Information and Software Technology 52 (2010) 888–901

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Software engineering research for computer games: A systematic review

Apostolos Ampatzoglou *, Ioannis Stamelos

Department of Informatics, Aristotle University, Aristotle University Campus, PO Address 54124, Thessaloniki, Greece

A R T I C L E I N F O

Article history:
Received 26 June 2009
Received in revised form 7 May 2010
Accepted 8 May 2010
Available online 13 May 2010

Keywords:
Software engineering
Computer games
Systematic review

A B S T R A C T

Context: Currently, computer game development is one of the fastest growing industries in the worldwide economy. In addition to that, computer games are rapidly evolving in the sense that newer game versions arrive in a very short interval. Thus, software engineering techniques are needed for game development in order to achieve greater flexibility and maintainability, less cost and effort, better design, etc. In addition, games present several characteristics that differentiate their development from classical software development.
Objective: This study aims to assess the state of the art on research concerning software engineering for computer games and discuss possible important areas for future research.
Method: We employed a standard methodology for systematic literature reviews using four well known digital libraries.
Results: Software engineering for computer games is a research domain that has doubled its research activity during the last 5 years. The dominant research topic has proven to be requirements engineering, while topics such as software verification and maintenance have been neglected up to now.
Conclusion: The results of the study suggest that software engineering for computer games is a field that embraces many techniques and methods from conventional software engineering and adapts them so as to fit the specific requirements of game development. In addition to that, the study proposes the employment of more elaborate empirical methods, i.e. controlled experiments and case studies, in game software engineering research, which, have not been extensively used up to now.

Contents

* Corresponding author. Tel.: +30 2310991923.
E-mail addresses: apamp@csd.auth.gr (A. Ampatzoglou), stamelos@csd.auth.gr (I. Stamelos).

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.05.004

1. Introduction

Currently, 3D applications appear to be one of the most modern and fast growing software fields from industrial point of view. The most popular subcategory among 3D applications is considered to be computer games. In 2007, the video game industry revenue was approximately $60 billion; that almost equals the size of the US Department of Defense expenditures on research [17]. Furthermore, the game industry is so innovative that in many cases, the hardware and software technological advancement, are applied to games before being adopted by other scientific domains [12,14]. Additionally, among the young game playing hours surpass television watching and listening to music. Finally, concerning open-source communities, games appear to be thriving [15].

Creating computer games is a very complicated task that requires the involvement of extremely skilled professionals from a wide spectrum of computer science [4]. Typically, computer games demand real time high quality performance. The main performance aspects are related to display frame rate, real time audio playback and processor response. Programming a game in low level is so complex that hundreds of thousands of code lines are required in order to implement a commercially viable game. The size of such programs, in combination with the evolving nature of the software, demands flexible design, maintainable implementation and straightforward documentation, in order to improve understanding among the development group and facilitate future developments. Consequently, game developers must employ specific software engineering techniques in order to achieve high quality levels.

In McShaffry et al. [13] the authors present a practitioner's approach to the aspects that differentiate game software engineering from classical software engineering. More specifically, games are products that have much more limited lifecycle than conventional software products. Games are usually developed in a smaller time period and all phases of the lifecycle need to be shrunk. In addition, the main maintenance activity for computer games is corrective maintenance, because most games, after being delivered to the market, have an average life of 6 months and in that time interval the next version of the program is created. During this period, the main maintenance task performed is bug fixing, which is typically provided without charge to the end-users. Consequently, the game development companies do not have corrective maintenance income [13]. However, successful games are often the basis for one or more sequels. If the sequel includes revisions to the user interface or game controls as a result of comments from users, this is a form of perfective maintenance. This procedure does not directly provide any income to the game development company which provides an additional revenue stream to the game developers. Additionally, another interesting characteristic of games is the fact that in many cases, the game development companies release purchasable extensions. These releases use the same game core in order for the "new story" to be told through the "old" game engine, which is already purchased from the end user. This process can be characterized as adaptive maintenance, as well.

Financial shortcomings that derive from the absence of income from the maintenance phase are balanced by game extensions and from marketing campaigns that aim at selling old game titles at lower rates. The extreme marketing demands of the game industry sometimes press the development companies to further shrink their development timetables. Often the schedules are overrun and poor release date estimation becomes an issue. Considering the above, game project management is a complicated task that diverges from traditional software project management.

Even though software engineering methodologies for game development is a field of great interest, there is no clear picture on the advancements in this field. Additionally, to the best of our knowledge no systematic reviews concerning game development has been reported in the literature. The purpose of this study is to summarize the existing evidence concerning software engineering techniques applied to computer games and identify possible gaps in current research, in order to suggest areas for further investigation and to provide background information to any relevant future work. Furthermore, the paper aims to describe the current state of the art in game software engineering with respect to research approaches and methods. In order to achieve this, the evidence-based research paradigm has been employed. The possibility of employing the evidence-based paradigm in software engineering has been proposed in [6,10]. This procedure aims to identify an answerable question that provides information, find evidence that answers the question and evaluate that evidence [1]. According to [1], a systematic review of the literature constitutes the first step of performing evidence-based research. Guidelines on conducting a systematic literature review are thoroughly explained in [1,8].

In the next section, we explain how we have used the methodology discussed in [1]. In Section 3, we present some statistics about the primary studies we analyze in this paper. In Section 4, we answer our research questions and in Section 5 we present possible threats to validity and our plans for future work. Finally, in Section 6 we present our conclusions.

2. Review methodology

Performing a systematic review can be decomposed into three main phases: planning, conducting and documenting the review [1]. Every phase is a combination of other simpler procedures.

Planning phase:

- specify research questions
- develop review protocol
- validate review protocol

Conducting phase:

- identify relevant research
- select primary studies
- assess study quality
- extract required data
- synthesize data

Documenting phase:

- write review report
- validate the report

According to [8,9], planning a review consists of six definitions as shown below.

- define research questions
- define search process
- define inclusion and exclusion criteria
- define quality assessment
- define data collection
- define data analysis

2.1. Research questions

In this study, we are planning to investigate several issues concerning research on game engineering. The main research questions addressed in the study are:

Q_1: Which is the intensity of the research activity on software engineering methods for game development?
Q_2: What software engineering research topics are being addressed in the domain of computer games?
Q_3: What research approaches do software engineering researchers use in the domain of computer games?
Q_4: What empirical research methods do software engineering researchers use in the domain of computer games?

With respect to Q_1, the research group has identified a corpus number of research publications. The graphical representation of the number of publications per year will indicate the existence of an increase or a decrease in the research activity. The slope of the line for game engineering is compared to the corresponding slope of the line representing research activity on traditional software engineering, as provided by [7].

To address Q_2, Q_3 and Q_4, each primary study has been associated to a research topic, to a certain research approach and to a specified research methodology. The classification of the primary studies into corresponding categories is discussed in detail in Sections 2.5.1 and 2.5.2. Concerning Q_2, the results will be compared to those concerning traditional software engineering as presented in [3], whereas the study results on Q_3 and Q_4, will be compared to those of [5,7].

2.2. Search process

A systematic review on a "new topic" should identify and highlight specific sources about the subject under study. However, no such sources exist in the domain of software engineering for computer games. Thus, related studies may be published in journals and conferences that are related either to "traditional software engineering" or to "entertainment computing". In the area of entertainment computing some dominant publishers such as Association for Computing Machinery (ACM), IEEE Computer Society (IEEE), Springer and Elsevier provide several journals and conferences, such as "Computers in Entertainment", "Entertainment Computing", "Conference on Advances in Computer Entertainment", "Conference on Foundations of Digital Games", "SandBox Workshop", "Conference on Future Play" and "Conference on Entertainment Computing".

The search procedure aimed at the identification of candidate primary studies that would be either included or excluded from the final set of the review studies. The search plan involved automated search into four well known digital libraries: ACM Digital Library, IEEE Digital Library, ScienceDirect, and SpringerLink. The search parameters included one keyword, i.e. *game*, which should be included in the article title and another keyword, i.e. *software*, which should be mentioned at least once in the article full text. In addition to that, before the manual observation of the studies, automated filtering was employed. In ScienceDirect the search was limited to subject "Computer Science" and in SpringerLink, the search was limited to subject "Software Engineering".

The article set that has been returned from the aforementioned query consisted of 3463 articles. However, the majority of these articles were identified marginally related to software engineering. The exclusion of irrelevant articles was manually conducted according to the inclusion and exclusion criteria defined in Section 2.3.

2.3. Inclusion and exclusion criteria

The papers that are selected as primary studies in the review had to be relevant to software engineering, i.e. the article should be classified under a software engineering topic, as described in Section 2.5.1. In line with [5], there are four stages in filtering the article set in order to produce the primary study data set. These stages are:

(a) identify relevant studies – search digital libraries (on the completion of this stage the article set consisted of 3472 articles),
(b) exclude studies on the basis of titles (on the completion of this stage the article set consisted of 223 articles),
(c) exclude studies on the basis of abstracts (on the completion of this stage the article set was consisted of 130 articles),
(d) obtain studies and select those relevant to software engineering on the basis of full text (on the completion of this stage the final primary studies dataset consisted of 84 articles).

All papers that have been considered in the primary study selection phase, after the completion of filtering out papers according to their title, are presented in the paper's website (http://sweng.csd.auth.gr/~apamp/survey_games.html). In line with [1] the title and abstract were examined by the first author whereas the full papers which were not rejected at the first three stages were examined by both authors. During the survey, we came up with a variety of interesting papers, but we preferred to include in the review only research that was closely related to software engineering issues. The main issues that are related to game development but are not related to software engineering are "game based learning", "artificial intelligence", "social impact of gaming", "networking" and "graphics algorithms". Finally, in the review only journal, full conference and workshop papers have been considered, conference short papers, smaller than four pages, and posters have been removed from the survey. The final primary study dataset consisted of eighty-four (84) research articles that are presented in an Appendix by the end of the paper.

2.4. Quality assessment

The quality assessment phase of the review procedure aims at validating that primary studies are solid in their methodology and results. Therefore, the results of the evidence-based procedure, which derive from analyzing the primary studies, will be solid as well, in the sense that the quality of the primary studies guarantees the integrity and accuracy of the data set of the secondary study. Considering the high standards of the review process of the selected journals and conferences we believe that the quality of the published articles is satisfactory. The workshop papers have not been excluded from the survey since they are selected through a high standard peer review process and in many cases they are published in sources specializing in entertainment computing. Additionally, it is expected that workshop papers represent research trends that are likely to be published in a journal or a conference in a later time.

2.5. Data collection

On the completion of the paper inclusion or exclusion procedure, a dataset of primary studies has been created. During the data collection phase the authors have extracted several attributes of the primary studies. The data extracted from every paper are:

- Type of Publication (journal, conference, workshop)
- Published in (journal or conference name)
- Publisher (ACM, IEEE, Elsevier, Springer)
- Year of Publication
- Country and Continent
- Classification of topic (as defined in Section 2.5.1)
- Classification of research approach (as defined in Section 2.5.2)
- Classification of research method (as defined in Section 2.5.2)

2.5.1. Classifying topics

This section of the paper aims at clarifying the categories that each study can be classified to, with respect to the software engineering issue it involves. The classification system that was employed is the *ACM Computing Classification System* that is widely adopted in many journals and conferences. The actual primary study classification has been made through the original classification of the study as presented in ACM digital library and therefore is considered safe. In addition to that, articles related to heuristics on user enjoyment are classified in D.2.1, since such heuristics are considered useful in the requirements elicitation phase. Finally, papers dealing with usability testing are classified in D.2.5. The same method has been employed in [3]. Table 1 presents the selected classification schema.

2.5.2. Classifying research approaches and methods

According to [7], scientific research papers can be characterized with respect to their approach and method. In the aforementioned work, an analysis of a classification scheme on handling research approach and methods is presented.

Table 1
Software engineering topics [3].

Software engineering	
D.2.0	General – miscellaneous
D.2.1	Requirements/specification
D.2.2	Design tools and techniques
D.2.3	Coding tools and techniques
D.2.4	Software/program verification
D.2.5	Testing and debugging
D.2.6	Programming environments
D.2.7	Distribution, maintenance and enhancement
D.2.8	Metrics
D.2.9	Management
D.2.10	Design
D.2.11	Software architecture
D.2.12	Interoperability
D.2.13	Software reuse

Table 2
Empirical methods.

Empirical method	Description
Experiment	A set of subjects is asked to perform a task in a highly controlled environment. The results are derived from observing of the subjects during the experiment, from inspecting the task outcome or from questioning the subjects at the end of the procedure
Survey	A set of subjects is asked to fill-in questionnaires either directly, or via internet. The results are derived from the valid answers to the questionnaire
Case study	A project, an activity or an assignment is monitored with respect to the methodology under study. Results are directly derived from project measurements

The main scientific approach categories mentioned in [7], are the descriptive approach, the explanatory approach and the empirical approach. A descriptive research paper typically describes a system, tool or method. In addition to that literature reviews are considered to be descriptive studies. On the other hand, exploratory research is conducted in a case where a problem was not clearly defined. Exploratory research helps determine the best research design, data collection method and selection of subject. Finally, empirical research produces its findings on direct or indirect observation of real subjects.

According to [16], there are three major methods of empirical investigation that are used in order to evaluate new techniques, methods and tools. These investigation types are *surveys*, *case studies* and *experiments*. Experiments are appropriate for exploring relationships, i.e. testing the correlation between two variables. On the other hand, surveys are a suitable way of investigating if the method under study is in use for a while. Additionally, case studies are used for similar reasons as experiments, but their level of control is lower in the sense that they are mostly observational studies [11,16]. The above mentioned empirical validation types are summarized in Table 2. The same classification on empirical methods was employed in [5].

2.6. Data analysis

The collected data that are mentioned in Section 2.5 were tabulated (*see* Appendix B) and statistically analyzed so as to investigate:

- The number of studies that are published per year (addressing Q_1).
- The number of papers that are published in journals and conferences, the journals or conferences that appear to be more active in game engineering research and the number of papers concerning game engineering that each digital library hosts (addressing Q_1).
- The number of studies that each country and continent produced during the last 5 years (addressing Q_1).
- The major software engineering topics that were investigated by game engineering research (addressing Q_2).
- The number of studies that employed each research approach and method (addressing Q_3 and Q_4).

3. Results

This section of the paper deals with presenting the results of the statistical analysis on the dataset of primary studies. At this point, it is necessary to clarify that a comparison between game engineering and other fields, accompanied with discussion on the research questions (Section 2.1) is provided in Section 4. The dataset of the primary studies characteristics are tabulated, so as

Table 3
Research activity per year and citation type.

Year								
Citation type	<2003	2004	2005	2006	2007	2008	2009	Total
Journal	2	3	1	1	8	6	5	26
Conference	2	4	6	6	5	16	15	54
Workshop	0	1	0	0	2	1	0	4
Total	4	8	7	7	15	23	20	84

for the reader to be able to trace the categories that each study is mapped to and the interested reader can refer to Appendix B.

Table 4
Journals/conferences number of publications.

Name	#Papers
Computers in Entertainment	6
Communications of the ACM	4
ACM Future Play	3
IEEE International Conference on Requirements Engineering	3
Science of Computer Programming	3
ACM Special Interest Group in Human–Computer Interaction	3
ACM Advances in Computer Entertainment	2
International Conference on Computer Graphics, Imaging & Visualization	2
Workshop on Design, Specification & Verification of Interactive Systems	2
International Conference on Entertainment Computing	2
IEEE Software	2
MindTrek Conference	2
Mobile Human–Computer Interaction Conference	2
Queue	2
ACM Special Interest Group in Computer Science Education	2

In Table 3, the publication count per year is being presented. Moreover, the aforementioned table provides information on the number of articles that each type of publication (conference, journal or workshop) is associated with, per year.

Furthermore, Table 4 presents the most active research publication outlets for game engineering studies. Moreover, Fig. 1 depicts the most active countries in game engineering research. Table 5 shows the distribution of the publication count in software engineering for computer games among the digital libraries. The data in Tables 3–5 and Fig. 1 will be the basis of the discussion in research question Q_1.

In Table 6, the results of classifying the primary studies into the ACM software engineering categories (Table 1) are presented. At this point it is necessary to clarify that the sum of this table does not equal the number of primary studies, since three studies have

Table 5
Number of publications per publisher.

Publisher name	# Publications
ACM	32
IEEE	25
Springer	19
Elsevier	8

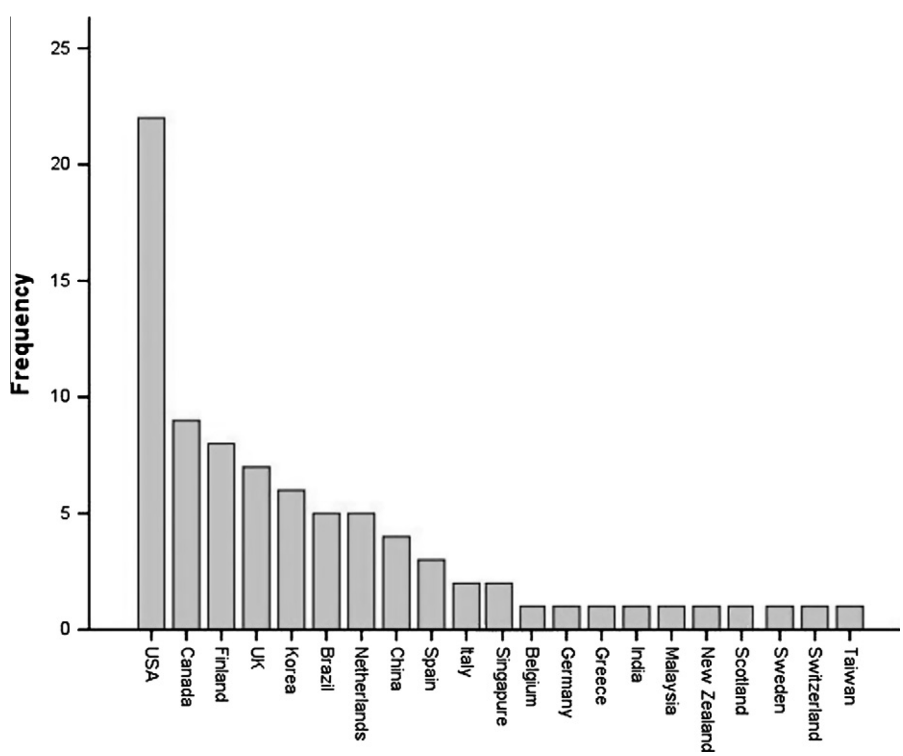

Fig. 1. Research activity per country.

Table 6
Game engineering research topics.

	Software engineering topic	Frequency	Percentage
D.2.0	General	4	4.76%
D.2.1	Requirements/specification	33	39.29%
D.2.2	Design tools and techniques	2	2.38%
D.2.3	Coding tools and techniques	10	11.90%
D.2.4	Software/program verification	0	0.00%
D.2.5	Testing and debugging	7	8.33%
D.2.6	Programming environments	8	9.52%
D.2.7	Distribution, maintenance and enhancement	0	0.00%
D.2.8	Metrics	2	2.38%
D.2.9	Management	10	11.90%
D.2.10	Design	3	3.57%
D.2.11	Software architecture	3	3.57%
D.2.12	Interoperability	0	0.00%
D.2.13	Software reuse	2	2.38%

Table 7
Game engineering empirical research methods.

Empirical Method	Frequency
Case Study	4/20
Experiment	6/20
Survey	10/20

Table 8
Game engineering research approach.

Research Approach	Frequency	pct
Descriptive	24	28,5%
Empirical	20	23,8%
Exploratory	40	47,7%

been classified in two categories each. The data from this table will be used in the discussion of research question ***Q*₂**, in Section 4.2.

In Tables 7 and 8, the results concerning the most popular research approaches and empirical methods are presented.

4. Discussion

This section of the paper discusses the finding of the review concerning the research questions specified in Section 2.1. The findings of our research on game engineering will be compared to the results of similar studies [3,5,7] in order to identify domain specific characteristics.

4.1. Game software engineering research intensity

The results of Table 3, clearly suggest that game software engineering research has increased during the last years. In Fig. 2, the increase in the research activity of software engineering and game software engineering are presented. The research activity increase for year$_{(i)}$ is calculated as a fraction. The numerator of the fraction equals the number of publications in year$_{(i)}$, whereas the denominator equals the number of publications in year$_{(0)}$.

The comparison of the research intensity between the two domains has been conducted in proportion to the first year of the study and not with absolute values, because research on game software engineering is a subset of research on software engineering. As shown in Fig. 2, the research activity of game engineering is growing at a higher rate than software engineering. This fact indicates that during the last 5 years game software engineering is an

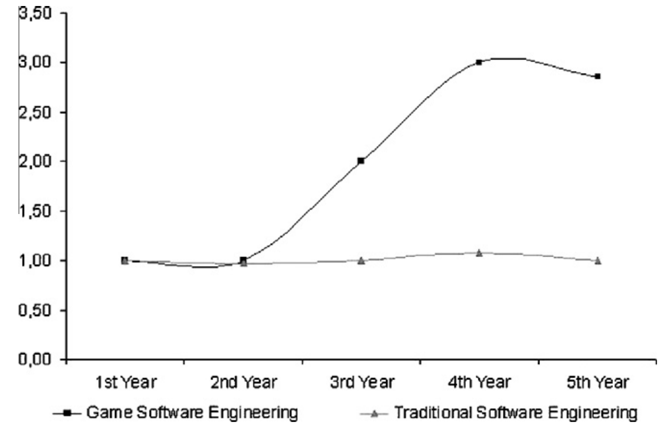

Fig. 2. Scientific domain research activity increase.

increasing research domain. More specifically, the number of game engineering publications has almost tripled up since 2005.

While examining research activity in different countries, the USA dominates software engineering gaming research since US researchers co-authored 26% of the primary studies. Furthermore, until 2004 the research on software engineering for computer games was dominated by the American continent (75%), but since then Europe seems to have published a similar number of research papers. Finally, concerning the most popular venues for game engineering papers, ACM seems to host the main bulk of publications (about 38%), followed by the *IEEE Computer Society*. The ACM journal entitled *Computers in Entertainment* has published 6 articles on software engineering for games followed by 4 articles in *Communications of the ACM*.

4.2. Game software engineering research topics

This section of the paper aims to identify the main research topics in game engineering and comparing them to the main research topics of software engineering. Table 6, clearly suggests that the category in which most game engineering takes place is D.2.1, i.e. *Requirements/Specification*, followed by D.2.3, i.e. *Coding Tools & Techniques*. On the other hand, categories concerning *Software Verification, Distribution, Maintenance, Enhancement and Interoperability*, appear not to have attracted the interest of researchers.

In Fig. 3, a comparison between each research topic activity in game engineering and software engineering is presented. The data concerning software engineering are extracted from [3]. The differences between game and software engineering are most probably caused by two facts. Firstly, it is caused because of the special needs and priorities of game development. Secondly, it is caused by the fact that game engineering is a young domain, which is in need of more fundamental research, such as research on requirements elicitation and specification, development and coding tools, etc. It is expected that when research on such subjects matures, research on other issues such as maintenance, verification, etc, will attract the interest of researchers.

Concerning game specific characteristics that have been discussed in Section 1, several issues in game development are not covered by conventional software engineering. Consequently, research on these topics appears to be more active. Because games are a form of entertainment, elicitation of user enjoyment requirements is more important to the development team than other aspects of game software. Additionally the requirements for a attractive user interface makes stringent demands on the game control systems and often need extensive prototyping of interface requirements. As a result, research effort on game requirements

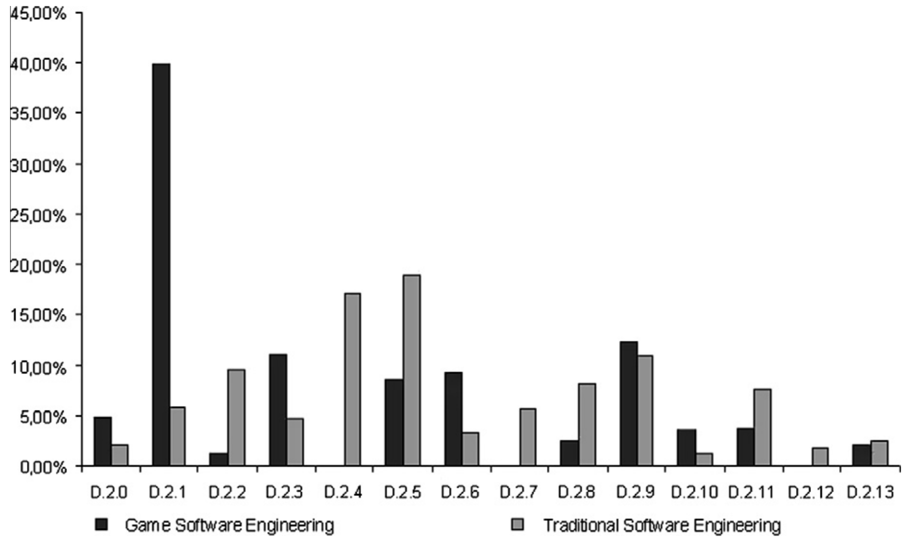

Fig. 3. Research topics.

engineering has increased. Furthermore, in [13] it is suggested that the implementation phase in computer games is smaller with respect to traditional software engineering, since the time to market for games is rather limited. Thus, we expected that categories D.2.3 and D.2.6 would exhibit higher research activity, since game development demands efficient coding and programming techniques. Moreover, the divergences of game project management tasks from conventional software management, which were discussed in Section 1, are supported by the relatively intense research efforts on this subject. Finally, the difference in research load between game design and generic software design (D.2.10 in Fig. 3) suggests the need for sophisticated design techniques, which are necessary to improve the "bad" internal quality reported in [13].

On the other hand, the amount of research on testing/debugging is not proportional to the intensity of testing effort in computer games. Based on the credits of a popular computer game, the development team appears to consist of 67 programmers while 53 people deal with quality assurance. However, much of quality assurance effort in gaming goes to game-play testing, which is considered to be a form of requirements elicitation. Game-play testers produce new requirements in the form of enjoyment heuristics or improve existing requirements.

In addition to that, Fig. 4 indicates the growth of each research topic during the period under study. In order for the graph to be more informative the 13 categories described in Table 1, are now merged in six, namely generic, analysis, design, implementation, quality and management. It is observed that in generic studies were mostly released during the early years of the research (about 50%). On the other hand, studies on game project management appeared mostly during the last 2 years (about 70%).

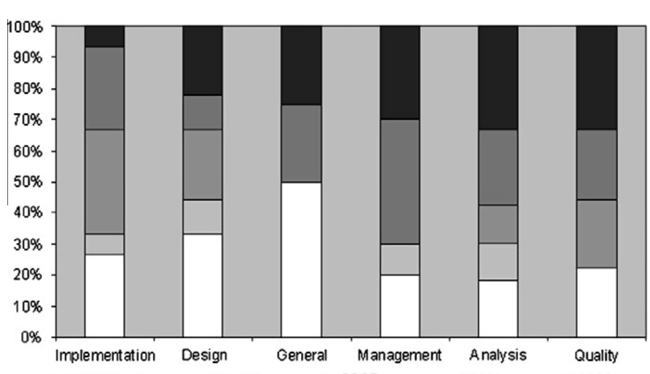

Fig. 4. Research topics growth timeline.

A significant part of a review on the research state of the art is the actual primary studies description. Additionally, according to [2], the primary studies of a mapping study should be clearly associated with the research topic they address. Next, a short description of the research effort on each game software engineering research topic is presented, accompanied with the full reference list for every topic. At this point it is necessary to clarify that, similarly to [6], the full list of primary studies is presented in an appendix, separately from the rest of the paper references.

4.2.1. General – miscellaneous

In [S7] the authors review several general issues concerning wireless game development. In [S9, S14 and S27], several issues such the difficulties of game development, the evolution of mobile games and a quality-driven game development process are discussed.

4.2.2. Analysis

In four studies it is suggested that game developers should put emphasis on the emotional effects of games on players, a point where games lag in comparison with movies, books and other forms of entertainment [S12, S68, S73 and S75]. Additionally, in [S11] the authors highlight the necessity of preproduction documents. The validation method of this study is extremely interesting since it is based on data from a special column of a game development magazine, where game development project managers submit failure reasons for computer game development projects. In four studies [S2, S22, S36 and S37] the authors investigate the special problems of developing mobile games in contrast to computer games with respect to both hardware and software aspects. In addition to that, four papers [S39, S50, S51 and S60] employ prototyping methods in game development so as to validate ideas and create new ones at an early development stage. Furthermore, computer games are created in order to entertain their players. Thus, in nine studies, the authors attempt to introduce user satisfaction heuristics that can be used in requirements elicitation phase [S10, S18, S20, S31, S38, S54, S58, S59 and S66]. In [S1 S23, S64 and S77], requirements for player–game interaction are discussed. Moreover, in [S30, S45, S61, S76 and S78], hardware requirements for computer games are presented. Finally, computer game logical flow is investigated in two studies, as an enjoyment factor [S15 and S72].

4.2.3. Design

In literature, two papers [S5 and S49] deal with evaluating the use of object-oriented design patterns in game development. In addition, in [S8 and S35] the authors propose design solutions for implementing game engines and bluetooth games.

In [S24, S26 and S41] innovative architectures that enhance the reusability of games and game engines are introduced. Such architectures produce more stable and extensible software, increase interoperability, improve robustness and scalability, loose coupling between modules and shorten the architecture's learning curve. Finally, in [S16 and S52] reuse opportunities in game development are examined.

4.2.4. Implementation

One scientific papers deal with automated software engineering for developing games through game mechanics design patterns [S19]. Most computer games employ some kind of language in order to implement the world logic. Several papers investigate the use of mark-up languages (e.g. xml, e-Game, etc.) in order to program game scripts [S46, S65, S79 and S80]. Furthermore, some papers investigate how agile programming methods [S71 and S81] and object-orientation [S83 and S84] can be used in developing games, game engines and mobile games. Finally, in [S13, S25, S28, S33, S40, S43, S44, S56, S62 and S63] the development of games through multi-language approaches, virtual reality authoring systems and toolkits is described.

4.2.5. Quality

Concerning the assessment of game quality, in [S48] an ISO quality model concerning game controls is proposed, whereas in [S82] the authors propose metrics that predict the enjoyment of the user. Moreover, seven publications focus on game usability testing issues [S6, S17, S21, S29, S32, S57 and S69].

4.2.6. Management

Five studies concern the whereabouts of game development companies, highlight the timeline of the development process, involve open-source methods and specify development stages [S42, S53, S67, S70 and S74]. Furthermore, in [S4] the authors discuss all possible activities and deliverables in the game development process. In [S34] the author discusses some unique characteristics of game development when compared to conventional software engineering. Additionally, in [S55] a survey dealing with the problems in the development process of electronic games is presented. Finally, in [S3 and S47] the authors explore application Software Product Lines in game development.

4.3. Game software engineering research approach

According to Table 8, most studies in game engineering employ an exploratory research approach. This observation is consistent with the results of [7] concerning software engineering. In Fig. 5, a comparison between the research approaches employed on software and game engineering is presented.

It is reasonable to expect that since game engineering is a younger research area, more papers should deal with the description of research problems, than in a more mature area. As shown in Fig. 5, this statement is not supported by the research evidence. More specifically, descriptive and evaluative strategies seem to be equally employed in both game software engineering and software engineering. This fact can be explained by the hypothesis that game software engineering researchers are also involved in the field of conventional software engineering research and therefore it is reasonable to expect that they employ similar approaches and methods.

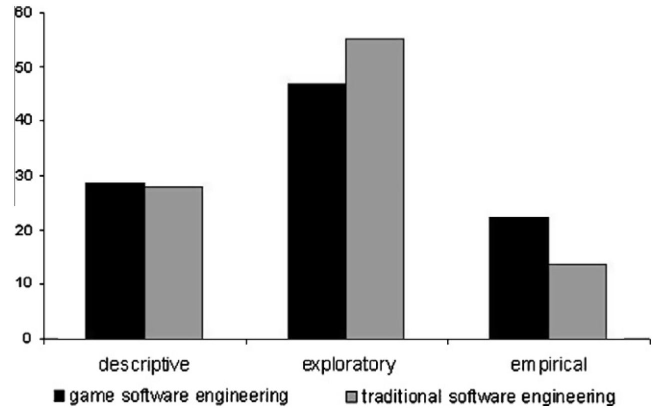

Fig. 5. Research approaches.

4.4. Game software engineering empirical research methods

In Fig. 6, a comparison between the empirical research methods employed in game engineering, conventional software engineering and agile software development is presented. The domains of the comparison have been selected as following: (a) Game software engineering is a subcategory of conventional software engineering. In addition to that, traditional software engineering is a mature scientific area. (b) Agile software development has been selected because it is a young domain and consequently it is a complement to traditional software engineering, with respect to area maturity.

The results suggest that case studies are more frequently employed in Agile Software Development research. In addition to that, surveys are more frequently employed in Game Development research and finally, experiments are more frequently conducted in Software Engineering research. This result can be explained by the level of maturity of the three domains under study. Thus, since software engineering is the most mature domain, more elaborate empirical studies are expected to be employed in it. According to [16] formal experiments are harder to organize and such endeavors demand significant experience in the field. In addition to that, game engineering research also has fewer case studies. A possible explanation for this is that performing a case study requires the existence of project data that should be monitored through a variety of measurements or inspections, and no research databases or repositories on computer games are available. Finally, concerning surveys, game engineering seems to outperform the other domains. This fact is considered reasonable in the sense that game engineering research, which is in its infancy, is attempting to produce knowledge by questioning domain experts, game users etc. Additionally, the increased

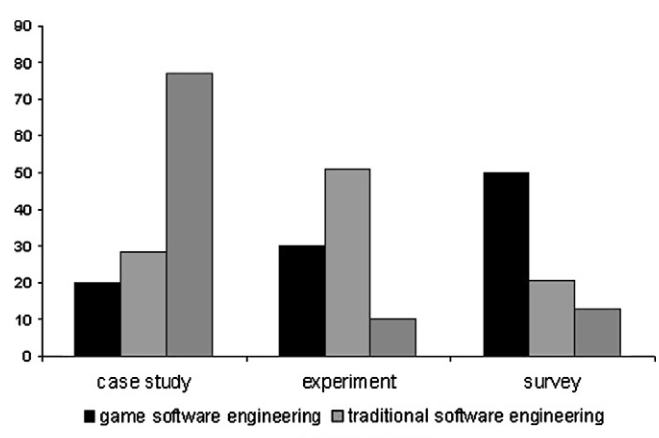

Fig. 6. Empirical research methods.

efforts that use surveys suggest that researchers attempt to seize the "know-how" experience that is accumulated in game industry.

In near future it is expected that case studies and experiments are going to appear more frequently in the research of software engineering for games, since game researchers will investigate deeper issues in a more thorough way, based on a greater variety of data that will be available to them.

5. Threats to validity

This section of the paper discusses possible threats to the validity of the paper and future work. In addition to that, studies that do not mention the word "game" in the title of the article have not been included in the primary studies set. Thus, it is possible for the search procedure to have missed a limited number of studies that refer to game development, but this is not referenced to its title. Moreover, the inclusion of workshop papers in the review might have altered the results of the review due to the nature of these studies, with respect to journals and conferences. Additionally, omitting a search of an SE indexing system such as SCOPUS, EI COMPENDIX or Web of Science means that papers in less known journals and conferences may have been omitted from the study. Finally, the employment of different classification schemas, might have led to different results. For example, if the research topics were not classified according to the *ACM Computing Classification System*, but with the phases described in the waterfall model, many topics would have been merged and the results would be different.

6. Conclusions and future work

This paper aimed at summarizing the current state of the art concerning scientific research on software engineering for computer games. In order to achieve this goal we performed a systematic literature review, which is considered to be the first step of the evidence-based research paradigm. The results confirmed that computer games are a fertile domain for applying software engineering technologies and that it is a research field that is growing more active year by year.

In addition to that, several research domains appear to be more active than others, mainly because of certain deviations of game development with respect to regular software development. The most active research topic appears to be *Requirements Engineering* and the main issues within such subject have proved to be the elicitation of game requirements according to user enjoyment heuristics, user emotions during game play and user interface attractiveness heuristics. Furthermore, research on the *Implementation* phase of the development, i.e. *Coding Tools* and *Programming Languages/Environments*, appears to be also very interesting, mostly due to of the fact that game programming is extremely demanding and the time period that is provided for game development is limited. However, *Testing/Debugging* research does not appear as intense as might be expected given the emphasis put on quality assurance in games development, but it has increased over the last year of the review.

Finally, the results of the review underline a certain lack of experiments and case studies in the field of game software engineering. This fact, although expected for a research subject in its early years, leads to the conclusion that there is room for improvement in the research evaluation process, through the employment of such research methods. More specifically, the plethora of open-source games that are available to researchers facilitate research based on case studies. Similarly, it is expected that more studies are going to use the data that are accumulated in game development magazines or evolution data of existing games as material for their validation.

As future work, a replication of the research in a next 5-year period will validate the expectations on the trend of game engineering research. Claims such as "*more elaborate empirical research methods are going to be employed*" and that "*a wider range of topics is going to be covered*" are going to be investigated. Finally, such future work will strengthen the fact that game engineering is a scientific domain rather than a soft skill topic with only a peripheral research activity.

Acknowledgements

The authors would like to acknowledge many valuable suggestions made by the anonymous reviewers with regard to the discussion on game requirements, game project management and game maintenance.

Appendix A. Studies included in the review

[S1] V. Abeele, B. Zaman, M. Abeele, The unlikeability of a cuddly toy interface: an experimental study of preschoolers' likeability and usability of a 3D game played with a cuddly toy versus a keyboard, in: Proceedings of the 2nd International Conference on Fun and Games, Springer-Verlag, Eindhoven, The Netherlands, 20–21 October 2008, pp. 118–131.

[S2] V. Alves, T. Camara, C. Alves, Experiences with mobile games product line development at meantime, in: Proceedings of the 12th International Software Product Line Conference, IEEE Computer Society, Washington, USA, 8–12 September 2008, pp. 287–296.

[S3] C. Alves, G. Ramalho, A. Damasceno, Challenges in requirements engineering for mobile games development: the meantime case study, in: Proceedings of the International Conference on Requirements Engineering (RE'07), IEEE Computer Society, Delhi, India, 15–19 October 2007, pp. 255–260.

[S4] Amanatiadou, I. van de Weerd, Extending the reference method for game production: a situational approach, in: 2009 Conference in Games and Virtual Worlds for Serious Applications, IEEE Computer Society, Coventry, UK, 23–24 March 2009, pp. 20–27.

[S5] Ampatzoglou, A. Chatzigeorgiou, Evaluation of object-oriented design patterns in game development, in: Information and Software Technology, vol. 49 (5), Elsevier, May 2007, pp. 445–454.

[S6] W. Barendregt, M.M. Bekker, D.G. Bouwhuis, E. Baauw, Identifying usability and fun problems in a computer game during first use and after some practice, International Journal of Human–Computer Studies 64 (9) (2006) 830–846.

[S7] D. Bendas, M. Myllyaho, Wireless games – review and experiment, in: Proceedings of the 4th International Conference on Product Focused Software Process Improvement, Springer-Verlag, Rovaniemi, Finland, 9–11 December 2002.

[S8] L. Bishop, M. Finch, M. Shantz, Designing a PC game engine, IEEE Computer Graphics and Applications, IEEE Computer Society 18 (1) (1998) 2–9.

[S9] J. Blow, Game Development – Harder Than You Think, Queue, Association for Computing Machinery, vol. 1 (10), February 2004, pp. 28–37.

[S10] M. Bond, R. Beale, What makes a good game?: using reviews to inform design, in: Proceedings of the 2009 British Computer Society Conference on Human–Computer Interaction, Association for Computing Machinery, Cambridge, United Kingdom, 1–5 September 2009, pp. 418–422.

[S11] D. Callele, E. Neufeld, K. Schneider, Requirements engineering and the creative process in the video game industry, in: Proceedings of the International Conference on Requirements Engineering (RE'05), IEEE Computer Society, Paris, France, 29 August–2 September 2005, pp. 240–252.

[S12] D. Callele, E. Neufeld, K. Schneider, Emotional requirements in video games, in: Proceedings of the International Conference on Requirements Engineering (RE'06), IEEE Computer Society, Minneapolis, MN, USA, 11–15 September 2006, pp. 292–295.

[S13] T.C.S. Cheah, K.W. Ng, A practical implementation of a 3-D game engine, in: Proceedings of the International Conference on Computer Graphics, Imaging and Vision: New Trends, IEEE Computer Society, Beijing, China, 26–29 July 2005, pp. 351–358.

[S14] F. Chehimi, P. Coulton, R. Edwards, Evolution of 3D mobile games development, in: Proceedings of the International Conference on Mobile Business (ICMB'05), IEEE Computer Society, Sydney, Australia, 11–13 July 2005, pp. 173–179.

[S15] J. Chen, Flow in games (and everything else), Communications of the ACM, Association for Computing Machinery 50 (4) (2007) 31–34.

[S16] H. Cho, J.S. Yang, Architecture patterns for mobile games product lines, in: Proceedings of the 2008 International Conference on Advanced Communication Technology (ICACT'08), IEEE Computer Society, Phoenix Park, Korea, 17–20 February 2008, pp. 118–122.

[S17] Y.J. Choi, Providing novel and useful data for game development using usability expert evaluation and testing, in: Proceedings of the 6th International Conference on Computer Graphics, Imaging and Visualization (CGIV'06), IEEE Computer Society, Tianjin, China, 11–14 August 2009, pp. 129–132.

[S18] S. Cornett, The usability of massively multiplayer online role-playing games: designing for new users, in: Proceedings of the Special Interest Group in Computer Human Interaction (SIGSHI'04), Association for Computing Machinery, Vienna, Austria, 24–29 April 2004, pp. 703–710.

[S19] M. Cutumisu, C. Onuczko, M. McNaughton, T. Roy, J. Schaeffer, A. Schumacher, J. Siegel, D. Szafron, K. Waugh, M. Carbonaro, H. Duff, S. Gillis, ScriptEase: a generative/adaptive programming paradigm for game scripting, in: Science of Computer Programming, vol. 67 (1), Elsevier, June 2007, pp. 32–58.

[S20] H. Desurvire, M Caplan, J.A. Toth, Using heuristics to evaluate the playability of games, in: Proceedings of the Special Interest Group in Computer Human Interaction (SIGCHI'04), Association for Computing Machinery, Vienna, Austria, 24–29 April 2004, pp. 1509–1512.

[S21] H. Desurvire, C. Wiberg, Game usability heuristics (PLAY) for evaluating and designing better games – the next iteration, in: Proceedings of the 3D International Conference on Online Communities and Social Computing, Springer-Verlag, San Diego, CA, USA, 19–24 July 2009, pp. 557–566.

[S22] H.B. Duh, V.H. Chen, C.B. Tan, Playing different games on different phones: an empirical study on mobile gaming, in: Proceedings of the International Conference on Human–Computer Interaction with Mobile Devices and Services (MobileHCI '08), Association for Computing Machinery, Amsterdam, Netherlands, 2–5 September 2008, pp. 391–394.

[S23] T.V. Evreinova, G. Evreinov, R. Raisam, Non-visual gameplay – making board games easy and fun, in: Proceedings of the 11th International Conference on Computers Helping People with Special Needs, Springer-Verlag, Linz, Austria, 9–11 July 2008, pp. 561–568.

[S24] E. Folmer, Component based game development – a solution to escalating costs and expanding deadlines, in: 10th International Symposium on Component Based Software Engineering (CBSE' 07), Springer-Verlag, Medford, MA, USA, 9–11 July 2007, pp. 66–73.

[S25] D. Frost, Ucigame, a java library for games, in: Proceedings of the Special Interest Group on Computer Science Education (SIGCHE'08), Association for Computing Machinery, Portland, OR, USA, 12–15 May 2008, pp. 310–314.

[S26] M. Furini, An architecture to easily produce adventure and movie games for the mobile scenario Computers in Entertainment, Association for Computing Machinery 6 (2) (2008) 1–16.

[S27] N. Graham, W. Roberts, Toward quality-driven development of 3D computer games, in: Proceedings of the 13th International Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '06), Springer-Verlag, Dublin, Ireland, 26–28 July 2006, pp. 248–261.

[S28] Ganchev, T. Kocsis, O. Papadopoulos, G. Fernandez-Aranda, F. Jimenez-Murcia, PlayMancer – a serious gaming 3D environment, in: Proceedings of the International Conference on Automated Solutions for Cross Media Content and Multichannel Distribution (AXMEDIS '08), IEEE Computer Society, Florence, Italy, 17–19 November 2008, pp. 111–117.

[S29] N. Graham, P. Curzon, G. Doherty, P. Palanque, R. Potter, C. Roast, S.P. Smith, Usability and computer games – working group report, in: Proceedings of the 13th International Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '06), Springer-Verlag, Dublin, Ireland, 26–28 July 2006, pp. 265–268.

[S30] J.H. Han, I. Kang, C. Hyun, J.S. Woo, Y.I. Eom, Multi-platform online game design and architecture, in: Proceedings of the 13th International Conference on Human–Computer Interaction (INTERACT' 05), Springer-Verlag, Rome, Italy, 12–16 September 2005, pp. 1116–1119.

[S31] G. Jacobs, B. Ip, Establishing user requirements: incorporating gamer preferences into interactive games design, in: Design Studies, vol. 26 (3), Elsevier, May 2005, pp. 243–255.

[S32] K. Jegers, Pervasive game flow: understanding player enjoyment in pervasive gaming, Computers in Entertainment, Association for Computing Machinery 5 (1) (2007).

[S33] Z. Jieyi, T. Min, T. Ruofeng, and D. Jinxiang, The platform of quick development of mobile 3D game, in: Proceedings of the 2008 International Symposium on Computer Science and Computational Technology (ISCSCT' 08), IEEE Computer Society, Shanghai, China, 20–22 December 2008, pp. 226–229.

[S34] C.M. Kanode, H.M. Haddad, Software engineering challenges in game development, in: Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations (ITNG' 09), IEEE Computer Society, Las Vegas, Nevada, USA, 27–29 April 2009, pp. 260–265.

[S35] S. Kaur, H.K. Kaura, Design and implementation of a multiplayer bluetooth game, in: Proceedings of the Second International Conference on Emerging Trends in Engineering & Technology, (ICETET' 09), IEEE Computer Society, Nagpur, Maharashtra, India, 16–18 December 2009, pp. 1080–1085.

[S36] H. Korhonen, J. Paavilainen, H. Saarenpää, Expert review method in game evaluations – comparison of two playability heuristic sets, in: Proceedings of the 13th International MindTrek Conference (MINDTREK' 09), Association for Computing Machinery, Tampere, Finland, 30 September–2 October 2009, pp. 74–81.

[S37] H. Korhonen, E.M.I. Koivisto, Playability heuristics for mobile multi-player games, in: Proceedings of the 2nd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA'07), Association for Computing Machinery, Perth, Australia, 5–7 September 2007, pp. 28–35.

[S38] H. Korhonen, E.M.I. Koivisto, Playability heuristics for mobile games, in: Proceedings of the International Conference on Human–Computer Interaction with Mobile Devices and Services (MobileHCI'09), Helsinki, Finland, 12–15 September 2006, pp. 9–16.

[S39] T. Larrabee, C.L. Mitchell, Gambit: a prototyping approach to video game design, IEEE Software, IEEE Computer Society 1 (4) (1984) 28–36.

[S40] S.H. Lee, G.H. Lee, H.H. Cho, D.H. Song, S.Y. Rhew, An empirical model of the game software development processes, in: Proceedings of the International Conference on Software Engineering Research, Management and Applications (SERA'06), IEEE Computer Society, 09–11 August 2006, pp. 371–377.

[S41] W.P. Lee, L.J. Liu, J.A. Chiou, A component-based framework to rapidly prototype online chess games for home entertainment, in: Proceedings of the International Conference on Systems, Man and Cybermetrics (SMC'06), IEEE Computer Society, Taipei, Taiwan, 8–11 October 2006, pp. 4011–4016.

[S42] H. Lee, T. Park, Design and implementation of an online 3D game engine, in: Proceedings of the 2004 International Conference on Computational Science and its Applications (ICCSA 2004), Springer-Verlag, Assisi, Italy, 14–17 May 2004, pp. 837–842.

[S43] M.W. Lee, J.M. Lee, Generation and control of game virtual environment, International Journal of Automation and Computing 4 (1) (2007) 25–29.

[S44] M. McNaughton, M. Cutumisu, D. Szafron, J. Schaeffer, J. Redford, D. Parker, ScriptEase: generative design patterns for computer role-playing games, in: Proceedings of the International Conference on Automated Software Engineering (ASE'04), IEEE Computer Society, Linz, Austria, 20–24 September 2004, pp. 88–99.

[S45] Mitchell, K. McGee, Designing storytelling games that encourage narrative play, in: Proceedings of the Second Joint International Conference on Interactive Digital Storytelling (ICIDS'09), Springer-Verlag, Guimarães, Portugal, 9–11 December 2009, pp. 98–108.

[S46] P. Moreno-Ger, J.L. Sierra, I. Martinez-Ortiz, B. Fernadez-Manjon, A documental approach to adventure game development, in: Science of Computer Programming, vol. 67 (1), Elsevier, June 2007, pp. 3–31.

[S47] L.M. Nascimento, E.S. de Almeida, S.R. de Lemos Meira, A case study in software product lines – the case of the mobile game domain, in: Proceedings of the 2008 Software Engineering and Advanced Applications Conference (SEAA'08), IEEE Computer Society, Parma, Italy, 3–5 September 2008, pp. 43–50.

[S48] Natapov, S.J. Castellucci, I.S. MacKenzie, ISO 9241-9 evaluation of video game controllers, in: Proceedings of Graphics Interface 2009 (GI'09), Association for Computing Machinery, Kelowna, British Columbia, Canada, 25–27 May 2009, pp. 223–230.

[S49] Nguyen, S.B. Wong, Design patterns for games, in: Proceedings of the Special Interest Group in Computer Science Education (SIGCSE'02), Association for Computing Machinery, Cincinnati, KY, USA, 27 February–3 March 2002, pp. 126–130.

[S50] T. Nummenmaa, J. Kuittinen, J. Holopainen, Simulation as a game design tool, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology (ACE' 09), Association for Computing Machinery, Athens, Greece, 29–31 October 2009, pp. 232–239.

[S51] E.M.I. Ollila, R. Suomela, J. Holopainen, Using prototypes in early pervasive game development, in: Computers in Entertainment, Association for Computing Machinery 6 (2) (2008).

[S52] E.B. Passos, J. Weslley, E. Walter, G. Clua, A. Montenegro, L. Murta, Smart composition of game objects using dependency injection, in: Computers in Entertainment, Association for Computing Machinery 7 (4) (2009).

[S53] M. Peltoniemi, Life-cycle of the games industry – the specificities of creative industries, in: Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era (MINDTREK 2008), Association for Computing Machinery, Tampere, Finland, 7–9 October 2008, pp. 54–58.

[S54] H. Peng, H. Song, Z. Zhang, Y. Chen, X. Zou, L. Xiao, A study on user experience of online games, in: Proceedings of the 2009 World Congress on Software Engineering, IEEE Computer Society, Xiamen, China, 19–21 May 2009, pp. 185–189.

[S55] Petrillo, M. Pimenta, F. Trindade, C. Dietrich, What went wrong – a survey of problems in game development, in: Computers in Entertainment, Association for Computing Machinery 7 (1) (2009).

[S56] A.M. Phelps, D.M. Parks, Fun and Games: Multi-Language Development, Queue Magazine, Association for Computing Machinery 1 (10) (2004) 46–56.

[S57] D. Pinelle, N. Wong, T. Stach, Heuristic evaluation for games: usability principles for video game design, in: Proceedings of the Special Interest Group in Computer Human Interaction (SIGCHI'08), Association for Computing Machinery, Florence, Italy, 5–10 April 2008, pp. 1453–1462.

[S58] D. Pinelle, N. Wong, T. Stach, Using genres to customize usability evaluations of video games, in: Proceedings of the 2008 Conference on Future Play: Research, Play, Share (Future Play'08), Association for Computing Machinery, Toronto, Ontario, Canada, 3–5 November 2008, pp. 129–136.

[S59] P. Piselli, M. Claypool, J. Doyle, Relating cognitive models of computer games to user evaluations of entertainment, in: Proceedings of the 4th International Conference on Foundations of Digital Games (FDG'09), Association for Computing Machinery, Orlando, Fl, USA, 26–30 April 2009, pp. 153–160.

[S60] E.M. Reyno, J.A. Carsí Cubel, Automatic prototyping in model-driven game development, IN: Computers in Entertainment, Association for Computing Machinery 7 (2) June 2009.

[S61] El Rhalib, M. Merabti, Y. Shen, Improving game processing in multithreading and multiprocessor architecture, in: Proceedings of the 1st International Conference on Edutainment, Springer-Verlag, Hangzhou, China, 16–19 April 2006, pp. 669–679.

[S62] T. Roden, I. Parbery, Designing a narrative-based audio only 3D game engine, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology (ACE'05), Association for Computing Machinery, Valencia, Spain, 15 June 2005, pp. 274–277.

[S63] T. Roden, I. Parberry, D. Ducrest, Toward mobile entertainment: a paradigm for narrative-based audio only games, in: Science of Computer Programming, vol. 67 (1), Elsevier, June 2007, pp. 76–90.

[S64] R.B. Rookhuiszen, M. Theune, Playful vs. serious instruction giving in a 3D game environment, in: Entertainment Computing, vol. 1 (2), Elsevier, April 2009, pp. 95–104.

[S65] Russell, A.F. Donaldson, P. Sheppard, Tackling online game development problems with a novel network scripting language, in: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games (NETGAMES' 08), Association for Computing Machinery, Worcester, Massachusetts, USA, 21–22 October 2008, pp. 85–90

[S66] J.L. González Sánchez, N. Padilla Zea, F. L. Gutiérrez, From usability to playability – introduction to player-centred video game development process, in: Proceedings of the 1st International Conference on Human Centred Design, Springer-Verlag, San Diego, CA, USA, 19–24 July 2009, pp. 65–74.

[S67] W. Scacchi, Free and open source development practices in the game community, in: Software Magazine, vol. 21 (1), IEEE Computer Society, January/February 2004, pp. 59–66.

[S68] R. Schaap, R. Bidarra, Towards emotional characters in computer games, in: Proceedings of the 7th International Conference on Entertainment Computing (ICEC'08), Springer-Verlag, Pittsburgh, USA, 25–27 September 2008, pp. 167–172.

[S69] T. Sellar, User experience in interactive computer game development, in: Proceedings of the 6th Asia Pacific Conference on Computer Human Interaction (APCHI'04), Springer-Verlag, Rotorua, New Zealand, 29 June–2 July 2004, pp. 675–681.

[S70] P. Stacey, J. Nandhakumar, Managing projects in a games factory: temporality and practices, in: Proceedings of the Hawaii International Conference on System Sciences (HICSS'05), IEEE Computer Society, Big Island, Hawaii, USA, 03–06 January 2005, pp. 1–10.

[S71] P. Stacey, J. Nandhakumar, Opening up to agile games development, in: Communications of the ACM, Association for Computing Machinery 51 (1) (2008) 143–146.

[S72] M.J. Taylor, D. Gresty, M. Baskett, Computer game – flow design, in: Computers in Entertainment, Association for Computing Machinery 4 (1) (2006).

[S73] T. Tijs, D. Brokken, W. IJsselsteijn, Creating an emotionally adaptive game, in: Proceedings of the 7th International Conference on Entertainment Computing (ICEC'08), Springer-Verlag, Pittsburgh, USA, 25–27 September 2008, pp. 122–133.

[S74] M.Q. Tran, R. Biddle, Collaboration in serious game development: a case study, in: Proceedings of the Conference on Future Play: Research, Play, Share (FuturePlay'08), Association for Computing Machinery, Toronto, Ontario, Canada, 03–05 November 2008, pp. 49–56.

[S75] M. van Lent, W. Swartout, Games: once more, with feeling, computer magazine, vol. 40 (8), IEEE Computer Society, August 2007, pp. 98–100.

[S76] J. Waldo, Scaling in games and virtual worlds, in: Communications of the ACM, Association for Computing Machinery 51 (8) (2008) 38–44.

[S77] D. Weibel, B. Wissmath, S. Habegger, Y. Steiner, R. Groner, Playing online games against computer- vs. human-controlled opponents: effects on presence, flow, and enjoyment, in: Computers in Human Behavior, vol. 24 (5), Elsevier, September 2008, pp. 2274–2291.

[S78] R. Wetzel, R. McCall, A.K. Braun, W. Broll, Guidelines for designing augmented reality games, in: Proceedings of the 2008 Conference on Future Play: Research, Play, Share (FuturePlay'08), Association for Computing Machinery, Toronto, Ontario, Canada, 3–5 November 2008, pp. 173–180.

[S79] W. White, C. Koch, J. Gehrke, A. Demers, Better Scripts, Better Games, Queue Magazine, Association for Computing Machinery 6 (7) (2008) 18–25.

[S80] W. White, B. Sowell, J. Gehrke, A. Demers, Declarative processing for computer games, in: Proceedings of Sandbox 2008: An ACM SIGGRAPH Videogame Symposium (Sandbox'08), Association for Computing Machinery, Los Angeles, USA, 9–10 August 2008, pp. 23–30.

[S81] S. Xu, V. Rajlich, Empirical validation of test-driven pair programming in game development, in: Proceedings of the International Conference on Computer and Information Science (ICIS'06), IEEE Computer Society, Honolulu, USA, 10–12 July 2006, pp. 500–505.

[S82] N. Yannakakis, J. Hallam, Capturing player enjoyment in computer games, in: Advanced Intelligent Paradigms in Computer Games, Springer-Verlag, June 2007, pp. 175–201.

[S83] W. Zhang, D. Han, T. Kunz, K.M. Hansen, Mobile game development: object orientation or not, in: Proceedings of the International Computer Software and Applications Conference (COMPSAC'07), IEEE Computer Society, Beijing, China, 24–27 July 2007, pp. 601–608.

[S84] W. Zhang, D. Han, T. Kunz, Object-orientation is evil to mobile game - experience from industrial mobile RPGs, in: Proceedings of the 3rd International Conference on Embedded Software and Systems (ICESS'07), Springer-Verlag, Daegu, Korea, 14–16 May 2007, pp. 1–12.

Appendix B. Study dataset

	Publisher	Citation_type	Topic	Country	Year	Method	Approach
S01	Springer	Conference	Requirements/Specification	Belgium	2008	Experiment	Emprical
S02	IEEE	Conference	Requirements/Specification	Brazil	2007		Exploratory
S03	IEEE	Conference	Management	Brazil	2008		Exploratory
S04	IEEE	Conference	Management	Netherlands	2009		Exploratory
S05	Elsevier	journal	Design	Greece	2007		Exploratory
S06	Elsevier	Journal	Testing/Debugging	Netherlands	2007	Experiment	Emprical
S07	Springer	Conference	General	Finland	2002		Descriptive
S08	IEEE	Journal	Design	USA	1998		Descriptive
S09	ACM	Journal	General		2004		Descriptive
S10	ACM	Conference	Requirements/Specification	UK	2009		Exploratory
S11	IEEE	Conference	Requirements/Specification	Canada	2005	Case Study	Emprical
S12	IEEE	Conference	Requirements/Specification	Canada	2006		Descriptive
S13	IEEE	Conference	Coding Tools & Techniques	Malaysia	2005		Descriptive
S14	Springer	Journal	General	UK	2008		Descriptive
S15	ACM	Journal	Requirements/Specification	USA	2007		Exploratory
S16	IEEE	Conference	Software Reuse	Korea	2008		Exploratory
S17	IEEE	Conference	Testing/Debugging	Korea	2009		Exploratory
S18	ACM	Conference	Requirements/Specification	USA	2004		Exploratory
S19	Elsevier	Journal	Coding Tools & Techniques	Canada	2007		Exploratory
S20	ACM	Work in progress	Requirements/Specification	USA	2004		Exploratory
S21	Springer	Conference	Testing/Debugging	USA	2009	Survey	Emprical

(*continued on next page*)

Appendix B (*continued*)

	Publisher	Citation_type	Topic	Country	Year	Method	Approach
S22	ACM	Work in progress	Requirements/Specification	Singapore	2008	Survey	Emprical
S23	Springer	Conference	Requirements/Specification	Finland	2008		Exploratory
S24	Springer	Conference	Software Architecture	USA	2007		Exploratory
S25	ACM	Conference	Programming Environments	USA	2008		Descriptive
S26	ACM	Journal	Software Architecture	Italy	2008		Exploratory
S27	Springer	Work in progress	General	Canada	2007		Descriptive
S28	IEEE	Conference	Programming Environments	Italy	2008		Descriptive
S29	Springer	Work in progress	Testing/Debugging	Canada	2007		Descriptive
S30	Springer	Work in progress	Requirements/Specification	Korea	2005		Descriptive
S31	Elsevier	Journal	Requirements/Specification	UK	2005	Survey	Emprical
S32	IEEE	Conference	Testing/Debugging	Sweden	2008	Survey	Emprical
S33	IEEE	Conference	Programming Environments	China	2008		Descriptive
S34	IEEE	Conference	Management	USA	2009		Descriptive
S35	IEEE	Conference	Design	India	2009		Exploratory
S36	ACM	Conference	Requirements/Specification	Finland	2006		Exploratory
S37	ACM	Conference	Requirements/Specification	Finland	2007		Exploratory
S38	ACM	Conference	Requirements/Specification	Finland	2009		Exploratory
S39	IEEE	Journal	Requirements/Specification	USA	1994		Descriptive
S40	Springer	Conference	Programming Environments	Korea	2004		Descriptive
S41	IEEE	Conference	Software Architecture	Taiwan	2006		Exploratory
S42	IEEE	Conference	Management	Korea	2006	Survey	Emprical
S43	Springer	Journal	Programming Environments	Korea	2007		Descriptive
S44	IEEE	Conference	Programming Environments	Canada	2004		Exploratory
S45	Springer	Conference	Requirements/Specification	Singapure	2009		Exploratory
S46	Elsevier	Journal	Coding Tools & Techniques	Spain	2007		Exploratory
S47	IEEE	Conference	Management	Brazil	2008	Case Study	Emprical
S48	ACM	Conference	Metrics	Canada	2009	Survey	Emprical
S49	ACM	Conference	Design	USA	2002		Exploratory
S50	ACM	Conference	Requirements/Specification	Finland	2009		Descriptive
S51	ACM	Journal	Requirements/Specification	Finland	2008		Exploratory
S52	ACM	Journal	Software Reuse	Brazil	2009		Exploratory
S53	ACM	Conference	Management	Finland	2008		Descriptive
S54	IEEE	Conference	Requirements/Specification	China	2009	Survey	Emprical
S55	ACM	Journal	Management	Brazil	2009	Survey	Emprical
S56	ACM	Journal	Design Tools & Techniques	USA	2004		Descriptive
S57	ACM	Conference	Testing/Debugging	USA	2008		Exploratory
S58	ACM	Conference	Requirements/Specification	USA	2008	Survey	Emprical
S59	ACM	Conference	Requirements/Specification	USA	2009		Exploratory
S60	ACM	Journal	Requirements/Specification	Spain	2009		Exploratory
S61	Springer	Conference	Requirements/Specification	UK	2006	Experiment	Emprical
S62	ACM	Conference	Programming Environments	USA	2005		Descriptive
S63	Elsevier	Journal	Programming Environments	USA	2007		Descriptive
S64	Elsevier	Journal	Requirements/Specification	Netherlands	2009		Exploratory
S65	ACM	Conference	Coding Tools & Techniques	Scotland	2008		Exploratory
S66	Springer	Conference	Requirements/Specification	Spain	2009		Exploratory
S67	IEEE	Journal	Management	USA	2004		Descriptive
S68	Springer	Conference	Requirements/Specification	Netherlands	2009		Exploratory
S69	Springer	Conference	Testing/Debugging	New Zealand	2004		Descriptive
S70	IEEE	Conference	Management	UK	2005		Exploratory
S71	ACM	Journal	Coding Tools & Techniques	USA	2008		Exploratory
S72	ACM	Journal	Requirements/Specification	UK	2006		Exploratory
S73	Springer	Conference	Requirements/Specification	Netherlands	2009	Experiment	Emprical
S74	ACM	Conference	Management	Canada	2008	Survey	Emprical
S75	IEEE	Journal	Requirements/Specification	USA	2007		Descriptive
S76	ACM	Journal	Requirements/Specification	USA	2008		Exploratory
S77	Elsevier	Journal	Requirements/Specification	Switzerland	2008	Experiment	Emprical
S78	ACM	Conference	Requirements/Specification	Germany	2008		Exploratory
S79	ACM	Journal	Coding Tools & Techniques	USA	2009		Exploratory
S80	ACM	Conference	Coding Tools & Techniques	USA	2008		Exploratory
S81	IEEE	Conference	Coding Tools & Techniques	Canada	2006	Case Study	Emprical
S82	Springer	Conference	Metrics	UK	2005		Exploratory
S83	IEEE	Conference	Coding Tools & Techniques	China	2007	Experiment	Emprical
S84	Springer	Conference	Coding Tools & Techniques	China	2007	Case Study	Emprical

References

[1] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (4) (2007) 571–583.
[2] D. Budgen, X. Turner, P. Brereton, B. Kitchenham, Using mapping studies in software engineering, in: Proceedings of Psychology of Programming Interest Group, Lancaster University, 2008, pp. 195–204.
[3] K.Y. Cai, D. Card, An analysis of topics in software engineering – 2006, Journal of Systems and Software 81 (6) (2008) 1051–1058.
[4] C.E. Crooks II, Awesome 3D Game Development, Charles River Media, Hingham, MA, 2004.
[5] T. Dyba, T. Dingsoyr, Empirical studies of agile software development: a systematic review, Information and Software Technology 50 (9–10) (2008) 833–859. August.
[6] T. Dyba, B.A. Kitchenham, M. Jorgensen, Evidence based software engineering for practitioners, Software Magazine, IEEE Computer Society 22 (1) (2005) 58–65.
[7] R.L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software Technology 44 (8) (2002) 491–506.
[8] B. Kitchenham, in: Procedures for Undertaking Systematic Literature Reviews, Joint Technical Report, Computer Science Department, Keele University.
[9] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering – a systematic literature review, Information and Software Technology 51 (1) (2009) 7–15.
[10] B. Kitchenham, T. Dyba, M. Jorgenson, Evidence based software engineering, in: Proceedings of the International Conference on Software Engineering (ICSE'04), IEEE Computer Society, Edinburgh, Scotland, 23–28 May 2004, pp. 273–281.
[11] B. Kitchenham, L. Pickard, S.L. Pfleeger, Case studies for method and tool evaluation, Software Magazine, IEEE Computer Society 12 (4) (1995) 52–62.
[12] M. Lewis, J. Jacobson, Game engines in scientific research, Communications of the ACM, Association for Computing Machinery 45 (1) (2002) 27–31.
[13] M. McShaffry, Game Coding Complete, Paraglyph Press, AZ, USA, 2003.
[14] T.M. Rhyne, P. Doenges, B. Hibbard, H. Pfister, N. Robins, The impact of Computer Games on scientific & information visualization: if you can't beat them, join them, in: Panel of the International Conference on Visualization (Vis'00), IEEE Computer Society, Salt Lake City, Utah, USA, pp. 519–521.
[15] S.K. Sowe, L. Angelis, I. Stamelos, I. Manolopoulos, Using repository of repositories (RoRs) to study the growth of F/OSS projects: a meta-analysis research approach, in: Proceedings of the International Conference on Open Source Systems (OSS'07), Springer, Limerick, Ireland, 11–14 June 2007, pp. 147–160.
[16] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering, Kluwer Academic Publishers, Boston/Dordrecht/London, 2000.
[17] M. Zyda, Creating a science of games, Communications of the ACM, Association for Computing Machinery 50 (7) (2007) 27–29.