Osborne-O'Hagan, A., Coleman, G., O'Connor, R. V., Software Development Processes for Games: A Systematic Literature Review, 21th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2014), CCIS Vol. 425, Springer-Verlag, June 2014, pp 182-193.

Software Development Processes for Games: A Systematic Literature Review

Ann Osborne O'Hagan[1], Gerry Coleman[1], Rory V. O'Connor[2]

[1] Dundalk Institute of Technology, Ireland
{ann.osborneohagan, Gerry.Coleman}@dkit.ie
[2] Dublin City University, Ireland
roconnor@computing.dcu.ie

Abstract. This paper describes the methodology and results from a Systematic Literature Review (SLR) of the software processes used in game development. A total of 404 papers were analyzed as part of the review and the various process models that are used in industry and academia/research are presented. Software Process Improvement (SPI) initiatives for game development are discussed. The factors that promote or deter the adoption of process models, and implementing SPI in practice are highlighted. Our findings indicate that there is no single model that serves as a best practice process model for game development and it is a matter of deciding which model is best suited for a particular game. Agile models such as Scrum and XP are suited to the knowledge intensive domain of game development where innovation and speed to market are vital. Hybrid approaches such as reuse can also be suitable for game development where the risk of the upfront investment in terms of time and cost is mitigated with a game that has stable requirements and a longer lifespan.

Keywords: Game Development, Software Process, Software Process Improvement (SPI), Software Engineering, Systematic Literature Review (SLR).

1 Introduction

Creating computer games is a complicated task that involves the expertise of many skilled professionals from various disciplines including computer science, art and media design and business. The pressure on game development to get to market as quickly as possible means that there are often schedule over runs with correspondingly poor time estimation. Classic software engineering issues associated with game development can include requirements management, configuration management, and verification and validation; these problems can be magnified by geographically distributed teams [1]. Typically the 5 phases involved in the process of creating a game are: concept; plan; design; develop and test [2]. Although best practices from traditional software development are adopted by game development [3], a fundamental difference is that game software aims to provide an experience rather than say produc-

adfa, p. 1, 2011.

tivity. This can cause a divergence in practices, usability testing is not always suitable for games, as game software often has the objective of providing increasingly difficult tasks that the user has to accomplish so that they feel appropriately challenged and eventually satisfied when they complete the challenge. In game development the emphasis is more on evaluating user experiences and using the feedback to drive design iterations. Callele et al.[4] identify clearly that it is necessary to extend the traditional techniques of requirement engineering to support the creative process of the electronic game development.

Developing software for the games industry is evolving rapidly and becoming ever more complex. A Systematic Literature Review (SLR) by Ampatzoglou and Stamelos [5] to assess the current state of the art on research in games development showed that research activity in game engineering is growing at a higher rate than software engineering. The aim of the present study is to assess and document the state of the art of the software processes used in game development. This could provide a foundation and direction for further research in game development processes. Section 2 outlines the research methodology used in this review. Section 3 provides an analysis of the results. Section 4 presents a conclusion of the review.

2 Research Methodology

The research process used has been taken from the guidelines set out by Kitchenham and Charters [6] for performing SLRs in software engineering, and the researcher (as a single researcher) has undertaken the '*light*' version of the review guidelines. The 3 phases of the review and the steps associated with each phase are shown in figure 1.

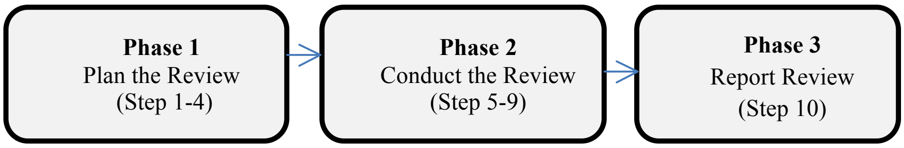

Fig. 1. Phases and Steps of SLR

Phase 1: Plan of the Review. The researcher demonstrated the need for the review (step 1) by searching the Google Scholar digital library and the Evidence Based Software Engineering (EBSE) website [7]. No such research was found by the researcher.

The PICOC (Population, Intervention, Comparison, Outcome and Context) criteria were used to help frame the research questions. Three primary research questions (RQ) were addressed by the review (step 2):

- **RQ1:** What Software Process Models are used in Game Development?
- **RQ2:** What Software Process Improvement (SPI) initiatives are in use in Game Development and to what extent is SPI used in practice?
- **RQ3:** What factors influence the adoption of Software Process Models and SPI in Game Development in practice?

A review protocol (step 3) was developed to reduce researcher bias and to ensure that the review could be replicated. The evaluation of the protocol (step 4) and the subsequent process of implementing the review iteratively improved the design of the review. The final review protocol is described in sections 2.1 to 2.5 (Steps 5-9 incl.).

2.1 Phase 2: Conduct the Review, Search Strategy (Step 5)

Creation of the search protocol consisted of a trial search similar to that performed by Unterkalmsteiner et al. [8]. A search was conducted with the keywords identified in the research questions and was compared against a known primary set of 25 papers requested from an expert in the field of process and process improvement. The keywords were iteratively improved until there was a >= 90% match rate. The terms computer, pc, mobile, software, video, online, console, serious, learning, educational, simulation, entertainment, role-playing, case study, engine, framework and interface were added to the intervention to ensure that the quota of papers were captured. The final search string captured all 25 reference papers.

A search of digital libraries was used to locate peer reviewed journal papers, conference proceedings and published books. The time period covered by the review was 2002 to 2013 (inclusive). The reference lists of primary studies were checked to find other primary studies. The search strategy showing population (P) and intervention (I) for each RQ is outlined in **Table 1**.

Table 1. Search Strategy

RQ1, RQ2, RQ3 (P)	RQ1 (I)	RQ2 (I)	RQ3 (I)
(game AND (development OR computer OR pc OR mobile OR software OR video* [1] OR online OR console OR serious OR learning OR educational OR simulation OR entertainment OR "role-playing" OR "case study" OR engine OR framework OR interface))	(process OR life-cycle or model OR method* OR requirement OR design OR management OR agile OR Scrum OR test*)	(innovation OR improve* OR SPI OR quality OR initiative OR strategy OR practice OR technique OR tool OR "lessons learned")	(motivation OR benefit OR advantage OR enable OR promote OR success OR barrier OR difficulty OR issue OR problem OR challenge OR disadvantage OR deter OR inhibit OR failure)

The searches revealed more than seven thousand publications and a procedure was put in place to help store, track and reference the studies in an organized and reproducible fashion. The following tools were used by the researcher: Microsoft excel was used to store search results; End note was used as a reference manager; and Atlas TI was used to store full text studies and to help with data extraction and categorization.

[1] * denotes a wildcard.

2.2 Primary Study Selection Criteria (Step 6)

A sample of the inclusion and exclusion criteria included peer reviewed journals, conference papers and published books showing qualitative and quantitative research relating to the development process in game development are included. The following studies were excluded: Non English texts; studies relating to Game Based Learning; Artificial Intelligence; networking; graphics algorithms; game theory; affective gaming; computational intelligence; human centered computing (HCI); user interaction; gamification; game based tools, and game based development approaches.

Study Selection Procedure. The researcher conducted a study selection pilot and a data extraction pilot to help with the problem of a single researcher applying inclusion/exclusion criteria and undertaking all the data extraction. This pilot helped ensure that the study selection criteria and the study classification were consistent between the researcher and the supervisor. There was satisfactory agreement, as illustrated by a Cohen Kappa [9] value of 0.64. Cohen's kappa coefficient is a statistical measure of inter-rater agreement for qualitative (categorical) items, and is thought to be more robust than percentage agreement as it takes agreement by chance into account. The equation for the coefficient is:

$$k = \frac{\Pr(a) - Pr(e)}{1 - \Pr(e)} \tag{1}$$

The hypothetical probability of chance agreement was $Pr(e) = 0.56$ and the relative observed agreement among the raters was $\Pr(a) = 0.84$. Conflicts in the results were resolved with a post mortem and this helped fine-tune the inclusion and exclusion criteria. The selection procedure started at this point. Searching the digital libraries was unlikely to find all relevant papers (see Section 2.5) and more papers were found by following up references in included papers [7], this is referred to as snowballing. A total of 7506 papers were retrieved from the searches. Duplicates and unavailable studies were excluded and each set was reduced to the full text studies of 404 papers, as illustrated in **Table 3**.

Table 3. Primary study selection

Digital Library	Initial number	Round 1 (Title, Keyword and abstract)	Round 2 (Full text) Duplicate[2] and unavailable removed	Final count
ACM	751	115	72	43
IEEE	2408	419	249	170
Science Direct	2204	111	42	69
Springer Link	1290	192	98	94
Wiley	232	37	17	20
IGI Global	583	30	28	2

[2] Studies that were listed in more than one digital library (duplicate) were resolved by keeping the copy that was most easily accessible to the researcher.

Inderscience	38	7	1	6
Total	**7506**	**911**	**507**	**404**

2.3 Study Quality Assessment (Step 7)

Study quality assessment can be used to guide the interpretation of the synthesis findings and determine the strength of the inferences according to [6]. The quality assessment performed in this review reports on reporting rather than study quality, as it is not possible to assess the authors' ability to address threats to validity [8]. Qualitative (**Table 4**) and quantitative (**Table 5**) studies had key questions answered during the data extraction.

Table 4. Quality assessment for Qualitative Studies

ID	Qualitative studies (361 Studies)	Yes	Partially	No
QA1	Clear unambiguous findings	165		196
QA2	Referenced Well	218	98	43

Table 5. Quality assessment for Quantitative Studies

ID	Quantitative studies (43 Studies)	Yes	Moderately	No
QA1	Aims clearly stated	42	N/A	1
QA2	Approach clearly explained	39	1	3
QA3	Context of research setting well described	31	10	2
QA4	Threats to validity considered	8	N/A	35

2.4 Data Extraction (Step 8) and Data Synthesis (Step 9)

The primary studies data were collected by means of a data extraction form initially; some of whose properties are shown in **Table 6**. The properties were extracted and tabulated to answer the research questions. The quantitative and qualitative studies were synthesized separately. The studies were classified according [8] to: Industry that refers to studies in where the research was performed in collaboration with or embedded in industry; or Non-industry that refers to studies performed in an academic setting or where the research environment is not properly described.

Table 6. Extracted Data

Property	RQ's
Identify the Software Process Models used in game development.	RQ1
Identify what SPI initiatives exist in game development.	RQ2
Identify the extent of SPI initiative being practiced in industry.	RQ2
What factors aid/deter the adoption of process models in practice?	RQ3
What factors aid/deter the adoption of SPI initiatives in practice?	RQ3

2.5 Phase 3: Report the Review, Study report and validation (Step 10)

The researcher identified 2 primary threats to the validity of the review: Firstly, telemetry, game metrics and data analytics would traditionally have had nothing to do with the game development process but from the research papers identified, the boundary is now shifting and this information is feeding into the development process. Secondly, there is no standard abstracting service, all the digital libraries use different interfaces, and there is a potential inconsistency in the search strategy.

3 Analysis of Results

A total of 404 primary studies were collated and analyzed as part of this review. There was an increase in publications from 2004 onwards with a peak in 2012. A total of 33 genres were recorded with serious games occurring most followed by generic/multi genre. Six types of platforms were recorded, 'mobile' occurring most often followed by the 'online' platform. There were many research methods recorded; case studies being the most frequently used. The majority (73%) of the primary studies were non-industrial (N) the balance (27%) was industrial (I).

3.1 RQ1: What Software Process Models are used in Game Development?

A total of 356 software processes were identified and grouped into 23 process models. The models belonged to either an agile (47%) or hybrid (mixture of traditional and agile) (53%) approach to game development. Agile and hybrid approaches differ on the expected amount and role of iteration. Development in the hybrid approach aims for a minimum number of iterations between phases, whereas development in the agile approach expects to return to the design and requirements stage, and there may be much iteration of the design and testing phase [10]. The primary studies were categorized according to quantitative (11%) and qualitative (89%) methods and each were analyzed according to development approach and context, examples are presented.

Analysis of the Quantitative Studies. There were fewer industrial (46.5%) than non-industrial studies (53.5%). In an Industrial context, agile accounted for 9% of the software processes and hybrid for 91%. In a non-industrial context, agile accounted for 41% of the software processes and hybrid for 59%.

Agile development in an Industrial context used Kanban and Scrum methodologies [11] and in a non-industrial context Rapid Application Development (RAD) [12] was used. Hybrid development in Industry used Component Based Development (CBD) [13], Modular Development [14], The Staged Delivery model (incremental) [15], and an empirical model of the game software development processes is proposed [16]. In a non-Industrial context Novak [17] proposes a generic model of the game development process.

The agile approach of XP [11] methodology and the hybrid approach of Reuse were common in both industrial and non-industrial studies. Reuse was the most commonly used software process in game development.

Analysis of Qualitative studies. There are substantially fewer industrial studies (25%) than the non-industrial studies (75%). In an industrial context 64% of the software processes were agile and 36% were hybrid. In a non-Industrial context 41% of the software processes were agile and 59% were hybrid.

Agile development in an Industrial context used Kanban [18], XP [19], and Ad-hoc development processes [20]. Hybrid development in Industry used IEEE SS&E [1], ETVX model [21], and SDLC [20]. In a non-Industrial context Model Driven Development (MDD) [22] and RUP [23]. Agile approaches which are evident in both contexts include: Evolutionary [24]; Spiral Process [2]; and Scrum [2]. Hybrid approaches evident in both industry and non-industry studies include: Modular [25] ; Reuse [26]; and Incremental model (Waterfall with iterations) [2].

3.2 RQ2: What SPI initiatives are used in game development? To what extent is SPI used in Game Development in practice?

A total of 148 SPI initiatives across both quantitative (17%) and qualitative (83%) studies were recorded in both industrial and non-industrial studies. There were no studies reporting the extent of SPI in practice. The SPI initiatives identified were grouped into requirement, design, development, evaluation and deployment. The initiatives were analyzed in terms of context; examples of SPI initiatives are presented.

Analysis of Quantitative Studies. There were fewer Industrial (48%) than Non-Industrial (52%) studies. The industrial studies contained design (67%) and development (33%) initiatives. The non-industrial studies contained design (77%), development (8%) and evaluation (15%) initiatives.

A development approach (I) concluded that Object Oriented development should be used with great care in the development of mobile games, and that structural programming can be a very competitive alternative [13]; An evaluation strategy (I) [27], used the MIPA (Middleware Infrastructure for Predicate detection in Asynchronous) framework to perform efficient evaluations to identify more usability defects.

Analysis of Qualitative Studies. The Industrial studies (36%) are substantially lower than the Non-Industrial (64%) studies uncovered by the research. The industrial studies contained Requirement (4%), Design (33%), Development (56%), Evaluation (5%) and Deployment (2%) initiatives. The non-industrial studies contained Design (62%), Development (23%), Evaluation (14%) and Deployment (1%) initiatives. The following examples of SPI initiatives are all from industrial studies.

A requirement approach [28], a design technique called a ‘game jam’ [29] and a deployment strategy are described [30]. A company transitioned from a “Laissezfaire" waterfall team to a simple and well-tuned Lean/Agile team by introducing agile and Kanban in [18]. Paring CMMI with IEEE CS SS&E standards in [1] is a development framework used by the US Defense Forces known as 'America's Army' gaming. A method for extracting a product line and evolving it, relying on a strategy that uses refactoring expressed in terms of simpler programming laws is described [31].

3.3 RQ3: Adoption Factors of Software Process Models and SPI in Practice.

Adoption Factors of Process Models. The factors that aid or deter the adoption of the process models described earlier are taken from the industrial studies and are described according to the approach taken agile or hybrid. The following factors are all from qualitative studies with one exception [11].

Agile Adoption Factors. Functional prototypes are useful for communicating requirements to a development team and the iterative approach is useful where organizations lack knowledge of another's area of expertise [32]. Organizations can play a role by fostering a collaborative spirit and providing the physical tools needed [33]. The XP methodology is by its nature suited to Bottom Up development, where requirements are likely to change and the build is incremental. The spiral process model [24] is suited to large projects: investment in training; having the right mix of people working together such as those with functional and gaming skills; a focus on features; a loose—tight discipline throughout the project; and quality and insurance against feature creep are all important for adoption. Scrum [11] is a suitable model: when requirements are hard to pre-define and are volatile; where product innovation and first-to-market thinking are a priority; and there is a desire to improve the quality and productivity of game development. Scrum in practice can cause problems especially the use of the sprint backlog, Lean principles such as Kanban can alleviate these issues in the production stage. The Scrum project management process requires flexible timetabling between designers and other stakeholders to implement [34]; the manager needs to be ready to move at the same or faster speed as the team to be in the lead [35]; and lessons learned in GameDevCo [36] report that to support their transition efforts to Scrum, the company retained an external consultant to mentor their Scrum masters. It was detrimental to the company that the consultant left before passing on the knowledge, which led to variations in the development process. Lack of training for contract employees and the lack of an effective tool to support the rapid development cycle time has also caused adoption issues of Scrum.

Hybrid Adoption Factors. Reuse processes such as the use of Mobile Games Product Lines can be incremental and may offer moderate costs and risks [31]. On the other side of this argument is that the proactive approach to product lines may be inadequate due to prohibitively high investment and risks. The hierarchical model of software product families is argued to be primarily suitable for large organisations with long-lived products. A considerable maturity with respect to development process and management is required. Systems with relatively stable requirement sets and long lifetimes are substantially more suitable than products whose requirements change frequently and drastically, such as due to new technological possibilities. Fathammer [37] has succeeded in creating a hierarchical software product family model that suits its needs very well. The demand for artistic vision, the need for novelty and the demand for creative designs are some of the unusual features of game development that promote the adoption of ISO/IEC 29110 [38]. A deterrent to using ISO/IEC 29110 is that it needs more support for iterative development to allow easier adaptation to real-life organizations.

Adoption Factors of SPI. The factors that aid and deter the adoption of SPI in game development were analyzed under the following headings: SPI Design; SPI Development; and SPI Evaluation. Some examples are described in this section.

A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects helped the process [39] as there was a need for this in requirements engineering, whereas designing a video game with a proposed Game Design Document (GDD) [28] required experience and training. The lack of version control on this GDD was a deterrent to adopting this process improvement.

SPI development approaches and frameworks have been adopted to improve the flexibility of a development team and help provide a sustainable iterative pace by integrating Kanban into the iterative process [21]. Having a good product owner and scrum master are critical for process improvement. Putting the required time and money into establishing these conditions is necessary for process improvement according to [40]. A paring of CMMI with IEEE CS SS&E standards in [1], helped to train staff and to improve SE practices.

4 Conclusion

The software processes identified by RQ1 were almost evenly distributed across agile and hybrid approaches, however the qualitative industrial studies reported almost double the use of agile processes, whereas the quantitative industrial studies were dominated (90%) by the use of hybrid processes.

Almost a fifth of the SPI initiatives identified by RQ2 emanated from quantitative studies, and there were a disproportionate number of industrial papers in the qualitative studies (half that of non-industrial). The qualitative studies contained a much broader range of SPI initiatives across all the development phases of the game development process such as Requirement, Design, Development, Evaluation and Deployment, compared to the quantitative studies that only reported SPI initiatives on the Design, Development and Evaluation phases. The industry quantitative studies had double the SPI Design and half the Development initiatives compared to the industry qualitative studies.

RQ3 highlighted how lightweight agile approaches such as XP, Scrum and Kanban are suitable where time to market and innovation are critical, the risk driven Spiral model is suitable for larger projects. Hybrid approaches, such as reuse, are needed when the investment in terms of time and cost are warranted by more stable requirements and products/games have longer lifespans. Good motivation and the provision of critical resources such as expert training were described as essential for SPI.

All the findings in this review are influenced by the predominance of non-industrial studies in the literature and the motivational differences between industry and research for using various process models. In academia, research rigor, rather than time-to-market, can be seen as more important, Model Driven Development (MDD) was used only in research [22]. The fact is that there are more studies available from the academic side. Many of the reports from industry exist in 'grey' literature, such as magazines, websites etc. This prompts future research to investigate what is actually happening on the ground in game development. Recommendations for future research would be the development of a best practice model for game development. A closer look at the game testing phase and how it is being conducted is also warranted.

References

1. Land, S.K., Wilson, B.: Using IEEE standards to support America's Army gaming development. Computer 39, 105-107 (2006)
2. Kanode, C.M., Haddad, H.M.: Software Engineering Challenges in Game Development. In: Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on, pp. 260-265. (2009)
3. Petrillo, b., Pimenta, M., Trindade, F., Dietrich, C.: Houston, we have a problem...: a survey of actual problems in computer games development. Proceedings of the 2008 ACM symposium on Applied computing, pp. 707-711. ACM, Fortaleza, Ceara, Brazil (2008)
4. Callele, D., Neufeld, E., Schneider, K.: Requirements engineering and the creative process in the video game industry. In: Proceedings of the 2005 13th IEEE International Conference onRequirements Engineering,. , pp. 240-250. (2005)
5. Ampatzoglou, A., Stamelos, I.: Software engineering research for computer games: A systematic review. Information and Software Technology 52, 888-901 (2010)
6. Kitchenham, B.A., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering V 2.3. School of Computer Science and Mathematics Keele University,Department of Computer Science University of Durham, (2007)
7. Evidence Based Software Engineering (EBSE), http://www.dur.ac.uk/ebse/
8. Unterkalmsteiner, M., Gorschek, T., Islam, A.K.M.M., Chow Kian, C., Permadi, R.B., Feldt, R.: Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review. Software Engineering, IEEE Transactions on 38, 398-424 (2012)
9. Emam, K.: Benchmarking Kappa: Interrater Agreement in Software Process Assessments. Empirical Software Engineering 4, 113-133 (1999)
10. Kasurinen, J., Maglyas, A., Smolander, K.: Is Requirements Engineering Useless in Game Development? In: Salinesi, C., Weerd, I. (eds.) Requirements Engineering: Foundation for Software Quality, vol. 8396, pp. 1-16. Springer International Publishing (2014)
11. Koutonen, J., Leppänen, M.: How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios. In: Baumeister, H., Weber, B. (eds.) Agile Processes in Software Engineering and Extreme Programming, vol. 149, pp. 135-149. Springer Berlin Heidelberg (2013)

12. Birchall, J., Gatzidis, C.: The Periodic Table of Elements via an XNA-Powered Serious Game. In: Pan, Z., Cheok, A., Müller, W., Liarokapis, F. (eds.) Transactions on Edutainment IX, vol. 7544, pp. 1-28. Springer Berlin Heidelberg (2013)
13. Weishan, Z., Dong, H., Kunz, T., Hansen, K.M.: Mobile Game Development: Object-Orientation or Not. In: Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, pp. 601-608. (2007)
14. Ampatzoglou, A., Chatzigeorgiou, A.: Evaluation of object-oriented design patterns in game development. Information and Software Technology 49, 445-454 (2007)
15. Petrillo, F., Pimenta, M.: Is agility out there?: agile practices in game development. Proceedings of the 28th ACM International Conference on Design of Communication, pp. 9-15. ACM, Brazil (2010)
16. Seung Hun, L., Gum Hee, L., Hyun Hoon, C., Doo Heon, S., Sung-Yul, R.: An Empirical Model of the Game Software Development Processes. In: Software Engineering Research, Management and Applications, 2006. Fourth International Conference on, pp. 371-377. (2006)
17. Francillette, Y., Gouaich, A., Hocine, N., Pons, J.: A gameplay loops formal language. In: Computer Games (CGAMES), 2012 17th International Conference on, pp. 94-101. (2012)
18. Polk, R.: Agile and Kanban in Coordination. In: Agile Conference (AGILE), 2011, pp. 263-268. (2011)
19. Musil, J., Schweda, A., Winkler, D., Biffl, S.: Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) Systems, Software and Services Process Improvement, vol. 99, pp. 83-94. Springer Berlin Heidelberg (2010)
20. Zaibon, S.B., Shiratuddin, N.: Towards Developing Mobile Game-Based Learning Engineering Model. In: Computer Science and Information Engineering, 2009 WRI World Congress on, pp. 649-653. (2009)
21. Pa, P.S., Su, T.-P.: Quality Control of Artistic Scenes in Processes of Design and Development of Digital-Game Products. In: Yan, X.-T., Ion, W., Eynard, B. (eds.) Global Design to Gain a Competitive Edge, pp. 103-113. Springer London (2008)
22. Cooper, K.M.L., Longstreet, C.S.: Towards model-driven game engineering for serious educational games: Tailored use cases for game requirements. In: Computer Games (CGAMES), 2012 17th International Conference on, pp. 208-212. (2012)
23. Gomez-Rodriguez, A., Gonzalez-Moreno, J.C., Ramos-Valcarcel, D., Vazquez-Lopez, L.: Modeling serious games using AOSE methodologies. In: Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on, pp. 53-58. (2011)
24. Walfisz, M., Zackariasson, P., Wilson, T.L.: Real-time strategy: Evolutionary game development. Business Horizons 49, 487-498 (2006)
25. Poderi, G.: Simple conversational practices in the case of free and open source software infrastructure. Proceedings of the 12th Participatory Design Conference: Exploratory Papers, Workshop Descriptions, Industry Cases - Volume 2, pp. 45-48. ACM, Roskilde, Denmark (2012)
26. Wu, Y.-h., Yao, X.-x., He, J.: Design and Implementation of the Game Engine Based on Android Platform. In: Internet Technology and Applications (iTAP), 2011 International Conference on, pp. 1-3. (2011)

27. Lee, J., Im, C.-Y.: A Study on User Centered Game Evaluation Guideline Based on the MIPA Framework. In: Kurosu, M. (ed.) Human Centered Design, vol. 5619, pp. 84-93. Springer Berlin Heidelberg (2009)
28. Salazar, M.G., Mitre, H.A., Olalde, C.L., Sanchez, J.L.G.: Proposal of Game Design Document from software engineering requirements perspective. In: Computer Games (CGAMES), 2012 17th International Conference on, pp. 81-85. (2012)
29. Musil, J., Schweda, A., Winkler, D., Biffl, S.: Synthesized essence: what game jams teach about prototyping of new software products. In: Software Engineering, 2010 ACM/IEEE 32nd International Conference on, pp. 183-186. (2010)
30. Alves, V., Cardim, I., Vital, H., Sampaio, P., Damasceno, A., Borba, P., Ramalho, G.: Comparative analysis of porting strategies in J2ME games. In: Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on, pp. 123-132. (2005)
31. Alves, V., Matos, P., Jr., Cole, L., Borba, P., Ramalho, G.: Extracting and Evolving Mobile Games Product Lines. In: Obbink, H., Pohl, K. (eds.) Software Product Lines, vol. 3714, pp. 70-81. Springer Berlin Heidelberg (2005)
32. Taylor, A.S.A., Backlund, P., Engstrom, H., Johannesson, M., Lebram, M.: The Birth of Elinor: A Collaborative Development of a Game Based System for Stroke Rehabilitation. In: Visualisation, 2009. VIZ '09. Second International Conference in, pp. 52-60. (2009)
33. Tran, M.Q., Biddle, R.: Collaboration in serious game development: a case study. Proceedings of the 2008 Conference on Future Play: Research, Play, Share, pp. 49-56. ACM, Toronto, Ontario, Canada (2008)
34. Pulman, A., Shufflebottom, M.: A Virtual Infection Control Simulation: The Development of a Serious Game in the Health-Care Sector. In: Petrovic, O., Brand, A. (eds.) Serious Games on the Move, pp. 43-56. Springer Vienna (2009)
35. Friis, D., Ostergaard, J., Sutherland, J.: Virtual Reality Meets Scrum: How a Senior Team Moved from Management to Leadership. In: System Sciences (HICSS), 2011 44th Hawaii International Conference on, pp. 1-7. (2011)
36. Srinivasan, J., Lundqvist, K.: Organizational Enablers for Agile Adoption: Learning from GameDevCo. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) Agile Processes in Software Engineering and Extreme Programming, vol. 31, pp. 63-72. Springer Berlin Heidelberg (2009)
37. Myllärniemi, V., Raatikainen, M., Männistö, T.: Inter-organisational Approach in Rapid Software Product Family Development — A Case Study. In: Morisio, M. (ed.) Reuse of Off-the-Shelf Components, vol. 4039, pp. 73-86. Springer Berlin Heidelberg (2006)
38. Kasurinen, J., Laine, R., Smolander, K.: How Applicable Is ISO/IEC 29110 in Game Software Development? In: Heidrich, J., Oivo, M., Jedlitschka, A., Baldassarre, M. (eds.) Product-Focused Software Process Improvement, vol. 7983, pp. 5-19. Springer Berlin Heidelberg (2013)
39. Laurent, P., Ma, x, der, P., Cleland-Huang, J., Steele, A.: A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects. In: Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on, pp. 35-44. (2010)
40. Kniberg, H., Farhang, R.: Bootstrapping Scrum and XP under Crisis A Story from the Trenches. In: Agile, 2008. AGILE '08. Conference, pp. 436-444. (2008)