Proceedings of I-KNOW '08 and I-MEDIA '08
Graz, Austria, September 3-5, 2008

Towards Continuous Integration of Knowledge Management into Game Development

Steffen Lohmann and Jörg Niesenhaus
(University of Duisburg-Essen, Germany
{steffen.lohmann, joerg.niesenhaus}@uni-due.de)

Abstract: Due to increasing professionalization and specialization in the development of computer and video games new challenges regarding the support of knowledge-intensive activities emerge. This paper aims at sensitizing and systematizing the needs and potentials for continuous integration of knowledge management into game development. It describes the interplay of development activities and involved parties with the knowledge creation process and provides insight into a tool-based approach that aims to support knowledge management in game development on three distinct levels.

Key Words: game development, knowledge management, continuous integration
Category: D.2.0, D.2.1, D.2.9, M.1, M.3

1 Introduction

The game industry has gone through an overwhelming economic growth within the past years and analyst reports foresee a strong growth in the nearby future as well. The branch of game development and publishing is already a major industry with its strongest markets in North America, Japan, and Europe. Along with this ongoing growth comes an increasing professionalization in the development process of digital games. Higher budgets and larger development teams also cause a growing specialization. Due to the outsourcing of development parts, the whole process tends to be more geographically dispersed. In addition, specific components (e.g., game engines, graphic assets, level artifacts, or development environments) are frequently bought from third-party developers [Saltzman 2003].

Along with the professionalization and specialization of the game industry, the documentation and maintenance of knowledge in game projects gets a higher priority. Complex dependencies have to be handled and not only data but also knowledge has to be transferred between project partners. Furthermore, a huge amount of knowledge on content-specific aspects of games is generated, such as the design of the game world, the storytelling, the basic game mechanics, and technical specifications (vgl. [Rollings 2003]). In simple terms: An essential part of game development is knowledge management.

However, the integration of knowledge management support into the development of digital games is confronted with specific challenges: Game development is characterized by very agile and creative processes. These processes should not be affected in their development by too systematic approaches or restricted methodical procedures. Thus, the key to success lies in an agile, minimal-invasive knowledge management approach having at best only little effect on the processes, not dictating strict procedures, coming along with minimal effort and simultaneously offering an immediate additional benefit to the project and all participants.

The rest of the paper is structured as follows: Based on a brief description of the state of the art in Section 2, a conceptual framework is presented and potentials for integrated knowledge management are identified in Section 3. Subsequently, a tool-based approach providing different levels of support is introduced in Section 4 and a short conclusion is drawn in Section 5.

2 State of the Art

So far, no established or even standardized continuous support for qualitative knowledge management exists in the area of digital games. Due to the many differences between game projects and work-related software projects, existing methods cannot be assigned to the area of game development without difficulty.

Normally, development studios use version control systems (such as *Perforce*[1], *Alienbrain*[2]) in order to administrate data and documents. Knowledge about technical and game mechanical issues is primarily stored inside the single documents and files of the respective version control system that are updated in variable time intervals. Outdated knowledge partially exists for any length of time – contrary and inconsistent conclusions across several documents are not rare.

In General, it is differentiated between runtime files and design documents. The game is generated out of the runtime files (e.g., engine, scripts, graphics) in its different design stages due to ongoing testing. The design documents describe guidelines and design decisions concerning the use of technology and game mechanics and define responsibilities.

The absence of a global document and knowledge management is clearly noticeable in this structure. Though it is common practice to set references within the design documents to other documents, this happens predominantly manually so that references often become obsolete or fragmentary as time passes by. In many cases, most files are insufficiently or not at all commented due to lack of time. Thus, the function of a file can often only be derived through its location in the file system, the version control structure, and its name.

[1] *Perforce* is a software configuration management system. More information at http://www.perforce.com

[2] *Alienbrain* is an asset management system that mainly supports graphic intensive projects. More information at http://www.softimage.com/products/alienbrain

It is primarily not the competitive situation of single team members or organizational units and teams between each other but rather the additional effort that prevents team members from externalizing their knowledge. No appropriate possibility is offered for documenting knowledge adequately and making it retrievable in later stages of development.

3 Potentials for Integrated Knowledge Management

By analyzing typical activities along the game development process, several connecting factors for knowledge management support can be identified.

3.1 Knowledge Management with Respect to the Development Process

At the beginning of the game development process in the phases of *pitching* and *pre-production* (see Figure 1) central activities are the generation of ideas and the creation of new game concepts: Numerous ideas are generated, developed, and discarded so that the shape of the game concept changes again and again. Converting these agile processes into permanent knowledge is of great value for a development team because it allows to reconstruct at a later time why ideas were discarded, why incompatibilities existed, and how a problem was finally solved. During the development of a game it occasionally happens that the team returns to an earlier point of discussion and reconsiders decisions on the basis of a new understanding of the game context.

In many cases, experiences of preceding projects are included in the considerations so that it is useful to activate knowledge of projects which have been accomplished in the past. Knowledge should always be connected with structures, files, and program code to enhance the chance of reusing already existing components [Rollings 2003].

With the beginning of the *production* phase, the demand of constant documentation increases with every generated version of the game. In this phase, priority should be given to the interconnection of file structures, data, and knowledge to make every step within the version control traceable and – if required – revocable at a later time.

After the completion of a game project the review and final feedback discussions start. This phase has come to be called *postmortem* in developer jargon. At this stage, the processes, problems, and experiences of the previous project are discussed in order to use this knowledge for future projects. In addition to the feedback given by the developers themselves, the experiences of the service units, the publisher as well as feedback of external experts, media representatives, and the community are brought into the discussion.

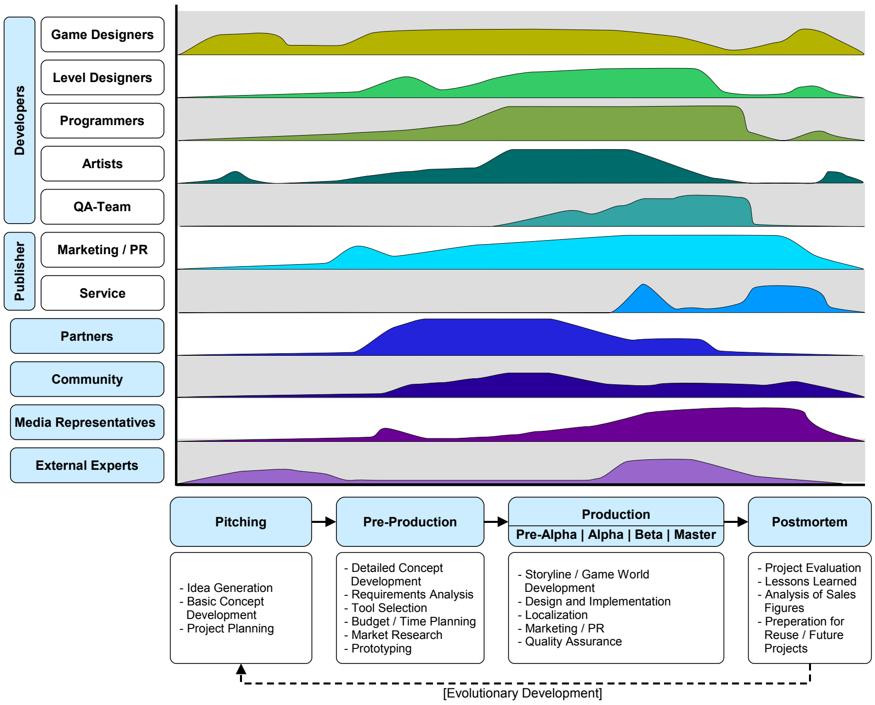

Figure 1: Typical game development process – The curves describe the distribution of intensity of the knowledge management activities.

3.2 Knowledge Management with Respect to the Involved Parties

The identification of participants involved in the development process clarifies the potentials of supporting knowledge management (see Figure 1).

The group of *developers* includes all participants actively involved in the product development such as game designers, programmers, graphic artists, level designers, etc. All developers contribute with their personal experiences and expertise to the project. Due to the high fluctuation of participants between development teams or departments it becomes eminently important to give insight into the individual knowledge of the team members of a project to prevent a loss of knowledge when a member leaves the team.

The *publisher* is responsible for the finance, placement, marketing, and distribution of the game. Besides the continuous dialogue with the developers, all knowledge intensive processes converge at the publisher making it possible to launch a product successfully. The coordination of marketing and public relations, the localization of a product for different markets, and the organization of the distribution are only a few examples for these processes.

Increasing project sizes cause a stronger involvement of outsourcing *partners*. This cooperation requires a particularly intensive form of knowledge exchange since it is essential to create a shared understanding of the project and to ensure that all externally developed components fit seamlessly into the overall product. Furthermore, the adjustment of the game to the current and future hardware demands an active knowledge exchange between game developers and hardware producers.

The *community* that gathers around published or announced game products is characterized by a high activity and a huge engagement when it comes to the critical review or discussion of a game or ideas for improvements and new features. There are numerous well-established community portals, boards, and weblogs that focus on digital games (e.g., Gamespot[3], IGN[4]). Thus, we consider it valuable to integrate the community as far as possible in the development process in order to gain new ideas, helpful suggestions, and feedback from outside (use of the *wisdom of the crowds* [Surowiecki 2004]). Some projects go as far as to give away basic decisions of the game design to the community and let the game fans actively participate in the development process (an example is the project *Top Secret*[5]).

The group of *media representatives* consists of journalists, editors, and producers who work for media formats in the area of digital games. Often, members of this group get the chance to test an early version of the game in order to prepare previews. The feedback of these previews is of great importance for the developers because the journalists are often the first external persons to see and play the game. Due to their broad experiences with digital games they often give valuable advice regarding bugs or weaknesses of the prototype game version.

Furthermore, *external experts* are involved in the project to assist the developers, for instance, when it comes to pedagogical or child-welfare issues. It is of great importance to the developers and publishers to receive early feedback on possible obstacles regarding age restrictions in order to still have an influence on changes. A high age rating (e.g., 'Mature' or 'Adult-Only') can have a negative impact on the sales numbers of a game and is an important factor for the calculation of a publisher. In addition, pedagogues might support the developers in designing games for special target groups in the areas of serious gaming [Michael and Chen 2005] and game-based learning [Prensky 2007].

[3] One of the biggest game portals for the American and European market – see http://www.gamespot.com

[4] Big American and British portal for interactive entertainment and new media – see http://www.ign.com

[5] *Top Secret* is a massive multiplayer online racing game which is developed under the lead of David Perry together with about 60.000 community members.

4 Continuous Integration of Knowledge Management

Against the background of the described dimensions of integration, we work on a solution approach in the context of the *SoftWiki* project[6] that supports knowledge management on three distinct levels: by a central knowledge repository, embedded feedback channels, and knowledge extraction mechanisms. In the following, we briefly describe these levels of support and illustrate our approach by means of tools we are currently developing within the project.

4.1 Central Knowledge Repository

A knowledge repository forms the central access point for all knowledge management activities. Our approach builds on the open-source framework *Powl* and the corresponding collaboration platform *OntoWiki*[7]. In cooperation with the department for Business-oriented Information Systems (BIS) of the University of Leipzig, we are working on an extension of this collaboration platform with the needs of game development in mind (see Figure 2a). Following the Wiki philosophy [Leuf and Cunningham 2001] the user interface allows intuitive, web-based editing of the game project's knowledge base and provides different access options and views. Besides the developer team, the publishers and partners have separate access rights and are enabled to adapt and update specific parts of the knowledge base. Among others, mechanisms for consistency checking and versioning as well as for voting and semantic search are provided.

In addition, the knowledge repository implements interfaces for syndication and further processing of parts of the knowledge base (such as web services and newsfeeds). That way a developer weblog (see Figure 2b) or mailing list can be easily connected to the repository. The other way around, external knowledge (e.g., provided by hardware producers) can also be easily integrated via appropriate interfaces according to the access rights. We work on an ontological schema that defines basic concepts regarding game development in order to enhance data exchange and semantic interoperability. Since digital games are a very fast-moving domain, we consider in particular easy adaptability and extension in all areas of development support.

4.2 Embedded Feedback Channels

The central knowledge repository is extended by decentralized feedback and communication channels that can be embedded directly into the development or run-time game environments. The general aim is to elicit feedback of users while

[6] Research project, funded by the German Federal Ministry of Education and Research (BMBF) - http://softwiki.de/

[7] http://ontowiki.net/

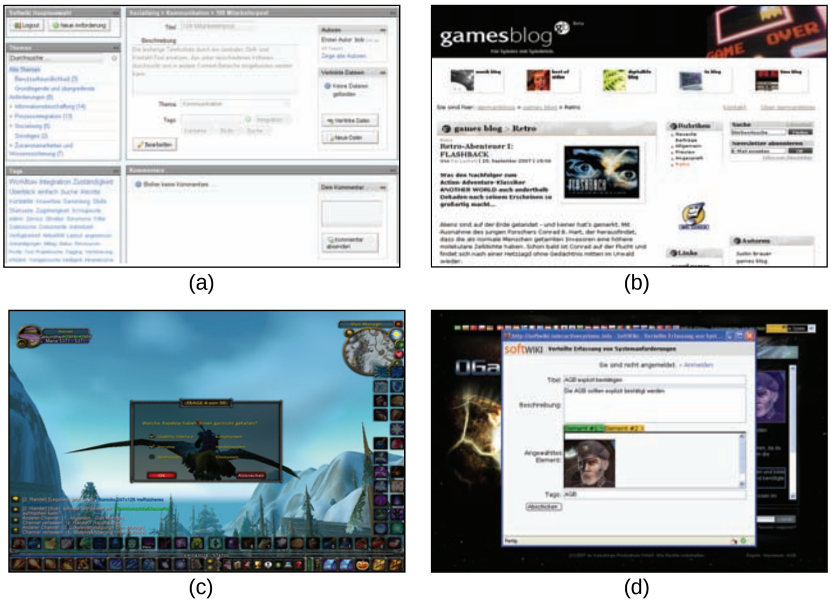

Figure 2: Examples of knowledge management support for game development.

they interact with the applications. Depending on the environment and state of development, different user groups (e.g., the QA-team, gamers, third-party experts, etc.) must be provided with adequate feedback channels. Furthermore, it must be defined if the user should give feedback in a self-directed way (e.g. when it comes to his mind) or if he is explicitly asked for feedback in certain situations.

Figure 2c shows an evaluation tool we implemented in the scripting language LUA [Ierusalimschy et al. 2007]. It can be seamlessly integrated into the game environment – in this case, the online game *World of Warcraft* (WOW)[8] – and enables event-based capturing of feedback within the game world. Figure 2d shows an alternative realization we implemented for web browsers based on web technology and the user interface markup language XUL [Feldt 2007]. It can be used to elicit feedback on browser games: If a bug or feature request occurs, for instance, the user presses a button in the web browser and a web form opens where he can enter a description of the bug or feature request.

4.3 Automated Knowledge Extraction

A third form of knowledge management support in our approach are automated knowledge extraction mechanisms. On the one hand, these mechanisms assist

[8] see http://www.worldofwarcraft.com or http://www.wow-europe.com

in using and analyzing the knowledge stored in the central repository. On the other hand, they can be used to augment the repository by externally available knowledge. For instance, we aim to develop intelligent agents that crawl the web for relevant content in community portals, discussion boards, newsgroups, and weblogs. Due to the increasing use of XML- and RDF-based formats for content representation, automated discovery and processing is facilitated.

5 Conclusion

We tried to point out that continuous integration of knowledge management into the development process of digital games is not only crucial for the success of large projects but also results in several benefits. These include easier adherence to the timetable and lower dependency on the knowledge of individuals reducing the risks and costs of development. Furthermore, continuous knowledge management facilitates the development of game series and secondary or downstream exploitation, for instance, if a similar game structure or the same game engine are used again. Besides its function as documentation, the knowledge base can also serve as a source of inspiration for subsequent projects. Our future work includes further development of the presented tools and their application in game development projects.

Acknowledgements

We thank our colleagues at the University of Leipzig, Thomas Riechert, Frank Schumacher, Sebastian Dietzold, and Sören Auer, for their input on this topic.

References

[Feldt 2007] Feldt, K.C.: “Programming Firefox: Building Rich Internet Applications with XUL”; O’Reilly Media (2007)

[Ierusalimschy et al. 2007] Ierusalimschy, R., de Figueiredo, L.H., Celes, W.: “The Evolution of Lua”; HOPL III: Proceedings of the 3rd ACM SIGPLAN Conference on History of Programming Languages, ACM (2007)

[Leuf and Cunningham 2001] Leuf, B., Cunningham, W.: “The Wiki Way: Quick Collaboration on the Web”; Addison-Wesley Longman (2001)

[Michael and Chen 2005] Michael, D.R., Chen, S.L.: “Serious Games: Games That Educate, Train, and Inform”; Thomson Course Technology (2005)

[Prensky 2007] Prensky, M.: “Digital Game-Based Learning”; Paragon House Publishers (2007)

[Rollings 2003] Rollings, A., Morris, D.: “Game Architecture and Design: A New Edition”; New Riders (2003)

[Saltzman 2003] Saltzman, M.: “Game Creation and Careers: Insider Secrets from Industry Experts”; New Riders (2003)

[Surowiecki 2004] Surowiecki, J.: “The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations”; B&T Press (2004)