Software Architectures and the Creative Processes in Game Development

by Alf Inge Wang and Njål Nordmark, 2012

About the article

- How game developers think about and use software architecture in development of games
- Questionnaires among 13 game developers, backed up by literature

Software Architectures and the Creative Processes in Game Development

Alf Inge Wang[1] and Njål Nordmark[2]

Norwegian University of Science and Technology,
N7491 Trondheim, Norway
alfw@idi.ntnu.no, njaal.nordmark@gmail.com

Abstract. Game development is different from traditional software engineering in that there are no real functional requirements and the customers buy and use the software only because it is engaging and fun. This article investigates how game developers think about and use software architecture in the development of games. Further, it looks at how creative development processes are managed and supported. The results presented in this article come from responses to a questionnaire and a survey among thirteen game developers. The research questions answered in this study are: what role does the software architecture play in game development, how do game developers manage changes to the software architecture, how are creative development processes managed and supported, and how has game development evolved the last couple of years. Some of our findings are that software architectures play a central role in game development where the focus is mainly on achieving software with good performance and high modifiability, creative processes are supported through flexible game engines and tools, use of scripting and dynamic loading of assets, and feature-based teams with both creative and technical professions represented, and game developers are incrementally using more game-specific engines, tools and middleware in their development now compared to earlier.

Keywords: Game development, Creative software development, Software architecture.

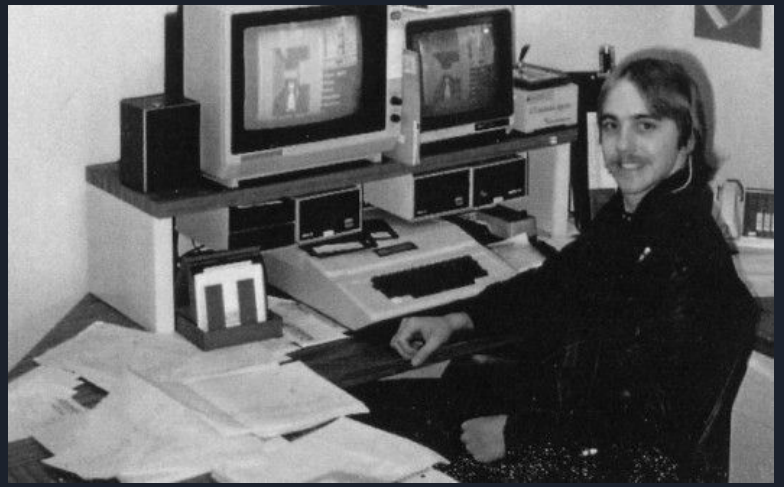
1 Introduction

Game development can be incredibly challenging as game technology such as game engines and game platforms changes rapidly, and code modules crafted for specific games offer less than 30 percent reuse [1]. In the early days of the video games era, game development was carried out by small teams, where the software architectures were made out of a few modules such as 2d graphics, simulation, sound, streaming of i/o and main. At this time, there was not much focus on software architecture and software engineering, but rather on how to how to create an interesting game with the limited hardware resources available. The success of the video game industry, the development of game technology and the increasing demands from the players have resulted in large and complex games developed by large teams of multiple

K. Chorianopoulos et al. (Eds.): ICEC 2015, LNCS 9353, pp. 272–285, 2015.
DOI: 10.1007/978-3-319-24589-8_21

Brief history

- Early days:
  - Small teams
  - How to create an interesting game with limited hardware
- Today:
  - Bigger teams
  - Complex software architecture



Research Goal, Questions and Methods

- **Research goal:** Examine how software architecture is used and how creative processes are managed from the point of view of a game developer in the context of video game development

- **RQ1:** What role does software architecture play in game development?
- **RQ2:** How do game developers manage changes to the software architecture?
- **RQ3:** How are creative processes managed and supported in game development?
- **RQ4:** How has game development evolved the last couple of years?

Design of Software Architecture (RQ1)

	Agree	Neutral	Disagree	N/A
1. Design of software architecture is an important part of our game development process.	69%	15%	8%	8%
2. The main goal of our software architecture is performance	54%	15%	23%	8%
3. Our game concept heavily influences the software architecture	69%	8%	15%	8%
4. The creative team is included in the design of the software architecture	69%	15%	8%	8%
5. Our existing software suite provides features aimed at helping the creative team do their job	92%	8%	0%	0%
6. Our existing software architecture dictates the future game concepts we can develop	15%	47%	38%	0%

Changes to the Software Architecture during Development (RQ2)

	Agree	Neutral	Disagree	N/A
7. The creative team has to adopt their ideas to the existing game engine	31%	46%	23%	0%
8. During development, the creative team can demand changes to the software architecture	61%	31%	0%	0%
9. The technical team implements all features requested by the creative team	69%	15%	8%	8%
10. It is easy to add new gameplay elements after the core of our game engine has been completed	70%	15%	0%	15%
11. During development, the creative team has to use the tools and features already available	47%	15%	38%	0%

Changes to the Software Architecture during Development (RQ2)

12. Who decides if change-requests from the creative team are implemented?

Technical team	Management	Creative team
10%	40%	50%

Supporting the Creative Processes (RQ3)

	Agree	Neutral	Disagree	N/A
13. Our game engine supports dynamic loading of new content	92%	8%	0%	0%
14. Our game engine has a scripting system the creative team can use to try out and implement new ideas	70%	15%	15%	0%
15. The creative team is included in our development feedback loop (e.g., scrum meetings)	86%	8%	0%	8%
16. Our game engine allows rapid prototyping of new levels, scenarios, and NPC's/behavior	86%	8%	0%	8%

Changes Over Time (RQ4)

	Agree	Neutral	Disagree	N/A
17. Today our company uses more 3rd-party modules than 3 years ago	46%	15%	8%	31%
18. It is easier to develop games today than it was 5 years ago	77%	8%	15%	0%
19. Middleware is more important to our company today than 3 years ago	55%	15%	15%	15%
20. Game development is more like ordinary software development today than 5 years ago	38%	24%	38%	0%

Conclusion

- **RQ1:** Software architecture is important in game development, and it is important for managing the complexity of game software.
- **RQ2:** The creative team has to some degree adjust their game play ideas to existing software architecture based on a cost/benefit analysis.
- **RQ3:** Current game engines enable creative processes through support of GUI tools, scripting, and dynamic loading of content.
- **RQ4:** There has been an increased use of third-party software, middleware has become more important, and it has become technically easier to develop games