**IGDV Development Management: A Critical Evaluation of Project Management Methodologies**

**Lewis Wilden**

MSc Games Development, University of Suffolk, Ipswich, England

S177026@uos.ac.uk

Abstract

In the game development industry, project management is a key factor to ensure a smooth development process. This report will discuss how data can be generated and collected through different project management methodologies (PMM), and how it can affect the success of the project. The results of this report will assist indie and AAA companies in making a more calculated decision on the most optimal management methodology for their project development process.

Keywords

Project management methodologies, Agile, Waterfall, Game development, Project success, Development process, Data, Communication

# Section 1.

Introduction

Project management methodologies are defined as a set of rules, methods & processes that are used to aid in the project development process. PMM gives project managers a framework for planning and managing the development process of a project. Current methodologies follow either an agile approach like Scrum or a more traditional approach like waterfall.

The digital creative industry has shifted from traditional software development to an agile based

development process (Parmentier and Mangematin,

2014). PMM needs to accommodate the new projects of today and respond to the increasing complexity and dynamic user requirements.

This study will provide a comparison between agile methodologies and traditional methodologies, the comparison will be based on how data is generated

and collected through these methods and the efficiency of the data. The results of this report will assist companies and project managers in choosing a PMM that’s optimal for

the project they are undertaking.

The rest of the paper is structured as follows. Section 2 discusses the literature review, followed by Section 3, which discusses agile and the most popular methodologies. Section 4 discusses traditional methodologies and the most popular. Section 5 will be the results collected from the report, followed by Section 6 which will be concluding by giving advice and opinions on the optimal methodology for games.

# Section 2.

Literature Review

When comparing agile and traditional methodologies it difficult to compare them as they are so different, however, project success needs to be investigated before starting a project and choosing a methodology. According to the 2011 Chaos report projects that used agile succeeded three times more often than waterfall, “software applications developed through the agile process have three times the success rate of the traditional waterfall method”(Standish Group, 2011). The results of this report come from projects conducted from 2002 through 2010, the total number of project in their database and the ratio of agile projects compared to the traditional projects is unknown as well as the reason for project failure.

The development process of a game is very similar to software development, these types of project require design, develop, test and iterate. (Kasurinen, Palacin-Silva and Vanhala, May 20, 2017). An issue with the game industry is that user requirement and technical requirements change, this forces the development process to adapt to these changes. If they do not adapt the project is more likely to fail (Standish Group, 2011).

# Section 3.

Agile Methodologies

Agile was created for projects that require significant flexibility and speed compared to more traditional management. Agile development follows an incremental and iterative development process and is highly flexible allowing for rapid adjustments throughout the project. Instead of massive planning like traditional methods, a project using agile will create a product backlog of tasks also known as story points that are prioritized in levels of importance, these tasks are then compiled in to

a sprint that can last up to 4 weeks. After the sprint is complete the team will review the sprint process, at this point it allows the team to identify issues and improvements they can implement for the next sprint.

An issue with agile development is the lack of detailed documentation and accountability for the project. It is more difficult to receive funding for agile managed project compared to traditional because traditional will have vast amounts of detailed documentation for the project agile will not.

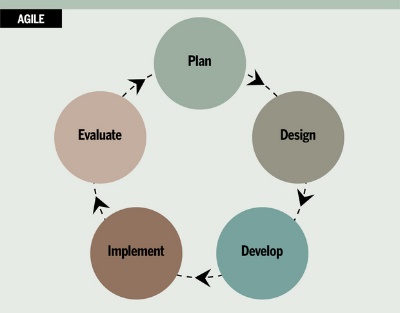


Figure 1: Agile Structure

Examples of methodologies that incorporate agile are as followed;

1. Scrum
2. Scrumban
3. Extreme Programming
4. Feature Driven Development
5. DevOps
6. Lean Software Development
7. Crystal Methodologies
8. Dynamic system development

Application lifecycle management (ALM) tools are a necessary tool when using an agile development approach, ALM provides a detailed insight into current tasks being worked on and the project backlog. Jira, Trello, Octane are examples of ALM tools, however, the “tools should be adaptable to your methodology and processes rather than the

other way around”(SpiraTeam, 2017). This also allows development teams to create work breakdown structures (Appendix A), this allows the team to estimate the amount of time they are going to need for a specific task. To do this you take a single task and break it down into separate elements. You then estimate how long each of the separate tasks will take to make.

**3.1 Scrum**

**Scrum is a methodology that uses a short iteration process called sprints, sprints can include all phases of the development process such as design, testing, implementation, etc.**

**The 12th annual state of agile report discovered that 56% of respondents used the Scrum methodology and 90% of respondents used a form of the daily stand-up meeting from the scum methodology (*12th Annual State of Agile Report*). Another report conducted by the scrum alliance shows that 54% of respondents use scrum in combination with other practices, while 42% reported exclusive use of Scrum. As well as this the report discusses the quality of life for the employees on the project, 87% of employees said it had improved work**

**life over the project development process (Scrum Alliance, 2015).**

**A Unique characteristic of scrum is a daily stand-up meeting or scrum meeting, the meeting is run by a scrum master and will ask three main questions to the team members, What I did Yesterday? What I’ll do today? And What’s in my way? This gives each member of the team a chance to communicate and evaluate the current progress of the sprint, this allows for issues or problems to be discovered quickly and adjusted and gives the entire team all the data. This can increase the productivity of the team members as well as the quality of the work as it reduces the chance of failed sprints.**

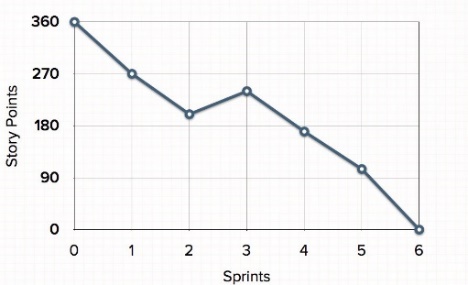
**A Scrum Master may create a burndown chart to track the progress of a project, the chart is updated after every sprint. The burndown chart is an essential tool for an agile managed project and is a way for the team to visualize the progress level after each sprint. Figure 1 shows an example of a burndown chart, as the dev team completes more sprints the number of task or story points should decrease. It can clearly show spikes in tasks or a slowdown in work being completed each sprint. **

Figure 2: Burndown Chart

**According to Schwaber and Sutherland, Scrum is comprised of three main roles which have different**

**tasks and goals in the development process (Schwaber and Sutherland, 2017). Appendix C shows these roles and their main responsibilities**

**3.2 Extreme Programming (XP)**

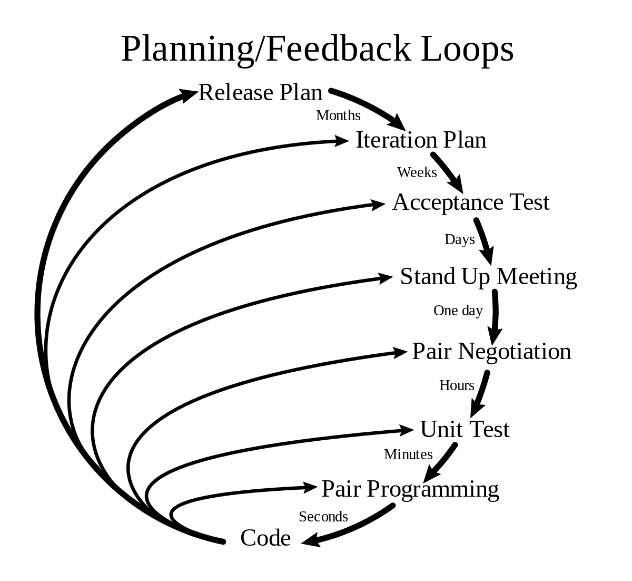
XP is a highly flexible agile based methodology that is used in the software development industry, the basis of XP development code-writing, test, iterate. Like many agile methodologies XP “Is primarily designed for smaller teams”(Fojtik, 2011). XP follows five core values;

Figure 3: XP Model

1. **Communication**

Poor communication is the reason behind many project failures, this can occur between developers and customers. XP uses a special role called a coach, the sole responsibility of a coach is to ensure good communication and detect any inconsistency in communication. Communication between developers is just as important as customers, XP uses an approach call pair programming in which two programmers use a single computer to create the code together. The communication between the two must be high and consistent throughout development to avoid issues and bug.

1. **Simplicity**

XP at tries to make the development process as simple as possible by only focusing on the most important aspect of the project, and not worrying about future addition/mechanics. “we should not create a more robust architecture than necessary for the moment” (Fojtik, 2011). It gives pair programmers the time to get the core functionality perfect before adding additional features to it.

1. **Feedback**

Constant feedback for any development project is important however XP requires feedback at every development stage, this is to ensure that the project is exactly what the customer wants. A study that evaluated XP development of specific software explore how feedback aided the development of software for disabled users “the main reason was the need of the developers to

become familiar with possibilities of users with specific disorders” (Fojtik, 2011).

1. **Courage**

Above all else, the developer must have the courage to re-write, remove and change the code to ensure as many errors are removed, and the program is running as efficiently as possible.

1. **Respect**

Development teams work closely together during projects, XP values connection between employees that emphasises and improves teamwork.

These five core values combine into a simple methodology that incorporates fast and efficient code generation that incorporates high levels of communication and interaction between employees and users (English, 2002).

**3.3 Kanban**

Kanban is a methodology that focuses on continuous delivery of data, whilst optimizing workload for the development team so they’re not overworked. The principles of Kanban are as followed

1. **Visual Workflow**

The Kanban system uses boards to visualize the current workflow of a project, these can be virtual or physical using sticky notes. In a virtual board, cards would be added to the board that represents a task, the task would move depending on the current state (Plan, in progress, validate, Completed). This allows the dev team to self-organise themselves and allows the team to observe current progress (Ikonen *et al.,* 2011).

1. **Limit work in progress (WIP)**

WIP is used to manage and limit the amount of work that employees can work on, this allows the dev team to focus on the most important task (Power and Conboy, 16 May, 2015). This attempts to manage the workflow and increase the quality of the most important features without having to worry about other tasks.

1. **Clear and explicit policies**

Knowing when a task is complete or is ready to move to the next step of development is necessary for the dev team to know and understand. It allows for accurate planning for the time required on the task and gives a clear criterion to be completed.

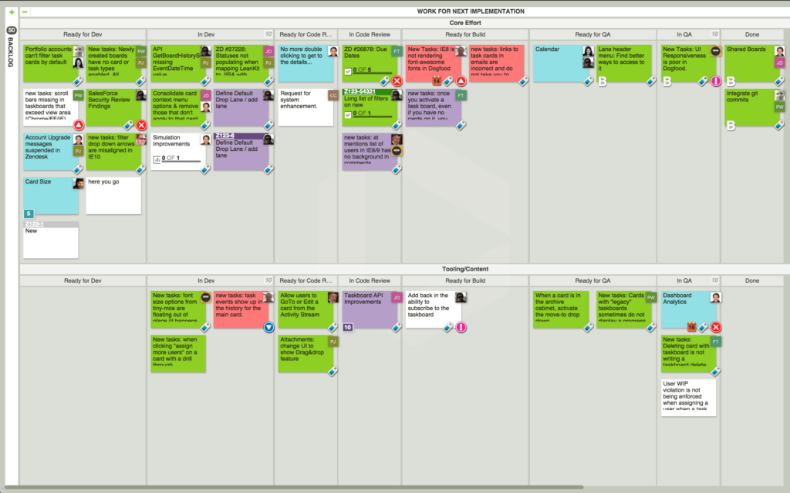


Figure 4: Kanban board to visualize tasks.

In my own experience, I have used agile scrum when developing projects, whilst using tools like Jira to set and manage workflow. This has allowed for a flexible development process with an emphasis on communication between the team members.

# Section 4.

Traditional Methodologies

In traditional project management, the development process is divided into static phases, these phases comprise of analysis, design, implementation, testing, deployment, and maintenance or commonly known as the waterfall method. These phases must be executed in this specific order, as it allows for increased control throughout each phase and offers a lot of formal planning before development is underway. Due to this massive planning, it can be beneficial for the directors of the company as they control the finances available to the company, they will be more inclined to give them the funds for the project if they have a clear plan for development. A tool that needs to be used for traditional methods are risk assessments (Appendix B), due to fact that traditional management does not allow you to go backwards in the development phase, the team will need to know the biggest risk for the project and pre-plan before development starts.

Traditional management is extremely inflexible due to the linear structured development process, it is possible to change aspects of the project during development however the changes must be justified and must go through control procedures.

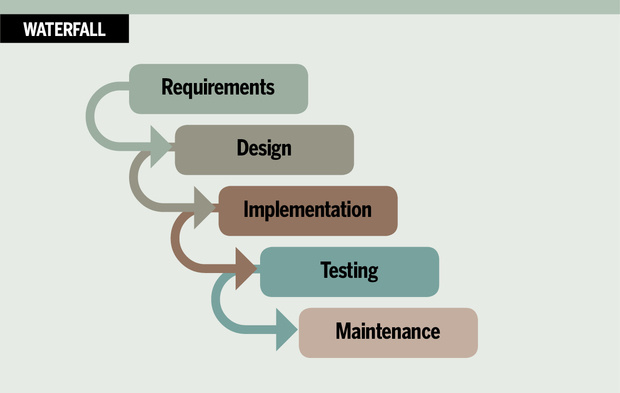


Figure 5: Traditional Structure called waterfall

Examples of traditional and structured management are as followed;

1. Structured System Analysis & Design Method (SSADM)
2. Waterfall
3. Prince 2
4. V Model
5. Rapid Application Development (RAD)

**4.1 V-Model**

V-Model is an extension of the waterfall methodology, instead of moving down like waterfall the process phases form a V shape.

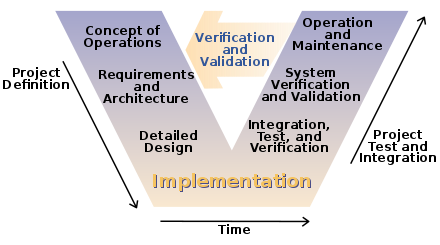


Figure 6: V-Model

V-Model is a well-structured methodology, with each phase of development having extensive documentation. The purpose of V-Model is to improve efficiency and effectiveness of software development by incorporating testing into the entire development process “Software testing is too important to leave to the end…incorporates testing into the entire software development life cycle” (Mathur and Malik, 2010). V-Model is highly effective in creating high quality, polished product however due to its traditional static phases you are still unable to change elements from the design phase of the project.

**4.2 Rapid Application Development (RAD)**

The biggest issue with traditional methods like waterfall is the inflexibility of the development process, RAD is a methodology that attempts to alleviate this problem, according to Hugh Mackay “it claims greater flexibility regarding changing requirements and the changing environments” (Mackay *et al.,* 2000).

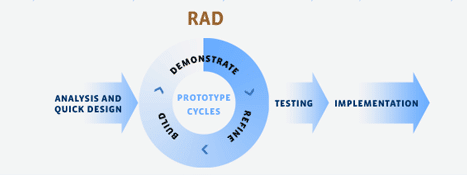


Figure 7: RAD Model

Although RAD is considered to be a traditional method like waterfall, the process of development is closer to an agile approach due to its quick iterative development, “a prototype of the product is quickly created, tested for usability, and then revised”(Lohr *et al.,* 2003). RAD still follows the static phases like other traditional methods that allows for extensive documentation and control of the project, however, it offers the possibility to break the boundaries between the dev team and the user. During project development users are involved at the start of the project, users can fluidly communicate with the developers about the good and bad points of the project which enables the dev team to build a project for the user “RAD offers explicitly the possibility of reconfiguring the boundary (between developer and users)” (Mackay *et al.,* 2000). The constant feedback from the user can allow the evolution of a project that is further targeted to their audience.

The cooperation between the dev team and the user can improve a projects quality and overall sales, however, it can become extremely expensive depending on development time to constantly bring in customers. RAD requires experienced staff when working on a project, it requires all the necessary skills that are required for the development process

# Section 5.

Results and Discussion

Both agile and traditional methodologies have their advantages and disadvantages, however in relation to game development agile supports key factors for development.

Traditional methodologies support vast documentation and accountability for a project, however, within game development, the constant change in hardware and customer requirements make it difficult to maintain this. RAD is a method that tried to fix by involving the customer into the development process, this however it still requires heavy documentation. RAD, however, can become expensive for smaller as they don’t have the financial support to bring in users. V-Model still uses the static phases of traditional development but increase testing and verification however it still fails to adapt to changes in the development process.

Agile on the other hand is an incremental and iterative development process that gives developers the ability to respond to changes in technology and user requirements. Scrum takes the principle of agile and improves teamwork, communication and flexibility through scrum meetings. These meetings give the team the ability to find and eradicate problem with the project early in the development phase. Kanban is a methodology that focuses on continuous delivery of data, whilst optimizing workload, Kanban limits work in progress tasks to improve the quality of the single most important. XP is a methodology that heavily focuses on high levels of communication and improved teamwork, however above all else the project must be delivered to a high standard.

# Section 6.

Conclusion

In this report, I have discussed how different methodologies collect data and how they adapt to changes in the development process. This report has shown that the traditional methodologies can work in project development but only if the company can guarantee the requirements of the project will not change during the development phase. In relation to game development this does not work, from personal experience and my research has shown the development process of a game requires a highly flexible, incremental and iterative development process.

Agile methodologies provide a framework that allows for these key factors of development. My recommendation is using a combination of Scrum and XP, this gives the development process a high level of communication and improved teamwork between employees. The use of scrum **can increase the productivity of the team members as well as the quality of the work as scrum meetings can discover issue quickly. With XP, a coach will purely focus on the communication between everyone included in the development process, this allows for clear and precise communication. The use of pair programming will also improve the overall quality of the core of the project.**

# References

*12th Annual State of Agile Report.* Collab.net VersionOne.com. Available at: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report> (Accessed:15.12.18) .

English, A. (2002) 'Extreme programming, it's worth a look', *IT Professional,* 4(3), pp. 48-50. doi: 10.1109/MITP.2002.1008537.

Fojtik, R. (2011) 'Extreme Programming in development of specific software', *ScienceDirect,* 3, pp. 1464-1468.

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P. and Abrahamsson, P. (2011) *On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation.* . 27-29 April 2011. IEEE, pp. 305.

Kasurinen, J., Palacin-Silva, M. and Vanhala, E. (May 20, 2017) *What concerns game developers?* . 23-23 May 2017. IEEE Press, pp. 15.

Mackay, H., Carne, C., Beynon-Davies, P. and Tudhope, D. (2000) 'Reconfiguring the User: Using Rapid Application Development', *Social Studies of Science,* 30(5), pp. 737-757.

Mathur, S. and Malik, S. (2010) 'Advancements in the V-Model', *International Journal of Computer Applications,* 1(12), pp. 29-34. doi: 10.5120/266-425.

Parmentier, G. and Mangematin, V. (2014) *Orchestrating innovation with user communities in the creative industries*.

Schwaber, K. and Sutherland, J. (2017) *The Scrum Guide.* Available at: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> (Accessed: 10.12.18).

Scrum Alliance (2015) *The 2015 State of Scrum Report.* Scrum Alliance. Available at: <https://www.scrumalliance.org/ScrumRedesignDEVSite/media/scrumalliancemedia/files%20and%20pdfs/state%20of%20scrum/scrum-alliance-state-of-scrum-2015.pdf> (Accessed:15.12.18) .

SpiraTeam (2017) *What is Application Life cycle Management?* Available at: <https://www.inflectra.com/spirateam/highlights/understanding-alm-tools.aspx> (Accessed:15.12.18) .

Standish Group (2011) *CHAOS Report.* The Standish Group. Available at: <https://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall/comments> (Accessed: 29.12.18).Stare, A. (2014) *Agile Project Management in Product Development Projects*

# Appendix A



Example of Work breakdown structure from my group project

# Appendix B

|  |  |  |  |
| --- | --- | --- | --- |
| Likelihood | Impact | Combined | Risk |
| 1 | 1 | 3 | No playtesters for the game |
| 2 | 3 | 4 | Unable to achieve specific emotion |
| 3 | 5 | 8 | 2 Week prototype goal |
| 4 | 2 | 7 | Over scope |
| 5 | 4 | 9 | Tools used in the project |

Risk Assessment: Example is taken from my group project

# Appendix C

|  |  |  |
| --- | --- | --- |
| **Product Owner** | **Development Team** | **Scrum Master** |
| **Clear product backlog** | **Self-organizing,** | **the scope is understood by everyone** |
| **Ordering backlog to best achieve goals** | **Cross-functional team members,** | **Facilitating Scrum meetings/ updating tools** |
| **Optimizing the value of dev teams work** | **Accountability belongs to the dev team as a whole** | **Coaching Dev team in Self-Organization** |
| **Ensuring dev team understands the backlog** | **No titles for dev team members, regardless of work** | **Leading and coaching organization in the scrum** |

Scrum Team

# Bibliography

*12th Annual State of Agile Report.* Collab.net VersionOne.com. Available at: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report> (Accessed:15.12.18) .

Alexander, M. (2018) *How to pick the best project management methodology for success.* Available at: <https://www.cio.com/article/2950579/methodology-frameworks/how-to-pick-a-project-management-methodology.html> (Accessed: Dec 20, 2018).

Anonymous (2009) 'An Agile Cure for All Ills?', *IEEE Software,* 26(6), pp. 8. doi: 10.1109/MS.2009.168.

Azanha, A., Ana Rita Tiradentes Terra Argoud, João Batista de Camargo Junior and Pedro Domingos Antoniolli (2017) 'Agile project management with Scrum', *International Journal of Managing Projects in Business,* 10(1), pp. 121-142. doi: //dx.doi.org/10.1108/IJMPB-06-2016-0054.

Ballard, G., Tommelein, I., Morten Schiefloe, P. and Rolstadås, A. (2008) 'International journal of managing projects in business', 7(4), pp. 638-660.

Beck, K. (1999) 'Embracing change with extreme programming', *Computer,* 32(10), pp. 70-77. doi: 10.1109/2.796139.

Brown, S.L. and Eisenhardt, K.M. (1995) 'Product development: Past research, present findings, and fu', *Academy of Management.The Academy of Management Review,* 20(2), pp. 343.

Bryde, D.J. (2003) 'International journal of operations & production management', *International journal of operations & production management,* 23(7), pp. 775-793.

Cohn, M. (2010) *Agile Game Development with Scrum.* 1st edn. Boston, Massachusetts, United States: Addison-wesley Professional.

Conboy, K., Coyle, S., Wang, X. and Pikkarainen, M. (2011) 'People over Process: Key Challenges in Agile Development', *IEEE Software,* 28(4), pp. 48-57.

Conforto, E.C. and Amaral, D.C. (2016) *Agile project management and stage-gate model—A hybrid framework for technology-based companies*.

Cugola, G. and Ghezzi, C. (1998) *Software Processes:  
a Retrospective and a Path to the Future.* . 14 -17 June 1998. Lisle, IL: .

Daly, L. *Scaling agile in large organizations.* Available at: <https://www.atlassian.com/agile/agile-at-scale> (Accessed: Jan 2, 2019).

English, A. (2002) 'Extreme programming, it's worth a look', *IT Professional,* 4(3), pp. 48-50. doi: 10.1109/MITP.2002.1008537.

Esperança Amengual Alcover, Antoni Jaume-i-Capó and Biel Moyà-Alcover (2018) 'PROGame: A process framework for serious game development for motor rehabilitation therapy', *PLoS One,* 13(5), pp. e0197383. doi: 10.1371/journal.pone.0197383.

Fernandez, D. and Fernadez, J. (2009) 'Agile Project Management - Agilism versus Traditional Approaches', *The Journal of Computer Information Systems,* 49(2), pp. 10-16.

Fojtik, R. (2011) 'Science', *Science,* 3, pp. 1464-1468.

Goh, J.C., Pan, S.L. and Zuo, M. (2013) 'Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives', *J. AIS,* 14, pp. 1.

Greaves, K. (2011) *Taming the Customer Support Queue.* . 2011. IEEE, pp. 154.

Grimstad, S. (October 24, 2005) *Understanding of estimation accuracy in software development projects.* . 19-22 Sept. 2005. Como, Italy: IEEE, .

Highsmith, J. and Cockburn, A. (2001) 'Agile software development: the business of innovation', *Computer,* 34(9), pp. 120-127. doi: 10.1109/2.947100.

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P. and Abrahamsson, P. (2011) *On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation.* . 27-29 April 2011. IEEE, pp. 305.

Joslin, R. and Müller, R. (2015) *Relationships between a project management methodology and project success in different project governance contexts*.

Kasurinen, J., Palacin-Silva, M. and Vanhala, E. (May 20, 2017) *What concerns game developers?* . 23-23 May 2017. IEEE Press, pp. 15.

Kostalova, J., Tetrevova, L. and Svedik, J. (2015) *Support of Project Management Methods by Project Management Information System*.

Livermore, J. (23 April, 2007) *Factors that impact implementing an agile software development methodology.* . 22-25 March 2007. Richmond, VA, USA: IEEE, .

Livermore, J. (15 May, 2006) *What Elements of XP are being Adopted by Industry Practitioners?* . 31 March-2 April 2006. Memphis, TN, USA: IEEE, .

Lohr, L., Javeri, M., Mahoney, C., Gall, J., Li, K. and Strongin, D. (2003) 'Using Rapid Application Development to Improve the Usability of a Preservice Teacher Technology Course', *Educational Technology Research and Development,* 51(2), pp. 41-55.

los Angeles, J.d. (2018) *Project Framework Comparisons: Agile vs. Waterfall vs. Hybrid vs. Lean.* Available at: <https://medium.com/@jdelosangeles/project-framework-comparisons-agile-vs-waterfall-vs-hybrid-vs-lean-dc6801d217e4> (Accessed: .

Mackay, H., Carne, C., Beynon-Davies, P. and Tudhope, D. (2000) 'Reconfiguring the User: Using Rapid Application Development', *Social Studies of Science,* 30(5), pp. 737-757.

Manole, M. and Avramescu, M. (2017) 'Comparative Analysis of Agile Project Management Tools', *Academy of Economic Studies.Economy Informatics,* 17(1), pp. 25-31.

Marcelino-Sádaba, S., Pérez-Ezcurdia, A., Echeverría Lazcano, A.M. and Villanueva, P. (2014) *Project risk management methodology for small firms*.

Matharu, G., Mishra, A., Singh, H. and Upadhyay, P. (2015) 'Empirical Study of Agile Software Development   
Methodologies: A Comparative Analysis ', *ACM SIGSOFT Software Engineering,* 40(1).

Mathur, S. and Malik, S. (2010) 'Advancements in the V-Model', *International Journal of Computer Applications,* 1(12), pp. 29-34. doi: 10.5120/266-425.

O'Sheedy, D. (2012) *A study of agile project management methods used for IT implementation projects in small and medium-sized enterprises.* . Southern Cross University.

Parmentier, G. and Mangematin, V. (2014) *Orchestrating innovation with user communities in the creative industries*.

Pathak, K. and Saha, A. (2013) 'Review of   
Agile Software Development   
Methodologies', *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(2), pp. 270-276.

Petrillo, F., Pimenta, M., Trindade, F. and Dietrich, C. (2009) 'What Went Wrong? A Survey of Problems in  
Game Development', *Computers in Entertainment,* 7(1).

Power, K. and Conboy, K. (16 May, 2015) *A Metric-Based Approach to Managing Architecture-Related Impediments in Product Development Flow: An Industry Case Study from Cisco.* . 16-16 May 2015. IEEE, pp. 15.

Rahman, A.A.U., Helms, E., Williams, L. and Parnin, C. (October 01, 2015) *Synthesizing Continuous Deployment Practices Used in Software Development.* . 3-7 Aug. 2015. Washington, DC, USA: IEEE, .

Rahman, N., Sharif, S. and Esa, M. (2013) 'Lean Manufacturing Case Study with Kanban System Implementation', *Procedia Economics and Finance,* 7, pp. 174-180.

Ramnath, V. (2010) 'APPLICATION OF KANBAN SYSTEM FOR   
IMPLEMENTING LEAN MANUFACTURING  
(A CASE STUDY)', *Journal of Engineering Research and Studies,* 1(1).

Rasnacis, A. and Berzisa, S. (2017) *Method for Adaptation and Implementation of Agile Project Management Methodology*.

Ribeiro, A. and Domingues, L. (2018) *Acceptance of an agile methodology in the public sector*.

Scheid, J. (2010) *Project Management Methodologies Comparison: Pros and Cons of Various PM Methods, Project Management Methodologies Comparison: Pros and Cons of Various PM Methods,* -03-24T00:00:00+00:00. Available at: <https://www.brighthubpm.com/methods-strategies/67087-project-management-methodologies-how-do-they-compare/> (Accessed: Dec 20, 2018).

Schmalz, M., Finn, A. and Taylor, H. (2014) *Risk Management in Video Game Development Projects.* . 6-9 Jan. 2014. IEEE, pp. 4325.

Schwaber, K. and Sutherland, J. (2017) *The Scrum Guide  
.* Available at: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> (Accessed: 10.12.18).

Scrum Alliance (2015) *The 2015 State of Scrum Report  
.* Scrum Alliance. Available at: <https://www.scrumalliance.org/ScrumRedesignDEVSite/media/scrumalliancemedia/files%20and%20pdfs/state%20of%20scrum/scrum-alliance-state-of-scrum-2015.pdf> (Accessed:15.12.18) .

Serrador, P. and Pinto, J.K. (2015) *Does Agile work? — A quantitative analysis of agile project success*.

Simon, L. (2006) *Managing creative projects: An empirical synthesis of activities*.

SpiraTeam (2017) *What is Application Life cycle Management?* Available at: <https://www.inflectra.com/spirateam/highlights/understanding-alm-tools.aspx> (Accessed:15.12.18) .

Špundak, M. (2014) *Mixed Agile/Traditional Project Management Methodology – Reality or Illusion?*.

Standish Group (2011) *CHAOS Report  
.* <https://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall/comments>: The Standish Group. Available at: <https://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall/comments> (Accessed: 29.12.18).Stare, A. (2014) *Agile Project Management in Product Development Projects*.

Stoica, M., Ghilic-Micu, B., Mircea, M. and Uscatu, C. (2016) 'Analyzing Agile Development - from Waterfall Style to Scrumban', *Informatica Economica,* 20(4), pp. 5-14. doi: //dx.doi.org/10.12948/issn14531305/20.4.2016.02.

Van Os, A., Van Berkel, F., De Gilder, D., Van Dyck, C. and Groenewegen, P. (2015) *Project risk as identity threat: explaining the development and consequences of risk discourse in an infrastructure project*.

Vaskimo, J. (2012) '  
  
Invitation to participate in project management methodology research', *PM World Journal,* 1(1).

Vijayasarathy, L. and Butler, C. (2015) 'Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?', *IEEE Software,* 33(5), pp. 86-94. doi: 10.1109/MS.2015.26.

Wells, H. (2012) *How effective are project management methodologies (PMMs)?* Limerick, Munster, Ireland.: Project Management Institute, .

Yadav, R. (2012) 'Improvement in the V-Model  
', *International Journal of Scientific & Engineering Research,* 3(2).

Yamada, S., Aoki, T. and Toyota, T. (2008) 'Quantitative Software Quality/Reliability Prediction Based on Project Management Data for Waterfall and Agile Development Paradigms', *Opsearch,* 45(4), pp. 391-404. doi: //dx.doi.org/10.1007/BF03398828.