INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Software Engineering Challenges in Game Development

Prof Manjula R[1], Chakradhar Raju M[2], Sai Chand M[3]

Computer Science Department, VIT University

Abstract— Game development is the software process that results in the production of a video game. In this research paper our main aim is to focus on various scenarios and challenges in game development. It is unique in that it requires a combination of work of teams from multiple disciplines (art, music, Graphics, programming, etc.,) and that engaging game play is sought after the use of prototypes and iterations. The challenges that rise from these differences are unique, but they can still be overcome by using some of the methods used in traditional software engineering. This paper describes these challenges and explains some engineering practices to help dealing with them.

Keywords—software development process, Prototype, Traditional software engineering, Software industry, Engineering practice

I.   INTRODUCTION

Game development is a process that starts from a core concept or an idea. Sometimes the idea might be on the modification of an already existing game. Designers often experiment with different combinations of genres as the game might involve different genres. Usually, the Game designer reports initial game proposal document that includes the game play, concept, feature list, story, target audience, requirements and schedule, staff & budget estimates.

Game development has many individual characteristics which will create many challenges for the management of the projects. Game developers need to do graphics, sound, music, combine code, etc. and create an entertaining and fun experience for the player. The variety and the shear amount of these advantages is a major problem. Now a days game projects are very large. Estimating the range of the project is becoming difficult and the problem is often made worse with feature creep functionality. A game project may involve thousands of people and take years to complete. People involved in project come from different places, which makes effective communication challenging. The developers are divided into teams by their specialty (programming team, animation team) which helps the sharing of information between people but makes communication between groups more difficult. Game development can be majorly divided into three phases: 1.preproduction, 2.production, and 3.testing.
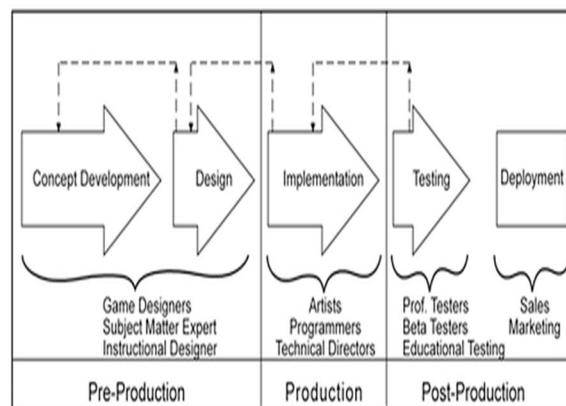

Fig. 1 different phases involved in game development.

Our prime focus is on the challenges faced in game development from the initial stages like the number of the developers available to the final outcome of the game, which includes initial project overview, software project models, functional and non-functional requirements, various test-cases and finally meeting the user requirements.

II.   RELATED WORK

Game development is a software evolution process, as a video game is software with gameplay, audio, and art. Formal software

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

development methods often fails to be noticed. Games with poor development methodology are likely to run over budget and time estimates, as well as containing a large number of bugs. Planning is very important for every individual and group projects similar. Altogether game development is not suited for software life cycle methods, such as the waterfall model, incremental model. One method which is employed for game development is agile development. It is mainly based on iterative prototyping, a part of software prototyping. Agile development depends on feedback and improvement of game's iterations with steadily increasing feature set. This method is successful because most projects do not start with a clear requirement profile. A favoured method of agile software development is Scrum. Other successful method is Personal Software Process (PSP) requiring additional instructions for employees to increase awareness of project's planning. This method is more expensive and requires dedication of team members. PSP can be enlarged to Team Software Process, where the entire team is acting freely and independently. Game development involves an overlap of these methods. For example, asset design may be done using waterfall model, because requirements and specifications are clearly expressed, but gameplay design could be done using iterative prototyping.
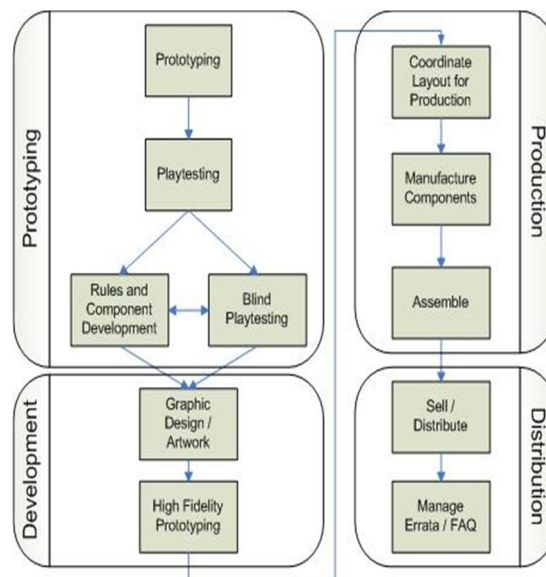

Fig. 2 stage wise functionality in game development.

Development of a commercial game usually includes the following stages:

A. Pre-production or design stage is a planning stage of the project focused on idea and concept development and manufacturing of initial design documents. The aim of concept development is to generate clear and easy to understand documentation, which describes all the problems, schedules for the development team. Before a thorough production can begin, the development team produces the initial version of a game design document including all or most of the information from the initial pitch; pitch is a short synopsis document considered to present the game's selling points and details about why the game would be profitable to develop. The design document describes the game's concept and crucial gameplay elements in particular. It may also include initial sketches of various features of the game. Design document is sometimes guided by functional prototypes of some parts of the game. Design document remains a living document throughout the development, often changed weekly or even daily.

B. Production is the main phase of development, when features and source code for the game are produced. Normal production is generally defined as the period of time when the project is completely staffed. Artists develop game features, such as 3D models, and programmers write new source codes. Sound engineers develops sound effects and composers develops music for the game. Level designers creates levels and writers writes dialogues for cut scenes and NPCs. Game designers continues to develop the game's design throughout production.

C. Testing is generally the last phase before the complete game is sent to the publisher. The testers look for defects and push the game to the limits. Testers start work once anything is playable. This may be one level or part of the game software that can be used to any reasonable range. Early on, testing a game occupies a relatively small amount of time. Testers may work on several games at a time. As development comes to a close, a single game generally employs many testers full-time (and often with overtime). They try hard to test new features and regression test existing ones. Testing is essential for modern, complex games

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

as single changes may lead to terrible results.

III.   PROBLEMS & SOLUTIONS

Issue 1: A contemporary game may use several input devices (such as keyboards, gamepads and joysticks) and may include different fields of science like advanced physics, 3D graphics, artificial intelligence ,digitised sound, complex strategy and may be playable against other people through the Internet or over a LAN. Each aspect of the game can require all skills of a programmer and, in many cases, several programmers. The number of programmers required for each module depends on the level of programmers' skills, mostly by the type of game being developed. Developer's involved come from different disciplines, which might make effective communication among them difficult and challenging.

Solution: The developers are often segregated into teams by their specialization (programming team, animation team). This aids in easy information sharing among people of same discipline but makes communication between groups more complex.

Issue 2: Generally, Third party software engines are used by Game developers for game engines, Artificial Intelligence (AI), etc. Game engine; a software framework for the creation and development of games. Developers use game engines to create games for mobile devices, consoles and personal computers. The core functionality provided by a game engine typically includes a 2D or 3D graphics rendering engine, a physics engine , animation, scripting, sound, networking, artificial intelligence streaming, memory management,   localization support, threading and may include video support for cinematics. The process of game development is often economized by adapting the same game engine to create different games, or for easy game porting to multiple platforms. This reduces the costs to a great extent but some issues are to be considered.

Solution: Careful decision should be taken on which engines to use as some engines are best used on certain types of games. Developers should be careful enough not to make their games very much similar to other games made using the same engine. Third party components must be carefully evaluated to determine which one is the best choice for the job.

Issue 3: Because the industry is so different from typical software industry, the managers need to understand the existing process and carefully consider which development processes works and which ones do not. They need to sort out which methodologies work best with their project teams. Spiral model methodology may work well for large projects. The cycles of the spiral model can be planned in a way so that the game can be tested at the end of each cycle. This would allow better testing of the game during development phase. Scrum is another development model that produces incremental iterations/cycles of the game. In scrum model the goal of every cycle is to have a reusable software. In production phase the more formal spiral process can be used.

Solution: Companies need to invest in their project managers training, so that they get the skills to evaluate different methods adapted and identify the ones that are right for their teams and the project. It would also help the team to see the game potential and to adapt to possible changes in the schedule and design of the game.

Issue 4: Most commercial computer games are written primarily in basic C, C++ and assembly language. Many games, specifically those with highly complex gameplay mechanics, require hardware to limit. As such, highly optimized code is essential for these games to run at an acceptable frame rate. Due to this reason, compiled code is typically used for critical performance-components, such as visual rendering. Also almost all PC games use either the OpenGL API's, DirectX or some wrapper library, so as to interface with hardware devices. Many web based games use java as it is cross-platform, doesn't require installation by the user usually, and poses fewer security risks when compared to a downloaded executable program. Java is also a preferred language for mobile based games. Java Script and Adobe Flash are popular development tools for browser-based games.

Solution: As games have grown both in size and complexity, middleware is becoming popular within the industry. Compared to standard lower level APIs such as DirectX and OpenGL, Middleware provides higher and greater level functionality and larger feature sets, such as skeletal animation. Some middleware also supports platform independent that would make common conversions much easier, for example Microsoft Windows to PS4.Middleware is essentially aimed at removal of redundancy in the development cycle as much as possible (for example, consider a studio development team writing new animation systems for each game), which would allow game developers to focus on new content.

Issue 5: Most of the games need to be translated into different languages, as the audience scope is wide and they speak different languages. Localization is the process of translating the linguistic assets of a game into other different languages. As the process is complex and time taking, it needs a little long time to translate the game's original language into respective regional languages.

Solution: The process of localization is a complex task; yet developers focus on it to expand their market and increase their accessibility to a wide range of users. Translation is of different levels, zero translation: no translation of the product and all things are raw, basic translation: includes translation of only few features of game and a full translation: includes complete translation of

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

the game. Localization also includes adjusting various game assets for different cultures so that the game can be reachable to all sectors of users in the world. Now-a-days most of the commercial game developers consider this factor into account and the games are being released in different languages simultaneously.

Issue 6: Testing/quality assurance is vital for modern, complex games as single change may lead to catastrophic consequences. As new features are added, subtle changes to the code base can produce unexpected changes in different areas of the game. This task/function is often overlooked by developers for several reasons. Sometimes when a feature in a game is implemented and tested, the feature is considered working for the rest of the project throughout the development phase and a very little attention will be given to perform repeated testing. Also, features that are added late to the game in the process of development are prioritized and existing features are neglected and often receive insufficient testing time. Proper testing is also becoming increasingly expensive as the number of features increases and is often scheduled incorrectly. Some game developers and publishers fail to test the full features of the game and ship it with bugs/errors. This can result in customer's dissatisfaction and failure to meet the goals.

Solution: Testers need to adopt testing strategies in a way that the testing covers entire features of the game. As an example regression testing can be carried out to make sure that features that have been in the game still operate correctly. Regression testing can be considered as one of the vital tasks required for effective game development. Most developers release patches for the games to fix the bugs and make the game fully playable again.

Issue 7: After the game is developed/shipped, the maintenance phase of the game starts. In the past there was no maintenance period for the video game consoles. As a result, the game would have bugs and updates that remain forever. All the consoles faced the same problem as they had identical or nearly identical hardware equipment and incompatibility levels. In such cases maintenance would only happen if there is a chance of making a sequel or if an updated version is being developed as the process includes reuse of original game engine and features. Contrary to that the maintenance phase begins immediately after the shipping of multiplayer online games (MMOG). Such games require continuous maintenance as the game scenario/interface is continuously changed along with which new features are added and it also requires a large number of maintenance staff.

IV.   SUGGESTIONS

Agile methods could work well for the preproduction stage for fast iterations and prototyping. At the production stage a more formal spiral model is advised. The Project manager can develop a sequence of tasks derived from the Game Design Document (GDD). GDD usually contains both stated and implied game requirements. The Project manager must clarify them for the production teams. Requirements engineering skills are very important at this phase. Errors that happen due to differences between designers and the production team can be reduced using requirements engineering. Risk management is also important to handle feature creep as additional features should be carefully considered as to how important they are and are they worth the extra effort. Risk management and Requirements engineering also helps with the project scope problems, which often result from translation of the Game Design Document (GDD) to a project plan. Different process methods can be combined to improve the development process. Third party components should be carefully analysed to determine which one is the best choice for the job.

V.   CONCLUSIONS

Game development is in many aspects different from the traditional software industry but adapting software engineering practices in the process of game development would definitely help in overcoming the challenges. . Management need to be instructed to understand different methodologies, so that they can select the ones that are righteous for their team and project. The scope estimations can help to predict success of budgets and schedules but they can never be exact. The tools and pipeline should be optimized for handling large amounts of diverse features. The teams need to understand that they need to work together. Game development is always a group effort.

REFERENCES

[1]   Adams, Ernest; Rollings, Andrew (2003). Andrew Rollings and Ernest Adams on game design. New Riders   Publishing. ISBN 159273-001-9.
[2]   Bates, Bob (2004). Game Design (2nd ed.). Thomson Course Technology. ISBN 1-59200-493-8.
[3]   ].Bethke, Erik (2003). Game development and production. Texas: Wordware Publishing, Inc. ISBN 1-55622-951-8.
[4]   Brathwaite, Brenda; Schreiber, Ian (2009). Challenges for Game Designers. Charles River Media. ISBN 1-58450-580-X.
[5]   Chandler, Heather Maxwell (2009). The Game Production Handbook (2nd ed.). Hingham, Massachusetts: Infinity Science Press. ISBN 978-1-934015-40-7.
[6]   McGuire, Morgan; Jenkins, Odest Chadwicke (2009). Creating Games: Mechanics, Content, and Technology. Wellesley, Massachusetts: A K Peters. ISBN 978-1-56881-305-9.
[7]   Salen, Katie; Zimmerman, Eric (2003). Rules of Play: Game Design Fundamentals. MIT Press. ISBN 0-262-24045-9.
[8]   Moore, Michael E.; Novak, Jeannie (2010). Game Industry Career Guide. Delmar: Cengage Learning. ISBN 978-1-4283-7647-2.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

[9]   Oxland, Kevin (2004). Gameplay and design. Addison Wesley. ISBN 0-321-20467-0.

[10]   Salen, Katie; Zimmerman, Eric (2005). The Game Design Reader: A Rules of Play Anthology. The MIT Press. ISBN 0-262-19536-4.

[11]   McShaffry, Mike (2009). Game Coding Complete. Hingham, Massachusetts: Charles River Media. ISBN 978-1-58450-680-5.

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)