**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 (150-250)

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 (200-400)

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 have to need to accommodate the new projects of today and respond to the increasing complexity of projects 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 my results and giving advice and opinions.

# Section 2.

Literature Review (500-700)

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.

# Section 3.

Agile Methodologies (600-900)

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 work/tasks that are prioritized in levels of importance, these tasks are then completed in sprints that can last up to 4 weeks, after the sprint is complete the team will demonstrate what they’ve completed. At this point it allows the team to identify improvements for the next sprint.

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).

**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 a chance to communicate with the entire team 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.**

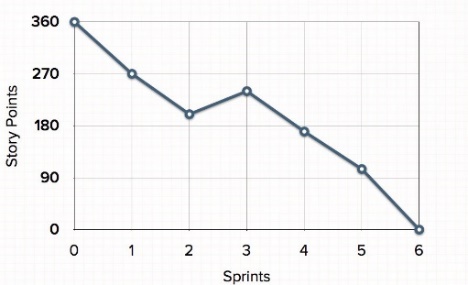
**A Scrum Master may create a burndown chart to track the progress of a project, the chart is updated after every sprint. The burndown chart is an essential tool for an agile managed project and is a way for the team to visualize the progress level after each sprint.**

Figure 1

**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).**

|  |  |  |
| --- | --- | --- |
| **Product Owner** | **Development Team** | **Scrum Master** |
| **Clear product backlog** | **Self-organizing,** | **Goals, 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** |

Figure 2: Table Showing main roles in scrum and their 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

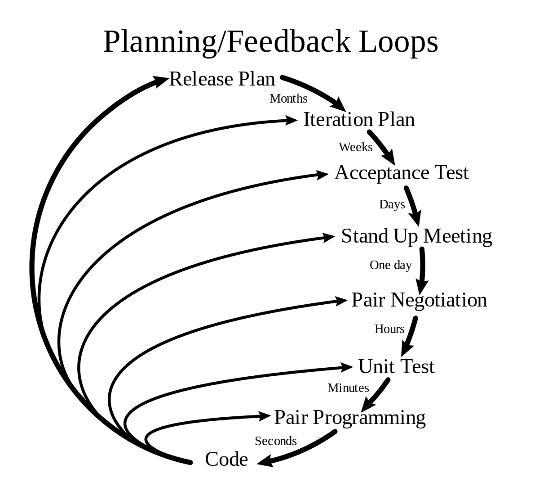
****

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 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 communication.

**3.3 Kanban**

# Section 4.

Traditional Methodologies (600-900)

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. Traditional management, however, 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. 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.

V-Model is a well structed 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).

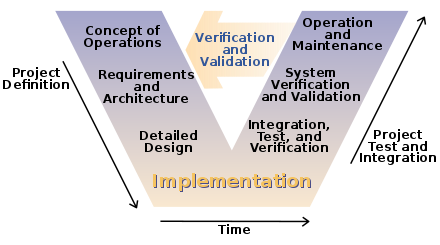


Figure 4: Basic V-Model

**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). 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 dev 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.

# Section 5.

Results and Discussion (200-400)

# Section 6.

Conclusion (200-300)

Key Consideration in choosing the methodology

* Organizational strategic goals and core values
* Constraints
* Stakeholders
* Risks
* Complexity
* Project Size
* Cos