

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/279998361>

# Incorporating a Game Design Document into Game Development Projects Deliverables

Conference Paper · July 2014

CITATIONS

0

READS

354

3 authors:



[Craig Marais](#)

Noroff Education AS

5 PUBLICATIONS 5 CITATIONS

[SEE PROFILE](#)



[Lynn Ann Futcher](#)

Nelson Mandela University

28 PUBLICATIONS 50 CITATIONS

[SEE PROFILE](#)



[Johan van Niekerk](#)

Nelson Mandela University

55 PUBLICATIONS 534 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



10th World Conference on Information Security Education - WISE10, 29-31 May, 2017, Rome (Italy) [View project](#)



BTech Project [View project](#)

# Incorporating a Game Design Document into Game Development Projects Deliverables

Craig Marais

Institute of ICT Advancement  
Nelson Mandela Metropolitan  
University

Port Elizabeth, South Africa  
+27 79 498 1898

s204005264@live.nmmu.ac.za

Lynn Futcher

Institute of ICT Advancement  
Nelson Mandela Metropolitan  
University

Port Elizabeth, South Africa  
+27 83 688 4002

Lynn.Futcher@nmmu.ac.za

Johan van Niekerk

Institute of ICT Advancement  
Nelson Mandela Metropolitan  
University

Port Elizabeth, South Africa  
+27 82 567 6523

JohanVN@nmmu.ac.za

## ABSTRACT

The computer gaming industry continues to show substantial growth. For this reason, computer game development is a viable career choice for Information Technology (IT) graduates specializing in software development. The challenge for institutions offering software development qualifications, however, is to prepare students for such a career. The School of Information and Communication Technology (ICT) at the Nelson Mandela Metropolitan University (NMMU) has been permitting students to develop computer games as part of their capstone experience for several years. However, game development projects do not fit the deliverables of the standard software development projects. The game development deliverables have therefore been reviewed and changed over the years using an 'action research like' approach. This paper reflects on this approach by presenting the lessons learned through the supervision of game development projects at the NMMU's School of ICT. The latest enhancement is the introduction of a contract in the form of a game design document (GDD) for student game development projects. The structure of this contract, together with other identified issues, is described in detail within this paper.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.9 [Software Engineering]: Management *Life cycle, Productivity, Programming teams, Process Models*

## General Terms

Management, Documentation, Experimentation

## Keywords

Capstone projects, Software development, Game development, Game Design

## 1. INTRODUCTION

Game development projects have become an acceptable outcome of the software development capstone experience. However, stakeholders involved in the supervision of such projects need to be prepared for the challenges specific to such gaming projects and not necessarily covered by "normal" software development

approaches. At the NMMU's School of ICT, this has been facilitated by the creation of a set of alternative project deliverables for the gaming projects. The driving force behind this division has been the growth of the computer gaming industry, specifically the 'Indie Paradigm'. However, this paradigm will not be the primary focus of this paper; instead it will seek to identify elements lacking from the existing gaming project requirements (deliverables) and recommend alternatives in this regard.

This paper forms part of an on-going 'action research like approach', the aim of which is to improve the overall standard of third year gaming projects. This approach has been followed for several years at the NMMU's School of ICT and was first reported on in a previous paper by Van Niekerk & Futcher (2011). In the previous paper, Van Niekerk & Futcher were able to successfully identify and address some of the issues facing third year gaming projects; these will be reflected on in Section 4. The approach adopted for this study is not once-off but on-going, requiring further reflection in order to evaluate success or failure. This reflection will facilitate further changes being made, thus iteratively improving the standard of the third year gaming projects.

The foundation of the third year projects (gaming and otherwise) is the ACM/IEEE 2008 guidelines for IT curricula, specifically the requirement for the students to undergo a 'capstone' experience. This experience should be designed to help students to integrate cross-disciplinary knowledge in preparation for responsibilities as a professional. In the case of gaming, the projects thus have to prepare students for a career as a game developer. Currently, South Africa does not have a well-developed (diverse) computer gaming industry, forcing many prospective game developers who seek employment in a formal capacity, to seek such employment abroad. The best alternative for these individuals is to pursue independent, entrepreneurial, game development as evident within the 'Indie Paradigm'. Many of the recommended gaming project deliverables are thus recommended based on their relevance to this 'Indie' approach to the industry.

This paper presents a case study of the on-going 'action research like' approach adopted by the supervisors of third year projects, specifically in the area of gaming projects. It describes the context of game development at the NMMU's School of ICT and presents a further cycle of improvements made to such projects, subsequent to the issues and resolutions presented by Van Niekerk & Futcher (2011).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

01

Ellen Haas

02

Ellen Haas

## 2. METHODOLOGY

Game development has become a viable career option for South African software developers. For this reason, students at the NMMU's School of ICT are actively supported and encouraged should they choose to pursue this option in their capstone project. However, the traditional business project deliverables, as proposed by Van Niekerk and Futcher (2011), do not fit gaming projects. An 'action research like' approach has therefore been used to address this dilemma. This is evident in the iterative, reflective nature of the research carried out on a specific case study, namely that of the third year software development projects of the NMMU's School of ICT. The need for this on-going research is supported by the increase in the number of students choosing to pursue game development together with the changing trends in the gaming industry.

The 'action research like' approach adopted follows the cyclic nature of the typical action research process as described by Kemmis (1990). According to this model, each cycle has four stages namely: plan, act, observe and reflect. This paper is a progression from Van Niekerk and Futcher (2011) and therefore addresses only the latest cycle in the research process.

This paper follows the structure of a case study as set out by Creswell (2007) namely:

- Entry Vignette
- Introduction
- Description of case and it's context
- Development of issues
- Details about selected issues
- Assertions
- Closing Vignette

The Introduction to this paper (Section 1) serves as the entry vignette and introduction, while the sections that follow match the case study structure as recommended by Creswell (2007).

## 3. CASE & CONTEXT

A traditional software development lifecycle does not create a suitable capstone experience for gaming projects (Van Niekerk & Futcher, 2011). The standard approach adopted by the NMMU's School of ICT for third year projects is that of an agile approach. The approach is adjusted to meet the requirements of grading such a project. These adjustments are in the form of specific requirements at predetermined times; this is done in the form of four distinct iterations throughout the course of the academic year. These iterations are currently mapped to similar iterations for gaming projects but using more gaming appropriate nomenclature. Table 1 lists the four 'agile' cycles as currently used for both software development and gaming projects.

Table 1: Lifecycle Comparison

Software Development Project	Gaming Project
Iteration 0	Pre-Alpha
Iteration 1	Alpha
Iteration 2	Beta
Iteration 3	Release

Iteration 0, or in the case of gaming the Pre-Alpha phase, requires students to draw up a 'contract', or for gaming projects a Game Design Document (GDD), which is then signed by all stakeholders. The stakeholders involved are typically four distinct groups, namely: the students, the user(s), project co-ordinator and the supervisor. This structure does map to gaming projects, as the user is replaced by the intended target audience, the players. Unlike users in software development projects, players rarely participate actively during the development phase of a game. Whereas the project coordinator is responsible for defining the project deliverables and deadlines, the supervisor is responsible for monitoring the progress towards these deliverables. This ensures that project submissions are of an acceptable standard.

In order to ensure the quality of project submissions, the supervisor must be familiar with both game design and the technologies typically used in the gaming industry. Additionally, the supervisor must be an individual who is familiar with game programming. Game design is the field of study encompassing all aspects of creating the mechanics and rules of the game to be played. These skills are not taught in a typical IT programing curriculum. The supervisor must therefore also have the necessary programming background to deal with the non-traditional challenges encountered by gaming projects.

In the past XNA, has been the favored tool for game development at the NMMU's School of ICT, being a natural middle ground between the industry standard DirectX and the preferred programming language of tuition, C#.

Through the years, the NMMU's School of ICT has seen the success of many gaming projects, but the ever shifting standards of industry and the drive to improve the quality of IT graduates has encouraged us to make changes to meet existing trends in the gaming industry.

## 4. EXISTING ISSUES

In their previous paper, Van Niekerk & Futcher (2011) identified a number of key issues to be address when supervising game development projects, three of their issues remain relevant to this paper. The others relate to placeholder graphics and artists.

The following issues have remained at the core of the gaming projects and will continue to do so for the foreseeable future:

- Differences between gaming and business project delivers should be clearly defined.
- A specialized interest group and/or dedicated student assistant is essential.
- Committed academic support is crucial.

There have been many other problems encountered with gaming projects through the years. It has only been through careful supervision that these projects have been able to succeed as they have. This supervision has served as the cornerstone of the agile software development lifecycle that is used at the NMMU's School of ICT.

### 4.1. Game Design


The agile development process is iterative. This means that requirements and solutions are defined as the project progresses; this is not suitable for grading as there is no way to compare two different gaming projects. In the past, this has been addressed by defining specific requirements for each iteration. This requires detailed involvement from the supervisor in order to create realistic requirements. In the case of gaming projects, requirements are derived through iterative play testing. In order to

achieve this, a certain level of flexibility needs to be catered for in the required documentation. However, all stakeholders involved need to agree on the scope of the final project.

## 4.2. User Requirements

The involvement of the supervisor is further emphasized by the lack of user, as the game projects are defined by genre rather than user requirements. The supervisor either needs to be heavily involved in drawing up requirements or the gaming group needs to act in independence from a user.

## 4.3. Development Platform

The user level help from the supervisor is extended to the other extreme too; the supervisor also needs to act at a 'design' level.  supervisor needs to understand the types of mechanics typically associated with the genre of game being developed. As mentioned before, these skills are not 'typical' for a supervisor to possess. It should be evident that a large amount of emphasis is placed on the supervisor of these projects to have experience with game development. One method of overcoming this is for students to rely on online forums for development help. For this reason XNA has been the platform of choice.

Unfortunately, Microsoft has recently indicated that XNA will not be developed further and will be 'discontinued' in 2014 (Roy 2012). This creates a troubling situation as an alternate platform must be found to encourage the passing of knowledge between existing students with future ones. This re-iterates that "committed academic support is crucial" (Van Niekerk & Fletcher, 2011).

## 4.4. Software Practices

The collaborative nature of students' projects requires that all stakeholders need access to a variety of resources at arbitrary times. This has in the past caused issues; with one team member working on outdated source code and rendering other team members parallel work moot. In the past this has been left for the students to deal with as they see fit, often causing strife within the group.

Summing up the identified problems, the following four key points are reached:

1. A flexible contract must be drawn up (unique per project)
2. Developers need to 'emulate' a user.
3. A suitable development platform must be identified
4. Sound software development practices must be enforced

Despite these problems, solutions have been found for these problems and gaming projects continue to be successful at NMMU's School of ICT.

## 5. DETAILS ABOUT SELECTED ISSUES

Through careful supervision and directed changes in deliverables, the gaming projects have remained relevant to industry and academically viable for grading. The agile-like approach will be continued into 2013, with the current five gaming groups comprising just less than two dozen students.

The requirements for gaming projects have been formulated using the following guidelines:

- Iteration 0 - Pre-Alpha

This iteration focuses on initial preparation and will focus on a detailed Game Design Document. Unique selling point identified.

- Iteration 1 - Alpha

The game must be playable by this point and the core mechanics should be implemented.

- Iteration 2 - Beta

Fully playable and featuring almost all intended features.

- Iteration 3 - Release

Completed game, ready for distribution. Marketing video with synopsis document.

## 5.1. Game Design Document

The initial emphasis for gaming projects is the drawing of a contract in the form of the Game Design Document (GDD); this contract will determine specifications for the project as a whole.

This will not be the only document required, but rather a central document, with all other documents being designated on an 'as needed' basis. A sample layout for this document is detailed in Table 2. The overall layout of this document has been distilled from the observation of several such documents which have been released by developers onto the Internet. Examples of such released documents include Grim Fandango, Leisure Suit Larry and Grand Theft Auto, additionally many game design textbooks include sample documents (Rodgers, 2010 and Rouse, 2005). Overall this document should be between 10 and 20 pages and should leave technical problems for supplemental documents.

Table 2 describes the layout for the GDD as used at NMMU. This layout has been structured to help the students meet the weekly deliverables, while maintaining relevance to details needed for game development.

 Table 2: Game Design Document Template

Section	Subsection	Description
Introduction	Document Scope	A brief description of the purpose of the document
	Authors	Biographical & Contact Information
Revisions	Game Design Document	Details of major document changes
	Additional Documents	References to separate technical documents
Development Specifications	Team Roles	Roles & Responsibilities
	Iteration Objectives	Project milestones
Gameplay Basics	Background Story & Concept	Verbose game description with story
	Player Roles & Objectives	Describe the players intended purpose
Technical Specifications	Target System	Hardware and software requirements
	Controls	Intended control scheme
Gameplay Specifications	Development System	Motivation and selection of tools to be used
	Genre	Describe the game style and theme (with examples of such)

World Environments	A detailed listing of all intended game worlds/levels
Character & Controls	Describe how the player will achieve his goals
Interface	Detail what will be visible to the player.
Gameplay Mechanics & Features	Detailed listing of all features and mechanics
Gameplay Mechanics & Features	Gameplay descriptions & AI review

### 1.1.1. Introduction

The introduction to the GDD identifies the purpose of the document, what it aims to achieve. Students are encouraged to motivate their choice of project at this point as well as addressing what they personally aim to learn from the project.

The authors are required to provide all applicable biographical and contact details at this point in the document.

### 1.1.2. Revisions

The nature of the GDD is that of a formalized contract, that should not be heavily edited through the year. Students are required to document in detail any deviation from this delineation.

### 1.1.3. Development Specifications

The student teams are required to describe the roles of each team member, this is intended to help the team identify possible skills that they lack.

The student team is also required to create a roadmap for their progress throughout the coming academic year, this plan is required to address what features will be present at each stage of development (iteration) of their game.

### 1.1.4. Gameplay Basics

The gameplay basics portion of the GDD address the *what* and *why* of the game.

The student team is required to provide the overarching story of the intended game. The players role within the story needs to be clearly delineated.

### 1.1.5. Technical Specifications

The student teams need to select a target platform for their game; this choice will also effect both the available control scheme (mouse, keyboard, gamepad, etc.), and the hardware requirements that the game must support.

The choices made in each regard must be motivated by the student team.

### 1.1.6. Gameplay Specifications

The Gameplay Specifications section (and subsections) feature all miscellaneous details of the intended game.

## 5.2. Development Blog

Parallel to this document will be an on-going development blog, in which the groups will be required to post fortnightly updates. These blogs will aid supervisors in evaluating groups over time, serving a similar role to the required time-sheets and minutes of meetings. Students are encouraged to use the blogs to voice their concerns and also to 'show-off' their achievements.

For 2013, we will be encouraging students to host these blogs publicly so that their progress is visible to any Internet user, this

will help the students to get user feedback from what they post. While having no designated user, they are able to gather public opinion of what they are doing. This will be to their advantage in the long run as they are able to easily extend such a blog to gather public funding should they wish to formally market their project after the year is done. This method of development has been used by many 'Indie' developers over the years; examples include Minecraft, Dwarf Fortress, Project Zomboid and many more.

These two forms of documentation will drastically reduce the load on the supervisor, the final hurdle being that of technical support. As stated before XNA has long been the defacto development tool at NMMU, unfortunately this may have to change in the near future.

## 5.3. Development Platform

Gaming development students have always been expected to learn addition skills in order to code gaming projects, however there has always been a transfer of skills between the years, allowing past students to assist those in the year below them. Future students may find themselves 'stranded' in this regard, thus it is crucial for academic support in this regard. Rather than force a single development platform, supervisors should familiarize themselves with the advantages and disadvantages of a variety of platforms. The supervisors should 'pitch' the different tools to the students and encourage self-education. Gaming students have long been expected to self-educate in this regard and it is a crucial skill for any industry programmer.

## 5.4. Version Control

The 'test bed' status of gaming groups is further extended to other software development tools, since 2012 gaming projects have also been encouraged to make use of an internal server running Subversion. Subversion is an open-source version control platform. Using this tool, students are able to collaborate on the same project even when they are not at the same geographical location. The students are able to connect to their own repository on the server via group specific credentials; likewise, the supervisor has global access to all repositories. The supervisor is able to retrieve the latest project version and monitor progress at will. The use of such version control software has completely eliminated incidents where students claimed that they lost their work due to an "update error". Such incidents were common prior to the use of Subversion.

The students are strongly encouraged to use either the available Subversion server or similar tools such as GitHub, Dropbox or Google Drive.

## 6. ASSERTIONS

The biggest change implemented over the last two years has been the more formalized inclusion of the GDD; while not a novel concept, the GDD as designed for the gaming projects should create a common vision between all project stakeholders. It is hoped that the GDD will create a solid basis for developers to work from and facilitate evaluation session criteria. This leads us to the first assertion:

1. A document like the GDD must form the basis of a game development project since this plays the role of a flexible contract.

The addition of a "Unique Selling Point" will help to enhance the variety of games that students produce, and will allow to similar themed games to be easily differentiated. The "Unique Selling Point" will also help students to market their finished games in a commercial sense. Thus, we can assert that:



2. Every game project must include one “Unique Selling Point” to differentiate it from other games.

The inclusion of development blogs will create transparency into group decisions and work ethic, further helping project invigilators to make informed decisions, and serve as focus for the identification of user requirements.

3. Each group should maintain a regular development blog. Responsibility for editing this blog should be shared.

The freedom of development platform may in future create difficulties for the supervising staff, but should ultimately help students to produce higher quality outputs.

4. Development platforms must remain current, and follow industry trends.

Finally, version control software is an industry standard that must be emphasized to the students. The students need to be familiar with the procedures and practices relating to good version control.

5. Students must make use of appropriate version control software; absolutely no excuse for lost updates should be tolerated.

It is expected that there will be further difficulties associated with the game development projects, but it is the authors’ hopes that by taking tried and tested methods from the game development industry at large, that the students will have a more meaningful capstone experience.

## 7. CLOSING STATEMENT

As the industry develops and new methods are adopted; so too must the education of future game developers change to accommodate these new trends. With the emergence of the ‘Indie Paradigm’ changes have been made to the gaming project deliverables in order to align the students’ experience with industry trends.

At NMMU’s School of ICT, the inclusion of a GDD in the requirements for game development projects’ deliverables has forced students to keep the relevance of their deliverables in line with industry trends. The GDD remains relevant to the needs of industry and thus enhances the capstone experience offered to students.

In following with the action research-like approach used by this paper, the study will proceed into a phase of observation. The outcomes of the assertions described and future changes in deliverables will be reported on in future publications.

## 8. REFERENCES

1. ACM, Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Retrieved 1 March 2013 from site: <http://www.acm.org/education/curricula/IT2008%20Curriculum.pdf>
2. Adams, T. and Adams, Z. Dwarf Fortress, Bay 12 Games, Retried 1 March from site: <http://www.bay12games.com/dwarves>
3. Ambler, S.W. The Object Primer, Cambridge University Press, New York, 3rd Edition, 2004
4. Apache Subversion, Apache Software Foundation, Retrieved 1 March 2013 from site: <http://subversion.apache.org/>
5. Cresswell, J.W. Qualitative Inquiry and Research Design: Choosing among Five Approaches. SAGE Publications, 2nd Edition, 2007
6. Daily, M. GTA, DMA Design, Retrieved 1 March 2013 from site: <http://www.flickr.com/photos/mikedailly/sets/72157602022230830/with/5548285262/>
7. GitHub, GitHub Inc, Retrieved 1 March 2013 from site: <https://github.com/>
8. Google Drive, Google, Retrieved 1 March 2013 from site: <https://drive.google.com/>
9. Kemmis. S. The action research reader. Victoria:Deakin University, 1990
10. Lowe, A. Game Designs, Retrieved 1 March 2013 from site: <http://www.allowe.com/gamesdesign/>
11. Mc Whertor, M. Tim Schafer Publishes Original Grim Fandango Design Doc, Kotaku, Retrieved 1 March from site: <http://kotaku.com/5077780/tim-schafer-publishes-original-grim-fandango-design-doc>
12. Minecraft, Mojang, Retrieved 1 March 2013 from site: <http://minecraft.net/>
13. Project Zomboid, Indie Stone, Retrieved 1 March 2013 from site: <http://projectzomboid.com/blog/>
14. Rodgers, S. Level Up: The guide to great video game design, John Wiley & Sons, Chichester, 2010
15. Rouse, R. Game Design: Theory & Practice, Wordware Publishing Inc., Sudbury, 2nd Edition, 2005
16. Roy, P. DirectX/XNA Phase Out Continues. Retried 1 March 2013 from site: <http://ventspace.wordpress.com/2013/01/30/directxxna-phase-out-continues/>
17. Van Niekerk, J. & Fletcher, L. Developing Criteria for the Supervision of Computer Game Development Projects, SACLA Conference, July 2011, Durban, South Africa



# Incorporating a Game Design Document into Game Development Projects Deliverables

Marais, Craig; Elizabeth, Port; Africa, South; Futchner, Lynn; Elizabeth, Port; Africa, South; Niekerk, Johan Van; Elizabeth, Port; Africa, South

- |                |            |        |
|----------------|------------|--------|
| 01             | Ellen Haas | Page 2 |
| 7/3/2019 20:04 |            |        |
| 02             | Ellen Haas | Page 2 |
| 7/3/2019 20:04 |            |        |
| 03             | Ellen Haas | Page 3 |
| 7/3/2019 20:03 |            |        |
| 04             | Ellen Haas | Page 3 |
| 7/3/2019 20:04 |            |        |
| 05             | Ellen Haas | Page 3 |
| 7/3/2019 20:04 |            |        |
| 06             | Ellen Haas | Page 4 |
| 7/3/2019 20:03 |            |        |
| 07             | Ellen Haas | Page 4 |
| 7/3/2019 20:04 |            |        |
| 08             | Ellen Haas | Page 4 |
| 7/3/2019 20:03 |            |        |
| 09             | Ellen Haas | Page 5 |
| 7/3/2019 20:03 |            |        |



10

Ellen Haas

Page 5

7/3/2019 20:03

11

Ellen Haas

Page 5

7/3/2019 20:04

12

Ellen Haas

Page 5

7/3/2019 20:04

13

Ellen Haas

Page 5

7/3/2019 20:04