effect of user influence and user—developer communication on user satisfaction were shown to be independent of the degree of user participation and higher emphasis on either of the factors translated into higher user satisfaction regardless of user participation.

Hwang and Thorn (1999) reviewed 25 studies released between 1974 and 1996 in an attempt to resolve inconsistent findings in prior studies. The effects of user involvement and participation were reflected against six system success variables. The meta-analysis confirmed that user involvement and participation are beneficial and positively correlate with system success. Participation was moderately and positively correlated with system quality, (system) use, user satisfaction, and organizational impact. However, the connection between participation and individual performance could not be established. Instead, the association of user involvement with system success was even stronger than that of user participation.

Kujala (2003) reviewed three streams of research of field studies, qualitative, and quantitative research published between 1986 and 2000 to build understanding on the benefits and problems of various user involvement approaches during the early

development process activities, especially in requirements gathering. The review confirmed the usefulness of involving users and its positive relationship with system success and user satisfaction.

He and King (2008) reviewed 82 studies on the impact of user participation on information system development and concluded that "user participation is somewhat beneficial in ISD". It has positive effects on behavioural outcomes (system acceptance) and positive yet weaker effects on productivity outcomes (system quality). However, the results of their study suggest that besides user participation other underlying factors have to be taken into account when evaluating users' influence on system success because mere user participation does not necessarily enhance ISD outcomes. It can be one of the potential strategies to reach successful system but in that case, user participation should be adjusted to reach the desired outcomes. For example, He and King (2008) mention that if aiming for system acceptance, "user participation should be designed to induce more psychological involvement among potential users" and if the goal is to enhance the productivity of the development process, user participation should aim to provide developers with the required domain knowledge.

Harris and Weistroffer (2009) reviewed 28 empirical studies published between 1996 and 2009 to solve the contradicting nature of "the importance of user involvement to successful system development". They confirmed the positive relationship between user involvement and system success and pondered that evolution of user engagement practices and modern complex systems, which benefit more from user engagement than earlier systems, could be possible explanations why previous meta-analyses have presented contradicting results.

Bano and Zowghi (2013) conducted a systematic literature review on user involvement and its effect on system success in software development reviewing 87 studies published between 1980 and 2012. 59 studies (68 %) showed a positive relationship, 7 (8 %) suggests a negative relationship, and the rest 21 (24 %) are uncertain. Thus overall, the relationship seems to be positive, but solely user involvement cannot be held accountable for system success since system development is a complex process with many factors influencing it. Bano and Zowghi (2013) also conclude that forming a meta-analysis of user involvement on system success is also difficult because of the huge variance in factors presented in studies.

Abelein and Pacch (2015) conducted a systematic mapping study on the effects of user involvement and participation on system success. They analysed 58 papers and

correlations from 86 studies and concluded that despite slight inconsistencies in results among studies, the number of positive correlations between system success and various aspects of user involvement and participation outweighs the inconsistencies. Out of the 14 studies that were classified as presenting a negative correlation, all except one was published before 2003, more than half of the studies presented a low correlation (<0,2), and the presented negative correlations were mostly between contextual factors and user involvement and participation, and therefore do not undermine the positive relationship between system success and user involvement and participation. Overall, it was confirmed that "user participation and involvement have a positive effect on user satisfaction and system use" and that those who are engaged in the software development process are more satisfied with the system and use it more frequently. Also, a positive connection was discovered between ease of use and user satisfaction. Summary of the previous meta-studies and literature reviews on user involvement and participation and their relationship on system success is presented in Table 4.

So far, the concepts of user involvement and user participation and their relation to system success have been considered separately and little value has been given to their internal relationship. Kappelman and McLean (1991) investigated the relationship of each construct towards system success and found out that the behavioural—attitudinal theory of IS success is better at describing the relationship of user participation with IS success than the behavioural theory is. In the latter, user participation is considered to be a direct antecedent of IS success whereas in the former user involvement is a direct consequence of user participation and mediates the user participation IS success relationship. Therefore, according to Kappelman and McLean (1991), user participation would cause user involvement, which may be a more important factor than user participation when trying to understand information system success.

Hartwick and Barki (1994) provide similar results in their study confirming user participation influencing system use through user involvement (as in system experienced as important and personally relevant) and attitude towards the system, making user involvement more important factor in explaining system use than user participation. However, it was noted that the participation—involvement relationship does not work the other way around as in initial involvement and attitude influencing the levels of user participation. Based on these, Hartwick and Barki (1994) highlight the importance of identifying other antecedents of user involvement besides user participation and list some potential candidates:

- personality characteristics
    - need for achievement
    - locus of control
    - dominance
- experience
    - education
    - amount of experience with IS
    - quality of experience with IS
- organizational status
    - organizational function
    - hierarchical level
- organizational culture

Hartwick and Barki (1994) identified that overall responsibility holds the greatest impact in forming involvement and attitude towards the system, meaning that handing out responsibility to users in system development related tasks, such as being in charge of a team or being responsible for a specific ISD task, would generate higher levels of involvement. When addressing multiple people or groups of individuals different tasks could be assigned to different people or single tasks to groups of individuals. Among voluntary users, the overall responsibility appeared to be strongly associated with users' attitudes, norms, intentions, and use, whereas among the mandatory users, user participation and involvement appeared to be unimportant and subjective norms (e.g. how users perceived they were expected to act) were most important in determining users' intentions.

Table 4 Relationship of user involvement and participation in system development success

Research	*Studies reviewed (between)*	*Focus on*	*Results*	*System success measures used*
Ives and Olson (1984)	22 (1959—1981)	Studies on user involvement	36 % (8) positive 32 % (7) mixed 32 % (7) negative/non-significant	System quality System usage User behaviour and attitudes Information satisfaction
Pettingell et al. (1986)	15 (1974—1986)	Studies on user involvement	Strongly positively linked with system success	Attitudinal Behavioural
Cavaye (1995)	19 (1982—1992)	Empirical studies on user participation	37 % (7) positive 47 % (9) inconclusive 16 % (3) negative	User information satisfaction System use
McKeen et al. (1994)	14 (1973—1991)	Empirical studies on user participation and involvement	57 % (8) positive 21 % (3) inconclusive 14 % (2) not reported 7 % (1) non-significant	User satisfaction System quality Successful implementation System usage
Hwang and Thorn (1999)	25 (1974—1996)	User participation and involvement	Positive correlation with system success	System quality and use Information quality User satisfaction Individual impact Organisational impact
Kujala (2003)	## (1986—2000)	User involvement in early stages of development	Positive relationship with system success and user satisfaction	System usage Personal relevance User satisfaction System acceptance
He and King (2008)	82 (1977—2006)	Empirical studies on user participation	"Minimally-to-moderately beneficial to ISD" with stronger benefit on attitudinal/behavioural outcomes than on productivity outcomes	User satisfaction Use intention System use Individual impact Team performance Organisational impact Project quality, success
Harris and Weistroffer (2009)	28 (1996—2009)	Empirical studies on user involvement	Positive correlation with system success	User satisfaction, impact System use, output, benefits, impact Software success Perspectives of social actors Impact on job Perceived data quality, usefulness, benefits, ownership Project completion, performance Process satisfaction Intention to use
Bano and Zowghi (2013)	87 (1980—2012)	Empirical studies on user involvement	86 % (59) positive 8 % (7) negative 24 % (21) uncertain	Not explicitly defined
Abelein and Pacch (2015)	58 (1984—2011)	Studies on user participation and involvement	Positive correlation with system success variables	User satisfaction Ease of use System use. quality Data quality Project in time and budget

2.2.2.2 Benefits

Fairly often cited benefits are related to gathering better and more accurate system and user requirements that lead to better quality systems (Damodaran, 1996; Kujala, 2003; Kujala et al. 2005; McGill and Klobas, 2008), avoiding unnecessary system features that could be costly to develop (Damodaran, 1996; Kujala, 2003) or facilitating acceptance of a new system (Damodaran, 1996; Kujala, 2003) and overcoming user resistance (Cavaye, 1995). These benefits are closely related to the functionalist approach that aims to deliver functionally efficient and correct systems (see Cavaye, 1995). Especially the case appears to be with user requirements that according to Kujala et al. (2005) are associated with project success and a lower ratio of requirements engineering costs of the total project costs. Other cited benefits are increased user satisfaction and individual impact leading to end-users being more involved with the developed system (McGill and Klobas, 2008) and to increased system usage either directly through user involvement or indirectly through increased user satisfaction (Baroudi et al. 1986), increased understanding of the system due to more efficient system usage (Damodaran, 1996) and potentially more democratic working environment through increased participation in decision-making (Damodaran, 1996; Kujala, 2003).

However, it is worth noting that many of the benefits may not necessarily be a direct consequence of user involvement or participation. For example, McGill and Klobas (2008) note that user satisfaction and individual impacts were indirect impacts through perceived system quality and Kujala (2003) notes that better user requirements were the reason for the benefits caused by user involvement. Also, Markus and Mao (2004) note that the basic assumptions of user participation inducing psychological buy-in among users may not be as relevant nowadays as it might have been during the early days of IS theory due to increased sizes and scopes of IS projects encompassing more users and user types. The theory of user participation leading to increased system success through increased psychological buy-in among users does not take into account the case of those who do not get the opportunity to participate in IS development. Likewise, Markus and Mao (2004) criticize the assumption of user participation leading to higher system quality through better user requirements stating that there's no guarantee of those requirements being incorporated in the system.

2.2.2.3 Challenges

The challenges in user involvement and participation found in the literature can be roughly categorized into user, developer, and process-related challenges, and mostly appear during interaction between these parties. Most of the challenges seem to appear in the relationship between users and developers (Heinbokel et al. 1996; Kujala, 2003). Users can be unpredictable in the process (Heinbokel et al. 1996), they may introduce new ideas to development or demand changes too late in the development process when it is no more feasible to implement (Heinbokel et al. 1996; Kujala, 2003) or they may be outright unwilling to participate due to fear of losing their job or having their working conditions weakened because of the new system (Heinbokel et al. 1996). There is also an issue of users' lack of understanding of the design and development activities, process, and its constraints and need for education and training (Damodaran, 1996; Wilson et al. 1997; Kujala, 2003) that could also contribute to previously mentioned challenges.

Because the user—developer relationship is bidirectional, some challenges can affect both sides. User involvement requires engagement from both users' and developers' parties (Kujala, 2003) and without it, the foundation for any benefits crumbles down. In this engagement, communication is a vital element in keeping the engagement alive but it is also prone to its problems. Unclear communication structures cause problems on both sides, such as users not knowing who to contact and how for their ideas or designers failing to get access to users to organize user involvement activities (Wilson et al. 1997). Problems may occur within the communication itself, when either of the parties fails to recognize or transmit necessary information, receiving end fails to interpret the information correctly, or information overload causes parties to fail to prioritize messages correctly (Gallivan and Keil, 2003). Maleej et al. (2009) suggests a more context-aware approach in providing input to mitigate chances of communication gaps between users and developers.

In terms of user involvement and participation process, the process itself is time-consuming and can potentially yield huge amounts of raw data to be processed (Kujala, 2003). Poor organization of user involvement activities and lack of information dissemination regarding the process and activities contribute to the failure of user involvement. Limited resources on user involvement may force designers to collect data from a handful of users and use user representation instead of exhaustive user involvement. (Wilson et al. 1997) Selecting possible user representatives can be challenging (Damodaran, 1996)

and uneven representation of the user population may lead to emphasizing certain areas of interests and ignoring others (Wilson et al. 1997). However, having users participate in issues in which they don't possess the expertise, could be useless or even harmful for the project (Doll and Deng, 2001).

Axtell et al. (1997) observed problems emerging in a situation where user participation was attempted to integrate into an already established system development process in a case organization. Similar problems to previous ones were mentioned, such as challenges in selecting users, limited resources, and additional training required for both users and developers, but in addition to those challenges emerged due to

- Lack of contact between users and developers (due to established working procedures formed ostensible walls between people or development phase inhibited contact)
- Users were given too much independence to operate, but not given clear goals, boundaries to work within, nor guidance or support, yet still expected to be knowledgeable of issues outside of their domain
- Lack of formal ownership of user participation process and varying degrees of commitment towards user participation among personnel
- Implementing user participation at untested scale and difficulties in getting access to users in widely-distributed user population in large-scale development

Finally, there's always a chance for internal strives in parties, such as gaining support from other users (Damodaran, 1996) or outright conflicts between different user groups with divergent interests. (Kujala, 2003) Conflict is defined as a state between two or more interdependent parties who share divergent interests, opinions, or goals, and whose actions would interfere with other parties from reaching their goals. Even though conflicts are usually considered bad and undesired situations, they do not inherently possess any moral status, but can lead to various outcomes, even positives, depending on how they are managed. (Barki and Hartwick, 1994b) In their study, Barki and Hartwick (1994b) found that user participation may lead to disagreements and conflicts. Disagreements can be born indirectly through users' influence exerted to further their interests which conflict with other users' interests, or directly when users come across with other users carrying a divergent, conflicting agenda. Disagreements may eventually escalate into conflicts, representing an indirect path from user participation to conflict, or user participation can directly lead to conflicts, for example, if people with previous unsolved confrontations

are brought together. Ability to influence was found to be the only factor related to satisfactory conflict resolution, suggesting that those with decision power can turn the tables for their favour. This could also explain why influence was negatively related to conflict: people with the ability to influence decision making are less likely to experience conflict as they can get what they want. (Barki and Hartwick, 1994b)

2.2.3 User engagement in practice

Until now various aspects of engaging users in information system development have been looked into. User involvement and participation have been defined and their characteristics, models, benefits and challenges have been examined. Having users participate in system development appears to be most suitable when task and system complexity are high. Uncertainty and ambiguousness require input from users and collaboration between users and system providers to be reduced and resolved. In contrast, when complexity is low, the need for user participation is likely to be minimal as well. (McKeen et al. 1994) Next, we will have a look at how user involvement and participation are applied in practice and what kind of potential activities can be executed to enable and facilitate user engagement.

2.2.3.1 Activities

Abelein and Pacch (2015) identified in their systematic mapping study several practices that enable and enhance user participation and categorized them by different software development project activities: project initialization, software specifications and requirements engineering, software design and implementation, software verification and validation, and software evolution. Because of its intuitiveness and practicality, the same structure will be used here.

The amount of activities users should participate in is dependent on the level of need for user participation. ISD projects with higher task and system complexity are likely to require more participation from users' part. However, users are not able to solve all the problems by themselves but require assistance from developers as well. Users are better at explaining what the system is supposed to do whereas developers are experts at deciding how the system should accomplish it. (McKeen and Guimaraes, 1997) However, involving all users is not practical especially in larger ISD projects and some form of user representation is usually employed. Still, users should be able to effectively participate

and genuinely influence both the design process and outcomes instead of merely rubber-stamping them. (Damodaran, 1996)

In the project initialization stage, the focus is on building a foundation which enables and supports productive user participation. Main practices revolve around identifying right users and assigning their roles, setting up communication structures which enable a seamless flow of information between stakeholders, setting up project management which manages project direction and vision based on users' needs, and setting up the project environment where users and developers can come together and collaborate (Abelein and Pacch, 2015). In addition to these, users should be included in feasibility analysis (Harris and Weistroffer, 2009), in the creation of project definition and approving the cost justification of the project, developing and approving project management schedules and progress reports, as well as selecting a potential user-liaison (McKeen and Guimaraes, 1997).

In the software specification and requirements engineering stage, the focus is on understanding users and their current situation, as well as extracting user and business needs from users to convert them into system requirements (Abelein and Pacch, 2015). These requirements should also get approval from users (McKeen and Guimaraes, 1997). Requirements can be formed for example based on issues users face and report to customer support, users can be asked to provide feedback on potential requirements formed by developers, or users can be involved to provide their suggestion for future requirements (Kabbedijk et al. 2009).

In the software design and implementation, the focus is on taking a user-oriented approach by engaging users in various tasks such as requirements prioritization, designing and experimenting with alternative solutions (Abelein and Pacch, 2015), helping to define the input and output forms, physical controls and security procedures, and helping to design user interfaces and report formats (McKeen and Guimaraes, 1997; Harris and Weistroffer, 2009). Requirements can be analysed based on their complexity, ambiguity, and completeness. Complex requirements may be worthwhile to divide into smaller requirements, ambiguous requirements need more clarification, and incomplete requirements need to be completed with additional information. Also, requirements that are similar to each other can be either merged, grouped, or linked together. Finally, feasibility analysis can be performed where out-of-scope requirements are rejected and system enhancing and complementing requirements are accepted. (Kabbedijk et al. 2009) Besides, it is important to have users review, validate and provide feedback on proposed solutions,

so that the design and implementation of the system prototype can be reiterated and incrementally improved (Abelein and Pacch, 2015).

In the software verification and validation, the focus is on the acceptance test of the system and testing with potential other users that were not taking part in the development process (Abelein and Pacch, 2015). Users should be included in the installation phase of the system, as its one of the basic activities that increase user satisfaction (McKeen and Guimaraes, 1997; Harris and Weistroffer, 2009). In the software evolution, the focus is on keeping the feedback channels open and encouraging users to voice problems, proposals, ideas, feedback, new features, changes, etc. as well as keeping users informed of various issues regarding the system such as future updates (Abelein and Pacch, 2015).

There also exist other ways to categorize various user participation activities. Barki and Hartwick (1994a) propose activity groupings based on user—IS staff relationships activities, project management and leadership activities, daily hands-on activities and later Bark and Hartwick (2001) proposed communication-related activities as a fourth grouping. Markus and Mao (2004) on the other hand propose that participation activities could be categorized by their type, richness, methods, and conditions. Activity type refers to either system design, system implementation, or project management participation activities, in which users may participate in. Each of the activity types contribute to their respective system outcome such as participating in system design activities like system requirements determination contributes to system development success whereas participation in system implementation activities such as training contributes to implementation success. Richness refers to the quality of an activity and can be more easily understood as the extent and depth of the experience provided to the participants. Methods refer to the techniques change agents utilize to engage participants in the participation activity. Conditions refer to factors either facilitating or constraining participation activities, such as the location of participation or available resources for participation, that change agents can attempt to manipulate to affect the effectiveness of participation activities. (Markus and Mao, 2004)

2.2.3.2 Best practices

There are several best practices, recommendations and ideas that have emerged in the literature concerning user involvement and participation, that are briefly mentioned here. Instead of blindly forcing users to participate and be involved in system development as much as possible, Harris and Weistroffer (2009) suggest that there exists certain optimal

level for user involvement after which additional involvement does not increase chances of success, but instead may be counterproductive and waste of resources. Users should be allowed to voice their opinions and make choices from predetermined alternatives to have the greatest effect on system success and to have users feel valued included, and possessing control over the outcome. Kujala (2008) notes that even though a potential user may be reluctant to contribute in the context of new product development, developers ought to nevertheless seek to collaborate with user representatives to gather information and feedback and to understand their needs and values. Especially users with functional expertise should particularly be included in the development process or otherwise, they are likely to develop negative attitudes toward the system if they feel they are ignored. Users could be motivated by explaining the benefits of user involvement and how it would benefit their work but it must also be acknowledged that motivation varies among people and positions and that arguments for user involvement should be tailored towards the people that are sought to be convinced (Wilson et al. 1997). Also having a people-oriented management is a must to ease communication with users. (Harris and Weistroffer, 2009)

In terms of the user involvement and participation process, informative user involvement is suggested in the context of new product development (Kujala, 2008) and better and more cost-effective methods for involving users are needed as well as the roles of users and designers have to be carefully considered (Kujala, 2003). A champion of user involvement should be chosen to promote the cause and potential benefits as well as help organizing the activities and facilitate communication between users and designers. In cases of user representation, a truly representative population should be gathered from a large pool of affected stakeholders, consisting of users with varying levels of seniority, expertise and service conditions. (Wilson et al. 1997) alongside with relevant external stakeholders to enhance the possibilities of participation activities reaching system development and implementation success (Markus and Mao, 2004). Sometimes it is rather clear user expectations than user involvement during the early stages of a project that can increase chances of success together with transparent bottom-up decision-making approaches (Xin and Myron, 2014). If the target of an ISD project is a collaborative work system, then the design process ought to be a collaborative activity itself (Doll and Deng, 2001). Kappelman and McLean (1991) note that users should be engaged during the installation phase of the system as it is strongly related to user satisfaction.

2.2.3.3 Measuring system success

Defining or measuring system success is no trivial task but user acceptance and user satisfaction are the most referred criteria (Bano and Zowghi, 2013) and often used synonymously in the literature (Harris and Weistroffer, 2009). User satisfaction is defined by Harris and Weistroffer (2009) as "the recipient's response to the use of the output of an information system". According to Abelein and Pacch (2015), there are several terms used to denote user satisfaction which mainly refer to the level of contentment the system and its mechanics and output delivers to the user as well as user's perceived usefulness of the system. Despite user satisfaction being a common mean to measure system success, having satisfied users does not necessarily indicate that a system is successful, especially if other measures are simultaneously employed. The underlying type of a system affects the applicability of system success measures. However, if several systems are equal in terms of functionality and performance, the one with the most satisfied users could be considered the most successful. (Harris and Weistroffer, 2009) Other frequently cited system success measurements are system use, ease of use, quality of the system, quality of data the system provides, and quality and successfulness of the project in terms of time, money, performance, satisfaction, completion, and implementation (Abelein and Pacch, 2015)

Despite various measures used in the industry, there has been at least one attempt to build a general understanding of IS success measurement. DeLone and McLean (1992) reviewed a large number of studies to build an understanding of metrics used to measure information system success. They found out that "there are nearly as many measures as there are studies" but managed to summarize the measures into six main categories: system quality, information quality, use, user satisfaction, individual impact, and organizational impact. Based on these categories, they built an IS success model of interdependent IS success dimensions, where system quality and information quality as independent factors contribute to system use and user satisfaction; where system use and user satisfaction affect each other either positively or negatively and contribute to individual impact, and where individual impact ultimately constructs the organizational impact. (DeLone and McLean, 1992)

The IS success model was constructed of three parts: system creation, system use, and system use outcomes. Later, DeLone and McLean (2002) revised the model based on the support and critique that had emerged in the literature. Service quality was included

as the third factor with system quality and information quality to reflect the shift in the role of IS in providing support and services to end-users. System use was reworked into 'intention to use' as an attempt to counter the multidimensional aspects of system use measurement. Individual impact and organizational impact were fused as net benefits to better describe the expected outcome and to allow the consideration of other stakeholders besides users (individuals) and organization. (DeLone and McLean, 2002) The IS success model is illustrated in Figure 4.

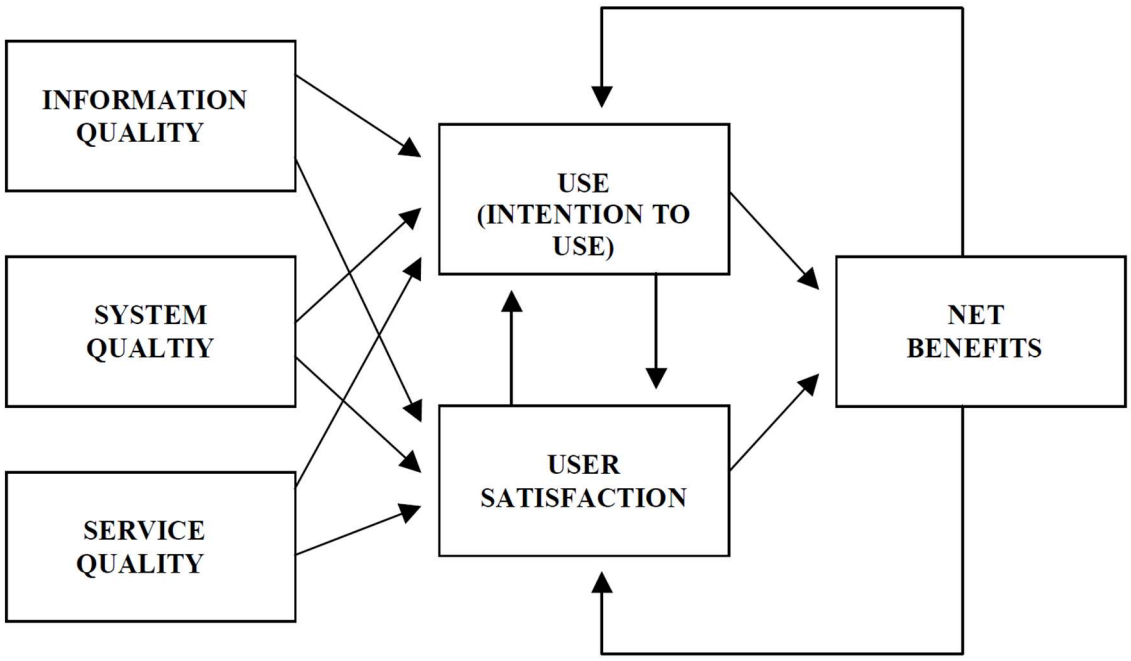

Figure 4 Reformulated IS success model (DeLone and McLean, 2002)

Markus and Mao (2004) took a different approach in their theoretical model and divided system success concept to system development and system implementation success. System development success refers either to the successful execution or to the successful outcome of the development process or both. System implementation success, on the other hand, refers either to a successful implementation process (namely successfully preparing a target community for the system adoption) or to successful system adoption outcome (users use and benefit from the system) or both. No necessary relationship between system development success and system implementation success exists, but they may be weakly related to each other so that system development success functions as an input to system implementation success. Markus and Mao (2004) argue that systems of higher quality and systems that are designed to fit the implementation context instead of

merely fulfilling the functional conditions are more likely to be implemented successfully than their counterparts. (Markus and Mao, 2004)

2.2.4 Packaged software development

So far user involvement and participation has been discussed in an implicitly assumed business-to-business (B2B) context. This approach has been sufficient as usually user engagement occurs when businesses develop custom information systems to other businesses. However, there are situations where information systems or applications are developed directly for the mass-market audience instead of narrow business segment, also known as off-the-shelf, commercial, or packaged software development. This market-driven environment calls for separate investigation on differences of user participation in the B2B context and business-to-consumer (B2C) context.

2.2.4.1 Characteristics

Several factors make packaged software development distinct from custom software development. A good typology is provided by Carmel and Becker (1995) where differences are classified into system-related and system development related characteristics.

The system, which has so far been considered from the business context, needs to be productized and viewed from consumers' point of view (Carmel and Becker, 1995). Software developers need to acknowledge that in the market-driven environment, there exists no single well-defined customer (Karlsson et al. 2002) but there are multiple potential user types to whom to develop a system for (Carmel and Becker, 1995). Developers should choose a specific target market they aim to address with the software or otherwise, they risk designing a generic system that requires high levels of flexibility and parameterization to be customizable for multiple user types. (Carmel and Becker, 1995) In the market-driven environment, customers are numerous and diverse in their preferences, which means that requirements may emerge from multiple directions at any time and cause volatility in requirements engineering (Karlsson et al. 2002). Customers are a good source of non-functional requirements (what kind of the system should be) whereas functional requirements (what the system should do and how) are usually suggested by internal staff, especially in new and technology-focused companies (Karlsson et al. 2007).

Besides considering their customers, developers need to take into account their product in relation to competitors' products as well i.e. product differentiation. There are many

aspects in which products may vary such as price, functionality, reliability, services, and image, and failing to differentiate a product may compromise its survival in the marketplace in the long run. (Carmel and Becker, 1995) Depending on for example the maturity of the market, developers may need to balance between requirements that emerge from the market environment and correspond to user needs, and innovative requirements, enabled by technology, that may provide a competitive advantage in a stable, mature market environment (Karlsson et al. 2002). However, there are risks in solely adopting a technology perspective in requirements gathering, because eventually, the offering has to be adapted to customers' needs (Karlsson et al. 2007).

The other side of the coin, besides to productizing the software, is productizing the software development process. Whereas custom software development process models implicitly assume the availability of users for requirements and feedback elicitation, in packaged software development the marketplace alongside the consumer exists outside of developers' control. Therefore, as a step in productizing the software development process, developers need to "find the remote customer". There may exist parties between the developers and customers, such as marketing department of the organization, or value-added reseller, inhibiting the direct interaction. (Carmel and Becker, 1995) If the developers are not in direct contact with stakeholders, the work of eliciting requirements for new products may be given to other departments that are more familiar with the characteristics of the potential target market, such as marketing department. (Karlsson et al. 2007) Depending on who is responsible for requirements engineering, there may exist diverse views on what constitutes as a good requirement (Karlsson et al. 2002) and how the requirements should be drawn and created (Karlsson et al. 2007). This can lead to requirements that are considered sufficient by their creator or by the management but which lack the necessary characteristics to be implemented in development such as requirements being independent, testable, clear and non-conflicting with each other (Karlsson et al. 2002; Karlsson et al. 2007). Several methods have been offered to narrow down this gap between different groups of professionals in an organization such as increasing communication and collaboration between involved groups (Karlsson et al. 2002), specifying a common language for conveying requirements such as unified modelling language (UML) and having a mediator between groups reformulating requirements from one group to correspond with other group's needs (Karlsson et al. 2007).

Once the customers have been identified developers need to involve them in the development process with their selected involvement methods. What makes packaged

software development so distinct from custom software development in this aspect is that customers are practically unknown before the software has been released and customers adopt it. (Carmel and Becker, 1995)

The context of the market environment and pressure of competitors forces to minimize the time-to-market and grab windows of opportunity in packaged software development to stay ahead of the competition. Also, as the revenue comes in this model from the sale of software licenses and upgrades, software companies are incentivized to develop more and rapidly. (Carmel and Becker, 1995)

Because the custom software development process models are usually driven by technical and behavioural requirements, they are lacking in the side of the market in the requirements. As the custom software development models are adapted to packaged context, a marketing interface needs to be manually integrated into the model to enable seamless collaboration with the marketing department of the organization and to incorporate their expertise into system development. (Carmel and Becker, 1995)

2.2.4.2 Models for packaged software development

Carmel and Becker (1995) proposed a process model for packaged software development consisting of two independent loops – requirements loop and quality loop – that are connected through (product) specification freezing stage. The requirements elicitation loop focuses on refining a product concept into a product specification through involving marketing, quality assurance and customers in the process and gathering and implementing feedback incrementally and iteratively. Simultaneously, risks are sought to be minimized through frequent concept evaluations with involved stakeholders, incremental commitment, and multiple continue/discontinue points. (Carmel and Becker, 1995) In a case study concerning the inclusion of users in the development of off-the-shelf software by Hansson et al. (2006) users participated in the development and provided feedback through attending user meetings, being in contact with customer support, and participating in online courses. The meetings were aimed at delivering news and facilitating discussion between developers and users regarding the development and questions about the system. It also made possible for users to get acquainted with other users enabling community creation. Customer support was used to elicit feedback and suggestions from users and was even handled by developers, thus removing any middle-mans between users and creators of the system.

Product specifications can be frozen after requirements have been sufficiently thoroughly refined into implementable state and when no more new requirements are expected to surface. The frozen specifications are the blueprint for designing and building the software in the quality loop as well as the standards which the software is tested against. The quality loop itself consists of sequential design, build, evaluate, and test phases which are repeated until the software fulfils specifications and the threshold defect standards. (Carmel and Becker, 1995)

Besides the process model proposed by Carmel and Becker (1995), there are also other potential ways how to organize packaged software development. Hansson et al. (2006) propose that two complementary development cycles could be run simultaneously, where one is faster and flexible, implementing fixes and small changes to the system, and the other one is larger and slower where major changes and improvements are developed. Karlsson et al. (2002;2007) mention a method to cope with the potential uncertainty in system development, is to leave a little room to manoeuvre and not complete 100 % of the product features as the product leaves for beta-testing, should customers change their minds. Another method to manage the complexity is to bundle together requirements that concern the same part of the system or affect the same code in the software. In this manner, conflicts or duplicates among requirements can be noticed more easily facilitating implementation. Hansson et al. (2006) suggest that while prioritizing users' proposals, its generality, implications to other functionality, degree of usefulness, and potential workload ought to be considered. In addition to that, technical aspects such as the effects of requirements on system architecture, potential risks in implementation, interdependencies within and between the system and requirements, and given deadlines; as well as business aspects such as the added customer and user value versus the development cost, and the long-term effects on strategic positioning in the market and market value; are also important factors to take into account (Karlsson et al. 2007).

There are also potential downsides to this kind of market-driven development approach. Only a certain portion of the community may be interested in participating in the development potentially leading to uneven representation of the user community (Hansson et al. 2006). If it is in the interests of the developers to cater to all user groups, more effort may be required from the developers' side to encourage the quieter side of the user community to voice out their opinions. Uneven representation may lead to wrong proposals being implements (Hansson et al. 2006). Also, the importance of giving feedback back to those who provide requirements suggestions and feedback to developers should

not be ignored as it may leave feedback providers feeling neglected and discourage them from providing further input in the future (Karlsson et al. 2002; 2007).

2.2.5 Summary of user participation in ISD

User involvement and user participation have been studied for almost half a century in the ISD literature. User participation refers to concrete activities users perform during ISD, whereas user involvement refers to users' mental stance towards the system in development. Together these can be addressed as behavioural engagement and attitudinal engagement towards the system and its development process.

Participation itself can be examined from different angles and categorized according to its type, degree, content, extent, formality, and users' influence ability. It can also be approached from a functionalist perspective, in which user participation is viewed more as a tool towards successful system development, whereas in neo-humanist view user participation is viewed as a method for users to affect their work life and environment. Various models of user participation or involvement has also been built to try to depict the relation between user participation and involvement, and ISD success.

The common conception is that user participation increases the chances of successful system development. Many studies and meta-studies have sought to confirm this relationship yet the results have mostly ranged between mixed and positive. Instead, a behavioural—attitudinal theory of IS success has emerged to describe the connection between user participation and ISD success. According to it, user participation leads to increased user involvement i.e. participation in the system development activities makes the system more personally relevant to users, and thus influences system use and facilitates system acceptance. Although, system success is highly dependent on how it is defined as there are multiple ways to measure it.

Since the beginning, various measures such as user satisfaction and system use have been used as a way to determine whether system development has been successful. However, no established procedure or set of measures has yet to emerge most likely due to each ISD project being a contextually tied instance. Instead, certain measurement categories have been identified that relate to system creation, system use, and system use outcomes. System development success and system implementation success has also been proposed as alternative success measurement categories.

The benefits and challenges of user participation and involvement have also been the interest of researchers for a long time. These are summarized in Table 5.

Table 5 Benefits and challenges of user participation and involvement

Benefits	***Challenges***
Gathering more accurate system requirements	Unpredictable users
Avoiding unnecessary system features	Users reluctant to participate
Facilitating system acceptance	Users' lack of understanding system design and development
Overcoming user resistance	Unclear communication structures
Increased user satisfaction	Failure to recognize, transmit, interpret, or prioritize necessary information
Increased individual impact	Time-consuming process
Increased system usage	The amount of data to process
Increased understanding of the system	Selecting user representatives
More democratic working environment	Uneven user representation
	Conflicts between user groups

User participation and involvement are usually employed in custom software development in a business-to-business context. However, when shifting towards business-to-consumer context different patterns need to be taken into account: the customers are more diverse and more difficult to find, the offering must be more accurately targeted towards the customers, competitors must be taken into account and products differentiated.

3 METHODOLOGY

As player participation development approach is still relatively new and unexplored, qualitative research approach was adopted in this study to enable constructing a more in-depth understanding of player participation, acquiring knowledge to map out different aspects of player participation, and validating the applicability of various user participation characteristics to game development context. More specifically, because player participation can be intuitively outlined both conceptually and in practice as a single entity, the case study approach was considered appropriate. Case studies are deemed "excellent in generating holistic and contextual in-depth knowledge through the use of multiple sources of data" (Eriksson and Kovalainen, 2016, 131) which serves the purpose of this study in building a general level understanding of player participation in game development. Data was gathered from game studios through semi-structured interviews and players were also recruited to answer a short online qualitative questionnaire of their player participation experiences.

3.1 Data collection

The target group of interviewees constituted of game studios that had involved players in the development process of their current or previous games. Game studios were searched from the listing of operators provided at the website of the umbrella organization of Finnish game development industry – Neogames (Neogames – Industry operators, 2020). Game studios located in the city of Turku were contacted due to their convenient and close location for an interview. E-mails were sent to these game studios asking whether players had been involved in their recent projects and if they would like to participate in the research. Out of 17 game studios contacted, four agreed to take part in an interview. Studios were small at size (mostly equal or less than 15 people) and focused on mobile as well as PC platforms. Target player groups of the game studios ranged between children, youth, and adults. Studios financed their game development through various means such as public grants, bank loans, sales revenue, and by funding from outside publishers. Games utilized freemium and premium monetization methods, and some games were completely free of charge. One studio requested for permission to participate in written format by e-mail. Their request was granted on the condition that more elaborative questions could be asked with subsequent e-mails to preserve some of the characteristics of face-to-face interviews.

Interviews were conducted at the premises of each studio in a one-on-one setting in Finnish. They lasted for about an hour and were audio-recorded for later transcription and analysis. Consent for voice recording was asked and concerns related to identifiable and sensitive information were addressed. Studios appointed their representative for the interview themselves. Interviewees worked either in leading, development-related, or community management roles. Interviews were conducted in semi-structured format to enable following the relevant themes identified in the literature yet allowing simultaneously to deviate from that same path to pursue new emerging unidentified topics relevant to player participation as necessary.

The interview themes were divided into three categories. The first theme attempted to clarify the characteristics and nature of player participation to build an overall better picture and definition of the phenomenon and to distinguish it from user participation. The second theme went deeper into the game development process itself and sought to understand how player participation takes place in practice in the game development project and whether it differs much from the requirements elicitation, and system testing phases that are common user participation phases in system development. Finally, in the third theme, the aim was to investigate what kind of effects player participation has both on the game, development, as well as on the financial outcome, and wrap up the overall impact of involving players in game development.

Besides the interviews with game studios, players' perspective for player participation was also considered to be a valuable and necessary addition to construct a more comprehensive picture of player participation. For this purpose, a qualitative questionnaire consisting of 12 mostly open-ended questions was constructed and distributed in English through relevant online forums on internet discussion platform Reddit. In the questionnaire amongst the other things, players were asked which game projects they had participated in. Based on that information, the questionnaire was distributed further to those game-related forums. To mitigate the risk of response fatigue, answering the questions was not mandatory apart from the initial demographic questions. 21 individuals with the following characteristics answered the questionnaire. Gender: 2 female, 18 male, 1 other. Nationality: 9 from the United States of America, 3 from Canada, 2 from Brazil, and 1 from each following country: Australia, Austria, Finland, Germany, Poland, Slovakia, and United Kingdom. The questionnaire is presented in Appendix 1. Player participation questionnaire for players.

3.2 Data analysis

In the data analysis section, the Gioia methodology was followed (Gioia et al. 2012). Interviews were transcribed, anonymized, and read through several times to identify emerging themes, patterns, and categories for codes. The literature review also functioned as a source for several potential codes. Transcriptions of the interviews and the players' answers for the questionnaire were gone through one research question at the time. Issues relevant to the specific research question were highlighted and later abstracted into descriptive and concise codes. Codes that handled similar issues or were related to the same topic area were bundled up together as higher-level categories. Eventually, these categories formed top-level topic areas that answered each research question and are represented as the main chapters of the findings section. The codes and coding categories are presented in Table 6 alongside a few examples. Not every code had a clear and concise example and therefore few of the fields have been left blank. The quotations from interviews in this paper were translated from Finnish to English while attempting to preserve as much of the original message and context as possible. The quotations from players were already in English and were preserved as such.

Table 6 Codes and coding categories for analysis

Dimensions of player participation	*Code categories*	*Codes*	*Example*
Actors	Player community	Player types	There are different desired player types, of course, we want players who somehow represent average players. *Interviewee #1*
		Internal dynamics	Plans written in Trello have received more publicity when players link the roadmap to each other. *Interviewee #4*
		Internal diversity	Blindly following each desire would not make the game good, because players have so different views on what is fun. *Interviewee #3*
		Motives for participation	"To help developers to improve their projects, to be not only a player but to be like a part of the project too." *Player*
	Game studio	Community manager	Affects work tasks in the sense that we have one person who handles community management. *Interviewee #3*
		Motives for involving players	Because it clearly leads to better chances in success when the goal is a well-selling game. *Interviewee #3*
Nature	Attributes (Caveye, 1995)	Type	
		Degree	[Degree of participation] depends on a player: some are so passionate that they may commit to test each new feature we release. *Interviewee #4*
		Content	It may, for example, be the general level of difficulty in the game. *Interviewee #3*
		Extent	
		Formality	
		Influence	We have our own vision of the game, to which we yet include players' feedback. We do have a stronger voice but if many players provide us with the same feedback, it is a clear sign for us to listen to that. *Interviewee #4*
	Voluntariness	Voluntariness	
		Participation threshold	People do not usually share their opinion of the game. Only is something is missing or something is broken. *Interviewee #1*
		Commitment	If the game does not please then people walk away and never come back. *Interviewee #1*
	Applicability	Monetization method	
		Genre of a game	"However, it [player participation] might not perfectly suit every project (such as i.e. story-driven experiences)." *Player*
		Stakeholders	It may be that publisher requires some sort of embargo from us, that they do not want that we announce it [the game] right away. *Interviewee #4*

Table 6 Codes and coding categories for analysis (*continued*)

Dimensions of player participation	***Code categories***	***Codes***	***Example***
Activities	Community management	Player acquisition	Player acquisition is aimed at primarily by discounts and exposure campaigns within a store. *Interviewee #4*
		Community maintenance	
		Participation maintenance	By creating methods through which players can provide feedback and then also by implementing part of the good ideas, that are provided by the community. *Interviewee #3*
	Communication	Active	There are friendly people [in Discord] who send bug reports to us. *Interviewee #1*
		Passive	And because people do not express their opinion, game studios usually analyse players' behaviour. *Interviewee #1*
		Channels	
	Testing	Local testing	
		Remote testing	
Outcomes	Benefits	Development related	Continuous validation of ideas and new functionality with real users is a tremendous benefit. *Interviewee #4*
		Game related	The product is simply different than it would be if players were not involved. *Interviewee #3*
		Business related	The benefit is that players like the game more and because of that it sells also a lot better. *Interviewee #3*
	Challenges	Community-related	Nowadays it is difficult to get anybody to even play a free game, and it is also very hard to get a player to provide feedback or to remain in a community. *Interviewee #3*
		Participation related	Considering that the game has been downloaded maybe over 100K times and there [in Discord] are a couple hundred people, it is not really a representative sample of players. *Interviewee #1*
		Workload related	That huge volume of text, a terrible amount of thoughts that have to be somehow handled and answered i.e. it creates work. *Interviewee #4*
	Other	Work-related changes	There is a strong bond with the community in the organization and we keep it always in mind when the game is developed. *Interviewee #3*
		Game industry problems related effects	I think it [player participation] may amplify them [certain problems in the game industry]. *Interviewee #4*

4 FINDINGS

The results of this study are organized in the following manner. In the first section, actors in the participation process are presented alongside with their motives and interests towards participation. In the second section, the nature of participation and the participation process is described and compared against the characteristics of user participation in information system development. Following that, activities identified in the participation process are described, and finally, the benefits, challenges and other effects associated with player participation are presented.

4.1 Actors in the participation process

The usual scene of video game development industry consists of video game producers i.e. game developers, publishers (who usually finance the game development project), distributors, and retailers (Zackariasson and Wilson, 2010). User participation in ISD on the other hand heavily concentrates on users and their role in the ISD process. Users or their representatives (Damodaran, 1996) can be categorized for example as primary or secondary users (Ives and Olson, 1984) or according to their position in the participation process as stakeholders or participants (Markus and Mao, 2004). The opposing side consists of various change agents that provide participation opportunities for the stakeholders such as IS professionals, as well as managers, HR professionals, and external consultants (Markus and Mao, 2004). Besides the usual actors in the game development industry, two new ones emerged clearly from the interviews. Those are player community – an aggregation of players – and community manager – representative of the game studio towards the players.

4.1.1 Player community

Player community was a concept that surfaced frequently during the interviews when interviewees talked about players. It was used as a way to address the whole player base as one singular unit instead of discussing individual players. There seemed to be no proper definition for it as in some cases player community appeared to be used to address the whole player population of a game and in other cases, only the participating players formed the player community. Therefore, it is difficult to say whether player community forms the stakeholders or the participants of player participation as according to the categorization by Markus and Mao (2004) because its use seems to differ according to the

context. However, what was evident is that player community is not a homogenous group of people, something that was hinted by Karlsson et al. (2002) stating that there exists no single well-defined customer in a market-driven packaged software development, and by Carmel and Becker (1995) stating that there are multiple potential user types whom to develop for. Player community consists of many diverse individuals with differing interests and motivations for participating, it is prone to its various internal dynamics such as supporting itself or sometimes ruining itself and has varying degrees of participation between the individuals.

> *Some like only to follow the interesting initial stages of a game. For them a regular development blog update is sufficient. Others, on the other hand, want to discuss with developers for example Discord functions well for this.*
>
> *–Interviewee #3*

4.1.1.1 Motives for participation

Players were asked about their motivations for participating in game development projects to understand why players want to experience and invest their time in unfinished products. Similar motives that Gandolfi (2018) found out such as being involved, opportunities to improve games, originality of the game idea, presence of friends, and developers' track record, were identified also in this study. However, motives such as testing before purchase, playing for free, and the genre of the game (Gandolfi, 2018) were not identified.

Some players had internal or personal motivations such as being part of the project or the community, assisting the development, and tinkering around with the game, whereas others were more influenced by external factors such as developers' promises and their vision of the game, the current state of a game and innovative game concepts, and the available reviews and feedback of the game. Some internal motivations appeared to be somewhat individual-oriented such as influencing the development direction, or abusing game exploits for personal enjoyment, whereas others were more collective in nature such as assisting development and the community and being part of the project and community. The external motivational factors, on the other hand, could roughly be categorized into game, developer, (promotional) material, or word-of-mouth related factors. Players' motives for participation are presented in more detail in Table 7.

Table 7 Players' internal and external motivational factors for participation

Internal motivational factors	***Example***
Being part of the project	"To be not only a player but to be like a part of the project too."
Following the development journey	"Chance to be in at the beginning and watch the game grow and/or provide feedback on game experience."
Assisting development	"To help developers to improve their projects." "Mostly because I work as a developer and want to help fix bugs, test stuff etc."
Influencing development	"…felt my input could help gear the game down a final path that I enjoyed."
Assisting the community	"Helping the games community."
Tinkering with the game	"I love playtesting and learning to abuse the game and being able to show my knowledge of the framework of a project to more casual users."
External motivational factors	***Example***
Developers' promises and vision	"Their promise to shape up and end as great games."
Developers' familiarity and interaction with the community	"I enjoy the devs at [game studio]." "Developer interaction with the community"
Price of the game	"Price, cool video and reviews on Steam and YouTube."
Innovative and different game concepts	"They looked promising and fun or did something different for gaming."
Available game content	"Plus, I gotta really like the concept of the game and what it currently offers in order to participate."
Current (and future) game state	"Game at current state looks good and promise of future updates"
Game related material	"I usually get convinced because of the gameplay trailers…"
Game reviews and recommendations	"…or because of friends that want me to play." "Very positive feedback from those who were already playing the game."

4.1.1.2 Player types

Players' participation motives also partially reflect what kind of player types exist in game development projects. Player types refer to various non-exclusionary characterizations on how players contribute to a project or how they act in a player community. For example, the two most common occurring player types in players answers were 'testers' who play a game and provide feedback and report bugs, as well as 'chatters' who engage in discussions with the community and developers on online forums. Other commonly identified player types were:

- authors, who come up with ideas and suggest new features
- creators, who create in-game content such as mods or translations
- tinkerers, who push the limits of a game or search for exploits
- voters, who voice out their opinions through polls and surveys
- financers, who participate through monetary contributions
- supporters, who help new players into the game and existing ones with issues
- reviewers, who write reviews of games

Also, game developers were inquired about the desired and non-desired player types to understand what kind of players are valued in game development projects. Similar player types were suggested such as testers and supporters but also various desired player characteristics and traits such as having good communication skills, interact with the community, understands about game design, and is a committed towards the project. Undesired players were essentially various troublemakers, who cause trouble in the player community, misbehave towards other players or game studio personnel and their associates, and are greedy, demanding, or otherwise unsatisfiable individuals who reward the game developers with negative reviews when the game is not developed according to their views. Factors such as having a bad day, anonymity in online discussion forums, and facelessness of the game studio which facilitates bundling up the company together with other ones, were suggested as reasons for the increase of misbehaving players.

Besides having different kind of players within the player community, some of these players may form various subcommunities within the bigger community. These subcommunities were discovered at the end of the research and could not be elaborated further but they likely unite players under more general level categorization such as a language.

> *The second biggest purchasing nationality of our game are Russians and we do not have systematic community management for our Russian community because we have nobody who speaks Russian. Russians have their social medias. It could be that in fact we have a quite large group [of players] there who think something and we do not know about it so much because only a fraction of them come to influence on our English spoken platforms in English. – Then next is China and you may only guess how much contacts we have there.*
>
> *–Interviewee #4*

4.1.2 Game studio

4.1.2.1 Community manager

The changes player participation brought into game studios were mostly noted as increases in the communications tasks with a player community. Sometimes these tasks were handed out to developers increasing their workload and potentially delaying some of their other work. For example, in one studio developers were responsible for the communication with the player community and answered players' questions when they had the time or felt like it. In another studio, a community manager was hired to systematically take care of the daily communication and interaction with the player community and manage other community-related tasks.

A community manager operates at the customer-interface and functions more or less as a representative of the game studio towards players, (and potentially a representative of players for the game studio). The difference with the ISD context is that in user participation literature representatives are usually discussed in terms of users as user representatives. A potential reason for this contrast is that in ISD context users are normally affiliated by an employer which creates a unifying context, purpose and direction for the participation and enables representation under the company mission. On the other hand, in the game development users i.e. players are non-affiliated individuals with own separate motives and interests for participating and playing, which hinders representation as there is less common ground between the players.

Besides hiring a sperate community manager, it is also possible to pick distinguished players from the community and promote them to handle community management tasks.

> *"Mostly I work on hunting bugs and troubleshooting errors. Either things in the code, or just assisting other players in getting their own issues resolved. This usually leads to me being raised to the lead tech support guru for the players. For some of the projects, I've been a community manager, and maintained the support section of their forums."*
>
> *–Player*

However, in this scenario, it should be carefully evaluated that the player is competent for the task and shares similar values and game vision as developers to mitigate any risks of a player getting drunk with power and hurting the community.

4.1.2.2 Motives for involving players

Kujala (2008) summarized the motives for user involvement into development-oriented motives focusing on delivering functional and correct system, democratic motives focusing on enhancing users' expertise and opportunities to influence work, and organizational motives aiming at successful implementation of the system. Respectively, this research set out to find motives behind involving players in game development, to understand what do game studios expect to gain through player participation. The motives identified are either game, development, community related or financial and are summarized in Table 8 with the respective aspects of game development they concern.

The most obvious reasoning for player participation was to create better games for players. This meant better games as a form of enjoyment as well as functionally and gameplay-wise that would lead to more successful game titles being produced.

> *It's [player participation] purpose, in my opinion, is to create a better game. A functional game technically but also a functional game in terms of game balance and financially.*
>
> *–Interviewee #1*

Another reasoning was that it would enable smarter and more cost-efficient development when players' feedback was listened to and development direction was steered accordingly.

> *We can make smarter decisions with much smaller costs when we listen to our players. – If we constantly receive feedback and opinions from our real userbase then it is a huge asset compared to trying to create something in a vacuum and then hoping that users like it when it exits the vacuum.*
>
> *–Interviewee #4*

This relates to the next motive that is targeting the game to its audience. When players are involved in the game development process, the game design choices can be validated with the actual players so that the game satisfies its players and players' functional requirements.

> *So we set off from wanting to see immediately how different children for example control [the game], how they would want to control, what is their intuitive way to control the game, and it [testing] must be done in a pretty early phase so that it cannot be programmed too complete – basically we aim for that any child could play our games without needing to learn how.*
>
> *–Interviewee #2*

Targeting was also supported in the literature concerning packaged software development because without targeting, a system would likely remain too generic and would require designing parametrization into the system to enable customizing it to various user types (Carmel and Becker, 1995).

The last couple motives were related to player retention and achieving exposure. Player participation was justified as a way to understand how players feel about the game so that would keep playing.

> *We want to, of course, know what they [players] think about it [the game] so that they remain involved with the game, want to play the game, like the game because in the end the longer you play a game the higher the likelihood is that at some point you buy something there with real money.*
>
> *–Interviewee #1*

In terms of achieving exposure, player participation was considered to contribute to it through multiple stages. By targeting a game towards its audience, the game would be more likely to receive better reviews that would potentially be favoured by the algorithms of a digital distribution platform in terms of gaining exposure. Respectively, when a game is liked by its audience it is also likely to be shared with friends and played more increasing the combined playtime, which can also be a factor in the game featuring algorithms. Even though these claims cannot be proven right unless the algorithms are made public, it cannot be denied either that featuring and selling a game which is played a lot and has been received well, would not benefit the distribution platform as well.

However, there was at least one argument against player participation stating that player participation would be somewhat redundant due to unbalanced participation which would skew developers' understanding of the player base.

> *Here is perhaps the reason why players are hardly involved that much because 99 % [of the players] do not express their opinion in any way and yet we are interested of course in their opinions also but they will not share it so what to do. Well, we do not want to give too much weight for that one per cent who yells out there about something because even though it, of course, represents many individuals' opinion … it does not necessarily indicate that it is the most important issue at hand, so because of that, too much weight cannot be given either.*
>
> *–Interviewee #1*

In the end, it is undeniable that behind every motive are financial interests. Smarter development and targeting contributes to the creation of a better game for the audience, which together with player retention methods ensures that the game lives up to its expectations and evolves as necessary. This then again leads to increased playtimes and positive word-of-mouth attracting more players and thus gaining more exposure, eventually translating into stronger financial returns.

Table 8 Motives for involving players and their relation to different aspects of game development

Motives	*Game*	*Development*	*Community*	*Financial*
Better game	X	X	X	X
Smart development		X		X
Targeting	X		X	X
Player retention			X	X
Exposure				X

4.2 Nature of participation

4.2.1 Voluntariness

The main difference between game development context and information system development context is voluntariness and obligatoriness of participation, which majorly dictates the characteristics of player participation and how it is realized in practice. Whereas the purpose of productivity applications is to provide functionality (Kasurinen and Laine, 2014; Kasurinen, 2016) to execute work tasks (Pagulayan et al. 2012) and the use of

productivity applications is usually obligated by an employer, the purpose of video games is to provide fun and enjoyment to players (Kanode and Haddad, 2009; Lewis and Whitehead, 2011; Pagulayan et al. 2012; Kasurinen and Laine, 2014; Murphy-Hill et al. 2014) and playing video games is a voluntary act (Pagulayan et al. 2012).

> *Information systems are used because you have work task you want to accomplish. Games, on the other hand, are entertainment, so you will drop a game quite faster whereas if you said at work that you do not feel like using this information system because the buttons are wrong coloured, then your boss would likely disagree with you at that point.*
>
> *–Interviewee #1*

These differences are also reflected in both participation contexts: Whereas at least some portion of the userbase is obligated to participate in an ISD project, in the game development context game studios do not possess similar power over players to force them to participate but must rely on other methods to attract players. A similar trend of lack of influence over users in contrast to custom software development was noted in packaged software development where it was mentioned that the marketplace and users exist outside the developers' control. Respectively the users of packaged software are unknown before the software has been released and adopted by its users. (Carmel and Becker, 1995)

As an example of the voluntariness, one interviewed game studio described a situation where they sought to recruit people over social media to playtest their games for free. They only managed to get people who were highly enthusiastic and interested in the game, whereas people who did not like playing or were sceptical of the game or playing generally, declined the offer. Another interviewee pondered that for a player to bother participating, player's interest and dedication towards the project must be higher than normal.

> *I think they [participants] are in certain way fans of the game. They are so enthusiastic about the game that they come to [our] Discord [channel]. It is already a sign that they are interested in the game more than normal.*
>
> *–Interviewee #1*

4.2.1.1 Participation threshold

Participation threshold is a concept derived from the voluntariness nature of player participation. It signifies the fact that each individual possesses a threshold that must be overcome until that individual chooses to participate. Some individuals have a lower tendency to participate whereas others may require more persuasion or are impossible to persuade for participation. The current trend appears to be that people inherently possess a high threshold for participation.

> *Nowadays it is difficult to get anybody to even play a free game, and it is also very hard to get a player to provide feedback or to remain in the community.*
>
> *–Interviewee #3*

That being said, participation threshold is an individual's attribute that may depend on various internal factors such as beliefs, or previous experiences of participation instances as well as on various external factors such as the genre of a game.

> *Gameplay seemed like my preferred style of gaming.*
>
> *–Player*

Developers' activity in the community.

> *Our presence [in the community] has clearly increased its activity. We show that we are present and we are often praised for it.*
>
> *–Interviewee #4*

Or even considerably annoying issues in the game.

> *...for example, this clan leader who is passive and causes a lot of trouble in that clan when new players cannot be accepted, so that was already enough annoying that somebody snapped.*
>
> *–Interviewee #1*

Intuitively, many of the factors presented in the motives for participation chapter, such as recommendations from friends, could also be associated as factors that lower players' participation threshold and encourage players to participate. Speculating a bit further, it could be even argued that motives for participation are factors that decrease players' participation threshold. Respectively, there are likely to be factors that increase

players' participation threshold as well i.e. reduce players' willingness to participate in game development projects, such as previous negative experiences. Some of these increasing or reducing factors can be more general, such as player's desire to contribute to something bigger or to feel being part of something bigger, or more directed towards the project such as the price of the game. Some factors may have long-lasting effects, for example, it could be argued that previous positive experiences on player participation instances would likely affect positively on future player participation opportunities, whereas some factors may only induce momentary participation such as the previously mentioned sufficiently annoying issues in the game that prevent players from reaching their goal.

The major difference of participation threshold to motives of participation is that, whereas motives list, and potentially categorize, various factors that contributed to starting participation, participation threshold takes the opposite approach and acknowledges that player's willingness or reluctancy to participate in a game development project is a variable depending on other various contextual factors. In some occasions, it might be easier to persuade a player to participate, on other occasions, it might be more difficult or even impossible. Sometimes it may require a combination of multiple factors to persuade a player hinting that not all factors are equal to each other. However, more research would be required in identifying more factors affecting players' participation threshold and how it would function in practice. One example of an assumed high participation threshold occasion was that it appeared to polarize participation instances into two extremes.

> *People do not usually express their opinion of that game … comments are usually at extremes: either something does not work, or something is super. From between it is quite difficult to get any information.*
>
> *–Interviewee #1*

Different levels and degrees of participation could also be examined for example: what does it take for a player to purchase or download a game, what does it take for a player to play the game and send feedback through a potential in-game feedback widget, what does it take for a player to search for game-related forums and engage in discussion with the community, what does it take for a player to send e-mail directly to the developers, etc.

4.2.2 Ownership and commitment

Barki and Hartwick (1994) suggested that "because of their participation, users may perceive that they have had a substantial influence on the development process and thereby develop feelings of ownership". Similarity was encountered during the interviews as one interviewee mentioned having observed occurrences of feelings of ownership and commitment among the active player base.

> *We have experienced, that because players are allowed to participate in the development and feel certain kind of ownership [over the project] – those, who have participated in the project the most – so I believe that we have a great number of very committed players, who for example with greater likelihood will become interested of our next games just because those are our games.*
>
> *–Interviewee #4*

The concepts of ownership and commitment were not discussed in the literature review and require additional clarification here. According to Pierce et al. (2001), psychological ownership refers to the possessiveness state of mind in which an individual considers that a target or an object of material or immaterial nature is theirs. It serves the purpose of enabling individuals to fulfil three basic motives: Being efficacious, expressing self-identity, and having a place. Experiencing ownership over a target allows an individual to alter that target and experience satisfaction through being efficacious and reaching desired outcomes. Targets that an individual considers their own can be used to build and express an individual's self-identity as well as preserving an individual's existence across time. Psychological ownership also serves a purpose in devoting a significant amount of resources in building a place that individuals can call home. (Pierce et al. 2001)

Pierce et al. (2001) propose that the feeling of ownership towards a target is caused by various factors such as being able to control the target, intimately knowing the target or being associated with it, and the extent individual invests themselves and/or their resources (labour, time, etc.) on the target. These factors and their strength on the feeling of ownership vary according to specific other factors, for example, control over the target depends on the power structures of the setting and the level of autonomy given to the individuals, being familiar with the target depends among the available information also on the intensity and length of association with the target, and the degree of self-investment

on the target is highly dependent on the complexity of the tasks at hand: creative, non-routine, expertise dependent tasks require more effort and individual's discretion to accomplish enabling more self-investment on the target. (Pierce et al. 2001)

Psychological ownership can be seen to lead to various outcomes. For example, individuals may expect to have a right to information regarding the target or right to affect the decision making concerning the target. It may give a rise to self-experienced, parental kind responsibilities towards the target such as protecting, caring, and nurturing the target. Depending on the context, psychological ownership affects how changes concerning the target are received. Changes that are self-initiated, evolutionary, and additive are likely to be received positively, whereas changes that are imposed, revolutionary, and subtractive are likely to cause negative emotions and resistance towards the change. Also, occasionally psychological ownership may lead to other counterproductive effects that, for example, can be harmful to teamwork such as failure to share responsibility or information among the individuals, damaging towards the target of ownership, such as sabotage in case of forceful separation of an individual from the target, or may harm individual's wellbeing through experienced feelings of frustration, stress, and alienation if individuals feel that they are losing control over the target. (Pierce et al. 2001)

Having defined psychological ownership, it's connection to commitment must be identified. Whereas the question, that psychological ownership seeks to answer, is "What do I feel is mine?" (Pierce et al. 2001), commitment or organizational commitment attempts to answer "Should I maintain my membership in this organization and why?" (Pierce et al. 2001; Dyne and Pierce, 2004) Dyne and Pierce (2004) confirmed psychological ownership being related to organizational commitment and argued that if individuals experience psychological ownership towards their organization, the organization is likely to fulfil the human motive of having a place or home, thus giving a direct reason for maintaining membership and committing to that organization.

Organizational commitment itself can be divided into three components namely: affective commitment that is based on emotional reasons for staying such as feelings of comfort; continuance commitment that is based on rational reasons for remaining such as related costs of leaving; and normative commitment that is based on experienced duty or obligation such as loyalty norm or owing a favour. Each individual would possess their commitment profile reflecting their grounds for maintaining membership in their organization. (Meyer and Allen, 1991)

During the interviews, it was noted that players with low levels of commitment or without it may drop the game at any time and may not make it to participate at all.

> *People do not really spontaneously report if there is something wrong somewhere maybe because the game is free of charge and when it is free people are like "I could not care less; I am downloading the next game".*
>
> *–Interviewee #1*

Intuitively this leads to the fact that studios would want to aim for higher player commitment levels to keep players engaged in the community participating and contributing to a game development project. Backtracking the progress from ownership to commitment, creation of commitment could be sought by facilitating the emergence of psychological ownership by addressing the routes that lead to psychological ownership. In practice, game studios could consider ways to allow the player community to have (more) control over the game or the development process, or different portions of them. If a game studio wants to reserve the creative control to themselves, more influence could be handed to players on matters that are less crucial or smaller at scale. Information regarding the game and its development process could also be handed out to players in higher amounts or in a more easily reachable and digestible format to allow players to get more closely associated with the target. Various methods to encourage players interaction with the game and its development process could be designed. Allowing the player community to get to know to the people behind the game could also be considered. Finally, allowing players to invest themselves in the development process for example by suggesting ideas and incorporating them into the game or by enabling the player community to create in-game content itself (Thominet, 2018) could be seen as ways for player community to invest themselves.

> *"Many requests I've made or seen which got eventually implemented into the game increased the quality of the experience more often than not. It has made me more proactive providing feedback."*
>
> *–Player*

> *"It was nice seeing bugs that I identified being ironed out as I play test. Made me want to help more."*
>
> *–Player*

Many games and player communities also thrive on community-made in-game content that has been created after a game has been released. These are called game modifications or 'game mods' in short, and creation of these could be officially endorsed and supported by the game studio to encourage further engagement with and within player community.

It would be interesting to study how the voluntariness nature of player participation affects players' commitment towards the development project. It could be argued that since participation is voluntary i.e. nobody is forced to participate against their will, and if game studio enables players to have some degree of control over the development, provides information about the development and game to the community, and allows players to invest themselves into the project and game, it would lead to higher levels of experienced ownership over the project and thus to higher commitment towards the game.

4.2.3 Applicability of participation

During the interviews, a theme began to emerge that player participation may not apply to game development in every possible scenario. In ISD, user participation appeared to be more appropriate when task and system complexity are high (McKeen et al. 1994). In the game development context, several potential scenarios were identified.

The applicability of player participation in the development of games that utilize freemium monetization methods was questionable. The interviewee noted that when players were given a chance to voice their opinions, they expressed their desire for more things for free.

> *Usually, people just want more things for free in the game which is, of course, an easy requirement but it does not necessarily mean that the game would function better for us financially.*
>
> *–Interviewee #1*

As the monetization of freemium games relies on alternative methods such as microtransactions or advertisements rather than traditional purchase fee, the game is usually provided to players free of charge in hopes of attracting huge numbers of players and turning them into paying customers. To encourage players to spend real money, some of the features may be locked behind a paywall, or progressing in a game can be made less tedious through microtransactions. Because players are unlikely to spend money for playing a game, at least in large amounts, developers must constantly keep monetization

perspective in mind while developing the game to ensure the continuance of the game studio. Incorporating a voice that only desires more for free into the development would not likely be beneficial for the game nor its development. Similar trend was also reported by Jacobs and Sihvonen (2011) who observed the forums of Facebook games created by Zynga, whose game mechanics revolved around the supply and demand of in-game resources. They noticed that players tend to participate in game development from the perspective of a player instead of a game designer, desiring things that are scarce and more tedious to acquire in a game. Developing a game according to players' desires in this scenario would eventually lead to unbalanced gameplay as players desires would undermine the core game mechanics.

Another major factor that most likely affects the suitability of player participation is the nature of a game. For example, games that rely heavily on the story may be troublesome to develop with players without spoiling the plot of a game.

> *If we do anything little plot-driven then we are spoiling it if we share it, so it depends on what kind the game will eventually become.*
>
> *–Interviewee #4*

Similar issue was also noted by a player.

> *However, it [player participation] might not perfectly suit every project (such as i.e. story-driven experiences).*
>
> *–Player*

Unfortunately, no other game dependent factors were identified. Interesting topic as it is, further research could be conducted on what kind of video game related characteristics affect the applicability of player participation and how. However, based on the previous analysis, it could be argued that for a player to participate the game ought to induce enough commitment so that player manages to overcome own participation threshold. One of the routes to psychological ownership, the antecedent of commitment, was the degree an individual is able to invest themselves on the target of ownership, namely the game and the development process. As the investment of oneself can take various forms such as "one's time; ideas; skills; and physical, psychological, and intellectual energies" (Pierce et al. 2001), it would make sense that a game whose gameplay, game mechanics, and environment are limited would not be a fruitful platform for player participation as players' self-investment opportunities would respectively be limited as well. Then again,

games that do not have such restrictions but possess broad and deep gameplay opportunities, such as open world sandbox games with multiple game mechanics, would potentially benefit better from player participation.

Finally, some other factors were mentioned that could affect the applicability of player participation. A potential publisher may require game studio to keep certain aspects of a game in secret thus limiting the opportunities for player participation, resembling the shroud of secrecy (Zimmerman, 2014) mentioned in the introduction chapter. Development team's attitude towards player participation and readiness to embrace player feedback plays also a major role. For example, in one game studio, challenges appeared when their art team had to orientate themselves into a new situation and accept feedback regarding an unfinished product – something that is probably quite unexpected and unheard of in the fields of art. Also, if the game studio possesses a strong vision of the game and its development direction and want to execute it that way, then player participation may be redundant.

> *If you want to develop a game only for yourself according to your own vision, then you might as well forget involving players.*
>
> *–Interviewee #3*

In the end, the purpose of player participation in reaching a successful development outcome should be kept in mind, and unless it does not contribute to that, the grounds for involving players are quite weak.

> *In our case, involving players must regardless support production and the operation of the company, and if for some reason or another the outcome was the opposite, participation would not probably be viewed as an intrinsic value.*
>
> *–Interviewee #4*

4.2.4 Attributes of participation

Cavaye (1995) provided a list of attributes used in characterizing and describing user participation instances. These attributes: type, degree, content, extent, formality, and influence of participation; were also used in this research to understand how this categorization behaves and applies to player participation context.

4.2.4.1 Type of participation

Type of participation referred to either direct participation, where all available users are involved in ISD, or indirect participation, where a proportion of users are involved and user representation is likely to be employed (Cavaye, 1995). In game development context, type of participation appeared to depend on the selected testing approach. In the case where one organization approached their focus group itself, exercised game testing in a small closed player group, and later released the game to wider audience, the type of participation corresponded indirect participation approach. In another case where an organization had distributed their game through a digital distribution platform and exercised open testing approach, the type of participation resembled direct participation initially. However, due to the voluntariness nature of participation, it is likely that some players will choose not to participate in the game development project, even though the option would be available for them, shifting the participation type towards indirect one on players' behalf. Also, one interviewee hinted that even though prerequisites for direct participation are present in open testing, the varying degrees of participation among players make some players' contribution more valuable, and thus developers' may be more prone to listen to those individuals, as well as that managing a smaller group of individuals is easier and enables more in-depth interaction than with a larger group of individuals.

> *How it is realized every day is that we have there a particular group who are familiar to us and also have a silent agreement for fast track influencing. In other words, it easily shifts towards it [indirect participation] because it is otherwise difficult to control. So that with the larger group the interaction is not so direct, it does not happen so much, even though we try to reach everybody but the interaction is much deeper with the smaller group.*
>
> *–Interviewee #4*

4.2.4.2 Degree and influence of participation

The degree of participation referred to the amount of responsibility that was handed out to users ranging from consultative roles to granting approval for decisions to being part of a design team and to having a total ownership of development (Cavaye, 1995). Although degree of participation is indicative of users' role and power it does not entirely

determine users' decision-making power, which is covered by influence of participation. For example, a user may be in consultative role yet possess proportionally higher ability to influence decision-making. Correspondingly, a user may be a part of a design team but user's contribution may be entirely ignored diminishing user's ability to influence. However, it would be intuitive to assume that responsibility and ability to influence would correlate with each other.

In game development context, players' degree of participation appeared to mainly focus on consultative roles, and in some special cases on granting approval for decisions. A couple game studios implied that they are happy to receive and listen to players' feedback but simultaneously want to reserve decision-making rights to themselves. If a suggestion does not contribute to game concept, it is likely to never see daylight, unless it is backed by sufficient number of players.

> *We have our own vision of the game, to which we yet include players' feedback. We do have a stronger voice but if many players provide us with the same feedback, it is a clear sign for us that it should be listened. Generally, we do as much [player suggestions] as we can, usually smaller features.*
>
> *–Interviewee #3*

On the other hand, when much more uncertainty was involved, for example regarding a certain decision, players were given more responsibility and ability to influence to support developers' decision-making.

> *Maybe in certain extreme cases when we are unsure about a decision and we are afraid of how players will react, then I could imagine that we would listen very carefully what players will say.*
>
> *–Interviewee #4*

This implies that even though players are involved in a game development project, the purpose of player participation from the perspective of game studio is to support the development process itself. Handing out too much power to a player community with diverse views on what is fun in a game, would perhaps have a quarrelsome effect on the development project if the community does not reach a consensus of development direction. This would most likely lead to development resources being exhausted before the game is finished and force abandoning the project.

> *Blindly following each desire would not make the game good, because players have so different views on what is fun. – If you let the feedback get better of you and start implementing everything, then you will of course run out of money and the game will never reach release state.*
>
> *–Interviewee #3*

4.2.4.3 Content and extent of participation

Content of participation referred to various aspects of ISD in which users may participate in. For example, besides the technical aspect of system design, users may also focus on social or human aspects as well. (Cavaye, 1995) Interviewees were inquired of the various sectors they had allowed or witnessed players to participate but it became quickly evident that players would autonomously participate in every aspect that was made available for them or not restricted outside of their domain. For example, players commented on gameplay features and mechanics, game difficulty and balancing issues, game controls or anything between the visual image, game world related, and actual code implementation.

> *Well they [players] can actually participate in all of that ... In other words, it can be anything between the quite shallow visual or game world related feedback and then again that very profound code evaluation.*
>
> *–Interviewee #4*

Therefore, it is likely that developers cannot restrict participation on anything that has been made available to players, but can nevertheless try to shift the focus of participation to some themes.

Extent of participation, close concept to content of participation, referred to the various concrete system development phases users may participate in. Most often user participation was exercised both during the early stages of ISD such as problem identification and requirements elicitation, and during the final stages of ISD, namely testing and installation stage. (Cavaye, 1995) In this regard, player participation barely diverges from user participation as player participation is mostly associated with the testing and feedback phase as well as with design phase where players come up with new ideas and suggestions for a game. This was also supported in players answers where testing and giving feedback was most commonly occurring themes in players' participation activities alongside generating new gameplay ideas.

4.2.4.4 Formality of participation

Formality of participation refers to the formality in organizing participation activities and the formality of the activities itself such as official meetings between formal teams (Cavaye, 1995). In game development context, the formality of participation was not explicitly discussed with the interviewees but the impression received hinted that player-led participation and interaction followed more or less everyday communication between people. However, the interaction and involvement procedures led by game studios appeared to be more formal the longer a game studio had involved players in game development projects, possibly due to game studios having had more time to experiment and finetune different involvement methods.

4.3 Activities of participation process

In ISD literature, activities related to user participation processes were rarely explicitly discussed. Usually users' association with the development process was expressed through indirect means such as users benefit the development through requirements elicitation and acceptance testing, suggesting that users mainly participate in defining requirements of a system and suggesting feature ideas, as well as confirming that the functionality of the developed system corresponds to the established requirements. In game development context, Thominet (2018) identified three main activities related to open video game development, namely: providing access to the game; transparency creation regarding the current game state, development, and future directions; and opening up feedback channels for communication.

This study identified three major activity categories that partly address the same activities covered by Thominet (2018) and derived from user participation studies. Community management category mainly concerns game studios, developers, and community managers, and cover various activities that are related to building up, managing, and maintaining a player community. Communication category consists of activities and factors that concern the interaction between player community and game studio. Finally, testing category refers to the activities and factors that are related to players playtesting a game and providing feedback.

4.3.1 Community management

4.3.1.1 Recruiting players

The purpose of community management is to build and manage a functional and fruitful player community platform that benefits the game and its development process. Contrary to custom software development where future users of a specific system are usually known and provided by a client organization, in packaged software development (such as game development), the future customers are unknown and exists outside developers' control (Carmel and Becker, 1995). Community management thus starts with player acquisition in which game studio attempts to gather enough players to initiate player participation. There are multiple ways in which players can be sought to recruit for example the more traditional methods of advertising, buying exposure, and providing discounts, but also by creating game related consumable content that could grab players' attention.

> *Player acquisition is aimed at primarily by discounts and exposure campaigns within a store. These typically have large but short-term impact, whereas more low-profile methods (e.g. blogs, own YouTube videos, participating in various discussions in social media or internet forums), that are primarily aimed towards player retention, can also result in these smaller player flows.*
>
> *–Interviewee #4*

It is important to have an initial rough build of a game so that it can be used as a basis for the game related content creation to demonstrate potential players what the studio is working on.

> *If the community does not exist from before, then a you must have some kind of interesting early game build, the first version. If you manage to get interesting screen captures or video material from this, it is possible to acquire followers.*
>
> *–Interviewee #3*

The early game could also be provided free of charge for selected number of active players that would potentially be interested in it, to build the initial player community or distributed more freely to acquire wider sample of an audience.

> *Then those games are tested so that we message in social media that a game build has become available for testing and we would like to receive feedback on it. Then we can distribute APKs [Android packages] to families that have signed up [for testing]. That is also a thing we practice much.*
>
> *–Interviewee #2*

If the current game is not the first one of the game studio, it is possible that the initial community is founded by the members of the communities from previous games, provided that previous communities exists and that those players are aware of the studio and its upcoming game projects. Developers can also take it to the streets and attempt to recruit players in their natural habitat, such as one studio whose target audience were children, approached players through schools and kindergartens. However, this method maybe expensive and inefficient unless the players are already assembled in common reachable location, or the studio is capable of organizing an event that would gather interested players under the same roof. Finally, studios may attempt to gain exposure through 3rd party content creators or influencers, whose impact may sometimes surpass what the studio is capable of, but cannot be exclusively relied on bringing players in.

> *YouTubers also played that our lead developer's first game a little bit so we gained of course more exposure from it and in similar manner few YouTubers have played this game already since the ancient times.*
>
> *–Interviewee #4*

However, at the end of the day, it was mentioned that the most important thing is an interesting game concept. Unless a game is able to capture player's attention, player is likely to drop the game for not being worth their time as playing and participating is voluntary and the market is full of alternative titles competing for players' attention.

4.3.1.2 Enabling and maintaining participation

Once the initial community has gathered enough members, the game studio has to continuously seek to maintain participation. Participation cannot be taken for granted as it is voluntary and players may choose to exit the community at any time contrary to ISD context where users are usually obligated to participate. Enabling participation begins by

providing players with an access to the game and by opening feedback channels as per Thominet (2018) and is maintained by reacting to the received feedback.

> *[How do you maintain and encourage player participation?] By coming up with ways that can provide them [players] a feeling that they are truly included … By creating methods through which players can provide feedback and then also by implementing part of the good ideas, that are provided by the community.*
>
> *–Interviewee #3*

By showing appreciation for players' feedback and by implementing their suggested ideas, developers acknowledge players' contribution and reinforce the participation cycle.

> *"Seeing devs respond to player feedback is one of the best things I have seen during game development, it gives the players a feeling that the developers are actually taking their feedback into consideration instead of viewing them as consumers that can be taken advantage of for monetary gain."*
>
> *–Player*

At the same time, acknowledging certain kind of contribution can function as an example of desired and needed participation and can guide community's participation towards that specific direction.

> *In case someone of them [participating players] writes something that we consider to be a good idea, then we may include it as it is and if possible, give thanks to that individual for the idea because it is not only about rewarding that individual for the work they have done, but also it guides the community in a larger scale towards the kind of ideas that we would potentially grab, so it can mould the conversation more beneficial for us.*
>
> *–Interviewee #4*

As a counter example, ignoring players' feedback is likely to leave players irritated and confused as players do not receive response for their contribution, do not understand whether their participation has been appreciated or not, do not understand developers' thought process, nor are able to correct their participation practices. Eventually this leads

to players losing their faith in developers and the project, and are forced to abandon the game. Thus, two-way communication and interaction between the player community and game studio is required.

> *"Sometimes a developer just won't listen to what the community is saying. Life is Feudal and Ark both had some real issues with this during the development, and is why I stepped away from both of those projects before they were finished."*
>
> *–Player*

Game related content creation was already mentioned in 'recruiting players' section as method to attract new players to community, but it is also deemed as a viable method for player retention. By providing players with new game related consumable content on a regular basis, developers can keep players interested and engaged in the project as well as informed of the latest news. Transparency related to future development plans can also function as consumable content and basis for conversations within the player community allowing players to discuss and speculate about the future direction and state of the game.

> *Disclosing development plans as transparently and tangibly as possible provides players with something new to discuss about and helps to keep interest levels high.*
>
> *–Interviewee #4*

At the same time, content creation and content consumption can be used as a way to measure community's engagement levels with the project. Whenever a video is watched, a hyperlink is clicked, or a message is written, engagement event can be recorded and compared to previous engagement levels to understand the general direction the player community is taking, thus enabling game studio to notice shifts in player community trends and to react to them accordingly. Finally, the effects of being present in the player community should not be underestimated.

> *Our presence [in the player community] has clearly increased activity there in such a way that we show that we are there and we are often praised for being developers who are present and listen to what players are doing.*
>
> *–Interviewee #4*

Part of maintaining player participation is also about nurturing the player community. Occasionally some members of the community start acting mean towards other members or even towards the studio and their partners, hinder participation process, and spread toxicity around the community. To prevent this kind of behaviour, clear rules should be laid out for the community, and those who fail to follow the rules would be eventually removed from the community.

> *The effects of these [mean community members] are mitigated by laying out clear rules, that everybody is respected, critique is allowed but in respectful manner. No bad-mouthing. If a player does not listen, they are removed from the community.*
>
> *–Interviewee #3*

4.3.2 Communication

Communication is the activity of exchanging information and takes place between a game studio and its player community. According to players it is the most important activity and often referred as one of the upsides of player participation if executed correctly.

> *[What upsides/good things have you encountered in a game development project as a player?] "Being able to speak with developers while playing their game or project is incredibly enjoyable." – "Good and clear communication between developers and players. The feeling that players are understood and their opinion is important."*
>
> *–Player*

On the other hand, if communication is executed horribly, such as communication from game studio is non-existent, players' feedback and suggestions are ignored, or game studio overpromises things and fails to deliver, players start to lose their faith in the project and in the game studio and may even abandon them.

> *"Really the key is communication with the user base. And that's not just sending out request updates in what's going on. That requires actually communicating with your players, and having conversations with them."*
>
> *–Player*

However, it should be noted that acknowledging players' contribution does not mean that it could not be disagreed on. By acknowledging players' participation, developers create a sense of appreciation and show that players' contribution is valued which is likely to reinforce further participation. But developers can still disagree with players and reject their suggestions if those do not fit in the game vision and contribute to development goals. This is likely to leave the player disappointed but it enables the player to revise their participation approach and alter it to fit better the game development vision in the future, provided that developers' decision is rational and their train of thought has been made explicit to players. It is also better than ignoring completely as ignoring may be experienced in various ways, such as in insulting behaviour, depending on personal and cultural settings.

Communication itself can take various forms and happen through several mediums. In this study, communication is categorized according to its nature: whether it is active or passive communication. Respectively, Thominet (2018) discussed of communicative feedback and passive feedback, and Jacobs and Sihvonen (2011) of direct participatory design and silent participatory design. Active communication is used to refer to a type of communication in which both parties take engage in conversation actively and exchange ideas with each other. An example of active communication could be a face-to-face dialogue between two individuals, but it can also involve more people or take place over different mediums. In passive communication the information flows from one party to another without needing to engage in active information sharing activities. A real-world example of passive communication could be a use of data gathering and analysing software that transmits data from one party regarding one's activities and does not require any actions from that party.

4.3.2.1 Active communication

As said, active communication takes place when two or more parties engage in information sharing activities with each other over a specific medium. In active communication, one constructs the message and shares it to its receiver. It does not matter whether there are multiple receivers, or if the message is consumed immediately after receiving or after a while, or whether the message composer expects to receive an answer for the message. It should also be noted, that as the message is actively constructed by an individual, the contents of the message reflect that individual's perception of the reality. Therefore, as objective as possible of reality the message appears to be, it should be kept

in mind that the message is closer to an opinion or information rather than an absolute truth. On the other hand, the passive communication methods such as data gathering produce data on specific topic and can be considered more truthful, but do not produce information.

> *These things [passive communication methods] produce data but they do not actually provide information, also if we look at that retention, we may receive data that 40 % of players returned the next day and maybe 10 % after seven days, but it still does not give us information why this happens. It only provides us with the raw data that now this happens but no actual reason to explain why.*
>
> *–Interviewee #1*

Various active communication methods were identified in the study. Players most commonly participated by providing feedback on game experience, encountered bugs, and suggested features. Players also engaged in discussions with other players and game developers through online discussion forums. Often cited mediums for feedback and discussions were marketplace forums such as Steam forums, or other discussion platforms such as Discord and Reddit, but some studios had included within their game a possibility to contact and send message or feedback through an in-game messaging service. Although these kinds of services may be unique and more easily approachable to some players, the execution of this in-game service frames the quality and extent of feedback and may provide a limited understanding of community's stance (Thominet, 2017).

Besides the feedback and discussions, players also write game reviews on the marketplaces where the game was available. However, as game reviews are a description of a game at specific time and players seldomly write multiple reviews of a game, once received negative reviews may not be reviewed by players even if the state of a game has improved. Therefore, game developers should pay attention to the reception of a game since the beginning. However, Lin et al. (2018) suggest, based on study on Early Access games on Steam platform, that players do not post that much reviews during the 'early access' stage and usually tend to be more forgiving towards unfinished games.

Even though game studios can also participate in active communication through discussions and responding in feedback, they can also construct polls or surveys on various topics or try to lay out their development plans and vision in easily accessible public format, that can act as a reference point for players on various issues.

> *We also have this kind of public roadmap in which we list that these are known issues, these are issues that we will fix in the next patch or update, these are things that we will do soon, here are the things we will do a little later, and here are the things we hope that we can do at some point but we'll see. In other words, we aim to keep our plans also in this public format so that we can guide players there because we cannot always answer to all player feedback, so it is easier if we have our core plans somewhere visible.*
>
> *–Interviewee #4*

If players are proactive, they may link these public plans to each other in attempts to answer other players' questions. Public plans also provide a ground for further discussion among the community and may help to build more interest towards the project. However, if some issues that are in public knowledge have been left unaddressed for a longer period of time, it may place the game studio in awkward position.

4.3.2.2 Passive communication

Passive communication was already discussed by Thominet (2018) as passive feedback and defined as "any form of feedback from players that did not require direct or intentional communication". It mainly concerns data gathering through telemetry and analytics which can be used for example for adjusting game difficulty and balancing issues as well as observing game stability issues through crash reports. Jacobs and Sihvonen (2011) called similar trend as silent participatory design in which "the logged actions of all players who access the game world contribute to the decisions made concerning the game design" and talked about using it in identifying underlying issues in the game design such as overly difficult game levels.

> *We have this kind of system called Crashlytics which sends a report every time the game crashes and that way we can understand why the game crashes at that point, what is the issue, and we can find and understand where the bug is.*
>
> *–Interviewee #1*

As noted previously, the passive communication provides data but no information on chosen topic. The information must be constructed by an individual interpreting the data.

Even though this approach leaves more work to be done on developers' end rather than asking directly players what is the issue, passive communication may be more suitable option when attempting to construct an overall picture of the whole community such as what is the retention rate of a game, are some parts of the game more popular or difficult than others, and so on. It may also be a suitable approach when player community has significantly high participation threshold and are not engaging in discussions.

> *And because people do not express their opinion, game studios usually analyse players' behaviour. So actually, games call home and simply tell what a player is doing.*
>
> *–Interviewee #1*

In case the testing of a game is not conducted as remote testing but as local testing, it opens up new opportunities for game developers to observe players' behaviour and reactions towards the game. Player's behaviour could also be considered a form of passive communication, as player's reactions are likely to be genuine if a player is absorbed in the game.

However, passive communication may rise concerns of privacy and data protection. As a player does not take active part in passive communication, the border of consent for such activities are more elusive than when a player is actively taking part in a discussion. Ones attempting passive communication practices ought to follow the relevant data collection and protection regulations. Passive and active communication could also be sought to combine in a way that builds on their strengths. For example, passive communication could be used to uncover and pinpoint underlying issues in a game, and active communication could be used to verify these issues with the community as well as ask for suggestions to solve the issues.

4.3.3 Testing

In the testing phase, players play the game and contribute to development either through active or passive communication practices. Provided feedback may concern game experience or game functionality issues, or may contain ideas and suggestions for new functionality. There appears to be variation among player communities when it comes to personal testing practices: some players take it as fun and leisure practice and may not necessarily contribute actively, whereas others may take more professional approach and aim to test a game more extensively.

[Please describe the things you have done during your participation]
"Playing the games, noticing the mechanics and where are the upsides and downsides of the game design, where need some improvements."
–Player

Some players may even go so far as to push the limits of a game in an attempt to find exploitable faults or bugs in game design and implementation. That being said, the testing phase appears to be limited to players playtesting a game and the variation in this activity on players part appears to be related to how seriously players approach this task and what parts of a game or how extensively players choose to test.

4.3.3.1 Local testing

Testing practices can be divided into public and closed testing, as well as in local and remote testing practices. Differences between public and closed testing were already discussed by Gandolfi (2018), thus the differences between local and remote testing will be discussed here. In local testing, both players as well as representatives from a game studio gather at same place at specific time to test the game, whereas remote testing usually takes place over the internet at asynchronous times convenient for each party.

In local testing, testing is possible earlier in the development phase while the game is in more incomplete state because the game studio has more control over the test scenario and can guide players to focus their attention on specific features and can help troubleshooting should problems arise. Besides interviewing players during or after the playtest sessions, developers can also observe players' behaviour, facial expressions, reactions etc. towards the game and attempt to identify issues that would otherwise likely remain hidden in remote testing.

However, the earlier a testing is conducted the more limited testing scenarios are likely to be as the number of implemented features is likely to be low as well. Local testing can also be conducted later in the development phase when the game is at better state, but in the end local testing is still limited by the triple constraint of the number of testers, time available for each tester, and resources required to organize the testing event.

4.3.3.2 Remote testing

In remote testing, the game usually has to be in more robust condition than what would be possible in local testing, as the game studio does not have control over the test scenario and how players choose to test the game.

> *It [the game] has to be after all in such condition that the game experience does not crash within the first ten seconds.*
>
> *–Interviewee #2*

In addition, the game has to be more polished than what would be acceptable in local testing, even though it would be useful to acquire feedback as early as possible in the development project.

> *We would like to push our game out as early as possible but there is no sense in pushing it out too early unless there are enough things in place, unless it is meaningful to test it.*
>
> *–Interviewee #1*

There has to be enough features and content implemented for players to even bother getting interested in the game.

> *"Sometimes when devs release it with literally no content just to be able to develop I always wait to get cause there's no reason I should pay 30 bucks in hopes you don't run out on us or even make a decent product."*
>
> *–Player*

The user interface and the controls of the game also have to be functional and practical, so that the interaction with the game is seamless.

> *Also, the user interface of the game has to be enough clear so that the game is somehow playable.*
>
> *–Interviewee #3*

The benefit remote testing appears to have over local testing is that it enables access to multiple number of players less expensively. Players can also playtest the game at their own pace in an environment where they feel comfortable, allowing potentially more relaxed and deeper interaction with the game and gives players more time to explore and reflect on the gameplay experience before sharing it.

> *Because then [during remote testing] we get different kind of information on their playing habits than when we are observing in person because the parents write about the experiences what the children have shared, which arises from the fact that they have time to talk about it and play more and then comes up completely different things.*
>
> *–Interviewee #2*

4.4 Outcomes of player participation

4.4.1 Benefits

The benefits often cited and associated with user participation in ISD were usually related in building a better system through better and accurate system requirements (Damodaran, 1996; Kujala, 2003; Kujala et al. 2005; McGill and Klobas, 2008) or avoiding unnecessary system features whose development would be a waste of resources (Damodaran, 1996; Kujala, 2003); thus factors that would contribute to system development success (Markus and Mao, 2004). On the other hand, the benefits were also related to factors that would contribute to system implementation success (Markus and Mao, 2004) such as facilitating acceptance of a new system (Damodaran, 1996; Kujala, 2003), overcoming user resistance (Cavaye, 1995), increasing user satisfaction, individual impact (McGill and Klobas, 2008), system usage (Baroudi et al. 1986), understanding of the system (Damodaran, 1996) and making the work environment more democratic through increased participation in decision-making (Damodaran, 1996; Kujala, 2003). In addition to player participation being beneficial for the development of a game, benefits of player participation were also found in the game itself as well as in the business aspect of game development.

4.4.1.1 Development benefits

The most self-explanatory benefits of player participation for the development of games are additional available resources for testing games as well as feedback, ideas, and suggestions received from the player community. More available resources mean that more people are looking out for possible bugs and game design flaws in the game thus allowing the game studio to use fewer resources on testing the game and shifting those resources to other necessary development tasks. In this sense, involving players in game development functions more or less as outsourcing the testing phase.

> *If we had no players playtesting [our game], then we would have to use more time in this house for testing, maybe procure it as outsourcing service. Of course, our publisher also does it but our resources would not last at all without our community for testing this game in a similar vein it is currently being tested.*
>
> *–Interviewee #4*

Besides utilizing the player community as a form of quality assurance, players are also eager to express their opinions of the game, provide feedback of its current state, and ideas and suggestions for any future course of development. Having uncompromised audience pointing out strengths and weaknesses of a game and contributing gameplay enhancing ideas is an invaluable asset when aiming for better games. Player community can also function as a platform for a game studio to validate development and game design ideas. By checking ideas with the community before and during implementation, developers may potentially save resources if they manage to avoid implementing features that would not contribute to better gameplay experience. Respectively in situations of greater uncertainty, developers may consult the player community to gain more confidence for development and facilitate decision making. In this sense, involving players in game development can function as a risk mitigation method during the development process.

> *Competitive advantage is also that we know what we are doing because we have like lights on and we are not travelling in the dark and trying to guess where we are going. – Even though in the end development is always done on [development] team's terms and not on players' demands, we naturally feel more self-confident about new content and functionality when we know it is a change desired by the players. Lights on means exactly this: We can validate development ideas before we start to implement them and also during the development process.*
>
> *–Interviewee #4*

Also, the reception of the game and received feedback may have positive effects on game studio employees' work morale and coping at work. However, it is a double-edged sword and may also have negative effects if feedback and reception is negative. The lessons learned and experiences from involving players can also be utilized in developing the following game titles and benefit game development retrospectively.

4.4.1.2 Benefits for game

Benefits that player participation brings for a game itself are mostly related to creating more targeted gameplay experience, and making the game technically more robust. Implementing features desired by players and developing the game towards the direction that pleases its player community targets the game and its gameplay experience at its existing audience; that is, the game fulfils better the needs and desires of its player community. How this appears in practice is entirely dependent on the feedback received. It may concern for example finetuning the general level of difficulty in a game, creating new complementary features, removing annoying features or making them optional, adjusting the user interface and gameplay controls, catering to different kind of player types, creating better rewarding gameplay experience, and so on. It can also help in identifying and correcting overlooked issues in game design such as features that should have been there but were potentially forgotten by developers or not deemed necessary.

More technically robust game refers to the game being more functionally stable as a software. Whereas the previous benefit was concerned about the contents of a game, this benefit covers the technical implementation aspect. When players report bugs while playtesting a game, they assist in making the game better not only for themselves but also for every other player. Bugs itself may, for example, be associated with graphical or audio related issues; bugs that break gameplay mechanics, prevent players from progressing, or allow players to play the game in unintended ways; or bugs that break the gameplay experience completely by crashing the game or by corrupting saved game data.

4.4.1.3 Business benefits

The business benefits of player participation are mostly related to more efficient development and potentially better sales. As noted in the development benefits section, involving players enables game studios to use fewer own resources for testing games. Validating ideas with the community and having them to tell what features they would like to see implemented, can make development more straightforward and remove uncertainty when developers do not need to consume too many resources on considering what players want.

> *On the other hand, we can also do things more cost-effectively because we are able to use our players in helping us with the development.*
>
> *–Interviewee #4*

Having a large community backing the development of a game is also beneficial if applying for financing from a publisher.

> *[Do you feel it has been easier to acquire financing or publishing on the basis that you already have this strong and functional player community?] It affected for sure in getting this publishing contract, in other words, it has been a selling point that we have a game that is unfinished but it has this and this many followers, our game has been added on the wishlist in Steam this and this many times even though we have never spent money on marketing because we have players so, it has affected yes.*
>
> *–Interviewee #4*

However, it is unclear what are the net benefits as player participation gives its rise to community management related tasks that brings its challenges.

The potential for better sales is attributed to the possibility of increased visibility as well as positive word of mouth. If a game is developed by listening to feedback from the community and implementing changes desired by players, the likelihood that players like the game and the game is generally well-received increases. If a game is liked and well-received, it is likely to be played a lot and receive favourable reviews, that are potential factors in marketplace algorithms determining exposure and featuring of games. Increased exposure then again acts as a platform for better sales.

> *Many factors are affecting how well games sell but one essential factor is players' feedback as reviews. If a game receives good reviews then the algorithms of the game marketplace favour this game in terms of gaining exposure. – Also, it is very likely that for example, Steam's algorithms favour games, that are being played for longer.*
>
> *–Interviewee #3*

It is also likely that favoured games are shared and recommended to friends whereas bad and disliked games are shunned and recommended to avoid.

Besides these benefits, it is also possible that well-managed player participation creates satisfied and committed player community that may become interested of the next game titles by the same studio, thus building a sense of continuity and fan base around the game studio itself.

4.4.2 Challenges

Besides the various potential benefits that can emerge from involving players, it is also prone to its challenges that must be taken into account when attempting to organize functional player participation. Those challenges somewhat follow the challenges present in user participation such as selecting users being challenging, users being unpredictable, introducing ideas and proposing changes too late, users' lack of understanding of development, problems in communication, participation process being work-intensive, and potential conflicts between participating parties.

However, before the player participation has even started, developers have their first challenge in determining whether the game is in such condition that it is ready to be handed out to the public. Besides being in enough stable condition so that the game does not crash every once in a while, the game must also include sufficient amount of content for testing purposes as well as to keep players interested and engaged. Even though under those circumstances, releasing too early rather than too late would perhaps be more disadvantageous, the earlier player feedback can be incorporated in development the better.

The next challenge lies in creating a functional player community. In the current competitive video game industry, studios are not only competing against each other for players but also against other forms of leisure. Gaining exposure for own game in this competitive world requires more and more creative ways. As playing and participating is a voluntary act and there is a plethora of games available for players, acquiring players may prove to be difficult. Not every player is even willing to purchase or play an unfinished product. Even if a studio manages to acquire a decent number of players for their community, another challenge is to get those players to participate actively and contribute to development let alone to keep them in the community.

> *Nowadays it is difficult to get anybody to even play a free game and it is really difficult to get players to provide feedback or remain in the community.*
>
> *–Interviewee #3*

Even if a game studio manages to gather a community of players around their game, it is unlikely that every player would participate and contribute to game development consistently. In these scenarios, the participating portion may not represent the whole community completely thus leading to the uneven representation problem. At worst, this

may skew developers' understanding of the community and lead to developing the game according to the views of the vocal portion of the community, neglecting the more silent portion. As it is likely to be in the interests of a game studio to cater to all players in the community and not to discriminate nor favour any player type, a challenge for game studio lies in determining what to do with the more silent community and how much weight should be given for the feedback and suggestions by the more vocal community.

Another factor, that may hinder players' quality of participation, is players' lack of understanding of game development and game design. Players are unlikely to understand what goes into developing a functional game and how it should be designed for the game to be compelling and interesting unless a player has previous experience with game development or player participation.

> *A dream picture of a game idea always springs to people's minds if they get really excited of it. The problem is that a regular player cannot comprehend at all nor evaluate a feature from every aspect.*
>
> *–Interviewee #3*

This lack of understanding may have various effects such as players ignore or may not notice issues that could be corrected and accept them as they are, or players fail to provide any meaningful feedback that would contribute to development.

> *"A lot of discussion is plagued by rants and non-constructive feedback, e.g. bug x appears please fix, or item y is OP [over powered] game is literally unplayable."*
>
> *–Player*

Furthermore, due to the heterogeneous nature of player communities, players have diverse and differing views on what constitutes as fun gameplay which also reflects in the feedback that players provide. Developers cannot blindly follow every suggestion without risking to lose the sense of development direction and compromising the success of development. Therefore, developers have to evaluate, analyse, and prioritize suggestions coming from the player community in terms of the game vision, which is a factor in increasing the workload. Also, the fact that sometimes players are very eager to produce suggestions etc. does not alleviate the workload at all.

> *Not everything can be implemented. You must be very selective. If you fail here, it may be that the game is never finished. Experience is important because it helps to perceive if there is any sense in a requested feature. Developer must have a strong vision of what benefits the game and what does not. Players come up with very arbitrary ideas, yet it is not good for the game to include every one of them. Only features that support the game idea are worth implementing. – On the other hand, there may be six other good features on a list, and these must be prioritized. You must also attempt to perceive how much time is required for implementation and what it gives to players in reality.*
>
> *–Interviewee #3*

Some challenges may arise in everyday interaction between people. Conflicts emerge between players due to differing opinions and requires managing, misbehaving players in the community must be kept in check or removed, and how to communicate issues that would cause dissatisfaction or disappointment among the player community such as undesired changes, delays in plans, or even broken promises. Also, in the case of passive communication, issues such as privacy and data protection must be taken into account.

4.4.3 Other outcomes and effects

The way player participation moulds game development is not only limited to benefits and challenges but it has a major impact in restructuring the work practices as well. First of all, the players' messages, feedback, and discussions increase the amount of communication activities. Hiring a community manager appears to be a common way to manage this increased workload. Secondly, it forces a customer-oriented approach into development: developers are no longer isolated from players and are required to manage their relationship with the player community. Players expect that their presence and contribution is acknowledged at a minimum. Otherwise, game studio may risk alienating their hard-earned players. Lastly, player participation appears to shift the emphasis of game development from the artistic and creative end towards more organized software like development. Whereas it may not be common in creative industries to let the audience experience unfinished products, in player participation it is a must and it may require developers and artists to re-orientate themselves to the new situation in which players question decisions and sometimes harshly criticize creations.

> *For the developers, there has also been a little bit learning what it means that the own unfinished work is out there in the open and people comment it. So in a way, you have to hold on to your own leadership after all and not get confused nor get disturbed of the discussion – you don't have much privacy to which developers have maybe become accustomed to, that you get to work in your own silent space, and then the PR-side and all the publicity is somebody else's trouble, but now they are approaching that side and I know that some have experienced it as disturbing, that it is weird to get feedback from the unfinished thing and maybe people become afraid to experiment with new things.*
>
> *–Interviewee #4*

One of the goals of this study was to also examine whether player participation would have any potential positive effects on the common problems in the game development industry, namely: unrealistic scope, feature creep, crunch time, cutting off features and so on. As a surprise, it appears that in some occasion player participation tends to amplify certain problems rather than diminish them. For example, the inclusion of players is ideal to promote feature creep when players introduce new ideas and suggestions to expand gameplay experience. In this case, developers need to keep the game vision constantly in their mind and pick those ideas that contribute to it. Otherwise, developers risk expanding their original game scope to a size that is not feasible to execute or may run into other problems such as not keeping to development schedule or having to cut off some features due to schedule and budgeting reasons.

Besides resource pressures, cutting off features may also occur due to players introducing substituting ideas that are better than the originals, or a majority of players desiring certain feature whose implementation is only possible by taking resources away from implementing some other feature.

> *If there is a feature that players desire a lot then it is really difficult to cut it off but on the other hand, if cutting off depends on something else than budget, for example, it depends on that maybe the feature was not such a good idea after all then that idea might have come from players as well.*
>
> *–Interviewee #4*

Scheduling problems depend on what or who sets the scheduling constraints. If the schedules are publisher, seasonal sales, or budget-related, player participation is likely to have some effects through feature creep. In the absence of such constraints, if game studio finances development with income received from selling and distributing the unfinished game, the budget may be more flexible and the schedule not so fixed. However, as this form of financing is dependent on how well the game succeeds, it should be not relied as a primary form of funding. Another twist that player participation brings to scheduling is that sometimes players are impatient for new content and eager to remind of missed deadlines. A recommendation by one interviewee was to not give any information in numeric form to players as those can backfire and can always be used against the game studio.

> *These dates we do not communicate to players. We have given estimates before to players for example of when the game would be available on Steam Early Access and when it was delayed, we noticed that people were not very pleased of it. But usually, we do not share any information in numeric form.*
>
> *–Interviewee #4*

Even though player participation appears to amplify certain problems one interviewee assessed that the effects would be temporary and diminish over time once player participation has been integrated into game studio's processes and workflow.

> *I think it [player participation] may amplify them [problems in game development industry] but once you have reached the kind of "we can" point in community management, then you can probably utilize it smartly in your normal workflow and maybe it alleviates certain problems but it requires processes and that you are capable of involving players. That you include the community does nothing immediately but you have to be able to integrate it as a part of your activities.*
>
> *–Interviewee #4*

Finally, there are always surprising outcomes. One studio had visited a kindergarten to test their game with children and to gather valuable feedback. Later they learned that the children had liked it so much that they started to do various game characters-related handicrafts, which settled the game characters in the daily life of the kindergarten.

5 CONCLUSION

The purpose of this research was to study the recent phenomenon of player participation in game development to build a general level understanding and to discover whether there are any real benefits in it to justify its existence among the more established game development methods. Contrary to the prior literature that appeared to emphasize development perspective, a different approach was adopted in this study to emphasize the collaborative nature of player participation. Therefore, player participation is defined as:

> *A collaborative game development method characterised by voluntariness, and emphasizing communication, and interaction between the player community and the game studio for creating a gameplay experience according to the emerging discourse between game studio's development vision and players' desires.*

The activities identified in this study complement the previously identified activities in the literature and provide a more comprehensive picture of player participation. Game studio –initiated player participation begins with community management followed by communication and testing activities. These activities are continuous and exclusive meaning that community management is mainly handled by a game studio, and testing is majorly done by the player community, whereas both parties are involved in communication activities. The previously identified activities fit into these categories, where providing access to the game could be considered to be one of the initial activities of community management, and creating transparency of the state of the game development process as well as gathering feedback would be related to the communication activity.

To summarize the outcomes, player participation appears to be beneficial for game development assuming that a game studio can manage the accompanying challenges. Via the testing and communication activities player participation benefits game development and its objectives with more efficient and smarter development, with more robust and targeted gameplay experience, and with potential for more favourable reception and increased exposure. The challenges are mainly related to creating, maintaining and managing a player community, persuading players to participate and gathering representative, constructive feedback, as well as managing increased workloads. Player participation may also amplify certain problems that are prevalent in the game development industry, such as feature creep. The findings of this study are summarized in Figure 5.

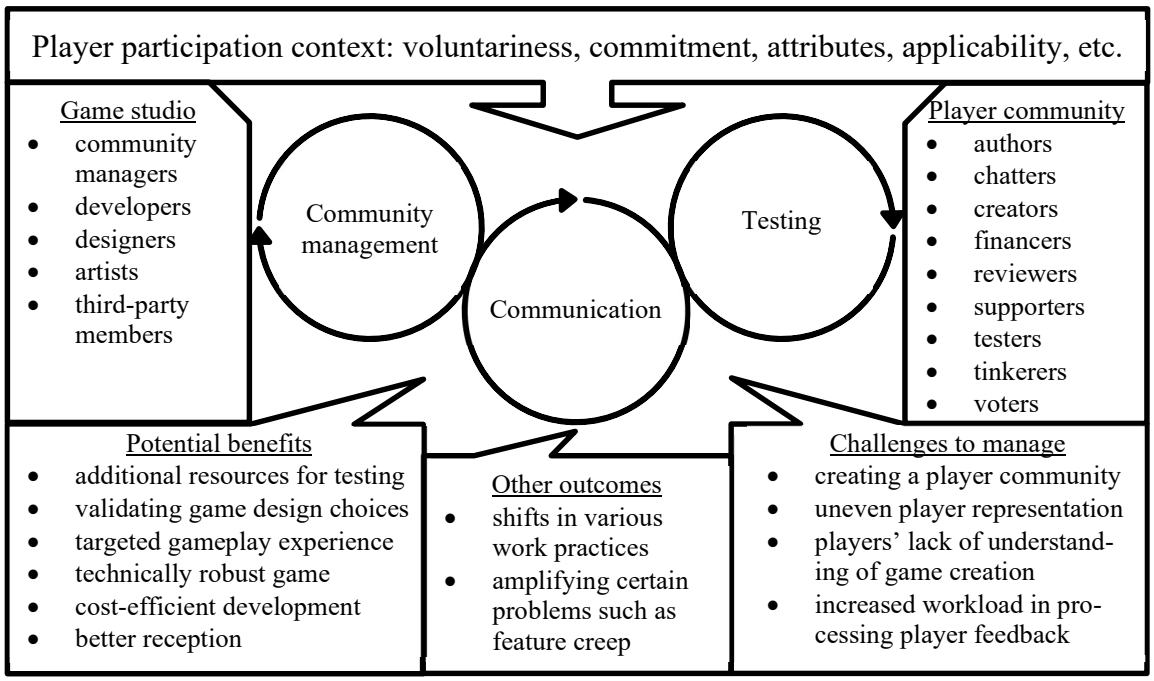

Figure 5 Summary of player participation

The benefits and challenges are similar to those of user participation with the small differences most likely attributable to the voluntariness nature of player participation and the fundamental difference in the purpose of information systems and games. Both approaches benefit stakeholders with more functional and targeted end-product. However, it seems that in player participation targeting through testing and incorporating feedback into the game carries more weight in relation to implementing players' feature suggestions, whereas in user participation it seems that users influence the development more through the requirements specifications phase in relation to testing which is utilized to validate that requirements are met. Potential reasons for this are that in ISD users supposedly are the experts of their domain and can better instruct system developers of what the system is supposed to do thus granting users more influence before the system development phase. However, in the game development players may not understand what goes into developing and designing a game to provide meaningful suggestions before the development and players may have diverse and occasionally conflicting views on what direction the game ought to take, making it potentially more productive for a game studio to follow their initial game vision and adjust it according to the emerging feedback.

User participation is also an important part in inducing user satisfaction and system acceptance towards a new system to counter potential problems emerging if users are forced to participate or utilize the new system. However, since player participation is

voluntary players may drop the game anytime thus decreasing the relevancy of such benefits in the game development context. On the other hand, voluntariness amplifies certain challenges such as maintaining participation that are perhaps less prevalent or non-existent in ISD context.

As for practical implications, a point has been presented here for the favour of player participation. Besides the potential benefits, player participation may be considered as an alternative risk management tool to the shift towards rationalization in game development practices, in which well-tried game genres and mechanics are favoured at the cost of creativity (Tschang, 2007). Through player participation, gameplay experiences can be targeted to better suit target audiences' preferences and thus can mitigate the risks of failure due to bad reception perhaps similarly or maybe even better than relying on old and proven game concepts. It also enables game studios to retain more creative control over the game compared to the state of rationalization. However, the potential downsides are that player participation may be more work-intensive in certain aspects and that it may push work practices towards more organized system development related practices such as revealing unfinished creations and encouraging feedback rather than focusing completely on artistic creative practices. Also, player participation may not be applicable in every situation and therefore future research should focus on identifying game or development related factors that determine the usefulness of player participation development approach. Future research could also focus on expanding various areas identified in this study, for example, it would be interesting to understand what it is the meaning of players' participation motives in their contribution value for the game development project i.e. do players with more collective participation motives shape the player community and game development project differently than players with more individualistic motives.

This study was limited essentially by two factors. The lack of prior research led to adopting a more general approach to the research topic and search for alternative literature from which to draw ideas. Therefore, the foundation of this study was not the most suitable. Secondly, the qualitative data used in this research was limited by the number of interviews (four) and the geographical location for the interviews (Finland). Players answers for the questionnaire also included high amounts of variation in terms of quality and a relatively small number of respondents (n=21). However, players who participated in the questionnaire represented various nationalities mitigating the potential impact of cultural differences. Future research could also attempt to validate the claims of this study with a wider sampling of interviewees and respondents.

REFERENCES

Abelein, Ulrike – Pacch, Barbara (2015) Understanding the influence of user participation and involvement on system success – a systematic mapping study. *Empirical Software Engineering*, Vol. 20 (1), 28—81.

Aleem, Saiqa – Capretz, Luiz Fernando – Ahmed, Faheem (2016a) Critical success factors to improve the game development process from a developer's perspective. *Journal of Computer Science and Technology*, Vol. 31 (5), 925—948.

Aleem, Saiqa – Capretz, Luiz Fernando – Ahmed, Faheem (2016b) Empirical investigation of key business factors for digital game performance. *Entertainment Computing*, Vol. 13 (1), 25—36.

Alex Wawro (2016) GOG launches its own take on Steam's Early Access service. <https://www.gamasutra.com/view/news/264517/GOG_launches_its_own_take_on_Steams_Early_Access_service.php>, retrieved 9.9.2019.

Amaya, George – Davis, John P. – Gunn, Daniel V. – Harrison, Chuck – Pagulayan, Randy J. – Phillips, Bruce – Wixon, Dennis (2008) Games user research (GUR): Our experience with and evolution of four methods. In: *Game usability: Advice from the experts for advancing the player experience*, eds. Katherine Isbister – Noah Schaffer, 35—64. Elsevier, Burlington.

Ampatzoglou, Apostolos – Stamelos, Ioannis (2010) Software engineering research for computer games: A systematic review. *Information and Software Technology*, Vol. 52 (9), 888—901.

Arafat, Mais – Qusef, Abdallah – Al-Taher, Samar (2019) Steam's Early Access model: A study on consumers' perspective. In: *2019 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT)*, 9—11 April 2019, Amman, Jordan, Jordan, 336—341.

Axtell, C. M. – Waterson, P. E. – Clegg, C. W. (1997) Problems integrating user participation into software development. *International journal of human-computer studies*, Vol. 47 (2), 323—345.

Bano, Muneera – Zowghi, Didar (2013) User involvement in software development and system success: A systematic literature review. In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, Porto de Galinhas, Brazil, April 14-16, 2013, 125—130.

Barki, Henri – Hartwick, Jon (1989) Rethinking the concept of user involvement. *MIS Quarterly*, Vol. 13 (1), 53—63.

Barki, Henri – Hartwick, Jon (1994a) Measuring user participation, user involvement, and user attitude. *MIS Quarterly*, Vol. 18 (1), 59—82.

Barki, Henri – Hartwick, Jon (1994b) User participation, conflict, and conflict resolution: The mediating roles of influence. *Information System Research*, Vol. 5 (4), 422—438.

Barki, Henri – Hartwick, Jon (2001) Communication as a dimension of user participation. *IEEE Transactions on Professional Communication*, Vol. 44 (1), 21—36.

Baroudi, Jack J. – Olson, Margrethe H. – Ives, Blake (1986) An empirical study of the impact of user involvement on system usage and information satisfaction. *Communications of the ACM*, Vol. 29 (3), 232—238.

Batchelor, James (2018) Games industry generated $108.4bn in revenues in 2017. gamesindustry.biz. <https://www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017>, retrieved 25.1.2019.

Blow, Jonathan (2004) Game development: Harder than you think. *Queue – Game Development*, Vol 1 (10), 28—37.

Callele, David – Neufeld, Eric – Schneider, Kevin (2005) Requirements engineering and the creative process in the video game industry. In: *13th IEEE international conference on requirements engineering (RE'05)*, Paris, France, August 29- September 2, 2005, 240—250.

Callele, David – Neufeld, Eric – Schneider, Kevin (2006) Emotional requirements in video games. In: *14th IEEE International Requirements Engineering Conference (RE'06)*, 11-15 Sept. 2006, Minneapolis/St. Paul, MN, USA.

Callele, David – Neufeld, Eric – Schneider, Kevin (2008) Emotional requirements. *IEEE Software*, Vol. 25 (1), 43—45.

Callele, David – Neufeld, Eric – Schneider, Kevin (2010) An introduction to experience requirements. In: 18th IEEE International Requirements Engineering Conference, 27 Sept.-1 Oct. 2010, Sydney, NSW, Australia.

Carmel, Erran – Becker, Shirley (1995) A process model for packaged software development. *IEEE transactions on engineering management*, Vol. 42 (1), 50—61.

Cavaye, Angèle L.M. (1995) User participation in system development revisited. *Information & Management*, Vol. 28 (5), 311—323.

Cheung, Gifford – Zimmermann, Thomas – Nagappan, Nachiappan (2014) The first hour experience: How the initial play can engage (or lose) new players. In: *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, 57—66.

Cox, Joe (2014) What makes a blockbuster video game? An empirical analysis of US sales data. *Managerial and Decision Economics*, Vol. 35 (3), 189—198.

Damodaran, Leela (1996) User involvement in the systems design process - a practical guide for users. *Behaviour & Information Technology*, Vol 15 (6), 363—377.

Davis, John P. – Steury, Keith – Pagulayan, Randy (2005) A survey method for assessing perceptions of a game: The consumer playtest in game design. *The international journal of computer game research*, Vol. 5 (1).

DeLone, William H. – McLean, Ephraim R. (1992) Information systems success: The quest for the dependent variable. *Information Systems Research*, Vol. 3 (1), 60—95.

DeLone, William H. – McLean, Ephraim R. (2002) Information systems success revisited. In: *Proceedings of the 35th Hawaii international conference on system sciences*, 10-10 Jan. 2002, Big Island, HI, USA.

Doll, William J. – Deng, Xiaodong (2001) The collaborative use of information technology: End-user participation and system success. *Information Resources Management Journal*, Vol. 14 (2), 6—16.

Draper, Stephen W. (1999) Analysing fun as a candidate software requirement. *Personal Technologies*, Vol. 3 (3), 117—122.

Dyne, Linn Van – Pierce, Jon L. (2004) Psychological ownership and feelings of possession: three field studies predicting employee attitudes and organizational citizenship behaviour. *Journal of Organizational Behavior*, Vol. 25 (4), 439—459.

Early Access FAQ (2019) Valve Corporation. <https://store.steampowered.com/earlyaccessfaq/>, retrieved 9.9.2019.

Eriksson, Päivi – Kovalainen, Anne (2016) *Introducing Qualitative Methods: Qualitative methods in business research, 2nd ed*. SAGE Publications, London.

Gallivan, Michael J. – Keil, Mark (2003) The user-developer communication process: a critical case study. *Information Systems Journal*, Vol. 13 (1), 37—68.

Gandolfi, Enrico (2018) Playing, debugging, learning: A proposal between game and instructional designs via extended prototyping. *E-Learning and Digital Media*, Vol. 15 (2), 67—92.

Gioia, Dennis A. – Corley, Kevin G. – Hamilton, Aimee L. (2012) Seeking qualitative rigor in inductive research: Notes on the Gioia methodology. *Organisational Research Methods*, Vol. 16 (1), 15—31.

Handrahan, Matthew (2018) Blizzard's Diablo mobile game meets community backlash. gameindustry.biz. <https://www.gamesindustry.biz/articles/2018-11-05-blizzards-diablo-mobile-game-meets-community-backlash>, retrieved 26.1.2019.

Hansson, Christina – Dittrich, Yvonne – Randall, Dave (2006) How to include users in the development of off-the-shelf software: A case for complementing participatory design with agile development. In: *Proceedings of the 39th annual Hawaii international conference on system sciences (HICSS'06)*, 4—7th January 2006, Kauia, HI, USA, USA.

Harris, Mark A. – Weistroffer, H. Roland (2009) A new look at the relationship between user involvement in systems development and system success. *Communications of the Association for Information Systems*, Vol. 24 (1), 739—756.

Hartwick, John – Barki, Henri (1994) Explaining the role of user participation in information system use. *Management Science*, Vol. 40 (4), 440—465.

He, Jun – King, William R. (2008) The role of user participation in information systems development: Implications from a meta-analysis. *Journal of Management Information Systems*, Vol. 25 (1), 301—331.

Heinbokel, Torsten – Sonnentag, Sabine – Frese, Michael – Stolte, Wolfgang – Brodbeck, Felix C. (1996) Don't underestimate the problems of user centredness in software development projects – there are many! *Behaviour & Information Technology*, Vol. 15 (4), 226—236.

Holmström, Helena (2001) Virtual communities as platforms for product development: An interpretive case study of customer involvement in online game development. In: *Proceedings of 22nd international conference on information systems*, 299—306.

Hwang, Mark I. – Thorn, Ron G. (1999) The effect of user engagement on system success: A meta-analytical integration of research findings. *Information & Management*, Vol. 35 (4), 229—236.

Ives, Blake – Olson, Margrethe H. (1984) User involvement and MIS success: A review of research. *Management Science*, Vol.30 (5), 586—603.

Jacobs, Melinda – Sihvonen, Tanja (2011) In Perpetual Beta? On the Participatory Design of Facebook Games. *Proceedings of DiGRA 2011 Conference: Think Design Play*.

Jöckel, Sven – Will, Andreas – Schwarzer, Florian (2008) Participatory media culture and digital online distribution – Reconfiguring the value chain in the computer game industry. *The International Journal on Media Management*, Vol. 10 (3), 102—111.

Kabbedijk, Jaap – Brinkkemper, Sjaak – Jansen, Slinger – Veld, Bas van der (2009) Customer involvement in requirements management: Lessons from mass market software development. *2009 17th IEEE international requirements engineering conference*, 31 Aug.-4 Sept. 2009, Atlanta, GA, USA.

Kanode, Christopher M. – Haddad, Hisham M.(2009) Software engineering challenges in game development. *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*, 260—265.

Kappelman, Leon A. – McLean, Ephraim R. (1991) The respective roles of user participation and user involvement in information system implementation success. In: *Proceeding ICIS '91 Proceedings of the twelfth international conference on Information systems*, New York, USA, 339—349.

Karlsson, Lena – Dahlstedt, Åsa G. – Natt och Dag, Johan – Regnell, Björn – Persson, Anne (2002) Challenges in market-driven requirements engineering - an industrial interview study. *Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality*, 37—49.

Karlsson, Lena – Dahlstedt, Åsa G. – Natt och Dag, Johan – Regnell, Björn – Persson, Anne (2007) Requirements engineering challenges in market-driven software development - An interview study with practitioners. *Information and Software Technology*, Vol. 49 (6), 588—604.

Kasurinen, Jussi – Laine, Risto (2014) Games from the viewpoint of software engineering. <http://www2.it.lut.fi/GRIP/publications/YTP_KasurinenLaine.pdf>, retrieved 25.4.2019.

Kasurinen, Jussi – Smolander, Kari (2014) What do game developers test in their products? In: *Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement (ESEM '14)*, Torino, Italy, September 18-19, 2014, article no. 1.

Kasurinen, Jussi – Maglyas, Andrey – Smolander, Kari (2014) Is requirements engineering useless in game development? In: *Requirements engineering: Foundation for software quality: 20th international working conference, REFSQ 2014*, Essen, Germany, April 7-10, 2014, 1—16.

Kasurinen, Jussi (2016) Games as software - Similarities and differences between the implementation projects. *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, 33—40.

Kerr, Aphra (2006) The business of making digital games. In: *Understanding Digital Games*, eds. Jason Rutter – Jo Bryce, 42—57. Sage Publications, London.

Kirriemuir, John (2006) A history of digital games. In: *Understanding Digital Games*, eds. Jason Rutter – Jo Bryce, 31—41. Sage Publications, London.

Kuchera, Ben (2018) Report: 7,672 games were released on Steam in 2017. Polygon. <https://www.polygon.com/2018/1/10/16873446/steam-release-dates-2017>, retrieved 25.1.2019.

Kujala, Sari (2003) User involvement: a review of the benefits and challenges. *Behaviour & Information Technology*, Vol. 22 (1), 1—16.

Kujala, Sari (2008) Effective user involvement in product development by improving the analysis of user needs. *Behaviour and Information Technology*, Vol. 27 (6), 457—473.

Kujala, Sari – Kauppinen, Marjo – Lehtola, Laura – Kojo, Tero (2005) The role of user involvement in requirements quality and project success. *Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering (RE'05)*, 75—84.

Laukkanen, Tero (2005) *Modding scenes: Introduction to user-created content in computer gaming.* Hypermedia Laboratory Net Series 9, University of Tampere.

Lewis, Chris – Whitehead, Jim (2011) The whats and the whys of games and software engineering. *Proceedings of the 1st International Workshop on Games and Software Engineering (GAS '11)*, 1—4.

Lin, Dayi – Bezemer, Cor-Paul – Hassan, Ahmed E. (2018) An empirical study of early access games on the Steam platform. *Empirical Software Engineering*, Vol. 23 (2), 771—799.

Maalej, Walid – Happel, Hans-Jörg – Rashid, Asarnusch (2009) When users become collaborators: Towards continuous and context-aware user input. In: *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, 981—989.

Markus, M. Lynne – Mao, Ji-Ye (2004) Participation in development and implementation – Updating an old, tired concept for today's IS contexts. *Journal of the Association for Information Systems*, Vol. 5 (11—12), 514—544.

McAllister, Graham – White, Gareth R. (2010) Video game development and user experience. In: *Evaluating User Experience in Games,* ed. Regina Bernhaupt, 107—128. Springer, London.

McGill, Tanya – Klobas, Jane (2008) User developed application success: sources and effects of involvement. *Behaviour & Information Technology*, Vol 27 (5), 407—422.

McKeen, James D. – Guimaraes, Tor – Wetherbe, James C. (1994) The relationship between user participation and user satisfaction: An investigation of four contingency factors. *MIS Quarterly*, Vol 18 (4), 427—451.

McKeen, James D. – Guimaraes, Tor (1997) Successful Strategies for User Participation in Systems Development. *Journal of Management Information Systems*, Vol 14 (2), 133—150.

Meyer, John P. – Allen, Natalie J. (1991) A three-component conceptualization or organizational commitment. *Human Resource Management Review*, Vol. 1 (1), 61—89.

Mollick, Ethan (2013) The dynamics of crowdfunding: An exploratory study. *Journal of Business Venturing*, Vol. 29 (1), 1—16.

Murphy-Hill, Emerson – Zimmermann, Thomas – Nagappan, Nachiappan (2014) Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? *In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, 1—11.

Musil, Juergen – Schweda, Angelika – Winkler, Dietmar – Biffl, Stefan (2010) *A survey on the state of the practice in video game software development*. Technical report no. IFS-QSE 10/04. Institute of software technology and interactive systems Vienna university of technology.

Neogames – Industry operators (2020). <https://www.neogames.fi/en/industry-info/operators/>, retrieved 30.3.2020.

O'Hagan, Ann Osborne – Coleman, Gerry – O'Connor, Rory V. (2014) Software development processes for games: A systematic literature review. In: *21st European conference on systems, software and services process improvement (EuroSPI 2014),* CCIS Vol. 425, Springer-Verlag, June 2014, 182—193.

Oxford English Dictionary (2020) Definition of the word 'dedication'. <https://www.oed.com/view/Entry/48552?redirectedFrom=dedication#eid>, retrieved 14.2.2020.

Pagulayan, Randy J. – Keeker, Kevin – Fuller, Thomas – Wixon, Dennis – Romero, Ramon L. – Gunn, Daniel V. (2012) User-centered design in games. In: *The human-computer interaction handbook: Fundamentals, evolving technologies, and emerging applications, 3rd ed.,* ed. Julie A. Jacko, 795—821. Taylor & Francis Group, Boca Raton.

Petrillo, Fabio – Pimenta, Marcelo (2010) Is agility out there? Agile practices in game development. In: *Proceedings of the 28th ACM international conference on design of communication* (SIGDOC '10), São Carlos, São Paulo, Brazil, September 27-29, 2010, 9—15.

Petrillo, Fábio – Pimenta, Marcelo – Trindade, Francisco – Dietrich, Carlos (2009) What went wrong? A survey of problems in game development. *Computers in Entertainment*, Vol 7 (1), 13:1—13:22.

Pettingell, Karen – Marshall, Thomas – Remington, William (1988) A review of the influence of user involvement on system success. *ICIS 1988 Proceedings*, 40, 227—236.

Phillips, Dicky (2017) Top 3 Challenges to Developing an Indie Game: As Voted by 9 Indie Devs. Rengen. <https://www.rengenmarketing.com/indie-devs-share-indie-game-development-challenges/>, retrieved 25.1.2019.

Pierce, Jon L. – Kostova, Tatiana – Dirks, Kurt T. (2001) Toward a theory of psychological ownership in organizations. *The Academy of Management Review*, Vol. 26 (2), 298—310.

Planells, Antonio José (2017) Video games and the crowdfunding ideology: From the gamer-buyer to the prosumer-investor. *Journal of Consumer Culture*, Vol. 17 (3), 620—638.

Politowski, Cristiano – Fontoura, Lisandra – Petrillo, Fabio – Guéhéneuc, Yann-Gaël (2016) Are the old days gone? A survey on actual software processes in video game industry. In: *Proceedings of the 5th international workshop on games and software engineering (GAS '16)*, Austin, Texas, May 14-22, 2016, 22—28.

Poretski, Leo – Arazy, Ofer (2017) Placing value on community co-creations: A study of a video game 'modding' community. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 480—491.

Reichert, Kate (2012) Top 5 Problems Faced by Indie Game Developers. Gamasutra.com. <http://www.gamasutra.com/blogs/KateReichert/20121101/180767/Top_5_Problems_Faced_By_Indie_Game_Developers.php>, retrieved 25.1.2019.

Ruffino, Paolo (2013) Narratives of independent production in video game culture. *Loading… The journal of the Canadian game studies association*, Vol. 7 (11), 106—121.

Sansone, Anthony T. (2014) Game design documents: Changing production models, changing demands. In: Computer Games and Technical Communication: Critical Methods & Applications at the Intersection, eds. Jennifer DeWinter – Ryan M. Moeller, 109—123. Ashgate Publishing Limited, Farnham, England.

Schmalz, Marc – Finn, Aimee – Taylor, Hazel (2014) Risk management in video game development projects. In: *Proceedings of the 2014 47th Hawaii international conference on system sciences (HICSS '14)*, January 6-9, 2014, 4325—4334.

Sinclair, Brendan (2018) Global games market to hit $137.9 billion this year – Newzoo. gamesindustry.biz. <https://www.gamesindustry.biz/articles/2018-04-30-global-games-market-to-hit-usd137-9-billion-this-year-newzoo>, retrieved 25.1.2019.

Stacey, Patric – Nandhakumar, Joe (2009) A temporal perspective of the computer game development process. *Information Systems Journal*, Vol. 19 (5), 479—497.

Steam Spy (2017) Sergey Galyonkin. <https://steamspy.com/genre/Early+Access>, retrieved 9.9.2019.

Taylor, T. L. (2006) Beyond management: Considering participatory design and governance in player culture. *First Monday Special Issue #7: command Lines: The Emergence of Governance in Global Cyberspace.* <https://journals.uic.edu/ojs/index.php/fm/article/view/1611/1526>, retrieved 29.3.2019.

Taylor, Haydn (2018) Activision Blizzard stock tumbles following Diablo Immortal backlash. gameindustry.biz. <https://www.gamesindustry.biz/articles/2018-11-06-activision-blizzard-stocks-tumble-amid-fan-backlash-over-diablo-immortal>, retrieved 26.1.2019.

Thominet, Luke (2018) How to be open: user experience and technical communication in an emerging game development methodology. *Communication Design Quarterly Review*, Vol. 6 (2), 70—82.

Thominet, Luke (2017) Tracing Player Experience: A Content Analysis of Player Feedback Tickets. *In Proceedings of SIGDOC '17, Halifax, NS, Canada*, *August 11-13, 2017.*

Tschang, F. Ted (2005) Videogames as interactive experiential products and their manner of development. *International Journal of Innovation Management*, Vol. 9 (1), 103—131.

Tschang, F. Ted (2007) Balancing the tensions between rationalization and creativity in the video game industry. *Organization Science*, Vol. 18 (6), 989—1005.

Wang, Alf Inge – Nordmark, Njål (2015) Software architectures and the creative process in game development. In: *Entertainment Computing – ICEC 2015*, eds. Konstantinos Chorianopoulos – Monica Divitini – Jannicke Baalsrud Hauge – Letizia Jaccheri – Rainer Malaka, 272—285, Spriner, Cham, Switzerland.

Washburn Jr., Michael – Sathiyanarayanan, Pavithra – Nagappan, Meiyappan – Zimmermann, Thomas – Bird, Christian (2016) "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*, 280—289.

Wilson, Stephanie – Bekker, Mathilde – Johnson, Peter – Johnson, Hilary (1997) Helping and hindering user involvement – A tale of everyday design. In: *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, 178—185, Atlanta, Georgia, USA — March 22 - 27, 1997.

Xin, James He – Myron, Sheu (2014) Efficacy of functional user impact on information system development. *Management Research Review*, Vol. 37 (10), 902—911.

Zackariasson, Peter – Wilson, Timothy L. (2010) Creativity in the video game industry. In: *Creativity: Fostering, Measuring and Contexts*, eds. Alessandra M. Corrigan, 109—120. Nova Science Publisher, Inc.

Zimmerman, Josh (2014) Psyche and Eros: Rhetorics of secrecy and disclosure in game developer-fan relations. In: *Computer Games and Technical Communication: Critical Methods & Applications at the Intersection*, eds. Jennifer DeWinter – Ryan M. Moeller, 141—156. Ashgate Publishing Limited, Farnham, England.

APPENDICES

Appendix 1. Player participation questionnaire for players

The purpose of this research is to investigate various issues regarding players participating in early videogame development projects (such as Steam's Early Access projects or crowdfunding projects in Kickstarter) and to build a more comprehensive picture of the trend. Through this questionnaire, we aim to understand what motivates players to participate in these projects, how do they discover these projects, what activities players do during the participation, and finally how players view players participation and what are their attitudes towards it. The questionnaire consists of 12 mostly open-ended questions out of which 3 questions are obligatory demographic questions that are solely used to argue reliability and credibility of the research findings. Even though the other 9 questions are voluntary, it would be most beneficial for the research to get as extensive and in-depth answers as possible. Thank you for your participation.

1. Gender
    - Female
    - Male
    - Other
2. Year of birth
3. Nationality
4. What game projects have you been participating in as a player? Please name the games or the game projects.
5. Through which (distribution) platforms have you participated in these projects?
6. How did you discover the game projects?
7. What motivated you to participate in the game projects?
8. Please describe the things you have done during your participation? How was your participation received and what kind of effects did it have? Could you give some examples of your participation and contribution to game development projects?
9. What upsides/good things have you encountered in a game development project as a player? What was special about them and why? How did they affect you and your participation?

10. What drawbacks/downsides have you encountered in a game development project as a player? How have those affected you and your participation? How could those issues be improved?
11. What do you generally think of being able to participate in a game development project as a player? (good/bad thing, interesting/boring, useful/unuseful, necessary/unnecessary etc.) Why?
12. Is there something else you would like to highlight, comment, criticize, or give feedback to the researcher?