A Report on Select Research Opportunities in Requirements Engineering for Videogame Development

David Callele
Department of Computer Science
University of Saskatchewan
Saskatchewan, Canada
callele@cs.usask.ca

Eric Neufeld
Department of Computer Science
University of Saskatchewan
Saskatchewan, Canada
neufeld@cs.usask.ca

Kevin Schneider
Department of Computer Science
University of Saskatchewan
Saskatchewan, Canada
kas@cs.usask.ca

***Abstract*—Requirements engineering research in videogame development is a relatively new field. This paper summarizes our work in the area then reviews the prior academic and trade press. We then present brief overviews of numerous research opportunities in the videogame domain that arose from our research program yet remain open questions with the goal of providing other researchers with a concise directory of interesting research possibilities.**

Keywords: Research opportunities, videogames, games, requirements

I. Introduction

Over the last 6 years, we have investigated many aspects of the videogame development process, with an emphasis on attempting to understand why so many videogame development projects experience significant production challenges, if not complete failure. Our research focused on the transition between the preproduction phase (the design of the videogame) and the production phase (the building of the designed videogame) and ways to improve this transition.

In this work we begin with a summary of our research program then review the related literature. We then summarize the opportunities presented for further research in our published work and supplement the summary with other opportunities extracted from our research notes. Our goal in this review is not to make a scientific contribution based upon specific results. Instead, our goal is to provide a concise summary of the research opportunities we have identified in the hope that it may motivate other researchers and practitioners to address these or related issues in this domain.

II. Research Program Summary

The work began with a study [5] of PostMortem reports from GameDeveloper magazine [1], [16]. A number of issues were noted, the most significant to our interest was the communications challenges observed between preproduction and production, and the problems generated by this disconnect. This work was followed up by Petrillo *et al.* [25], [26] who performed finer-grain analyses of the issues, particularly within the production phase.

For pragmatic reasons, our research program in this area proceeded in a breadth-first rather than a depth-first manner, exploring the domain and identifying major elements of the domain rather than focusing on a specific topic within the domain. However, the research did progressively reveal the principle contribution of the effort, the identification and definition of experience requirements for this domain. Experience requirements are a model for the elements that compose the experience, a framework that provides guidance for expressing the experience, and an exemplary process for the elicitation, capture, and negotiation of the requirements for an experience. Experience requirements capture the designer's intent for the user experience; they represent user experience goals for the artifact and constraints upon the implementation. Experience requirements are evolutionary in intent – they incrementally enhance and extend existing practices in a relatively lightweight manner using language and representations that are intended to be mutually acceptable to preproduction and to production.

The research program also generated a number of interesting other venues of research that illustrate the breadth of the domain but we did not have the resources to pursue them beyond the published work. A summary of these opportunities is presented in Section V.

The first type of experience requirement, the emotional requirement [6], [8], was introduced as a mechanism to address some of the communications issues between preproduction and production. Emotional requirements are used to capture the game designer's intent for the player emotional experience in a manner that provides appropriate guidance for the production effort and which includes visualizations such as the emotional intensity map and emotion timeline. The interactions between emotional requirements and security requirements were investigated further [9] and we showed that emotional requirements can dominate security requirements in some cases. In other words, stakeholder needs addressed by security requirements may be subjugated to stakeholder needs captured by emotional requirements such as player satisfaction. We also proposed a justice-system inspired technique for runtime requirements negotiation [7].

We revisited industry [10], [11] and learned that specific production issues were not yet addressed by emotional requirements – in particular, guidance as to *how* and *where* the emotion was to be induced was needed by the production team. The visualizations for emotional requirements were extended to provide greater production guidance [11] and Smith's work [29] was incorporated to add narrative descriptions of the actions that the player would take and the goals that the player would formulate in response to the emotional stimuli. The formalism for emotional requirements was extended to incorporate the (cue, action, goal) information from Smith and to include validation guidance from the game designer.

Physualization [4], an enhanced form of paper-prototyping incorporating a wide variety of common office materials into the process, was introduced as a requirements engineering tool that could be used in the domain as an alternative to investment in developing domain-specific software tools.

Emotional requirements were then generalized to experience requirements [13], requirements that captured and represented three principle experience types: emotional, gameplay (including cognitive and mechanical subtypes) and sensory (including visual, auditory, and haptical subtypes). A more detailed exploration of cognitive gameplay requirements was performed and an initial model was proposed [12]. In general, experience requirements could be considered the requirements engineering counterpart to the work of Marc Hassenzahl [1].

III. Prior Work (Academic Research)

We began our research program with an initial academic literature review that focused on identifying prior work in videogame production (the practice of managing the development (production) of videogames). At the start of the research project (2004-2005) we were unable to locate any significant academic research work in the area of videogame production – the closest areas of academic research were film production and ludology (the study of games as entities unto themselves). While significant work appeared in both related domains, we were unable to identify specific guidance that would be useful in the requirements engineering or software engineering aspects of videogame production.

In parallel with the start of our work, videogame development consultant Jonathon Blow pointed out that videogame development is "harder than we think" [3], specifically drawing attention to "problems due to highly domain-specific requirements. [p.29]". In the examples given in this work, the requirements are related to the skill sets needed by the development team, rather than the requirements for the videogame itself.

Other researchers investigated software engineering aspects of the domain. For example, Prayaga [27] reports on mapping the seven basic software engineering principles identified by Ghezzi *et al.* [19] (Rigor and Formality, Separation of Concerns, Modularity, Abstraction, Anticipation of Change, Generality, Incrementality) to the teaching of videogame development. They report success using game development as a motivating factor for students learning software engineering. Given that the courses were focused on videogame development rather than videogame design, effort on the creative aspects of preproduction and the game design document were highly constrained. The creative aspects were captured in the initial storyline, story boarding and a sketch or concept art of their videogame. These creative materials were then formalized as a gameplay flowchart. In essence, the students produced a minimalist game design document from their preproduction efforts.

Furtado *et al.* [18] look at the application of Domain Analysis to Digital Game Software Product Line (SPL) development. They advocate the creation of a "game design vision" that guides the Domain Analysis – explicitly identifying the elements that will, and will not, be part of the SPL. They do not appear to be aware of our work and refer to experience requirements as "psychochemical requirements". They make fundamentally the same observations we made earlier [5]:

> The development of a digital game is not a direct outcome of user requirements or business needs, which may not even exist. Games are not focused on solving user problems, but to entertain them. On the other hand, psychochemical requirements such as immersion, surprise, delight and nostalgia are present in digital games. This way, traditional Requirements Engineering cannot be applied as is to game development. For example, the well-known concept of use cases, with well-defined roles, flows and input/output artifacts may not make sense to a game development process. Game Design documents and experimentation processes are more realistic in this area. [p. 318]

They further note that the experience aspects that were the focus of our work (or alternatives) are a necessary element for SPL planning efforts.:

> *Unless problem domain psychochemical features are refined and made more concrete, modeling them does not seem to be useful to the game SPL, due to their subjective and cross-discipline nature.* Typical problem domain feature examples for that are "appealing physics" and "nostalgia", which can only benefit from Domain Engineering processes if their understanding and underlying requirements are really made specific. [p.323]

Kanode and Haddad [20] discuss software engineering challenges in videogame development, building on our early work and the follow-on work by Petrillo *et al.* [25], [26].

[1] http://hassenzahl.wordpress.com/

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

They advocate the use of agile methods during preproduction to help ensure that the resulting gameplay experience is as intended. They identify the need to put SE principles into practice and the need for sound project management skills. Contrary to apparent popular belief, they identify that agile development methodologies do not appear to deliver the expected benefits when a thorough preproduction process has been followed.

IV. Prior Work (Trade Press)

Given the dearth of academic research into videogame production at the start of this research program, we chose to survey the trade press, specific to the videogame industry, under the hypothesis that a general statement of industrial practices could be explicitly located in the materials or, if necessary, a general synthesis of reported industrial practices could be performed as part of the research effort.

We learned that the videogame development effort is typically modeled (in the trade) as iterations over a sequential, two phase process composed of preproduction and production efforts. In preproduction, the game design is developed and captured in the Game Design Document (GDD) or equivalent, a mechanism used to capture and communicate the designer's vision for the production (implementation) phase. The purpose of this artifact strongly resembles the purpose of a requirements specification in traditional software engineering and we initially guided our research from this perspective.

There are numerous iterations within and across the preproduction and production phases but the essentially linear, waterfall *model* (but not necessarily practice [24]) for the two phases derives heavily from the film industry and appears to be consistent throughout the videogame industry.

The application of various agile methodologies within the videogame industry has increased the visibility of the iterations but does not appear to have fundamentally changed the model: plan what to create (preproduction) then create what was planned (production). For example, a movie production may have (a target of) a single iteration between preproduction and production, essentially following a waterfall model in the large, while iterating in the small (*e.g.* shooting and re-shooting a specific scene). A large videogame development effort may follow a similar model, capturing the broad description in a GDD while leaving the details to be elaborated in subsequent iterations.

We note that agile methods have become more prevalent in industrial practice over the course of the research program. Our observations have been that the practice of generating a (near) complete game design document appears to be on the wane, at least in those groups with whom we work. Petrillo *et al.* [24] continued to extend our post-mortem analysis work to determine what, if any, agile practices have been adopted by the videogame development community. A number of practices commonly associated with SCRUM and XP were identified and correlations between these practices and those reported by development teams were identified.

While we do not have enough data points from which to generalize, there appear to be a number of factors impacting the GDD. One factor is an industry shift toward games developed for mobile platforms. Many of these videogames are self-funded and self-published through avenues such as the Apple App Store and Android Market. The average revenue generated by such games is reportedly quite small – we have had values in the range of $3,500 to $10,000 reported to us by industry participants. Assuming that these values are valid then investment in preproduction is likely to be as low as possible and investments in (more formal) documentation structures such as the GDD (or other traditional software engineering documents) are less likely to be made.

Another factor is time-to-market pressure. A small studio generating a mobile game must create content with the expectation of very limited "shelf life"; the attention focused on each game is soon diverted to the "next new thing".

A. Internet Review

Materials available on the Internet were reviewed first, under the assumption that the most current material would be available by electronic publication. We expected that the most focused and pragmatic explanations would also be in this form since there was no need to meet the demands of peer-review or a publisher requiring sufficient content to create a commercial product. Our review of the electronic media showed that many authors agree upon what should be in the game design document but not necessarily what needs to be in the GDD to support production efforts. Further, (effectively) no author addressed how the material should be represented in the game design document to support the needs of the production team. For example, a number of templates are available for download, but none of them provide substantive details as to how to represent a videogame design in a manner that can be used to direct or manage a production effort.

The review of materials on the Internet provided less guidance than expected. We then turned our attention to books from the videogame trade press, reviewed for the practitioner perspective on the game design document and captured the requirements for the production effort.

B. Trade Press Review

Michael, in *The Indie Videogame Development Survival Guide* [23] describes the necessity for a game design document and associated project plan but provides few details. "The design details section describes the game in its entirety, from beginning to end. Cover all relevant topics in enough detail to be useful later on." [p. 84] However, Appendix E in the same book does provide a sample game design document [pp. 349-370] that captures only the minimum amount of

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

information that the author has deemed necessary to capture their vision.

Meigs, in *Ultimate Game Design* [22] provides a review of many of the elements that compose a game design, principally from the perspective of a level designer. Approximately one page of content is dedicated to the game design document [pp. 316-317]. Meigs notes that game design document templates are available on the Internet, that game design documents vary in size from 20 pages to over 500 pages, and advocates the use of HTML based documentation.

Crawford, in *Chris Crawford On Game Design* [15] provides no direct guidance with respect to videogame production but significant guidance with respect to videogame design and, by implication, what should be captured in a game design document. Crawford draws attention to the videogame industry infatuation with movie production: "Storyboards are de rigeur among many games people, even though they really don't add anything to the design process. The word "cinematic" seems to be more common in game design discussions than "interactivity", even though the latter is central to game design and the former is peripheral." [p.182]

Koster, in *A Theory of Fun for Game Design* [21] provides no direct guidance with respect to videogame production either, but significant guidance with respect to videogame design and again, by implication, what should be captured in a game design document. Koster proposes a theory that player engagement in the game, and their long term satisfaction, is strongly related to the learning process.

Rollings and Adams, in *Andrew Rollings and Ernest Adams On Game Design* [28] provide minimal guidance with respect to videogame production but significant guidance with respect to videogame design and, by implication, what should be captured in a game design document. Appendix A provides samples of two game design related documents, the high-concept document and the design treatment; they also provide a reference to a design script or design bible template from Chris Taylor. The design script [pp. 569-587] is equivalent to the document we refer to here as the game design document.

Taylor provides a *Game Design Document Template*, a Microsoft Word document template that can be downloaded from various Internet locations[2]. Major topics covered by the template include a game overview, a description of the feature set and of the game world, a description of the game engine(s), characters, the user interface, sounds, weapons, play modes, etc. As appears to be typical of the domain, the template provides guidance on what to capture, but little or no guidance on how to capture the information or to what level of detail to capture the information.

[2]Downloaded from http://www.runawaystudios.com/articles/ctaylordesigntemplate.doc

Flynt, in *Software Engineering for Game Developers* [17] provides an introduction to classic software engineering topics set in the context of videogame development. Flynt provides the most detailed treatment of software engineering topics, explicitly addressing requirements with their own chapter. This book has a greater focus on production than on preproduction and it includes a detailed set of design documents on the included CD-ROM. Flynt advocates a development methodology based on stripes, which are equivalent to components, cards or user-stories in agile development methodologies. Again, there is little or no guidance on how to capture the information or to what level of detail to capture the information.

Bethke, in *Game Development and Production* [2] provides the most detailed reference with respect to preproduction and production topics. This work covers a broad range of materials in a relatively compact manner. As a result, no topic receives an extensive treatment and the author appears to expect that the reader is an experienced developer, or an experienced project manager. Bethke describes a process composed of phases that include a game concept document, a game vision document, a game design document and a technical design document. The work provides practical guidance as to what needs to be captured, including (in some cases) to what level of detail. Bethke advocates the use of UML for capturing the game designer's vision, at least in the technical design document and explicitly addresses requirements issues, but in a somewhat superficial manner.

The works of Flynt and Bethke were most relevant to our research program but neither address issues of acceptance of their proposed methodologies by members of the preproduction team. Their proposals are solidly rooted in traditional software process paradigms and, as such, may not be acceptable, or even comprehensible, to some members of the preproduction team. For the most rapid introduction to the domain, we recommend the works of Bethke and Koster.

This review of established practices in videogame development, as expressed in electronic publication and the trade press, established a baseline for videogame development practices. The investigation revealed that there was general agreement as to what the preproduction team should generate and capture within the game design document. However, there was little evidence that this agreement was at any other than a high level abstraction – *e.g.* describe the gameplay, describe the world, *etc.* Definitions for the necessary information, and techniques for capturing the necessary information, were either vague or they relied upon traditional software engineering methodologies such as UML. Further, we did not observe concrete evidence that the design of the player experience was explicitly rooted in psychological principles which may also contribute to the perceived erratic success of the design efforts.

We do not advocate that there is a single *correct* way to handle the transition from preproduction to production.

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

However, we do note that the game design document descriptions exhibit characteristics of a *push* of information that the preproduction team deems important to the production team. We do not observe similar evidence that the game design document contains the information that the production team would *pull* from preproduction because production deems it important. In other words, the traditional GDD appears to serve the producers of the document more than the consumers of the document.

V. Research Opportunities Identified In Our Work And Research Notes

In this section we present a selection of research opportunities identified during and within our work, organized by topic. Note that this is not a general review of research opportunities in the field, these opportunities identified during our work are summarized here to provide a concise starting point for researchers who may wish to perform investigations in the indicated directions. We employ a mix of descriptive statement and rhetorical question to present the opportunities and assume that the reader is conversant with the domain terminology utilized in the works cited earlier.

Process management Preproduction is an overhead activity. If it works well, then production can proceed with greater certainty and, hopefully, with less wasted effort than may otherwise occur. But how much preproduction is *just enough*? Is there a role for some form of agile preproduction and if there is, what are the benefits and tradeoffs?

An in-depth field investigation of established software engineering and production management practices at studios with established track records would better situate the academic research within the domain. Of particular import are means for identifying critical production elements and determining an appropriate balance between tightly-specified and loosely-specified elements. Formalized design review techniques and practices for gameplay, media assets, storyline, *etc*. could make a significant contribution to risk reduction. There appear to be opportunities for reuse in media assets as well as software assets. Can reuse planning be an effective part of preproduction?

Involving production personnel in the preproduction process may lead to more efficient development or it may lead to reduced creativity; further investigation is needed to quantify the effects and tradeoffs. Industrial case studies of return on investment for experience requirements and other software engineering techniques would increase the relevance of the research to practitioners.

Explicit guidance for all aspects of gameplay testing is needed, particularly to determine whether the intended gameplay experiences are actually induced in the player. Gameplay testing theory and practice generally assumes the videogame engine exists before testing begins. The requirement for testing on the actual engine can increase risk be forcing the testing team to wait for the engine to be (sufficiently) complete. Guidance for gameplay testing before the engine is available could reduce the risk that the engine works as expected but the gameplay does not entertain, at least in the player's perception. Finally, we have observed that experience requirements are often emergent from design reviews and gameplay testing. Means for addressing emergent experience requirements are still to be addressed.

A general evaluation of, and guidance for, the design of experiments for this field is needed. Our field experience has shown a significant lack of perceived credibility for software engineering research; credible research that is meaningful to academia *and* to practitioners is needed. It remains an open question whether industrially credible research in this area can be performed within the constraints of academia.

Requirements engineering in games The definitions for functional requirements and non-functional requirements in the context of a game remain open issues. For example, a requirement for a gameplay element such as "an attacking monster" is functional in the context of the game experience but would probably be considered non-functional in the context of the software implementation of the game engine. Experience requirements attempt to address these issues to some degree but there may be a need for our fundamental definitions for requirements to change to include contextual information such as context or point-of-view. If the definitions do change, are common requirements tasks such as modeling, negotiation, and triage affected?

Experience requirements Our work has addressed only some aspects of experience requirements [10], [12], [13]. Opportunities exist for lightweight methods for cognitive gameplay and mechanical gameplay contributions, particularly in puzzle capture, representation, and validation efforts. The contribution of software tools that support experience requirements, particularly as extensions to storyboards, requires investigation. These tools could investigate the utility and communication effectiveness of visualizations such as emotional intensity maps and emotion timelines and could address complexity issues for multiple, simultaneous stimuli within a scenario.

There are many other opportunities to develop tools to support both preproduction and production efforts. Of particular interest are tools that support traceability (although the degree of traceability that is needed is unknown) and capture rationale (for making design choices) without unduly disturbing the creative process. Other tools could provide support for early evaluation of the development effort (*e.g.* computational and rendering complexity) associated with a given requirement: creativity without a reality check on production constraints can lead to features (and chains of dependencies) that are not technically feasible.

Experience requirements and interactions with other requirement types Our work [7], [9] has shown that emotional requirements could be used pro-

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

actively to enhance security requirements by capturing and modeling threat identification (such as griefers (those who play a game with the intent of interfering with the gameplay experience of others) and other destructive (motivated, but hostile) stakeholders) and motivation. Emotional requirements may also be able to be used to help prioritize security requirements.

Security requirement modeling utilizes antirequirements, misuse cases, *etc*. Are there equivalents in experience requirements?

An in-depth study of destructive stakeholders is needed since these stakeholders have the potential to corrupt, if not destroy, the intended user experience. As a result, one can argue that their actions can make a game a failure in practice – negating all the invested effort. The alternative is to invest, as necessary, in protecting the experience from these stakeholders.

Formulating requirements for appropriate responses to the threats posed by destructive stakeholders requires an understanding of issues such as motivation, negotiation and amelioration. Complicating matters are issues of measurement and interpretation. For example, how can one (through passive, third-party observation of the game world only) discriminate between an inept player who accidentally kills another player and a destructive player who deliberately kills another player? We want to support the inept player and help them to get better; the destructive player may require corrective action such as warnings or even blocking from participation.

We proposed the use of runtime requirements negotiation to facilitate self-policing within the player community, providing support for community modification of, and enforcement of, the rules of the game. Determining the value of runtime requirements negotiation to a community of interest, such as players participating in a common game, remains an open question. The role of the community as a self-policing entity is worthy of further investigation, particularly with respect to the effects on the stringency necessary for the security requirements for that community: if the players will self-correct, it may not be necessary to invest as heavily in security infrastructure.

The effects of real-time requirements negotiation on other game requirements, particularly security requirements, are unknown. We have posited that real-time requirements negotiation targeting destructive shareholders could lead to an inherently unstable system, characterized by skirmishes between destructive stakeholders and others. The validity of this hypothesis remains untested.

Requirements prioritization and negotiation usually assumes a stakeholder community with extant power structures. Real-time requirements negotiation leads to the question: Who should have control over how the game is played? Is it the designer, the publisher, the player, or someone else? Appropriate mechanisms for resolving conflicts between these stakeholders have not yet been developed. Should alternative play (such as that enabled by *hacks* or *mods*) be considered a security threat?

USER EXPERIENCE DESIGN, GAME DESIGN Experience requirements could be applied outside of the videogame domain, forming a contribution to the more general field of user experience design. Perhaps there are elements of experience that can be "engineered" through the use of artifacts like style guidelines, patterns, and GDD content checklists. Are there identifiable patterns that can used to make experience design more of a science than an art? If these patterns exist, are they unique to a genre, perhaps acting as differentiators between games?

One area that our work has not addressed is that of "serious games", particularly in business and government domains. There can be (potentially severe) real-life consequences as a result of serious-games-based training or simulations. We expect that additional research will be necessary to help us understand requirements in this domain.

FILM AND OTHER CREATIVE / MEDIA PRODUCTIONS A thorough survey and analysis of established practices in media production may identify new opportunities to expand or apply experience requirements. Can the practical techniques developed as part of the experience requirements methodology be successfully applied to other forms of media production? Just as experience requirements are generalizations of emotional requirements, perhaps there are generalizations beyond experience requirements.

VALUE AND ECONOMIC ANALYSIS Understanding the stakeholder value propositions for different gamer demographics and different game genres could lead to improved requirements for these games. For example, do the players within a particular community value interactions between real-world capital systems and game-world capital systems and economies? Can the nature of these interactions be predicted or even controlled? Do concepts such as Consalvo's gaming capital [14] apply in practice?

Further work on the economic interactions between "better experiences" during gameplay and customer purchasing behavior could provide guidance regarding investment in new game titles. Example questions could include: What is the importance of the story behind the game experience? Why do players tend to bond to some genres and not others? And, what is the role of social interaction in the promotion, acceptance, and adoption of new games?

LANGUAGE AND ONTOLOGY Our earliest work in the area identified implication in the GDD as a significant risk factor [5]. Three types of problematic implication have already been identified; are there other types of implication not yet identified? A cost-benefit analysis of the threats posed by the implication types would better quantify the consequences of failing to address the issue.

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

VI. Conclusions

This work has presented a summary of our research program into challenges in videogame development, particularly our focus on those challenges associated with communications between the preproduction and production phases of the development process. We have proposed experience requirements as a contribution toward resolving some of the challenges.

To situate experience requirements within the greater domain of videogame development, we have presented herein a review of academic and trade press contributions to the area of requirements engineering in videogame development. We have presented examples of academic research interest in videogame development and pragmatic perspectives from the trade press. However, none of the reviewed work focuses on the videogame production challenges that we have addressed.

We have summarized herein the opportunities presented for further research in our published work and supplemented this summary with other opportunities extracted from our research notes. These opportunities are broadly grouped into the categories of process management, experience requirements, requirements engineering in games, interactions between experience requirements and other types of requirements, user experience design and game design, film and other creative/media productions, value and economic analysis, and language and ontology.

Our goal with this review was to provide a concise summary of these opportunities that may help other researchers and practitioners find interesting places to start their own work in this domain and perhaps even identify opportunities for collaboration. The breadth of domain is vast and we believe that the opportunities presented here are but a small portion of those available.

References

[1] Various Authors. Postmortem column. *Game Developer*, 6(5) through 11(6), May 1999 - June 2004.

[2] Eric Bethke. *Game Development and Production*. Wordware Publishing, Inc., 2003.

[3] Jonathan Blow. Game development: Harder than you think. *Queue*, 1:28–37, February 2004.

[4] David Callele. Physualization: Going beyond paper prototyping. In *MERE '10: Proceedings of the 5th International Workshop on Multimedia and Enjoyable Requirements Engineering*, pages 1–10, Sydney, Australia, 1-1 2010. IEEE Computer Society.

[5] David Callele, Eric Neufeld, and Kevin Schneider. Requirements engineering and the creative process in the video game industry. In *RE '05: Proceedings of the 2005 13th IEEE International Requirements Engineering Conference*, pages 240–250, Paris, France, 2005. IEEE Computer Society.

[6] David Callele, Eric Neufeld, and Kevin Schneider. Emotional requirements in video games. In *RE '06: Proceedings of the 2006 14th IEEE International Requirements Engineering Conference*, pages 292–295, Minneapolis, MN, USA, 2006. IEEE Computer Society.

[7] David Callele, Eric Neufeld, and Kevin Schneider. Balancing security requirements and emotional requirements in video games. In *RE '08: Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*, pages 319–320, Barcelona, Spain, 2008. IEEE Computer Society.

[8] David Callele, Eric Neufeld, and Kevin Schneider. Emotional Requirements. *IEEE Software*, 25(1):43–45, 2008.

[9] David Callele, Eric Neufeld, and Kevin Schneider. Requirements in conflict: Player vs. designer vs. cheater. In *MERE '08: Proceedings of the 3rd International Workshop on Multimedia and Enjoyable Requirements Engineering*, pages 12 –21, Barcelona, Spain, 9-9 2008. IEEE Computer Society.

[10] David Callele, Eric Neufeld, and Kevin Schneider. Augmenting emotional requirements with emotion markers and emotion prototypes. In *RE '09: Proceedings of the 2009 17th IEEE International Requirements Engineering Conference*, pages 373 –374, Atlanta, GA, USA, August 2009. IEEE Computer Society.

[11] David Callele, Eric Neufeld, and Kevin Schneider. Visualizing emotional requirements. In *Requirements Engineering Visualization (REV), 2009 Fourth International Workshop on*, pages 1–10, Atlanta, GA, USA, 1-1 2009. IEEE Computer Society.

[12] David Callele, Eric Neufeld, and Kevin Schneider. Cognitive gameplay requirements. In *MERE '10: Proceedings of the 5th International Workshop on Multimedia and Enjoyable Requirements Engineering*, Sydney, Australia, 2010. IEEE Computer Society.

[13] David Callele, Eric Neufeld, and Kevin Schneider. Introducing experience requirements. In *RE '10: Proceedings of the 2010 18th IEEE International Requirements Engineering Conference*, Sydney, Australia, 2010. IEEE Computer Society.

[14] Mia Consalvo. *Cheating: Gaining Advantage in Videogames*. The MIT Press, 2007.

[15] Chris Crawford. *Chris Crawford on Game Design*. New Riders Publishing, 2003.

[16] Auston Grossman Editor. *POSTMORTEMS from Game Developer*. CMP Books, 2003.

[17] John Flynt and Omar Salem. *Software Engineering for Game Developers*. Nelson Thomson Learning, New York, 2004.

[18] Andre Furtado, Andre Santos, and Geber Ramalho. Streamlining domain analysis for digital games product lines. In Jan Bosch and Jaejoon Lee, editors, *Software Product Lines: Going Beyond*, volume 6287 of *Lecture Notes in Computer Science*, pages 316–330. Springer Berlin / Heidelberg, 2010.

[19] C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of Software Engineering*. Prentice-Hall, 1991.

978-1-4577-0935-7/11/$26.00 © 2011 IEEE

[20] C.M. Kanode and H.M. Haddad. Software engineering challenges in game development. In *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*, pages 260 –265, april 2009.

[21] Raph Koster. *A Theory of Fun*. Paraglyph Press, 2005.

[22] Tom Meigs. *Ultimate Game Design*. McGraow-Hill/Osborne, 2003.

[23] David Michael. *The Indie Game Development Survival Guide*. Charles River Media, Inc., 2003.

[24] Fabio Petrillo and Marcelo Pimenta. Is agility out there?: agile practices in game development. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 9–15, New York, NY, USA, 2010. ACM.

[25] Fábio Petrillo, Marcelo Pimenta, Francisco Trindade, and Carlos Dietrich. Houston, we have a problem...: a survey of actual problems in computer games development. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 707–711, New York, NY, USA, 2008. ACM.

[26] Fábio Petrillo, Marcelo Pimenta, Francisco Trindade, and Carlos Dietrich. What went wrong?; a survey of problems in game development. *Comput. Entertain.*, 7(1):1–22, 2009.

[27] Lakshmi Prayaga. Mapping software engineering principles to stages in game development. *J. Comput. Small Coll.*, 26:208–214, December 2010.

[28] Andrew Rollings and Dave Morris. *Game Architecture and Design, A New Edition*. New Riders Publishing, 2004.

[29] Greg M. Smith. *Film Structure and the Emotion System*. Cambridge University Press, 2007.

978-1-4577-0935-7/11/$26.00 © 2011 IEEE