

Scrum in video games development – Literature review

Bartosz Pałka

Abstract

The video game industry is one of the fastest-growing industries in the world. In consequence, projects of video game development get progressively bigger and more complex. Scrum is the most commonly used methodology in software development; therefore it is often used in video games development as games are essentially combination of software, art and sometimes science. The aim of this re-view is to provide overlook on Scrum as a methodology used in Software development, describe in short what characteristics the video games development has and how well Scrum fits into the current state of video games development. It also presents advantages and disadvantages noticed by previous researchers with using Scrum inside and outside of video games industry. The information contained in this article may be useful for people that already work with Scrum outside of video games industry, as well as people working in video games industry not working with Scrum. Both groups may get familiarized with challenges and benefits of using Scrum in context of video games industry and with review of articles based on real life examples contents of this article contains tips valuable to people struggling with Scrum implementation in their video games development projects and how to enhance this methodology with other methodologies to achieve most beneficial outcome.

1. Introduction

In the last two decades, we have observed a rapidly growing video game market that is no less significant for today's economy than any other entertainment industry. More than 10 years ago in 2012, it surpassed revenue of regular software revenue by three times (Kasurinen et al., 2017). Regarding the current state of the video game industry, it is forecasted by the Statista portal that it will surpass 385 billion dollars in revenue in 2023. Projects generating this much revenue consist of many individuals and are as big as consisting of hundreds of people (Egenfeldt-Nielsen et al., 2019). In that case right project management method should be applied to enable its success. Software development was revolutionized with a „rugby” approach a new for that time methodology. That enabled software developers to create software faster with a more flexible approach (Takeuchi & Nonaka, 1986). Later it was developed further and dubbed Scrum by Schwaber and Sutherland in 1997.

Video games, which are essentially software, have one difference in their development. Game development is a combined effort of software engineers, audio, and video engineers, (voice) actors, and sometimes even scientists. This kind of software has no other function than providing customers with entertainment (Koutonen & Leppänen, 2013). That kind of multi-layered collaboration influences coding in video game development. Multiple articles reviewed present the benefits and challenges of using or implementing Scrum methodology in both regular and video game software development. Is Scrum the best option to choose in video game development? What potential challenges with Scrum are recorded by experienced video game developers? What could be changed to enhance Scrum functionality in video game development? The first part of the article focuses on Scrum in software development. In the second part milestones in the video game history and video game development are presented to give more context to the last third part where Scrum is revised as a methodology used in video game

development. In the same part literature on real-life examples was studied to show not only the pros and cons of Scrum in video game development, but also solutions or enhancements to the methodology that gave better results in said examples.

This article is a comprehensive literature review that aims to determine whether the Scrum methodology is a suitable one for video game development in current times. It also questions whether it is agile enough to enable cooperation between that many roles in the project and if it can be enhanced – what needs to be changed. To ensure its validity a variety of literature covering Scrum, video game development, and Scrum in video game development was analyzed. To add more context to the video game industry a timeline of the most important events is presented in this article.

2. Methodology

The methodology used in this article is a literature review. The main goal of the chosen methodology is to hand-pick the best source of information regarding the title of the article. A review of such a source of information will enable us to highlight the most recurring scenarios present in the use of SCRUM, AGILE methodology in the industry of video game development. In a comprehensive analysis of already existing literature on the topic, the article will summarize the knowledge gathered by previous researchers and therefore aspires to be a starting point for discussion of possibilities on usage of SCRUM in the future of the video game industry. The literature review serves as an essential step in the research process that allows mapping and assessment of previous research in a field and provides the foundation for the study's research question.

A literature review enables the author to provide an overview of the topic and/or to conduct an assessment of current knowledge. It also enables the evaluation of historical data in the context of today's knowledge (Snyder, 2019). On top of that it is a useful tool to identify potential gaps in the knowledge and encourages a circle of people interested in

the topic to seek out new enhancements for re-researched topic.

The main advantage of such a method is that it follows a procedure that enables an environment where the researcher does not have to focus on the methodology itself, but rather on the quality assurance of the review to deliver it with utmost quality and care (Koch et al., 2023). The procedure of literature review starts with extensive research on the viable articles on the video game development industry, SCRUM methodology, and the use of the said methodology in the mentioned industry. Refer to the map below for the detailed steps of conducting the research.

For research following databases were used: Google Scholar, ResearchGate, and EBSCOhost, to identify the material that is needed for the literature review, the tags for SCRUM were used as such: Scrum in Game Dev, Scrum in Software Development, Professional Scrum, history of SCRUM. The articles on video game development were searched with phrases as follows: Agile in video game development, Video game programming, Scrum in video game development. In the research phase around 45 pieces of literature worth analysis emerged. In this literature review the materials used are scientific publications, conference papers and on top of that, research found two books on the topic.

To ensure a broad overview of the topic sources track back to the emergence of the Scrum methodology in project management. Older sources were used to depict the core values of the methodology and give more context to the latter sources. Newer sources were used to highlight the usage of Scrum in video game development and indicate the current state.

The analysis of the source material, if it is suitable for this review, was happening through the screening with the use of the following questions:

1. Does the abstract and contents indicate that relevant data is contained in the article?
2. Does the author have first-hand experience with Scrum or Video Game Development?
3. Is the article in English, Polish, or German?
4. Is the data provided in the article for the literature review up to date for its purpose?

Then all the materials were sorted as follows:

- If the abstract and content indicated the presence of the necessary information the articles were gathered.
- If the author had first-hand experience, it source was marked as a priority source and if not it was marked as a source of secondary importance.
- Articles without translation that seemed comprehensive were scrapped from the list of useful data.
- If the data was outdated for its purpose it was deleted from the list of useful information.

To ensure relevance actions were taken as follows: If the author of the article did not have firsthand experience in the field of a researched topic, the information in the article was revised and if they were significant and correct, they were marked as useful. If not, a bibliography was used to trace back to the comprehensive author with relevant information on the topic. If the abstract and contents indicated the

presence of the relevant information the article was thoroughly analyzed.

The analysis indicated that only 23 pieces of literature were applicable to this review. Articles were rejected for the following reasons:

- Language not being English, Polish, or German
- Translation looked not accurate enough to the original language.
- Scarcity of information on researched topic
- Outdated information
- Too much theorizing due to the second-hand experience of an author

The main benefit of using this methodology is that data is gathered from a different multiple sources presenting standpoints of experienced individuals in the usage of Scrum in video game development and revised under the vigilant eye of previous researchers. With every methodology, there is always a downside. Here problem might be the outdated information on such a fast-growing industry, and the credibility of the chosen literature - even after a meticulous screening process. Another issue contrasted with the video game industry a topic of Scrum, where an over-burdening number of articles were found. It made it impossible to review every one of them.

To minimize the downsides of the methodology the review leaves a place for doubt and self-reflection on Scrum in video game development to let one assess with own experience in Scrum if the source is credible enough to mark it as truthful. What is more the highest care was present during tracing back to the source material of reviewed literature and research of materials was done. Then it was established if their footprint on the gaming industry was recognizable enough to label them as the expert in their field. Basically, the bigger the success of the product (game) was the more credibility have the source.

3. Literature review

Scrum in software development

The history of Scrum can be traced back to the publication by Takeuchi and Nonaka from 1986, which provided valuable insight and influenced subsequent thinking of agile methodologies. In the publication, it was highlighted that product developers need speed and flexibility to prevail in the changing market. Thanks to that new approach emphasizing not only quality but also speed, flexibility, and autonomy companies like Canon, Honda, and Xerox benefited and can be labeled as pioneers in using agile methodology.

In 1997 Ken Schwaber published his work on a new software development approach called „SCRUM Development Process” where he did not hide the fact that Scrum is highly influenced by Takeuchi and Nonaka. Term scrum was first used in the work of Takeuchi and Nonaka to compare this approach to a rugby game where the cooperation of each team member is essential to achieve the same set goal.

Schwaber highlighted that SCRUM is an enriched version of the agile approach that develops and delivers object-oriented software iteratively in increments. Later in 2001 Schwaber together with Sutherland highly contributed to the Manifesto for Agile Software Development including

SCRUM methodology. That year was also beneficial for software development due to the establishment of Scrum Alliance, a platform that was set out to train and certify people eager to learn SCRUM ways, by the forefather of SCRUM itself Ken Schwaber. In 2009 Schwaber left the Scrum Alliance and created Scrum.org, a site dedicated to helping people across the world to learn SCRUM. According to their home page, their mission is to help people and teams solve complex problems by using the power of said methodology and acquire success in software development.

The Scrum approach has been widely adopted by leading software companies and has demonstrated significant success. It is believed that Scrum can also benefit other software development organizations, seeking to leverage the advantages of Object-Oriented techniques and tools (Schwaber, 1997). Even before Schwabe's publication, it was recorded that by adopting a scrum approach (Takeuchi & Nonaka, 1986) companies like Canon, Xerox, and Honda have benefited from it immensely.

At its core, SCRUM emphasizes dividing the development process into short iterations dubbed sprints, typically lasting not more than four weeks. The end of each sprint should bring a usable product increment that can be delivered to the end user. Before the start of each sprint, a collaborative sprint-planning meeting takes place, where developers and stakeholders work together to select the tasks for the upcoming sprint. The impact of such a development approach was highlighted by participants of Alami and Krancher's (2022) study. What stood out for them was a sense of modularity brought to the project that enabled developers to focus more on the quality of the code in an atomistic way, which translates to control over the process (Alami & Krancher, 2022) in a mostly autonomous project development where HQ involvement is limited to funding and support (Takeuchi & Nonaka, 1986).

In SCRUM, the customer's perspective is represented by the product owner, who captures requirements in the form of user stories and maintains a prioritized product backlog. This backlog is a dynamic document that continually evolves to reflect the current understanding of user needs. Actually, in the role of Product Owner lies the flexibility of managing incoming stakeholder needs with current items in a backlog and prioritizing the most important items that need to be included in the next iteration (Cho, 2010)

Scrum is primarily designed for small, interdisciplinary teams consisting of approximately six to nine developers. A fundamental aspect of Scrum teams is self-organization, empowering them to make decisions on how to achieve sprint objectives (Schwaber, 1997). Within small teams, there exists a heightened sense of ownership and responsibility among individuals for their respective contributions. Consequently, this fosters an environment of increased accountability and a pro-found commitment to the delivery of exceptional outcomes with uncompromising quality (Anderson & West, 1998). Small teams exhibit enhanced agility and adaptability when confronted with evolving requirements. They demonstrate a remarkable ability to swiftly respond to emerging information, seamlessly adjust their plans, and

make well-informed decisions with expediency (Cockburn & Highsmith, 2001).

To ensure effective coordination and adherence to the Scrum process, every SCRUM team requires a Scrum Master. The Scrum Master imparts knowledge, offers training, and provides mentorship to steer teams toward the successful adoption and proficient implementation of Scrum practices (Rubin, 2012). By fostering an environment that nurtures self-organization, promotes autonomy, and empowers collective decision-making, the Scrum Master significantly enhances team performance and cultivates a culture of collaboration (Kniberg & Skarin, 2010). With effective communication at the core of Scrum, the Scrum Master plays a vital role in facilitating transparent and open channels of communication within the team, expertly organizing regular meetings such as daily stand-ups, sprint planning, and retrospectives, to encourage collaboration and the sharing of vital information (Schwaber & Sutherland, 2020). Furthermore, the Scrum Master actively encourages a culture of continuous improvement within the team, skillfully facilitating retrospective sessions where the team reflects on their performance and identifies opportunities for enhancement, resulting in increased efficiency and productivity over time (Schwaber & Sutherland, 2020). Additionally, the Scrum Master's unique perspective offers a comprehensive overview of the process and its output, enabling them to assess its impact on achieving higher software quality (Alami & Krancher, 2022).

Daily standup meetings are conducted to maintain a rapid work pace, allowing team members to share updates on their progress and tasks for the day. The daily stand-up meetings in Scrum, as highlighted by Ionel (2009), are recognized as valuable for clarifying development goals and promoting better team coordination. These brief meetings provide a platform for team members to synchronize their efforts, address any obstacles, and align their activities toward project success.

Additionally, retrospectives are held after each sprint to provide an opportunity for reflection and improvement of the team's work practices. The availability of artifacts for review and retrospectives enables a thorough examination of the accomplishments made during a sprint. Participants of the Alami and Krancher project indicated that transparency is an inherent characteristic of the Scrum framework (Alami & Krancher, 2022). What is more, according to participants of the same project personal pride arises when presenting the results. They took great pride in their code as it fulfilled all the requirements and functions flawlessly. This sense of accomplishment adds value to our personal growth, boosts morale, and encourages them to consistently produce even better code." (Alami & Krancher, 2022). Overall, Scrum provides a structured framework that promotes iterative development, customer collaboration, self-organization, and continuous improvement in software development projects.

On the other hand, one needs to keep in mind that Scrum is not an immaculate methodology and has its challenges to it. Based on the reviewed literature challenges were grouped as follows:

- People challenges,

- Methodology understanding challenges,
- Product delivery challenges.

Anand and Dinakaran indicated in their work that teams may face an issue inflicted by geographical differences. Different time zones may create problems in exchanging information on time and availability for different meetings. On top of that, any cultural differences or language barriers may also interrupt the workflow of the project. Scrum does not provide a solution to that challenge (Anand & Dinakaran, 2015). Any type of meeting, for example Stand-up meeting to start the work, can start only when every team member starts to work on global projects as well as in projects working in a flexible schedule may interrupt work for the people who are already working. Also, Zada and Shazad have pointed out the issue of hindered collaboration and communication flow for the team whenever they are placed all over the world. A survey made by Akif and Majeed showed results that indicated problem with managing Scrum projects that consist of multiple teams especially when they have different locations and in the Cho article developers stated that using media to communicate is not as effective as interactions in the same room. There are also no guidelines included in Scrum for managing that kind of project (Akif & Majeed, 2012).

The next challenge indicated is a change in people structure. Whenever there is a change due to people changing workplaces or projects, especially when management is changing the team will focus less on the work and friction may arise due to going through a four-step team relationship-building process (Anand & Dinakaran, 2015). The structure may also have a size challenge. In larger projects, people may tend to bend the rules of Scrum and form teams larger

than eight people, which will create more complexity in communication and lower the number of interactions between members impairing the empirical knowledge acquisition process (Cho, 2010).

With a review of selected materials, a challenge with self-management was identified. It is difficult to move in one direction to aim for the same achievement. Not all team members will instantly understand that the common goal is one pillar of cooperation in Scrum (Akif & Majeed, 2012). An example from Cho's work is that developers personally answered in the survey that they prefer the governance of a team leader over their work to compensate for not-so-great self-managing team members and rely on a team lead to push the team in the right direction. This particular challenge type also influences delivery. Lack of well-structured self-organization also creates a situation where developers themselves do not know when to stop coding and deliver the product.

The next set of challenges spotted in the reviewed literature regards understanding Scrum methodology by people involved in the development process. The previously mentioned challenge with self-governance is a sign of misalignment to Scrum rules. Cho mentioned in his work that the team loses trust whenever an increment is not delivered. This may also be connected to the issue of developers not knowing when to stop coding (Anand & Dinakaran, 2015). To some extent, Zada and Shazad picked up on the same issue in their work. They stated that inexperienced developers are vulnerable to not having a common goal. Where there is a lack of training a focus on standards and methodological solutions may dilute in overall development process. Challenges will have a lesser chance to be avoided when leaders

Table1. Examples of leaders that are connected to misunderstood Scrum

Product Owner (PO)	Scrum Master (SM)
Not following Scrum principles when it comes to new tasks, may cause a scenario where tasks are distributed directly to developers. That kind of approach omits a Product Backlog(PB), which creates confusion and team members will spend time on tasks that are not in the Product Backlog. It will make the organization of PB only harder with each sprint. (Akif and Majeed, 2012)	Daily Scrum turns into a meeting where people try to find a solution to problems. This causes to extend the duration of the meeting (Anand and Dinakaran, 2015). This occurrence was also observed by Cho.
Inexperienced PO and SM may cause confusion by not organizing Product Backlog in a clear manner. What is more, lack of experience may cause a situation where backlog is organized in a wrong way. (Anand and Dinakaran, 2015)	
PO and SM often disturb the work of developers to ask for updates. According to Akif and Majeed team members dissatisfied with interruptions among surveyed people is around 44%	
PO that does not understand one role may not organize Daily Scrum well. This may and in people not feeling encouraged to give feedback in a meeting (Zada and Shazad, 2015). Even worse when a lack of Scrum knowledge leads to the team organizing Product Backlog instead of PO without knowing customer needs.	SM that does not provide his team with information on time may impact the work of the team. Whenever items are not delivered by SM it may even halt progress (Cho, 2010)

Source: (Akif & Majeed, 2012; Anand & Dinakaran, 2015; Cho, 2010; Zada & Shahzad, 2015)

in the project are not well-trained in Scrum (Zada & Shahzad, 2015).

The initial effectiveness of Scrum may be limited in the first months of implementation. Any type of meeting might not be as effective as expected. Team members may feel uncomfortable with that many meetings or the flow of the process of the Scrum. Some of the developers indicated that meetings like Daily Scrum and Sprint Review Meeting are counterproductive and too long. The general atmosphere expected of Scrum, which is reasonable freedom and self-supervision may make people feel too easy-going about increment delivery. Any Backlog Item that gets delayed for the next sprint may be stuck in limbo for a long time, as no one wants to take responsibility for finishing it (Cho, 2010).

All the challenges intertwine with each other and one leads to another in a cycle creating more and more challenges. At the end of a development, a delivery of the product should happen. The next set type of challenges identified during the review are challenges with delivery. As Scrum forces to deliver an increment of work at the end of the sprint the quality of code might be compromised, thus creating more items in a Backlog that are essentially fixes for bugs originating from poor hurried code (Akif & Majeed, 2012). The previously mentioned challenge with the self-governing process also impacts delivery. Not knowing when to stop the work can potentially create issues of not delivering increments on time or delivering it in a poor „quick and dirty” way (Anand & Dinakaran, 2015). Having a set time to deliver a working product can create an environment where developers are working overtime in order to fit into a planned time frame of delivery. This makes developers burn out faster and impacts their capabilities of delivering code free of bugs or any further quality-related problems (Akif & Majeed, 2012). Cho observed that often items are carried over from a Sprint to the latter Sprint. This is a challenge to organization of a Sprint as, according to Cho, nobody wants to take responsibility for the postponed Backlog Items. A project manager in Cho's work brought attention to a possible cause of not taking responsibility in following up on carried-over cases is connected to the too relaxed atmosphere in Scrum methodology and PO not having enough authority to pressure the team to work. The next challenge within delivery is connected to the SM role. As developers rely on Scrum Master to push further and SM does not provide what is necessary then it results in a team not delivering a distrust may rise between all the people in the project (Cho, 2010). Scrum has benefits that need to be leveraged and challenges that should be mitigated. In each article both advantages and disadvantages repeat themselves, thus we can conclude that they are connected to Scrum itself and not only to oneself experience with this methodology.

Video Game Development

According to Oxford Dictionary a game is essentially an image that is produced by a program on a computer that is manipulated by a player to influence events presented on the screen in order to navigate through the game world to have fun and win. The outcome is visible on any type of compatible display and input is generated with additional peripherals like gamepads, mouse, and keyboard or more game-

type specific controllers (Koutonen & Leppänen, 2013). What characterizes a game is that it is a product sold directly to the customer (Egenfeldt-Nielsen et al., 2019) and the value for the customer is pure entertainment and in some instances educational value (Kasurinen et al., 2017).

The gaming industry evolved from a small entertainment market to a market so big that revenues surpassed other markets like the movie industry or software industry. One may say that it all started in 1961 when Steve Russel, an MIT student, created the first video game on a PDP1 computer. Eleven years later a company named Atari created a game called “Pong”, the first game which succeeded commercially. After that other companies like Nintendo, Capcom, Midway, and Namco emerged throughout the seventies and eighties, gaining significance in the gaming market. Video Game Market was saturated with games like Pac-Man, Space Invaders, The Legend of Zelda, Donkey Kong, and Tetris. They attracted many players and with new hardware in the form of consoles, it was possible to bring this hobby home. In the 90s and 00s bigger companies started to invest in this market. 1994 brought to life a PlayStation gaming console by Sony and in the years 2000 and 2001 second iteration of this console. In the same years, Microsoft released the first Xbox console and Nintendo released their next console called GameCube. This is the time that can be deemed as the start of the console war. After that, until the present day new generation of consoles are released every few years with updated hardware that enables games to be more complex. The biggest growth was seen during the 90s and 2000 s and even though the financial crisis in 2007 caused some trouble to the industry, it thrived (Egenfeldt-Nielsen et al., 2019). Ever since the video games industry is experiencing rapid growth and is considered one of the fastest-expanding sectors in the international media landscape. It has established itself as a significant contributor to the global entertainment economy. Additionally, the video games industry plays a crucial role in modern software development, generating revenues three times higher than software retail already in 2012. With more investment in this field to create more capable hardware for pure entertainment the process of developing software started to be more and more complex.

The Video Game Market is a dynamic and thriving ecosystem surrounding the entire process of creating, distributing, and profiting from video games. It includes various platforms such as console, PC, and mobile games, including highly profitable online games, along with associated hardware. With a massive global audience of passionate gamers, the industry is experiencing remarkable growth, generating billions of dollars in annual revenue.

Within the Video Game Market, there are several distinct segments worth exploring, including Mobile Games, PC and Console Games including Online Games, and Cloud Gaming, each having a dedicated market and audience. To gain a comprehensive understanding of the market, key indicators such as revenue figures, user metrics, average revenue per user, and penetration rates are essential. Revenue generation predominantly hinges on in-game purchases, consumer expenditure on games, and subscription fees on top of paying retail prices. In this vibrant industry, innovation and

competitiveness are driven by prominent industry players like Activision with its popular Call of Duty franchise, EA with sports games, Microsoft with Cloud gaming services trying to reach a wide range of players, Tencent trying to dominate a mobile gaming part of the market, and Sony competing with Microsoft and Nintendo to dominate the console gaming. Nowadays it is common that one company to buy another or form a strategic alliance to create a supply for the demand of gamers (Egenfeldt-Nielsen et al., 2019). The most recent example can be Microsoft acquiring Activision in 2023.

In a time of such rapid growth and when the hardware is getting more advanced the projects of video game software are also getting exponentially more complex. Currently, a large triple-A games project that consists of hundreds of people working on it is nothing exceptional (Egenfeldt-Nielsen et al., 2019). There also exists smaller projects often called indie games where sometimes even one developer produces a game. An example here might be still in development game called *Out of Action* or gaining popularity three-man project called *Battlebit Remastered* that in June 2023 generated revenue of 16 million dollars.

A commercial game's development is often broken up into stages that are identified by milestones as in any other project. These milestones form the basis of contracts between publishers and developers. The game concept is first formulated in a concise manner, conveying the core idea, platform, and sometimes concept art. A more detailed game proposal is then created to attract funding and plan production, covering market analysis, technical issues, budget projections, and the game's audiovisual style. Functional and technical specifications are described in great detail in the design document, which can be extensive document, pages long for triple-A titles. Agile methods such as eXtreme Programming (XP), DSDM, FDD, Kanban, Lean and naturally Scrum are commonly used. The design document itself includes descriptions, illustrations in the form of concept art, and object lists, and it is constantly updated during development. Based on the design document, the game engine is selected, either by building a custom engine or acquiring a license for a third-party engine. Today in massive projects, investors require playable demo versions of games before giving project approval once the prototype is created and approved then the production begins, involving coding, graphics creation, sound development, and specialized teams for large-scale productions.

The presence of different kinds of artists, developers and sometimes even scientists is dictated by the needs of the project. Game development needs a team with diverse skill sets. According to Egenfeldt-Nielsen et al., the project needs roles like:

- Game Designer who provides vision with a description of details to stay true to the planned game identity
- Programmers from various backgrounds. Beginning with the ones that will be responsible for Engine and Creation Tools, going through networking programmers with knowledge of how to manage infrastructure for online games, and ending with AI programmers that will ensure that the world reacts in an intelligent way to the player's

input. Between you will find other programmers who will create and manage audio, graphics, or physics in a game.

- Producer who will be managing and overseeing the production – a function similar to project manager.
- Graphic and audio artists who will create visuals and sounds for the game.
- Animators and motion-capture artists.
- Voice actors, actors, and stuntmen that will be a voice, a face or movement template for the game characters.

Those are just examples of important team members needed for game development, a real list would be much longer than that, including specialists from different fields whenever a game needs that kind of involvement. During the lengthy process of production that takes months or even years, challenges may arise, leading to changes in game elements or even to start over from the beginning (Egenfeldt-Nielsen et al., 2019). Later, when the game reaches maturity and the first playable barebone version of a fully playable game can be put to the test, a release of an alpha version that are usually used for internal testing to fish for any bugs that may impact on game in a way that it is not playable. After the alpha version is tested some more polishing and adjusting takes place and once it is done the next version is released. The beta version is used to test the game internally as well as externally. Sometimes developers decide to invite beta testers chosen from fans, gamers, or influencers to test the real reception of the game in addition to test of the technical state. Beta testers play the game for extended time and they provide feedback for the development team to make necessary corrections to the game. When it comes to online games it is common for developers to let everyone play an open beta version to have more feedback and stress test the capabilities of the servers (Egenfeldt-Nielsen et al., 2019). After beta tests, more polishing work is done upon a feedback from the testers. Once the game is ready it acquires a „gold” status, which means that the game is ready for release to the customer. Some additional work is done like the preparation of physical copies of the game. Nowadays when the game is in gold status it does not mean that the work is finished. Developers use the time between gold status till the premiere to improve the game even further and release a day one patch and fix other problems that were found in the game after the gold status was announced and the support in the form of patches is an ongoing process throughout a life of the game. The longer the game lives and is played by gamers the longer it will get the support from developers. Prior to the mid-1990s, completing a gold master version of a game frequently meant the developer was moving on to other projects. However, in modern times, it is customary for game studios to jump right into the creation of a sequel or extra content packages (Egenfeldt-Nielsen et al., 2019).

An exception to this general flow of game development is Early Access. It was inspired by Minecraft's success initiative on the Steam platform where unfinished games could have been published when still in the beta or alpha phase of development. This lets the developers effectively manage the budget and more easily tailor the game to user needs (Lin et al., 2018). In a nutshell, unfinished product with a promise to deliver it in a finished state sometime later is bought by

the customer and the customer can provide feedback directly to the developers so they are able to change the game constantly upon feedback. This solution of iterative delivery benefits developers as they can gather money for the development and manage the budget on real money, not only on promise to buy a game (pre-orders) and main benefit for the customer is that they are able to oversee the progress of the awaited by them game and provide feedback for developers on what they like and dislike. The downsides of such an approach are that only one-fourth of games in Early Access were actually released and in some cases, games were just cancelled and promises to deliver broken (Lin et al., 2018). Some successful games that were in Early Access and stood up to the expectations are DayZ, Minecraft, and Subnautica.

Scrum in video game development

If we put Scrum and video game development together it seems reasonable to use this Agile methodology. Having in mind that game development is done by multiple people relying on each other, it is iterative and the project requirements may change during the development process, it sounds even more sane to use it. According to Kristiadi et al. work Scrum is simple enough to learn and deploy methodology effortlessly in the project. In the described situation the outcomes of Scrum implementation were appraised by the team (Kristiadi et al., 2019). That is why many companies from the video game industry tend to use and recommend it for their projects. According to Koutonen and Leppänen's article in 2013 already more than half of the companies used Scrum. The benefits of implementing Scrum in video game development are aligned mostly with the benefits of Scrum in retail software development. It is said that delivery time is shortened, which helps to deal with ever-changing requirements in the early stages of development (Kristiadi et al., 2019). Especially in the phase of prototyping – creating a demo version for stakeholders (McKenzie et al., 2021). For this phase it is common to deliver features iteratively therefore Scrum puts the team of developers at a right pace to balance backlog and in the end, this solution does not put out the flame of ambition by demotivating them with undelivered features that decrease morale (Koutonen & Leppänen, 2013). On top of that it is deemed that Scrum seems perfect for core game mechanics and post-launch support as there is more creative work and ongoing change (McKenzie et al., 2021). Comparing Scrum to Waterfall it was spotted that Scrum is more effective, as Waterfall makes people rely on one another in a way that when one of the developers fails to deliver another one cannot start their work. This situation creates slowdowns in a development. Additionally, it impairs the ability to monitor expected delivery time. On the other hand, Scrum through iterative development not only lets the project manager be on track, but also divides the work in such a way that people rely less on each other (McKenzie et al., 2021). According to McKenzie et al, the more aligned with Scrum values and agile approach of the team was, the better was the reception of Scrum methodology in general, and Kristiadi et al, mentioned that it is most reasonable to use Scrum to have better visibility in the project when producing completely new and unique features for the game.

The downside of using Scrum in video game development is aligned to the possible challenges mentioned in the first sub-chapter. Kristiadi et al, mentioned that before deciding to use Scrum it has to be ensured that developers understand Scrum in the first place. It was also suggested that training in Scrum might bring some benefits for developers, but the best approach since they are game developers, is the game, and an example here given was a card game called Playscrum. One of the emphasized issues is the adaptation to the methodology by newcomers to the project and their training (Al-Azawi et al., 2014). More disadvantage to that issue is the fact that often companies resort to hiring contract or part-time employees. In consequence, companies might not be enthusiastic about investing in such team members to provide them with essential Scrum training, therefore they might be a bottleneck to the flow of development in Scrum (Osborne O'Hagan et al., 2014). If we take a look at the project as one body that incorporates Scrum it can be spotted that whenever the Scrum implementation gets further from the original Scrum values, a team makes some tweaks to the methodology as they think their project needs it, then the overall reception of Scrum is seen negatively (McKenzie et al., 2021). Another issue with Scrum in video game development that repeats itself among reviewed literature is the issue with sprint backlog when in the production phase (Osborne O'Hagan et al., 2014) and Koutonen and Leppänen indicated that for example, Kanban might be a better solution to manage tasks.

McKenzie et al. pointed out that using Scrum methodology might be problematic for different groups of people within a game development project. Whenever developers, that Scrum suits well, are attending Daily Scrum they can only be discouraged from participating mainly when the meeting is not conducted well enough. On the other hand, artists are working in different pace to the developers and while they enjoy the freedom of self-governance in Scrum, they may perceive mandatory Scrum events like Daily Scrum as an effort to control their work, which in some instances might be met with disapproval from their side and a challenge for the Scrum Master, to encourage them to be positive about the meeting. In the reviewed articles downside in regard to product itself emerged. As previously stated, the end product is fun and entertainment, which cannot be quantified easily and measured due to it being subjective criteria. An assumption of Scrum is that anybody can do anything is not applicable in video game development even amongst the programmers. They are specialized in different fields and even if coders that deal with physics can give a hint to AI programmers it is unlikely that they will be able to create a code for the AI programmers as effectively (McKenzie et al., 2021). Lastly, in contrast to what Koutonen and Leppänen stated about the emphasis on team members and their learning from an information exchange over documentation being a profit for the team using Scrum, Al-Azawi et al had the opposite view of lack of documentation being a root cause of miscommunication issues. Studios tend to lean toward agile methodologies like Scrum as it is natural for them as the teams in order to cooperate already adopted some parts of those methodologies. Although literature

presents the pros and cons of Scrum in game development the reality is that game studios try different approaches to implementing Scrum. Kristiadi et al stated that there were some instances where sprints were redesigned to tailor the needs of the projects and recorded an improvement over standard Scrum, by cutting the time of the sprint from 30 days to a duration between two and four weeks. What is more in a survey studied in Kristiadi et al piece of work it was established that half of studios implementing Scrum have modified it to be more flexible. As Koutonen and Leppänen concluded not a single company is able to use only one model of project management as the standard, where one methodology is lacking agility a combination with other methods might be a solution to an identified gap. They have also stated that the most commonly used methodologies in game development are a combination of Scrum with either Kanban or Extreme Programming. The same situation happens in research done by McKenzie, where the company decided to modify Scrum extensively due to it not being agile enough to manage their project. They have used a combination of Scrum and Kanban, Scrum and Waterfall, or Scrum and XP. Al-Azowi et al suggested that modification of agile methodology in the form of having documentation for bigger projects might be a beneficial form of communication. Politowski et al. in their work pointed out important economic factors that might create issues with implementing Scrum. Once the contract with publishers, for example during the development process of indie games, is signed, the structure of Scrum gets damaged due to milestones and delivery dates being set. Freedom as a benefit of Scrum will fade away and pressure of delivery will take its place. It was also pointed out that milestones are usually underestimated. One important factor noticed in his piece is that although video game development is different from regular software development, Scrum can be used as a philosophy rather than methodology and benefit team members who are taking part in video game development but are not producing any code. Example here are various artists that produce assets for video games that can adopt some of the Scrum principles in their daily work without adopting Scrum as a whole process. This emphasizes and encourages tweaks in Scrum to assist in the development process that it should.

4. Discussion

In this study most important question to consider is whether Scrum is a sufficient and agile enough methodology for the present video game industry. Literature review showed that throughout time Scrum has been recognized as a valued methodology to manage a project of software development with features like self-governance, small teams exchanging information constantly, iterative development of increments and delivery, and additional benefits like having Product Owner that is an effective connection between customer and developers, Scrum Master that oversee the development process and assist, smaller organizational bits like daily stand-up meetings or Sprint retrospective. It also highlighted challenges with social barriers including language, culture, work culture, issues with self-governance, and gaps in Scrum and agile principles knowledge due to weak

implementation. In an industry that is currently one of the fastest growing and still developing such as the video games industry, there is a need for agile solutions. Projects have grown from a few people projects to global multi-team, multi-disciplinary undertakings, where a few hundred people working on a single game is no surprise. Therefore Scrum in some instances has been proven not enough as a single method to drive the project. In contrast to the benefits presented by Kristiadi et al. Koutonen and Leppänen and McKenzie et al, like better visibility over the progress of the project, iterative development for the prototyping phase, or core mechanics development (this phase requires more coding), there are downsides to using Scrum in video game development. The same scenarios that occur during regular software development can be spotted, such as issues with training the teams on what is Scrum and how should it be applied to their everyday work (Kristiadi et al., 2019). Kristiadi et al suggested that learning through playing games might be beneficial for team members since they are probably interested in playing games. Especially new and seasonal employees might benefit from that. Another issue that was pointed out in McKenzie et al paper directly answers the question if the cooperation between many roles in video game development projects can be easily managed with Scrum. McKenzie et al explained that for example, Daily Scrum might be discouraging for artists to participate in it as they perceive it as a control mechanism for their work. It implies that Scrum might be not agile enough to cooperate with artists and some future research might even enlarge the group of people for which Scrum mechanisms are not fit to their needs in daily work. Another issue with interdisciplinary teams is that they will not be able to fulfil another teams' tasks due to being educated in completely another field of coding. Example here is AI programmers and physics programmers. They might be able to assist the other team with their knowledge on common grounds of coding, but doing tasks for other teams might be challenging or in many instances impossible.

The solution to issues with Scrum that appear multiple times in reviewed literature is combining Scrum with other methodologies like Kanban, Extreme Programming, or even Waterfall. It was implemented in many projects described in reviewed literature and the benefits of such an approach were significant. Some companies even cannot imagine working solely in Scrum. Another solution proposed is tailoring Scrum to the needs of the project. If Sprint time is too long for one phase of the project it can be cut even in half. If the project is large enough that to be able to transfer information it is beneficial to create documentation than it should be created. When it comes to interdisciplinary team management tailoring Scrum to the team's needs and having Scrum in the background as a philosophical value was also proposed as a solution.

To finish the discussion part it is worth mentioning that a fairly new approach named Early Access might be an aid to Scrum in video game development. Since in video game development product is sold directly to the customer, which makes them key stakeholders and value is brought to them, it is reasonable to ask them for their opinion on the product

during development and have a direct customer as a tester. This solution demands immense amounts of trust in the process on both sides, but in many instances, it turned out to be the most profitable approach for both sides. Customers – gamers had a direct influence on the development of the product that they actually desired and the developer studio did not have to rely on investors that might want to control their artistic vision.

5. Conclusion

The video game industry is quite different from regular software development even though differences exist development teams often resort to Scrum as a good way to develop software. On top of the overall goals, the aim of this literature review was to assess the Scrum methodology in the context of software development and how it works in the booming video game industry and encourage further researchers to study the topic of project management in video game development especially focusing on agile methodologies such as Scrum. To enable that a brief overview of the game development process and Scrum was presented.

References

- Akif, R., & Majeed, H. (2012). Issues and challenges in Scrum implementation. *International Journal of Scientific & Engineering Research*, 3(8), 1–4.
- Alami, A., & Krancher, O. (2022). How Scrum Adds Value to Achieving Software Quality? *Empirical Software Engineering*. <https://doi.org/10.1007/s10664-022-10208-4>
- Al-Azawi, R., Ayesh, A., & Obaidy, M. A. (2014). Towards agent-based agile approach for game development methodology. 2014 World Congress on Computer Applications and Information Systems (WCCAIS), 1–6.
- Anand, R. V., & Dinakaran, M. (2015). Issues in scrum agile development principles and practices in software development. *Indian Journal of Science and Technology*, 8(35), 1–5.
- Anderson, N. R., & West, M. A. (1998). Measuring climate for work group innovation: Development and validation of the team climate inventory. *Journal of Organizational Behavior*, 19(3), 235–258. [https://doi.org/10.1002/\(SICI\)1099-1379\(199805\)19:3<235::AID-JOB837>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1099-1379(199805)19:3<235::AID-JOB837>3.0.CO;2-C)
- Cho, J. J. (2010). An exploratory study on issues and challenges of agile software development with scrum. All Graduate theses and dissertations, 599.
- Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131–133.
- Egenfeldt-Nielsen, S., Smith, J. H., & Tosca, S. P. (2019). *Understanding video games: The essential introduction*. Routledge.
- Kasurinen, J., Palacin-Silva, M., & Vanhala, E. (2017). What concerns game developers? A study on game development processes, sustainability and metrics. 2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WET-SoM), 15–21.
- Kniberg, H., & Skarin, M. (2010). Kanban and Scrum making the most of both. *InfoQ*.
- Koch, J.-A., Sagarra, M., Amirbagheri, K., & Gomber, P. (2023). A Methodology for Systematic Bibliometric Literature Reviews: General Approach and Application to Digital Finance and FinTech Literature. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4477127>
- Koutonen, J., & Leppänen, M. (2013). How are agile methods and practices deployed in video game development? A survey into Finnish game studios. *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings* 14, 135–149.
- Kristiadi, D. P., Sudarto, F., Sugiarto, D., Sambera, R., Warnars, H. L. H. S., & Hashimoto, K. (2019). Game Development with Scrum methodology. 2019 International Congress on Applied Information Technology (AIT), 1–6.
- Lin, D., Bezemer, C.-P., & Hassan, A. E. (2018). An empirical study of early access games on the Steam platform. *Empirical Software Engineering*, 23, 771–799.
- McKenzie, T., Morales-Trujillo, M., Lukosch, S., & Hoermann, S. (2021). Is agile not agile enough? A study on how agile is applied and misapplied in the video game development industry. 2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE), 94–105.
- Osborne O'Hagan, A., Coleman, G., & O'Connor, R. V. (2014). Software development processes for games: A systematic literature review. *Systems, Software and Services Process Improvement: 21st European Conference, EuroSPI 2014, Luxembourg, June 25-27, 2014. Proceedings* 21, 182–193.

The result of the literature review has indicated both the perks and challenges of Scrum in software development as well as video game development. In the reviewed materials are embedded implications that Scrum might not be enough for video game development in a current state and it needs alterations to it. The best way suggested in research papers was to combine Scrum with other methods of managing a project like Kanban, eXtreme Programming, or in some instances even Waterfall. The challenges identified revolved around people and the methodology itself. It was pointed out in the researched papers that inadequate training on Scrum is provided to the teams and that methodology itself when used as a sole method is not enough and needs to be combined with other ones to fill gaps of Scrum not being enough. The limitation to this study was the low number of research done on Scrum in the context of video game development. Video game development is a fairly new industry therefore research is mainly done by enthusiasts who also happen to have knowledge about project management and want to share their observations.

- Politowski, C., Fontoura, L., Petrillo, F., & Guéhéneuc, Y.-G. (2016). Are the old days gone? A survey on actual software engineering processes in video game industry. *Proceedings of the 5th International Workshop on Games and Software Engineering*, 22–28.
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley.
- Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings* 16 October 1995, Austin, Texas, 117–134.
- Schwaber, K., & Sutherland, J. (2020). *The scrum guide*. Scrum Alliance, 21(1), 1–38.
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard business review*, 64(1), 137–146.
- Zada, I., & Shahzad, S. (2015). Issues and implications of scrum on global software development. *Bahria University Journal of Information & Communication Technologies (BUJICT)*, 8(1).