

Developing a Reference Method for Game Production by Method Comparison

Inge van de Weerd, Stefan de Weerd and Sjaak Brinkkemper
Department of Information and Computing Sciences, Utrecht University,
PO Box 80.089, 3508TB Utrecht, The Netherlands
{i.vandeweerd, sweerd, s.brinkkemper}@cs.uu.nl
WWW home page: <http://www.cs.uu.nl/>

Abstract. In this research, we use a formal method comparison approach to construct a reference method for game production. First, we analyze four game production methods by using a meta-modelling technique: three documented methods and one method obtained via a case study at a game production company. By developing a super method, containing all activities and concepts of four analyzed methods, we compare the four methods. Based on the super method, a reference method is constructed to give a complete overview of all possible steps and deliverables in a game production process: the reference method for game production.

1 Method Comparison

Several motives exist for evaluating and comparing methods. In literature many reasons are listed; see for example [1] and [2]. We can order these motives according to the point of view from the actor that is involved. From the method user's point of view, method comparison can aid in selecting the best method for a particular situation. A tangle of methods exists in the IS development world. To know which method is best in a certain situation, one has to know the strengths and weaknesses of candidate methods. Furthermore, method users might want to use a tool to support their method. Method comparison makes it easier to select the right tool.

From the developer or researcher's point of view, comparing methods leads to a better understanding of methods and their rationale. Also, existing methods can be improved and new situational methods can be assembled. Ultimately, it allows the researcher to develop a reference method, which can be used to identify the similarities and differences between the various methods in a systematic way.

Several empirical and non-empirical approaches exist for method evaluation [1]. Empirical approaches for method evaluation are often time-costly. A laboratory setting, for example, is almost unfeasible. Case studies take a lot of time to get

Please use the following format when citing this chapter:

van de Weerd, I., de Weerd, S., Brinkkemper, S., 2007, in IFIP International Federation for Information Processing, Volume 244, Situational Method Engineering: Fundamentals and Experiences, eds. Ralytė, J., Brinkkemper, S., Henderson-Sellers B., (Boston Springer), pp. 313-327.

enough case studies for a reliable result. Another drawback from some empirical approaches is that the evaluation of the method can be highly influenced by the performers and their experience, and the domain or project in which the method is used. Non-empirical methods are in general less time-costly.

In this paper we use a qualitative and formal approach to develop a reference method for game production; namely a comparison approach based on conceptual differentiation of meta-models, as described in [3]. In this comparison, we compare three documented methods for game production. A fourth method is obtained via a case study at a game developer company. We develop a super method, based on the four analyzed methods, which we use for the method comparison. Finally, we construct a reference method that can be used to a) give an overview of the steps and deliverables in a game production process, b) develop a uniform terminology field within the game production domain, c) serve as input for a public knowledge infrastructure on development methods, and d) give recommendation to the game production company that was researched in the case study.

The remainder of this paper presents our approach to the development of the reference method. Related work in the method engineering and method comparison domain is described in Section 2. Then, in Section 3, we describe our approach. We present the resulting reference method for game production in Section 4, and discuss the results in Section 5. Finally, in Section 6, we present our conclusions and future research.

2 Related Work

Siau and Rossi [1] give an extensive overview of empirical and non-empirical method evaluation techniques. They distinguish the following empirical techniques: surveys, laboratory experiments, field experiments, case studies and action research. The non-empirical methods are: feature comparison, meta-modeling, metrics approach, paradigmatic analyses, contingency identification, ontological evaluation and approaches based on cognitive psychology. They state that none of these techniques is inherently superior to others, but that the choice to use a certain technique should be based on the research questions, the environment, the strengths of the researchers, and the opportunities available. Also Fettke and Loos [2] compare the different approaches on evaluation. They propose a framework for the multi-perspective evaluation of frameworks, in which the same perspectives as Siau and Rossi are used. However, the framework is extended with an economic-based evaluation, a master reference model-based evaluation, and a plain text-based evaluation.

In [3], a formal approach to the comparison of six object-oriented analysis and design methodologies is presented. From all six methods a meta-process model and a meta-data model is created in order to obtain a uniform and formal representation of the methods. The meta-models are then used to compare the analysis and design steps, the concepts, and the techniques provided in each method. The result is a set of tables that reveal the similarities and differences between the methods.

Several method comparison frameworks have been developed; see for example the ACRE framework for selecting the right requirements acquisition method [4], the Method Characteristics Framework for evaluating information engineering methods [5], and the Cataloging Framework for software development methods [6]. All these frameworks use a number of features or properties which are used to characterize the methods.

3 Method Comparison: A More Formal Approach

In this section we describe the approach that we followed to come to a systematic method comparison. The approach we use was first applied in [3] for the comparison of Object-Oriented methods as described in Section 2.

3.1 Approach

Based on the formal approach for method comparison, proposed by Hong, van den Goor and Brinkkemper [3], we use the following steps to come to a complete method comparison:

1. *Method selection*

In this research we compare four methods. Three of these methods are documented in game production literature, namely Game Development and Production [7], Introduction to Game Development [8], and The Game Production Handbook [9]. The reason for this choice lies in the fact that all three books are written from a management perspective, rather than a technical development perspective. All three methods have received good reviews and they complement each other in the topics that are covered. The fourth method is proprietary method used at a game production company.

2. *Method modeling*

For the analysis of methods, we use process-deliverable diagrams (PDDs), a meta-modeling technique that is based on UML activity diagrams and UML class diagrams. This meta-modeling technique is clear, compact and consistent with UML standards. The resulting PDDs models the processes on the left-hand side and deliverables on the right-hand side, see for examples figure 1 and 2. Details on this modeling technique can be found in [10] and [11]. The process and deliverables are explained by accompanying activity and concept tables, in which all activities and deliverables are described.

3. *Development of super method*

The four methods, modeled in PDDs, are decomposed in activities and concepts. From both activities and concepts a comparison table is created that lists all activities and concepts of all four methods, using a similar approach as is described in [3].

4. *Comparison of methods*

The method comparison is performed by filling in the fields in the comparison tables with comparison symbols: an '=' symbol to indicate that the concept or activities are the same; the '<' and '>' symbols to indicate whether an activity in

the super method comprises more or less than the activity in the concerning method; and the '><' symbol to indicate that the activity in the supermodel partly overlaps the activity of the process model. In case a field in the comparison table is left blank it means that the activity or concept is not present in the concerning method.

The comparison is as formal as can be at the moment at a large scale. Even more formal would be that all concepts tabulated in step 2 are formally described using an ontological language like [12]. The activities of step 2 can be sequentially formalized as manipulations (create, modify, delete) of concepts. Then, the comparison of concepts and activities can be executed at the *most* formal level. However, it is debatable whether this most formal comparison would really provide valid results, as the field of game production is young and dynamic, which makes the formal comparison outdated the moment it is presented.

3.2 Meta-models of documented methods

In this section, an overview is given on the game development methods derived from the theories in the three books. Altogether, the analysis of game development methods resulted in 13 PDDs.

Chandler [9] describes in 'The Game Production Handbook' four main processes in the game production cycle. These are Pre- Production, Production, Testing and Post-Production. The pre-production phase encompasses the definition of the game concept, the definition of the game requirements and the definition of the game plan. After pre-production there is production, in which builds of the game are created and localized. In the successive testing phase quality assurance tests are performed and the final game code is released, after which in the post-production phase post mortems are conducted and closing kits created. In Figure 1, the PDD of the Game concept definition is depicted to give an example of a PDD.

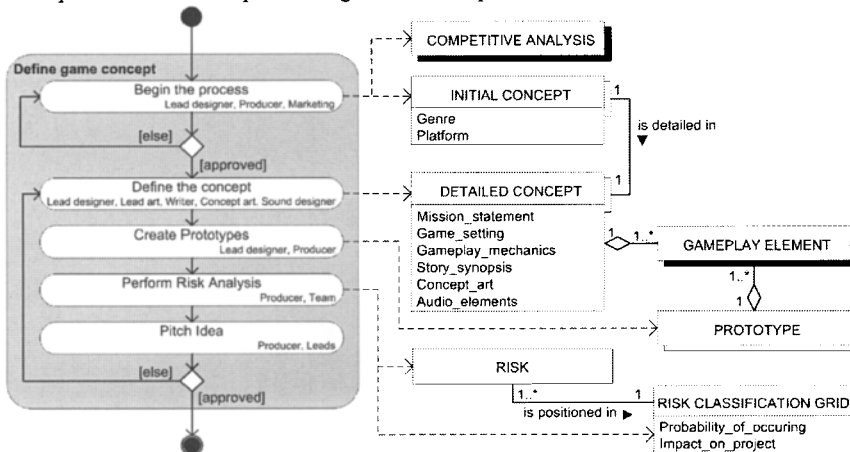


Fig. 1. Process-deliverable diagram of the Concept Phase in "The Game Production Handbook" [9]

The main activity *Define game concept* consists of five sub activities that all result in one or more deliverables and/or a decision point. In Tables 1 and 2, we describe the activities and the deliverables respectively.

Table 1. Activity table for Define game concept

Activity	Sub-Activity	Description
Define Game Concept	Begin the Process	At the start of the process, the Lead designer, Producer and Marketing manager develop an INITIAL CONCEPT for the new game. In the INITIAL CONCEPT the genre and platform on which the game is supposed to run are described. They also perform a COMPETITIVE ANALYSIS.
	Define the Concept	When the INITIAL CONCEPT is approved, the Lead designer, Lead art, Writer, Concept art, and Sound designer define the concept in a DETAILED CONCEPT.
	Create Prototype	The Lead designer and Producer create a PROTOTYPE, based on the DETAILED CONCEPT.
	Perform Risk Analysis	The Producer develops together with the rest of the Team a RISK CLASSIFICATION GRID, in which all RISKS are plotted.
	Pitch Idea	The Producer and Lead pitch the idea to the management. When it is approved, they can carry on with defining the game requirements.

Table 2. Concept table for Define game concept

Concept	Definition
INITIAL CONCEPT	A not detailed concept of the game that needs to present a compelling goal for the game to achieve.
DETAILED CONCEPT	A definition of the concept of a game that specifies the game mechanics, setting, characters, storyline, and major features.
COMPETITIVE ANALYSIS	An identification of the strengths and weaknesses of your game’s competition, market opportunities for your game, and any threats that might impact the game’s success in the market.
GAMEPLAY ELEMENT	Elements that are defined in the detailed concept, like mission statement, game setting, gameplay mechanics, story synopsis, concept art, and audio elements.
PROTOTYPE	An original type, form, or instance serving as a basis or standard of the full game for later stages.
RISK CLASSIFICATION GRID	The result of performing risk analysis, where risks have been identified, analyzed and classified on probability of occurring and impact on the project.
RISK	The possibility of suffering harm or loss.

The second method, described in ‘Game Development and Production’ (Eric Bethke [7]) recognizes four main steps in game development project life cycles; Business Context, Game Design, Game Implementation and Post Release Support.

Finally, in ‘Introduction to Game Development’, edited by Steve Rabin [8], the game development process is divided in five phases; the Concept Phase, Preproduction Phase, Production Phase, Postproduction Phase and After-Market Phase.

In Table 3 we give an overview of the amount of activities, sub activities and concepts per method. The method derived from Game Development and Production

[7] is referred to as GD&P; Introduction to Game Development [8] is abbreviated to ITGD; and The Game Production Handbook [9] is referred to as TGPH.

Table 3. Method statistics

	GD&P	ITGD	TGPH
Activities	6	10	9
Sub activities	23	37	36
Concepts	36	42	49

3.3 Processes of the Zylom case study

We carried out a case study at a developer and publisher of casual games: Zylom, which is part of RealNetworks Inc. Zylom develops retro arcade games, new games inspired by retro games, card and board games, puzzle games, and the like. These games are often referred to as “casual games” because it is possible for the casual consumer to pick them up and learn to play quickly [13]. The Zylom Media Group was established in 2001 in the city of Eindhoven, the Netherlands, and still has its headquarters there nowadays. The company currently employs over 60 people.

3.3.1 Case Study Design

The case study was carried out in a period of three weeks. Resulting from the case study, activities and deliverables are identified, which are used to compare to literature on game development. The research is done by means of an exploratory-explanatory case study. The case study concentrates on the game development process, from the initial idea for a game until the final release of it. The primary goal of this case study is to obtain an overview of the game development processes at Zylom at present. The procedure in the case study is as follows:

- a. Perform explorative interviews
- b. Analyze documentation
- c. Perform feedback interviews to affirm and explain results

The explorative interviews were conducted with seven employees: a game designer, three game developers, an employee of the localization department, an employee responsible for Q&A and support and a member of the Management. Each employee answered the following questions

- What are the chronological steps taken in your specific part of the game development process?
- What are typical activities that are performed in your department?
- What are the dependencies between these activities?
- What deliverables are created and/or used by your department?

Complementary documentation was provided by access to the internet and via email.

After modeling the activities and deliverables in PDDs, a second interview session was carried out to check the results. Eight employees were interviewed, of which 4 were the same employees as in the first interview round.

3.3.2 Results

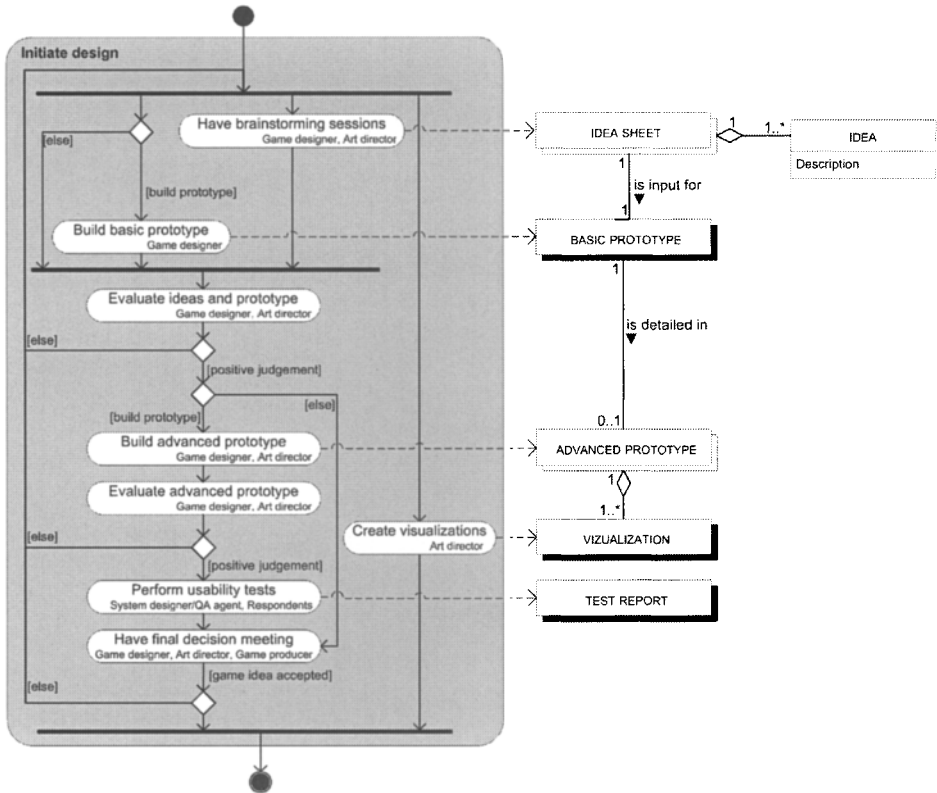


Fig. 2. Process-deliverable diagram ‘Initiate design’

The game production process as it is carried out at Zylom is divided into four phases, namely Design, Development, Quality assurance and Localization. The method process consists of 11 activities, 59 activities, and 60 concepts.

In Figure 2, the PDD that illustrates the activity of the Design activity, namely ‘Initiate design’. We identify one main activity, ‘Initiate design’ and eight sub activities. Each sub activity results in a deliverable (e.g. BASIC PROTOTYPE), or proceeds to decision point (e.g. ‘Evaluate ideas and prototype’). In each sub activity, roles are added, which describe the actor(s) that carries out the activity. Roles in the ‘Initiate design’ activity are: Game designer, Art director, System designer, QA agent, Respondent, and Game producer.

3.3.3 Validity Issues

To make sure that the gathered information about Zylom and the game development theories is valid, the research design applies to the case study tactics defined by Yin [14]. These case study tactics encompass various facets that underwrite the validity of four research design tests; construct validity, internal validity, external validity and reliability. By making sure that the collected data from the research met the case study tactics, the validity of the scientific research is vouched for.

The case study that was performed at Zylom can be classified as a single-case study design, since there is no more than one source for the data collected from the case study. Moreover the case study design is a single-case embedded design as multiple units of analysis, in this case departments of the company, are included. Various sources of evidence were used to collect the data for the research. Information was gathered from documentation, archival records, interviews and – to some lesser extent – observations.

Because the character of this case study is exploratory, the internal validity of the design is irrelevant [14]. In order to vouch for construct validity, external validity and reliability, the case study tactics were applied. This implies that multiple sources of evidence were used and a chain of evidence was established during data collection. Having key informants review draft case study reports increased the validity of the composition of the data. External validity is difficult to obtain in a single-case study. Yin [14] claims that external validity could be achieved from theoretical relationships, and from these generalizations could be made. However, due to the type of game production company Zylom is, namely a casual game company, limitations exist to the extent we can generalize the research. Nevertheless, we do not believe this is a major issue, since the Zylom case study is only one of the four sources that is used for the method. We believe it is complementary to the other three methods. Finally, the reliability of the case study is obtained by using a formal case study protocol and developing a case study database.

3.4 Super method: tabulation and comparison

The second step in the comparison of methods is the tabulation of the analyzed methods, leading to a so-called super method. Two tables were created. For the activity table the procedure is as follows: In case a field in the comparison table is left blank it means that the activity on that particular row is not present in the process model of the corresponding column. When a field is not blank, these are the notations that describe the comparative relationship between two methods:

- An '=' symbol indicates that a similar activity to the one in the super method is available in the concerning method.
- The '<' and '>' symbols indicate whether the activity in the super method comprises more than the activity in the concerning method or less than the activity in the concerning method, respectively.
- The '><' symbol is used when a part of the activity in the supermodel overlaps a part of the activity of the process model, but other parts don't overlap.

For the comparison of concepts a similar approach is used; a super set of concepts is derived from the meta-deliverable models and forms the basis for the comparison of concepts. The notation is somewhat different than for the activities. A blank field is representing that a concept from the super method is not available in the concerning method. Other notations used in the concept comparison table are:

- The '=' symbol is still used to indicate that a concept in the super method is also included in the concerning method.

A string in a field indicates the same; however the naming for the concept is different than in the super method.

In Tables 4 and 5, we show excerpts of the resulting activity comparison and concept comparison table respectively. We use the same abbreviations as in Section 3.2 to refer to the different methods.

Table 4. Activity comparison table (excerpt)

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
<i>2.1 Create game design</i>				
2.1.1 Brainstorm	=			
2.1.2 Delegate design	=			
2.1.3 Write game design document	=	>		=
2.1.4 Evaluate game design document				=
2.1.5 Write technical design document	=	><		<
2.1.6 Create visualizations				=
2.1.7 Present game design				=
2.1.8 Evaluate technology			=	
2.1.9 Define tools and pipeline			=	
2.1.10 Create documentation			=	

Table 5. Concept comparison table (excerpt)

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
<i>2.1 Create game design</i>				
2.1.1 GAME DESIGN DOCUMENT	=	=	DESIGN DOC.	=
2.1.2 CORE GAMEPLAY	=			
2.1.3 CONTEXTUAL GAMEPLAY	=			
2.1.4 STORY	=			
2.1.5 TECHNICAL DESIGN DOCUMENT	=	=	TECHNICAL DOC.	=
2.1.6 REQUIREMENT	ASSET	FEATURE	FEATURE	ASSET
2.1.7 VISIBLE REQUIREMENT	=			
2.1.8 NONVISIBLE REQUIREMENT	=			
2.1.9 FEATURE LIST			=	ASSET LIST
2.1.10 DOCUMENTATION			=	
2.1.112 PROTOTYPE				=

The super method consists of four phases, in which we can identify 13 activities and 96 sub activities. These sub activities have the following distribution over the four comparison methods: GD&P: 26; ITGD: 37; TGPH: 37; and Zylom: 40. The activities result in a total of 117 concepts, which are distributed as follows: GD&P: 32; ITGD: 40; TPGH: 49; and Zylom 46. Please note that these statistics do not match the statistics described in Section 3.2. This is due to two reasons: a) only relevant activities and concepts are included in the comparison table, and b) some activities in the comparison methods are listed more than once, due to the fact that this activity covers more activities in the reference method.

One important observation from the comparison of activities of the different methods is that some methods cover areas or phases in the game production process

that are neglected by other methods. It can be concluded that GD&P is clearly written from a management perspective; most processes in the method focus either on the preparation, measuring or monitoring of tasks. The more basic tasks that directly relate to game development are mainly omitted. The focus in the theory is also clearly on deliverables that need to be completed in game production projects.

The ITGD method is the only one to address in detail the hiring of staff members as part of the game production process and performing marketing and sales related activities. Activities are not only management specific; also the actual game development process is covered in quite some detail. However, the localization of finished games is described very briefly and is therefore too basic to be really useful. Still, ITGD covers many important parts of the game production process and is therefore quite complete.

In TGPH some extra activities in the pre-production phase are suggested; the evaluation of technology, creation of documentation and definition of tools and pipelines to be used during the project are available in this method. Management related activities in this method are quite basic and high-level. The TGPH method excels when it comes to the localization process and the finalization of the game production project. The localization steps and the definition and creation of closing kits are thoroughly discussed. Finally, the part of the project in which code is released is also well explained in TGPH compared to other methods.

The Zylom method contains very detailed descriptions of the localization and quality assurance testing activities. Besides that, prototyping during several phases in a game production processes and usability testing using these prototypes are fragments that are only available in Zylom and valuable for the reference method.

3.5 Determination of reference method

The final step is the creation of the reference method. This reference method is an executable method that includes the best method fragments from the super method, based on the comparison of the four methods. In the next section we present the method.

4 Reference method

4.1 A Reference Method for Game Production

The resulting reference method for game production consists of thirteen main activities, expressed in four PDDs, which correspond to the four production phases. In Figure 4, we provide a high-level overview of the game production process, comprising four phases: Concept phase, Pre-production phase, Production phase and Post-production phase. The reference method consists of 13 main activities and 69 sub activities, which result in 93 deliverables. In the concept phase first the business parameters are identified, which comprises, among others, the budget, recourses and competitive analysis. A detailed game concept is defined and presented by means of

prototypes. Then, in the pre-production phase, the game design document is developed, which described the story, gameplay and requirements. Also, a project plan and staffing plan is created. This late moment for project planning and staffing has to do with the nature of the game production process. Rather than the straightforward development process from requirements to implementation that is used in the information systems domain, first the game concept needs to be approved. Developing this game concept is a creative activity, comparable with the production of movies. Someone has to approve or invest in the idea that is developed in this phase, before you can think about a project and staffing plan.

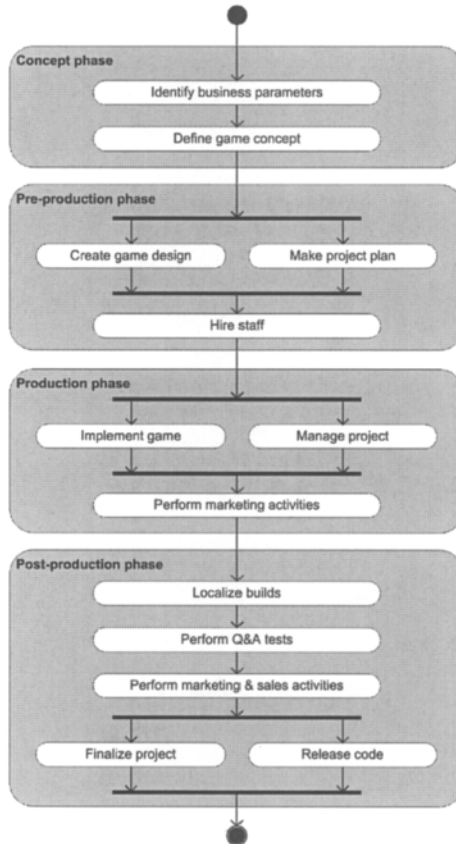


Fig. 3. Reference Method for Game Production: Process Overview

Next, in the production phase, the game is implemented in a working game; the final version. During the implementation scope changes are managed. At the end of this phase, a demo is developed for marketing purposes. Finally, the fourth phase comprises the typical post-production activities. The game needs to be localized so it can be released in different countries, QA tests are done and promotion material for the marketing department is created. Finally, the project concludes with developing a closing kit and releasing and shipping the game.

4.2 Define game concept

To elaborate the reference method a bit further, we illustrate one of the main activities, namely ‘Define game concept’ (Figure 4). When we compare this PDD with the PDD of Zylom’s first development activity, described in Section 3.3.2, we see that the first has more activities and deliverables. The process is more structured and there are less decision points. Furthermore, usability tests are included very early in the process. An important issue is the formal definition of a game concept, which is missing in the Zylom method.

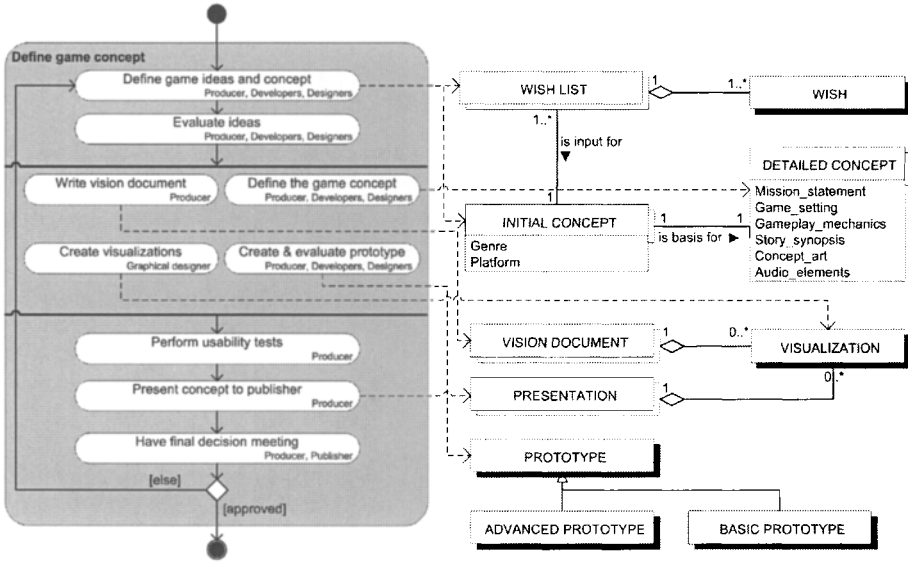


Fig. 4. Process-deliverable diagram of ‘Define game concept’

5 Discussion

5.1 Reference Method

Theories on game production are not unambiguous; there exist quite some differences in elements of the game production domain between the various sources. For example, the importance of localization of a game was stressed in some theories, while the localization process was completely omitted in others. Also, the importance of defining your business parameters before commencing with the design phase was not indicated in all methods. However, globally there are many similarities between the methods when it comes to the main game production process. All fragments from the methods have been included in the reference method, leaving the duplicates out. In case there were more than one fragments representing one and the same activity, the clearest and most comprehensible

fragments have been selected and included in the reference method. This has resulted in a complete reference method of the game production process that should quickly provide people with insight in what game development projects exactly encompass. Note that we do not want to prescribe this method as the best game production method, but rather as a complete reference of all activities and deliverables, showing how the method could be organized.

5.2 Public Knowledge Infrastructure

The result of this research, the reference method and the method comparison can be of great value for game companies. Various employees of Zylom that were interviewed indicated that task descriptions and templates would be useful to include in the project planning. Providing employees with proper, unambiguous descriptions of the tasks that they need to perform in a specific period of time can improve the process. In our vision this is realizable by developing a public knowledge infrastructure for game production methods, consisting of process descriptions, templates and best practices. Currently, we are working on such an infrastructure in the domain of product software [11].

5.3 Uniforming Terminology

During the analysis of the four game production methods, we encountered a wide range of terms that often are used for the same concepts. Especially in the comparison of the deliverables this was a problem, since it was difficult to detect whether two concept with different names, meant the same thing and vice versa. A de facto standardization of the terms would make the comparison of methods considerable easier.

5.4 Method Comparison

The comparison technique that is used in this research has some great advantages. The visibility and intelligibility that was already signaled in the comparison study of object-oriented analysis and design methodologies [3] also shows in this research. Moreover, the use of the activity and concept tables, improved the comparison method. However, the actual comparison of activities and concepts proved to be quite complex. When comparing activities, it was hard to identify whether two activities were totally similar, or if one activity encompassed more or less than another. For concepts it was often not easy to see whether or not two concepts with different names actually represent one and the same thing, due to the different terminologies that were used. Finally, after completing the comparison process, the resulting tables provide a good overview of the differences between the various activities and concepts.

5.5 Recommendations Zylom

A major observation from the processes at Zylom is that there are many decisions in a game development project whether or not to continue with the project. Especially in the designing phase of development projects these so called go – no go decisions occur often which might result in starting over the whole design process from scratch. In some cases it might prove useful not to reject a game idea as a whole, but refine or redefine a game idea by adjusting a prototype or brainstorm about other features or concepts.

In the current situation at Zylom the project plan is defined after the design phase. This means that design activities are not as properly planned as the rest of the game development activities. A suggestion is to create the project plan before the designing of a game takes place, in order to improve the managing and progress tracking of projects. It was indicated by various interviewees that game development projects are commonly significantly over time and thus that predefined targets in terms of numbers of games that need to be completed are not met.

5.6 Game Production versus Standard Information Systems Development

When it comes to the differences between managing the game production process and software development for other types of software, there are some differences that are identified. First of all, the concept phase in game production projects is quite different and more complex, because a game's atmosphere has to be defined and an initial concept version of a game is far less representative for the final game than in other software engineering projects. Especially the determination of the future 'look-and-feel' of a game and its atmosphere implies that the concept is likely to be changed more often during the concept phase than in other types of software products. The inventive and creative character of games also results in a design phase that is different from other software production projects. Because the atmosphere of a game needs to be defined, lots of art work and gameplay prototypes are created already during the design phase. Because creativity in game production plays an important role, and (elements of) games can be seen as kind of art, defining a basic concept with some wishes for requirements is not enough. Game production projects contain many decision moments during the design phase whether or not to cancel a project and start up with a complete new idea, while ideas for other types of software products are more easily defined in a product design and then developed.

6 Conclusions and Future Research

In this paper we presented an approach to the development of a reference framework for game production. We used a more formal approach to method comparison by using activities and concept tables. We analyzed three documented game production methods and carried out a case study at a game production company to analyze a fourth method. By developing meta-models with accompanying activity and concept

tables, we were able to carry out a method comparison and develop a reference method.

We can conclude that all methods are written from different perspectives, which causes a difference in the activities and concepts they contain. The reference method integrates these perspectives and can be used as a complete reference of how a method could be organized.

The use of activity and concept tables improves the method comparison approach as described in [3]. In future research, the method can be improved by making it more formal by using ontologies. This will also solve the problem of the ambiguous terminology that is used in the domain. Finally, we aim to develop a knowledge infrastructure for game production methods, containing process descriptions, templates and best practices.

References

1. K. Siau and M. Rossi, Evaluation of Information Modeling Methods - A Review, *Proceedings of the 31st Annual Hawaii International Conference on System Sciences* 5, (1998), p. 314.
2. P. Fettke and P. Loos, Multiperspective Evaluation of Reference Models - Towards a Framework, *LNCS* 2814, 80-91 (2003).
3. S. Hong, G. van den Goor, and S. Brinkkemper, A Formal Approach to the Comparison of Object-oriented Analysis and Design Methodologies, *Proceeding of the 26th Hawaii International Conference on System Sciences* 4, (1993), pp. 689-698.
4. N.A.M. Maiden and G. Rugg, ACRE: Selecting Methods for Requirements Acquisition, *Software Engineering Journal* 11(3), 183-192 (1996).
5. R.D. Hackathorn and J. Karimi, A Framework for Comparing Information Engineering Methods. *MIS Quarterly* 12(2), 203-220 (1988).
6. G.M. Karam, R.S. Casselman, A Cataloging Framework for Software Development Methods, *Computer* 26(2), 34-45 (1993).
7. E. Bethke, *Game Development and Production* (Wordware Publishing, Inc., Plano, Texas, 2003).
8. S. Rabin, *Introduction to Game Development* (Charles River Media, Boston, 2005).
9. H. Chandler, *The Game Production Handbook* (Charles River Media, Boston, 2006).
10. I. van de Weerd, S. Brinkkemper, J. Souer, and J. Versendaal, A Situational Implementation Method for Web-based Content Management System-applications, *Software Process: Improvement and Practice* 11(5), 521-538 (2006).
11. I. van de Weerd, J. Versendaal, and S. Brinkkemper, A Product Software Knowledge Infrastructure for Situational Capability Maturation: Vision and Case Studies in Product Management, *Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06)*, Luxembourg (2006)
12. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, From SHIQ and RDF to OWL: The Making of a Web Ontology Language, *Journal of Web Semantics* 1(1), 7-26 (2003).
13. G.E. Mills et al., 2005 Casual Games White Paper (IGDA Casual Games SIG, 2005).
14. R.K. Yin, *Case Study Research: Design and Methods* (Sage Publications, Thousand Oaks, CA, 2003).