Cowboys, Ankle Sprains, and Keepers of Quality: How Is Video Game Development Different from Software Development?

03/06/2025

Emerson Murphy-Hill, Thomas Zimmermann and Nachiappan Nagappan
Presenter Paul Wooseok Lee

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Why Compare Game and Traditional Software Development?

- **Video games are a major part of the software industry.**
- **Game revenue surpasses traditional software.** (At least when the paper was published)
- **Yet, little research has been done on software engineering practices in gaming.**

How Was the Study Conducted?

- **Interviews**: 14 game developers with traditional software experience.
- **Surveys**: 364 developers and testers from Microsoft (Games, Office, Other).

		Games	Non-Games
Median years of development experience		8.5	8.5
Number of interviewees with "extensive" experience in…	Programming	10	12
	Design	6	5
	Management	7	4
	Audio/Visual	2	3
	Testing	3	5

	Games	Office	Other
Mean years at Microsoft	4.4	7.1	5.1
Mean years of development experience	10.7	11.0	8.8
Number of engineers	113	61	82
Number of testers	32	39	37

Game Development Encourages 'Cowboy Coding'

- **Creativity is valued over structured processes.**
- **Game designers often modify requirements based on player experience, making formal requirements less stable than in other software fields.**
- **Developers often work long hours "crunch time".**
- **Game studios have a fast-paced, high-pressure environment.**

Game Testing Is More Manual, Less Automated

- Unit tests and automated testing are rare in game development.
- Testing focuses on gameplay experience, not functional correctness.
- Game testing relies on human playtesters rather than automated scripts.

Non-Technical Managers and Teams

- Game studios often have non-technical managers.
- Communication between developers and management can be challenging.
- Game teams are highly interdisciplinary, requiring developers to collaborate closely with artists, designers, sound engineers and even specialists.
- Communication between developers and disciplines can be challenging.

Code and Tool Reuse in Games vs. Traditional Software

- Code reuse is rare due to game-specific performance optimizations.
- Games emphasize unique experiences, making standardization difficult.
- Reuse occurs more in tools (e.g., game engines) rather than in code.

Positive Points

- I appreciated that the study didn't just focus on engineers but also included testers. They provided valuable insights into the differences and the challenges of game development compared to traditional software engineering.
- Even with a relatively small number of interviewees, the paper presented meaningful observations.

Negative Points

- **The paper focuses so heavily on how game development differs from traditional software engineering that it almost presents them as entirely separate fields.**
- **The study mainly sampled developers from Microsoft, missing out on a broader industry perspective. It would have been more insightful to include developers from various companies, and game genres (e.g., single player, MMORPGs, indie games) to provide a more comprehensive view of game development practices.**

Future Work

- **The paper mentioned how game studios develop in-house tools for their needs. Future work could explore how these tools are developed and maintained, and whether standardization could benefit the industry.**
- **Investigating sustainable development practices for healthier work environments.**
- **Enhancing collaboration between Engineers & Creatives in game development.**
  - **Understand the challenges and dynamics between the two.**
  - **Develop and test a structured framework or methodology to improve collaboration.**

Rating

Discussion Points and Comments

1. **What specialists can you think of that are required to make games? To elaborate on their Diablo III example, many games do require specialists not typically found in traditional software teams due to the game's specific domain. A great example is Sid Meier's Civilization series, which relies on accurate historical facts.**
2. **Game company exploited passionate gamers by hiring them at low wages and overworking them. Is this true?**
3. **Should game development borrow more structured engineering practices, or would that limit creativity?**
4. **Any potential methodology for collaboration between Creatives & Engineers?**