Challenges and Opportunities on Software Engineering for Computer Games

Fabio Petrillo
LIF/DIM/UQAC
Seminar LATECE/UQAM
September 2020

Fabio PETRILLO

- **Associate Professor at DIM - UQAC (July/2018)**
  - **SE for Computer Games, Software Quality and Architecture**
  - **Member of “Laboratoire d’Informatique Théorique (LIF)”**
- Master and Ph.D. in Computer Science (UFRGS/Brazil)
  - **Agile methods for computer game**
  - Analysis, comprehension et visualisation of software
  - **Swarm Debugging**
- Lecturer at Polytechnique Montréal
- Postdoctoral Fellow at Concordia University (Montréal)
  - UBISOFT Montréal (MITACS) - Logging analysis and anomaly detection
- > 30 years of experience in Software Engineering
  - **Software developer and architect**
  - **Manager** and agile coach
  - Experience on complex and critical-mission systèms

Is **game industry** important?

Game industry is billionaire, greater than cinema and music together

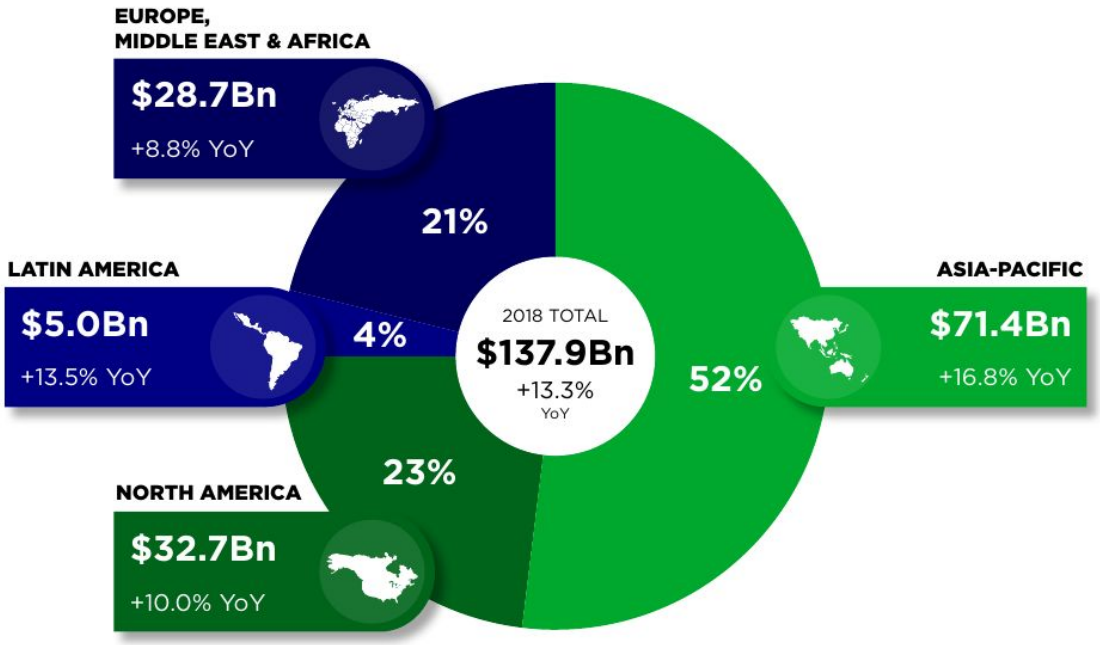
(NEWZOO,2016)

GLOBAL GAMES MARKET 2018

PER REGION

THE GLOBAL GAMES MARKET

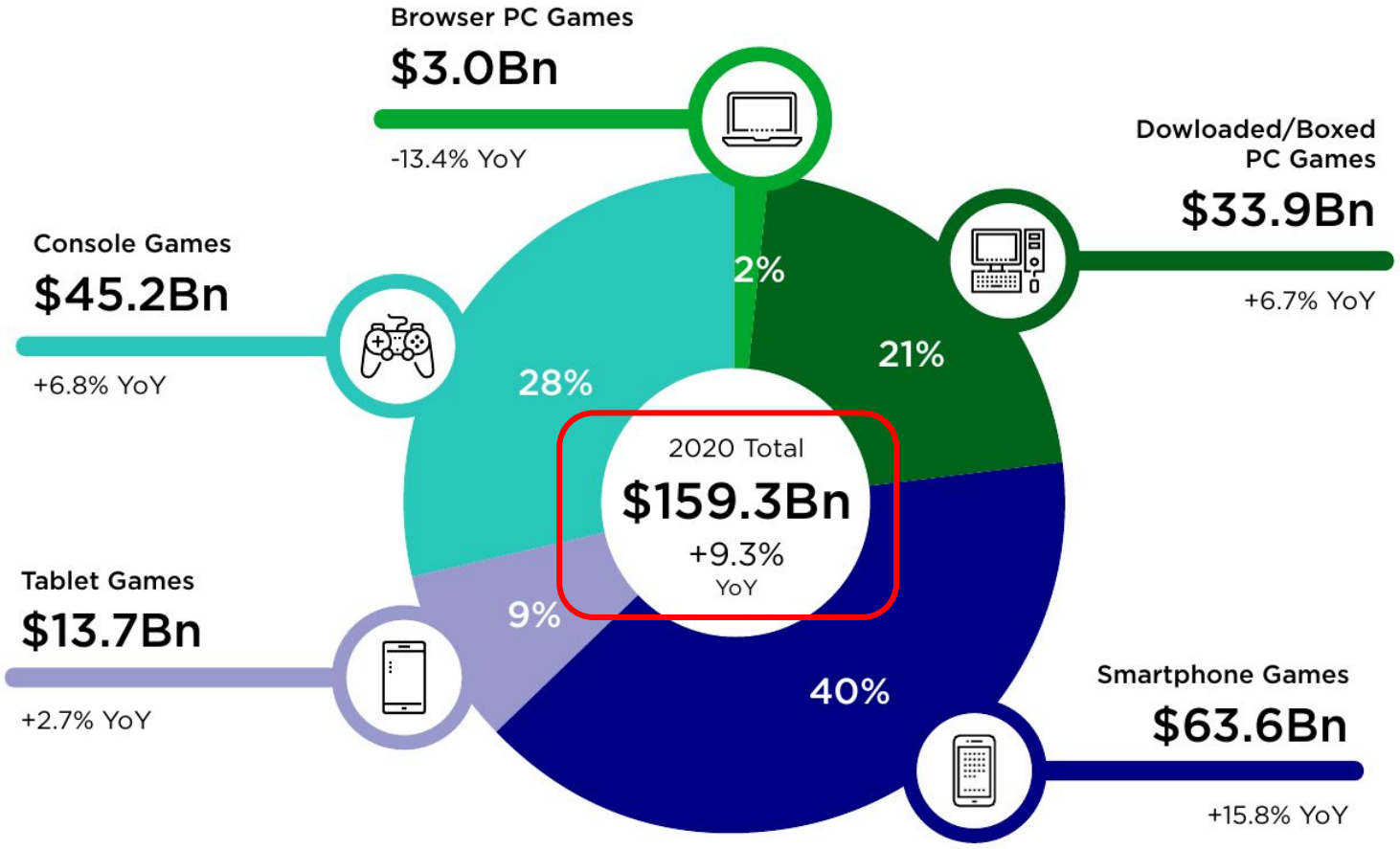
PER REGION



Newzoo 2018 Global games market report, Trends, insights, and projections towards 2021

2020 Global Games Market

Per Segment



Top 50 Public Companies 2019

Rank	Company	HQ	Q1 ($M)	Q2 ($M)	Q3 ($M)	Q4($M)	2019 ($M)	YoY Growth
1	Tencent	CN	5,096	4,936	5,213	5,300	20,545	10%
2	Sony	JP	3,452	2,865	3,047	3,769	13,133	-8%
3	Apple	US	2,435	2,564	2,946	2,887	10,832	14%
4	Microsoft	US	2,243	1,980	2,219	2,831	9,273	-4%
5	Google	US	1,737	1,788	1,948	1,877	7,350	13%
6	NetEase	CN	1,725	1,665	1,680	1,690	6,759	16%
7	Activision Blizzard	US	1,706	1,279	1,107	1,749	5,841	-15%
8	EA	US	1,238	1,209	1,348	1,593	5,388	2%
9	Nintendo	JP	885	769	1,109	2,191	4,954	13%
10	Bandai Namco Entertainment	JP	909	632	726	701	2,968	2%

Top 50 Public Companies 2019

Rank	Company	HQ	Q1 ($M)	Q2 ($M)	Q3 ($M)	Q4($M)	2019 ($M)	YoY Growth
1	Tencent	CN	5,096	4,936	5,213	5,300	20,545	10%
2	Sony	JP	3,452	2,865	3,047	3,769	13,133	-8%
3	Apple	US	2,435	2,564	2,946	2,887	10,832	14%
4	Microsoft	US	2,243	1,980	2,219	2,831	9,273	-4%
5	Google	US	1,737	1,788	1,948	1,877	7,350	13%
6	NetEase	CN	1,725	1,665	1,680	1,690	6,759	16%
7	Activision Blizzard	US	1,706	1,279	1,107	1,749	5,841	-15%
8	EA	US	1,238	1,209	1,348	1,593	5,388	2%
9	Nintendo	JP	885	769	1,109	2,191	4,954	13%
10	Bandai Namco Entertainment	JP	909	632	726	701	2,968	2%

Top 50 Public Companies 2019

Rank	Company	HQ	Q1 ($M)	Q2 ($M)	Q3 ($M)	Q4($M)	2019 ($M)	YoY Growth
1	Tencent	CN	5,096	4,936	5,213	5,300	20,545	10%
2	Sony	JP	3,452	2,865	3,047	3,769	13,133	-8%
3	Apple	US	2,435	2,564	2,946	2,887	10,832	14%
4	Microsoft	US	2,243	1,980	2,219	2,831	9,273	-4%
5	Google	US	1,737	1,788	1,948	1,877	7,350	13%
6	NetEase	CN	1,725	1,665	1,680	1,690	6,759	16%
7	Activision Blizzard	US	1,706	1,279	1,107	1,749	5,841	-15%
8	EA	US	1,238	1,209	1,348	1,593	5,388	2%
9	Nintendo	JP	885	769	1,109	2,191	4,954	13%
10	Bandai Namco Entertainment	JP	909	632	726	701	2,968	2%

The Canadian Video Game Industry 2019

ENTERTAINMENT SOFTWARE ASSOCIATION OF CANADA

Nordicity

November 2019

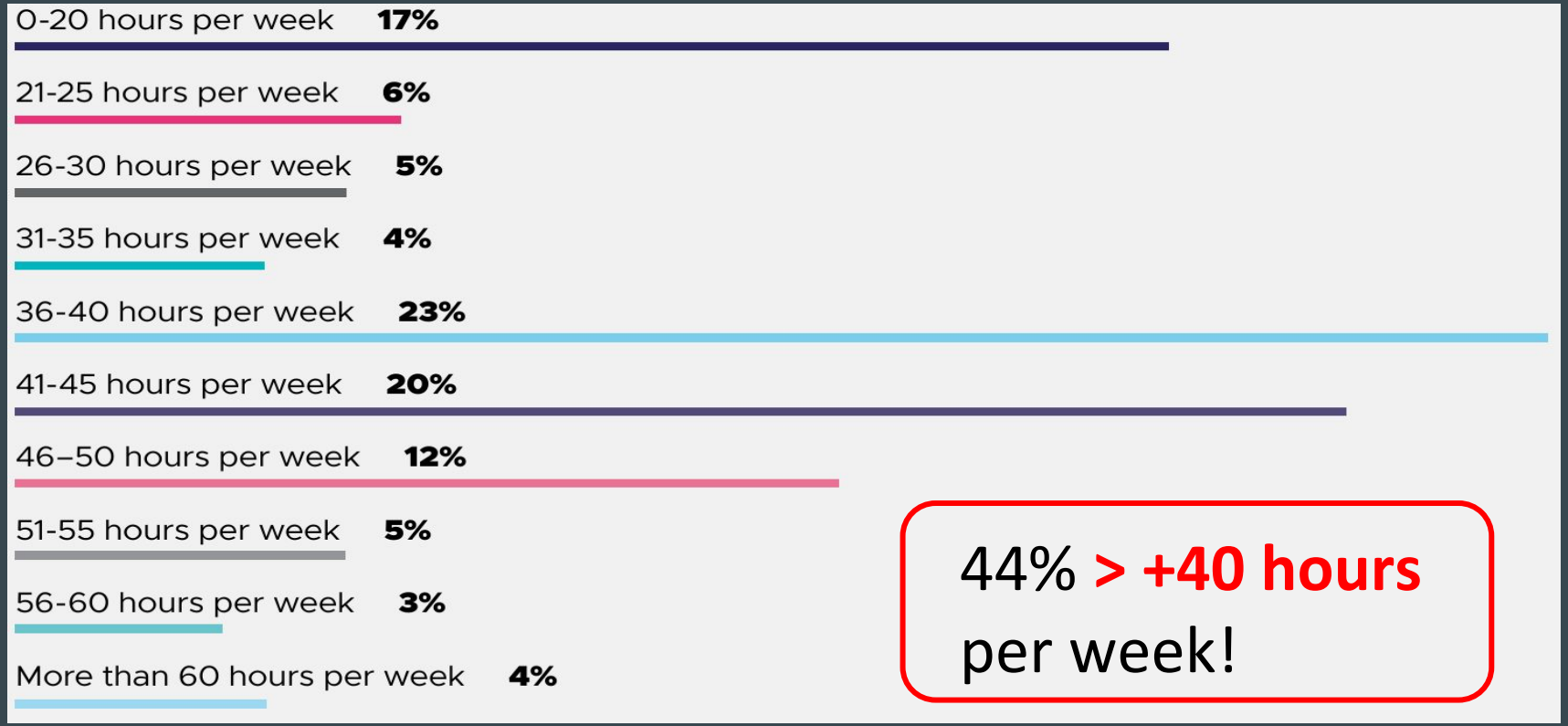
Number of video game companies in Canada

FIRM SIZE	2017	2019	CHANGE
Micro (less than 4 employees)	228	377	+65%
Small (5 to 25 employees)	241	213	-12%
Medium (26 to 59 employees)	62	35	-44%
Large (60 to 99 employees)	39	30	-23%
Very Large (more than 100 employees)	26	37	+42%
Total	**596**	**692**	**+16%**

Employment Impact in Canada

	EMPLOYMENT 2017 (FTES)	% INCREASE 2017-19	EMPLOYMENT 2019 (FTES)
Video game industry	21,700	28%	27,700
Indirect impact	8,800	11%	9,800
Induced impact	10,200	4%	10,600
Total impact	40,600	18%	48,000

Average of worked hours per week in the last 12 months



Source: GDC 2020 State of the Industry Report

Why we are here?


We
VIDEO
GAMES!

Computer Games and Software Engineering

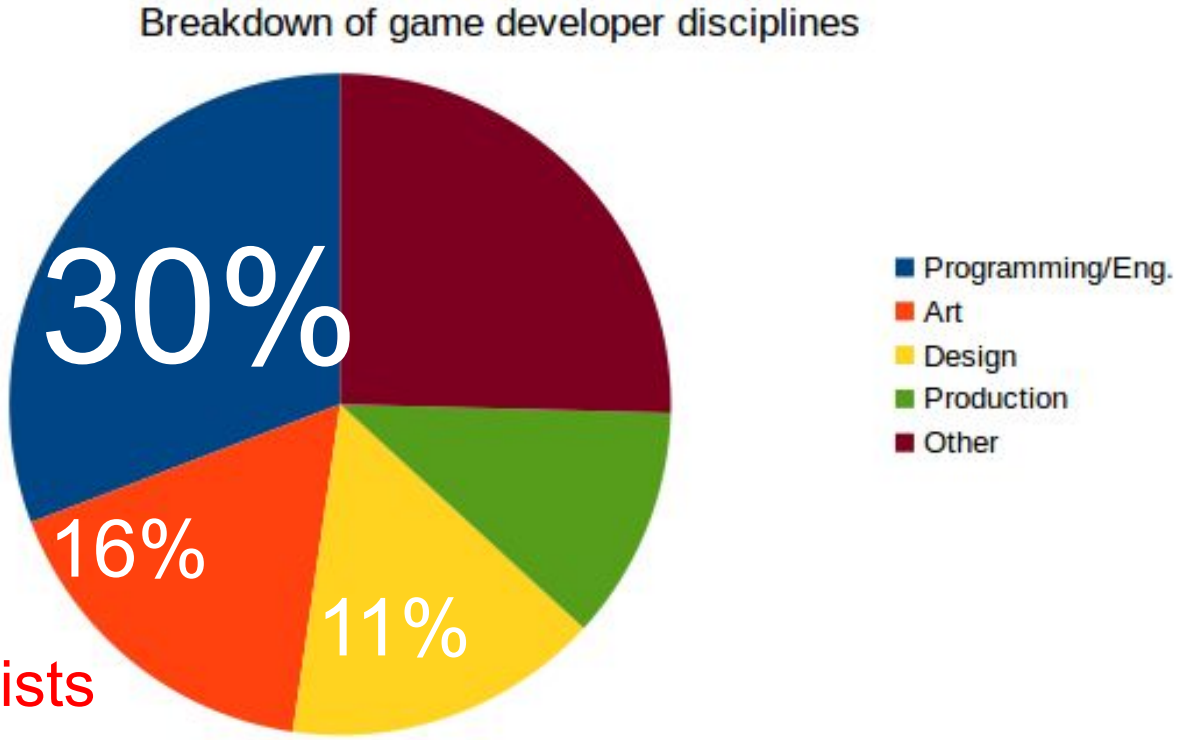
- Computer Games (CG) are simultaneously **advanced software** products and complex works of **creativity and art** [Engstrom, 2018]
- Varying requirements and business goals [Kasurinen, 2017]

Is a computer game more a piece of **software** or a piece of **art**?

GDC 2016 State of the Industry Report

Programmers

Artists



Computer Games and Software Engineering

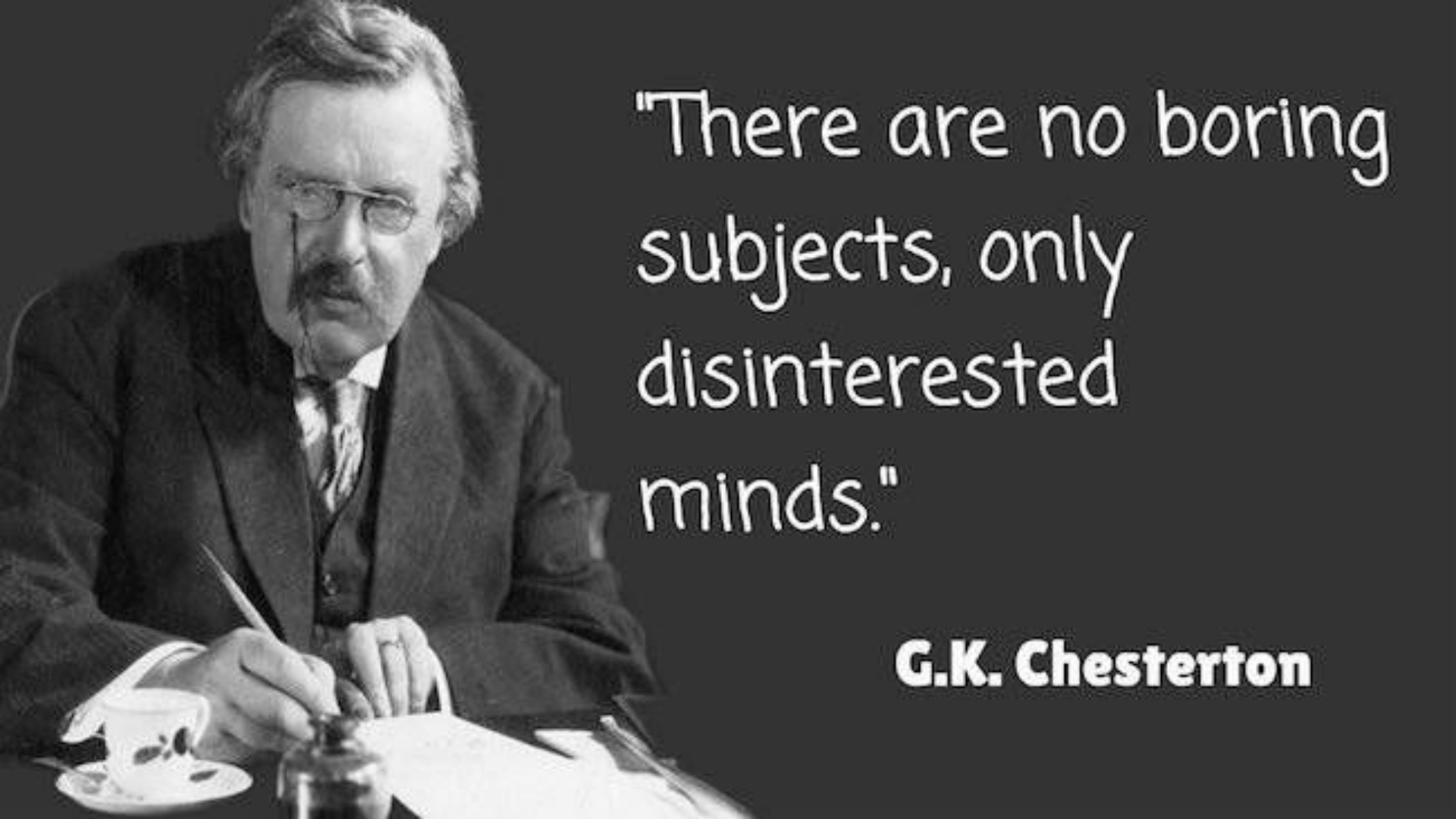
- Nevertheless, the software engineering community **rarely studies CGs** [Murphy-Hill, 2014].

Why do I start to work on Computer Games?

2006/7
I was in a master student seminar when...


BRACE YOURSELVES
A BORING PRESENTATION IS COMING
memegenerator.net

However, it was not exactly that ...


"There are no boring subjects, only disinterested minds."
G.K. Chesterton

We started a discussion:
Waterfall vs Agile for
game development?

So, I decided to investigate Agile and Computer Games!!!

2008
Master Results

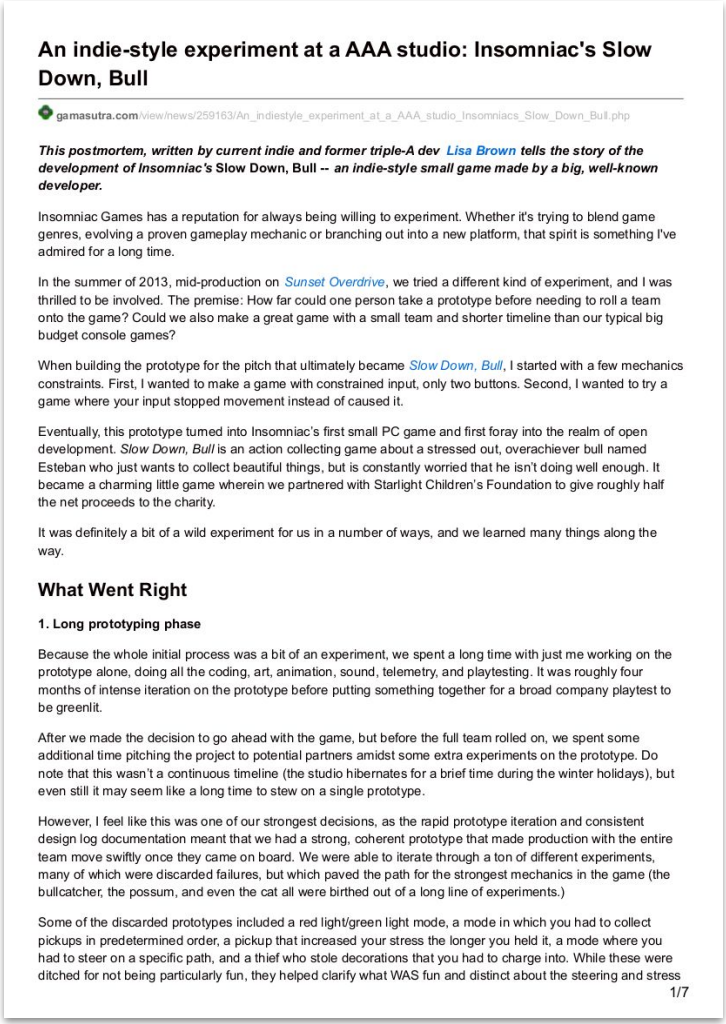
Does game industry have the same problems that "traditional" software industry?


20 game industry
postmortems
DIABLO II

Postmortems

What went right discusses the best practices adopted by developers, solutions, improvements, and project management decisions that help the project.

What went wrong discusses difficulties, pitfalls, and mistakes experienced by the development team in the project, both technical and managerial.



An indie-style experiment at a AAA studio: Insomniac's Slow Down, Bull

gamasutra.com/view/news/259163/An_indiestyle_experiment_at_a_AAA_studio_Insomniacs_Slow_Down_Bull.php

This postmortem, written by current indie and former triple-A dev Lisa Brown tells the story of the development of Insomniac's* Slow Down, Bull *-- an indie-style small game made by a big, well-known developer.

Insomniac Games has a reputation for always being willing to experiment. Whether it's trying to blend game genres, evolving a proven gameplay mechanic or branching out into a new platform, that spirit is something I've admired for a long time.

In the summer of 2013, mid-production on *Sunset Overdrive*, we tried a different kind of experiment, and I was thrilled to be involved. The premise: How far could one person take a prototype before needing to roll a team onto the game? Could we also make a great game with a small team and shorter timeline than our typical big budget console games?

When building the prototype for the pitch that ultimately became *Slow Down, Bull*, I started with a few mechanics constraints. First, I wanted to make a game with constrained input, only two buttons. Second, I wanted to try a game where your input stopped movement instead of caused it.

Eventually, this prototype turned into Insomniac's first small PC game and first foray into the realm of open development. *Slow Down, Bull* is an action collecting game about a stressed out, overachiever bull named Esteban who just wants to collect beautiful things, but is constantly worried that he isn't doing well enough. It became a charming little game wherein we partnered with Starlight Children's Foundation to give roughly half the net proceeds to the charity.

It was definitely a bit of a wild experiment for us in a number of ways, and we learned many things along the way.

What Went Right

1. Long prototyping phase

Because the whole initial process was a bit of an experiment, we spent a long time with just me working on the prototype alone, doing all the coding, art, animation, sound, telemetry, and playtesting. It was roughly four months of intense iteration on the prototype before putting something together for a broad company playtest to be greenlit.

After we made the decision to go ahead with the game, but before the full team rolled on, we spent some additional time pitching the project to potential partners amidst some extra experiments on the prototype. Do note that this wasn't a continuous timeline (the studio hibernates for a brief time during the winter holidays), but even still it may seem like a long time to stew on a single prototype.

However, I feel like this was one of our strongest decisions, as the rapid prototype iteration and consistent design log documentation meant that we had a strong, coherent prototype that made production with the entire team move swiftly once they came on board. We were able to iterate through a ton of different experiments, many of which were discarded failures, but which paved the path for the strongest mechanics in the game (the bullcatcher, the possum, and even the cat all were birthed out of a long line of experiments.)

Some of the discarded prototypes included a red light/green light mode, a mode in which you had to collect pickups in predetermined order, a pickup that increased your stress the longer you held it, a mode where you had to steer on a specific path, and a thief who stole decorations that you had to charge into. While these were ditched for not being particularly fun, they helped clarify what WAS fun and distinct about the steering and stress

What are the most important problems in game industry?

"*All the **main problems** of the **traditional** software industry are also found in the **game** industry*"

Petrillo, F., Pimenta, M., Trindade, F., & Dietrich, C. (2009). What went wrong? A survey of problems in game development. *Computers in Entertainment*, *7*(1), 1.

1) Unrealistic scope
2) Feature creep
3) Cutting features

Petrillo, F., Pimenta, M., Trindade, F., & Dietrich, C. (2009). What went wrong? A survey of problems in game development. *Computers in Entertainment*, *7*(1), 1.

*"...the traditional and game software industries do not suffer mainly from technological problems, but from **management problems**."*

Petrillo, F., Pimenta, M., Trindade, F., & Dietrich, C. (2009). What went wrong? A survey of problems in game development. *Computers in Entertainment*, *7*(1), 1.

So, how can we mitigate these problems?

"*We believe that adoption of **agile practices** in game development can achieve promising results.*"

Petrillo, F., & Pimenta, M. (2010). Is agility out there? Agile Practices in Game Development. In *Proceedings of the 28th ACM International Conference on Design of Communication - SIGDOC '10* (p. 9). New York, New York, USA: ACM Press.

Fábio Petrillo

FOLLOWING

Associate Professor - Université du Québec à Chichoutimi
Verified email at petrillo.com - Homepage

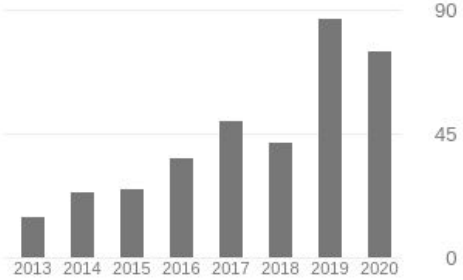
Software Engineering

TITLE	CITED BY	YEAR
What went wrong? A survey of problems in game development F Petrillo, M Pimenta, F Trindade, C Dietrich Computers in Entertainment (CIE) 7 (1), 1-22	108	2009
Houston, we have a problem... a survey of actual problems in computer games development F Petrillo, M Pimenta, F Trindade, C Dietrich Proceedings of the 2008 ACM symposium on Applied computing, 707-711	58	2008
Is agility out there? Agile practices in game development F Petrillo, M Pimenta Proceedings of the 28th ACM International Conference on Design of …	47	2010
Are REST APIs for cloud computing well-designed? An exploratory study F Petrillo, P Merle, N Moha, YG Guéhéneuc International Conference on Service-Oriented Computing, 157-170	25	2016
Are the old days gone? A survey on actual software engineering processes in video game industry C Politowski, L Fontoura, F Petrillo, YG Guéhéneuc Proceedings of the 5th International Workshop on Games and Software …	25	2016

Cited by

VIEW ALL

	All	Since 2015
Citations	400	316
h-index	9	8
i10-index	8	8



Co-authors

EDIT

- Yann-Gaël Guéhéneuc, Concordia University
- Pimenta, Marcelo S., Informatics Institute INF/UFRGS
- Foutse Khomh, Professor of Software Engineerin…

2016,
Ten years later...


KINGDOMS OF AMALUR
RECKONING

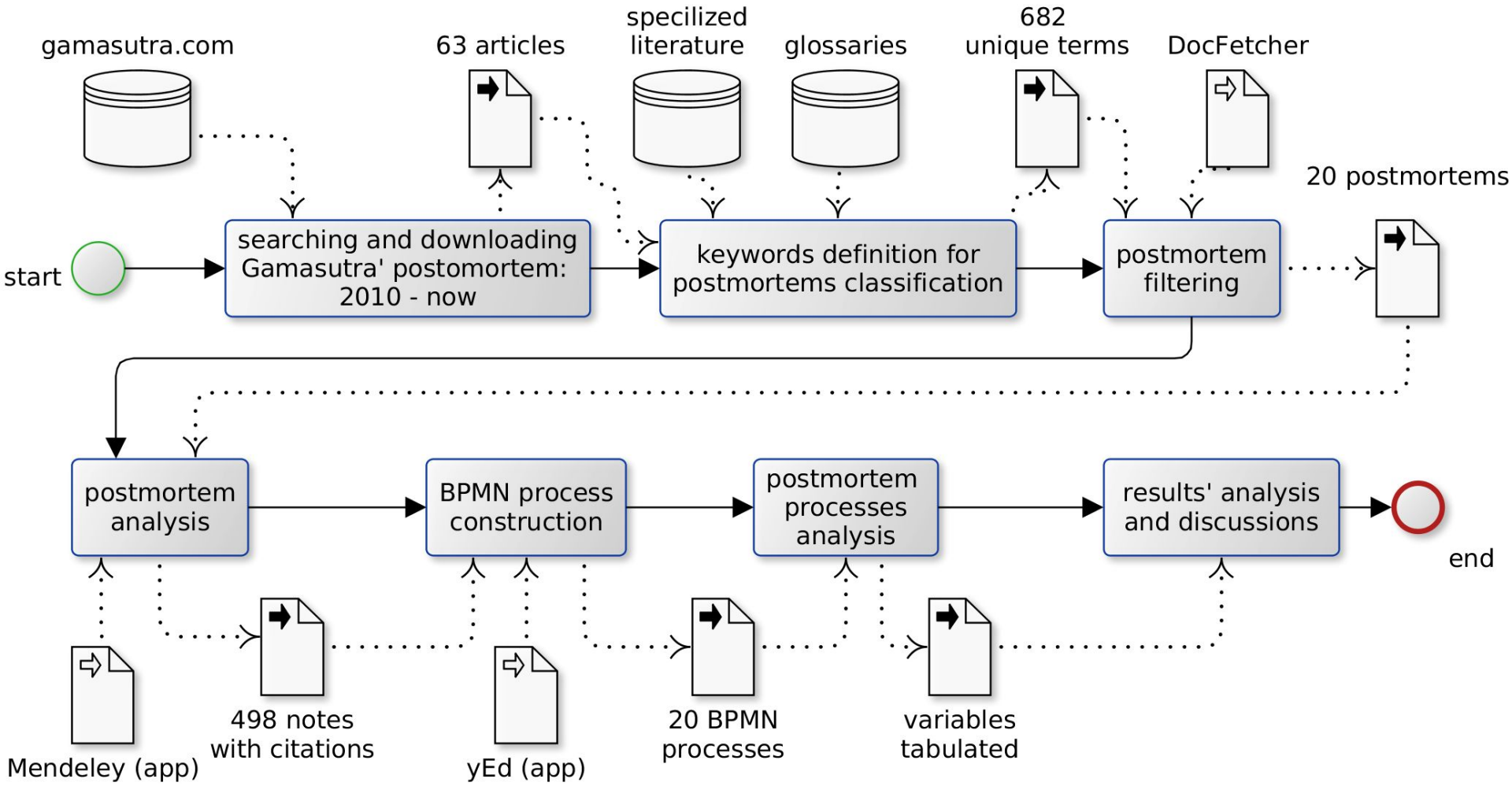
"*The old days are gone.* You can't expect producers or leads to come up with a huge **waterfall** of everything they thought would get done over the *next three years*. In the game development business, it's *insane* to think you have any insight into what your team will be doing one year from now. You can set *major milestones* with hard dates, but filling in all the details between those points is an exercise in **futility**."

Fridley, M. (**2013**). Postmortem: Kingdoms of Amalur: Reckoning. Retrieved from http://www.gamasutra.com/view/feature/197269/postmortem_kingdoms_of_amalur_.php

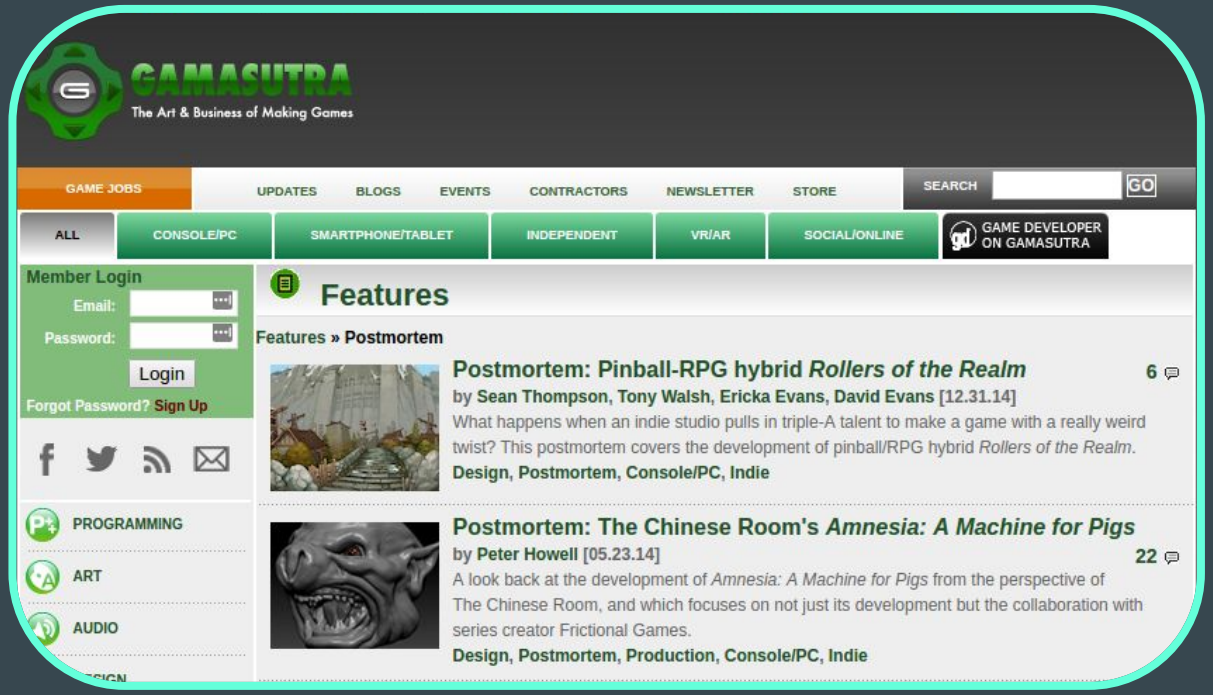
Are these claims general or a *"cherry picking"* cases?

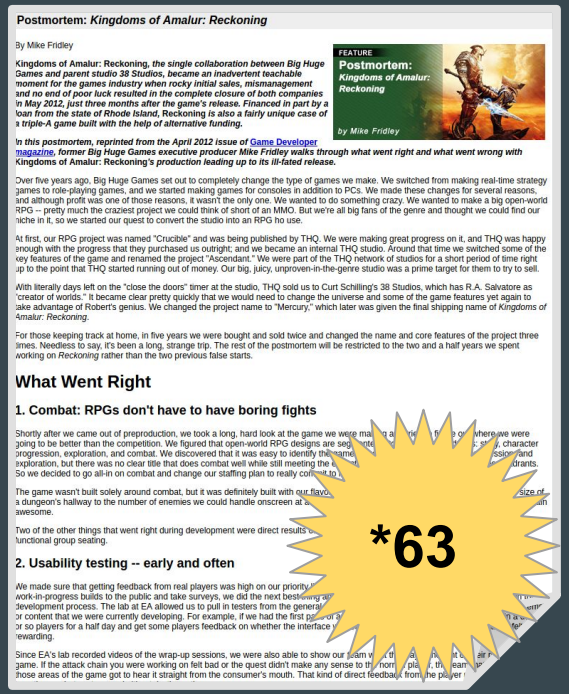
RQ: Are "the old days" really gone in video game industry?

Methodology

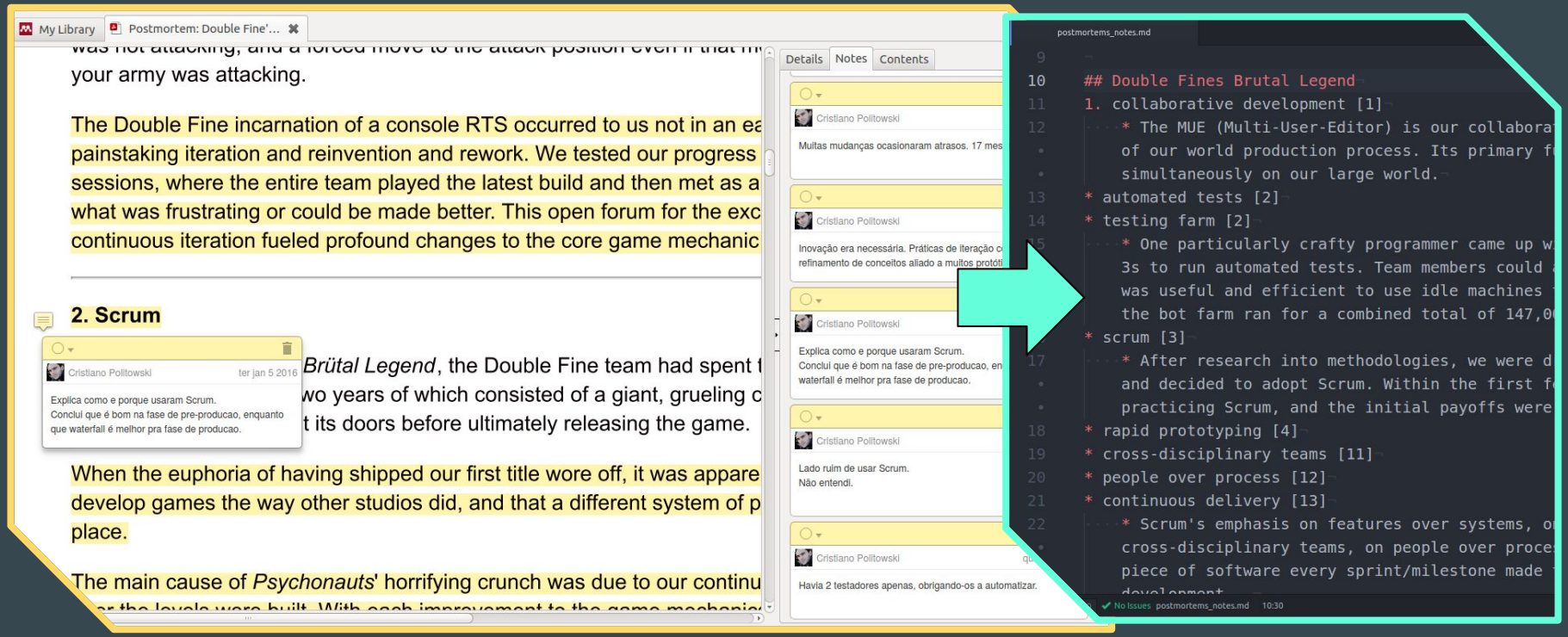

gamasutra.com
63 articles
specilized literature
glossaries
682 unique terms
DocFetcher
20 postmortems
start
searching and downloading Gamasutra' postomortem: 2010 - now
keywords definition for postmortems classification
postmortem filtering
postmortem analysis
BPMN process construction
postmortem processes analysis
results' analysis and discussions
end
Mendeley (app)
498 notes with citations
yEd (app)
20 BPMN processes
variables tabulated

Postmortem Search (2010 - 2016)

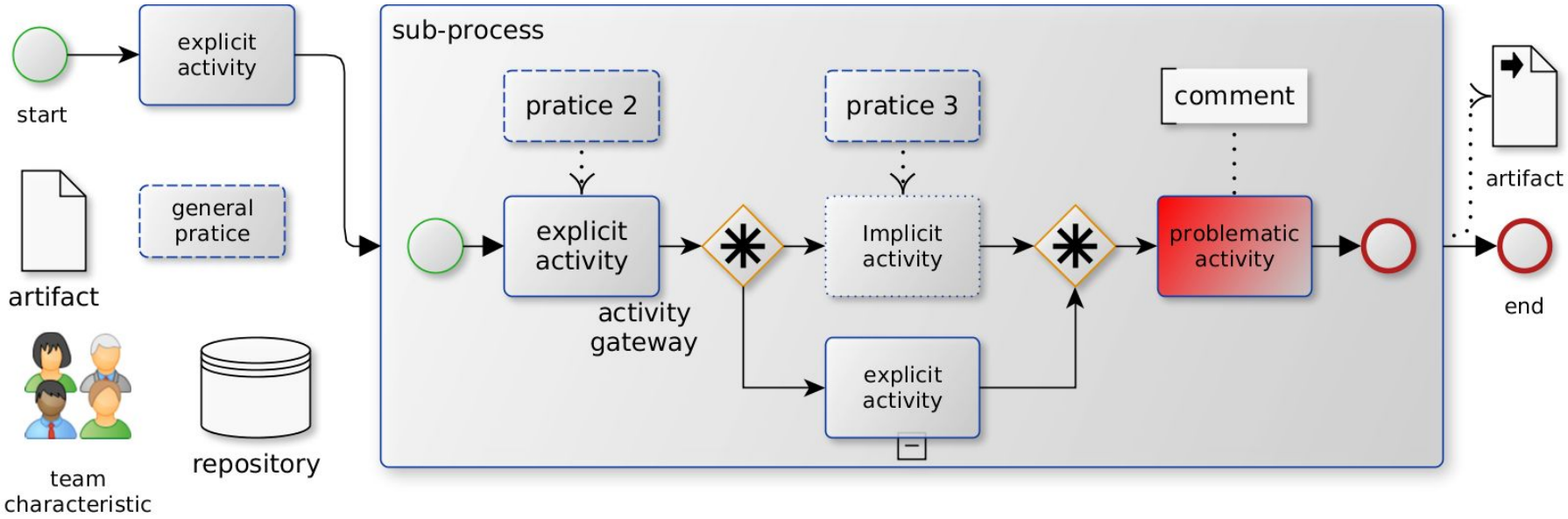




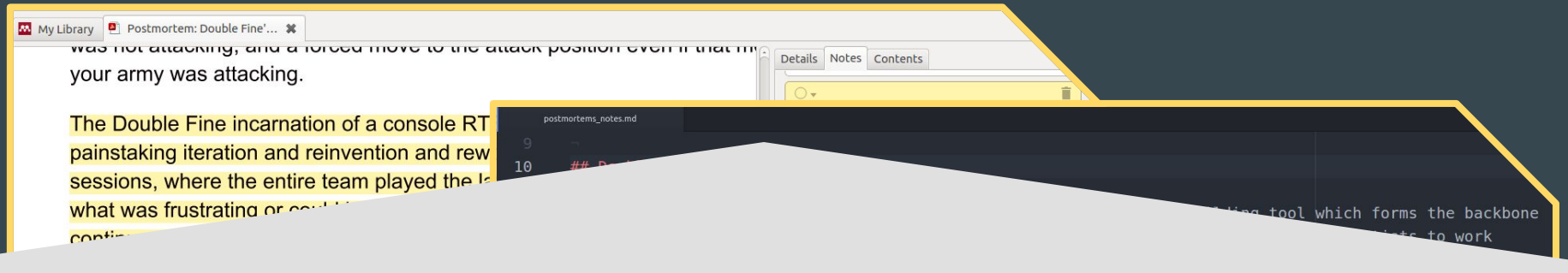
Postmortem Analysis (20 articles)



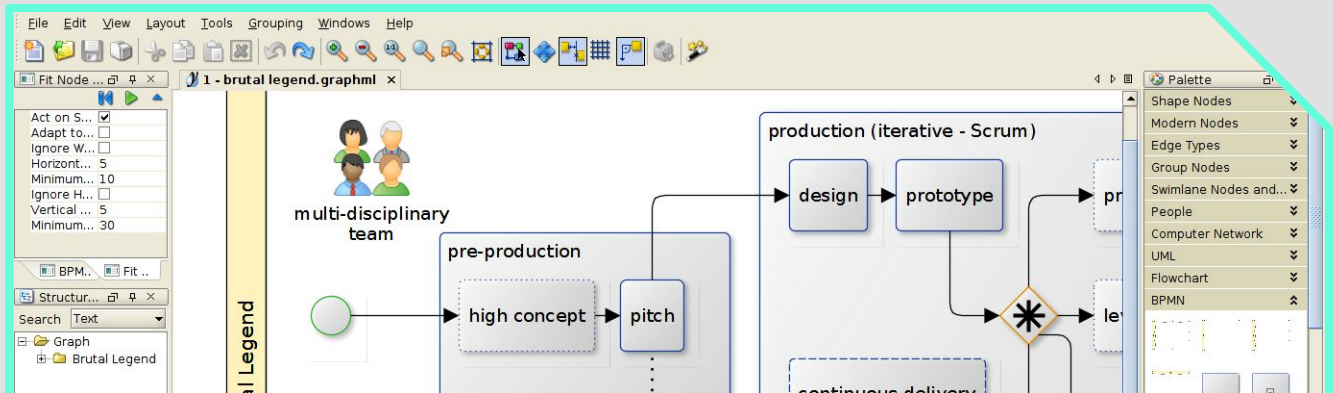
Process Metamodel



Postmortem Analysis (20 articles)

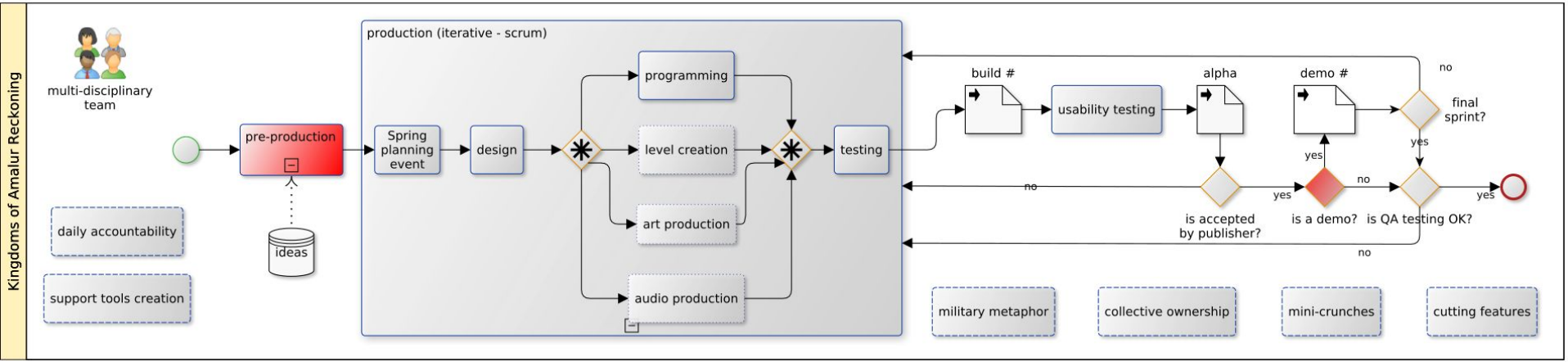


BPMN Process Construction

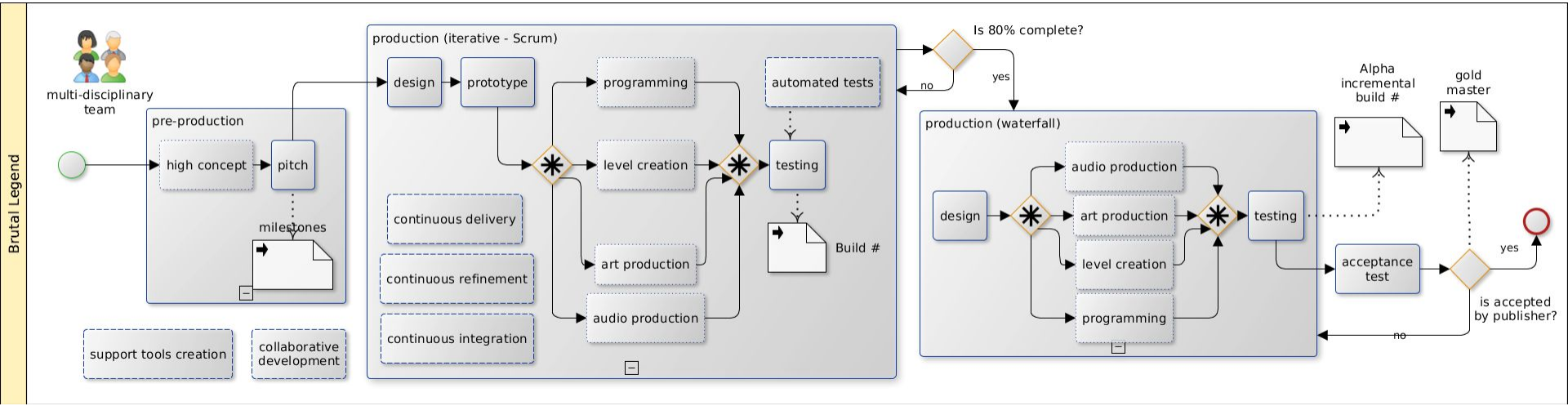


20 process models

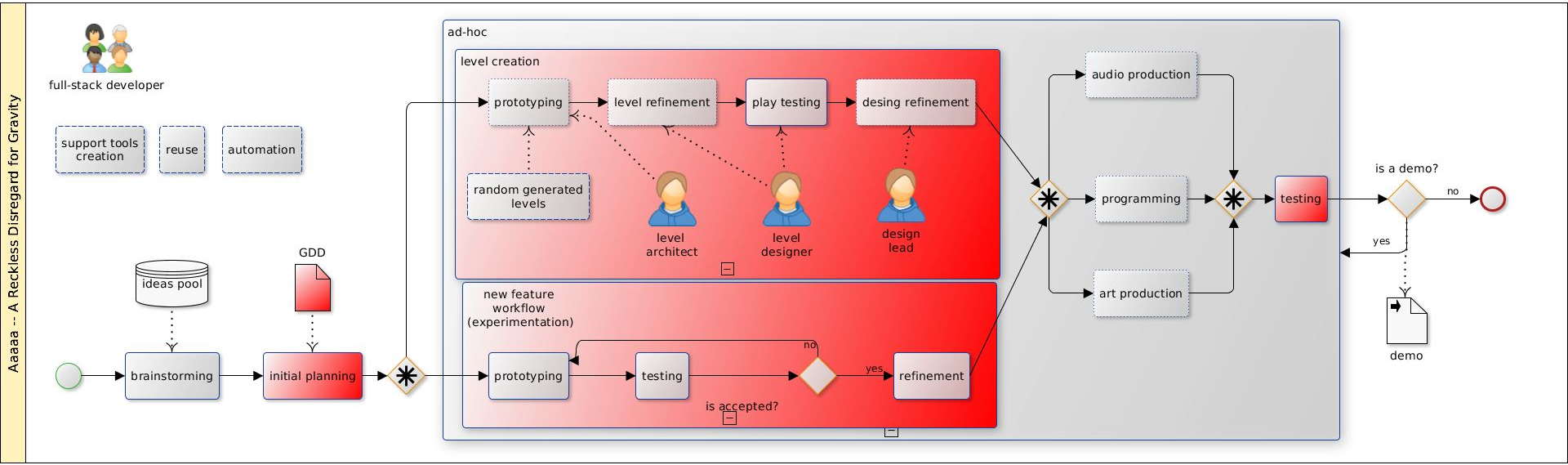
Iterative Process - *Kingdoms of Amalur: Reckoning*



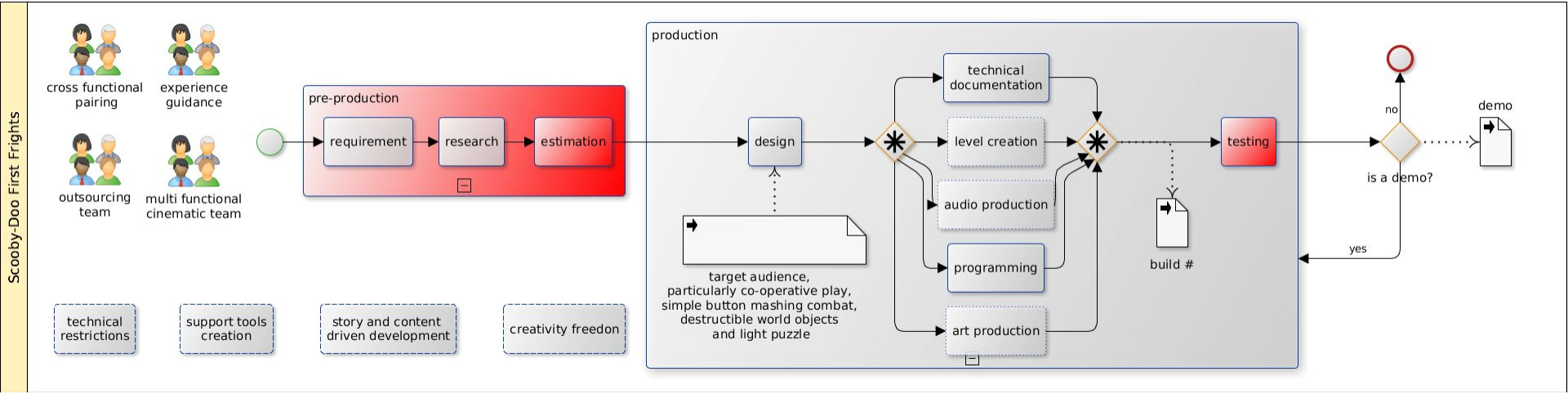
Hybrid Process - *Brutal Legend*



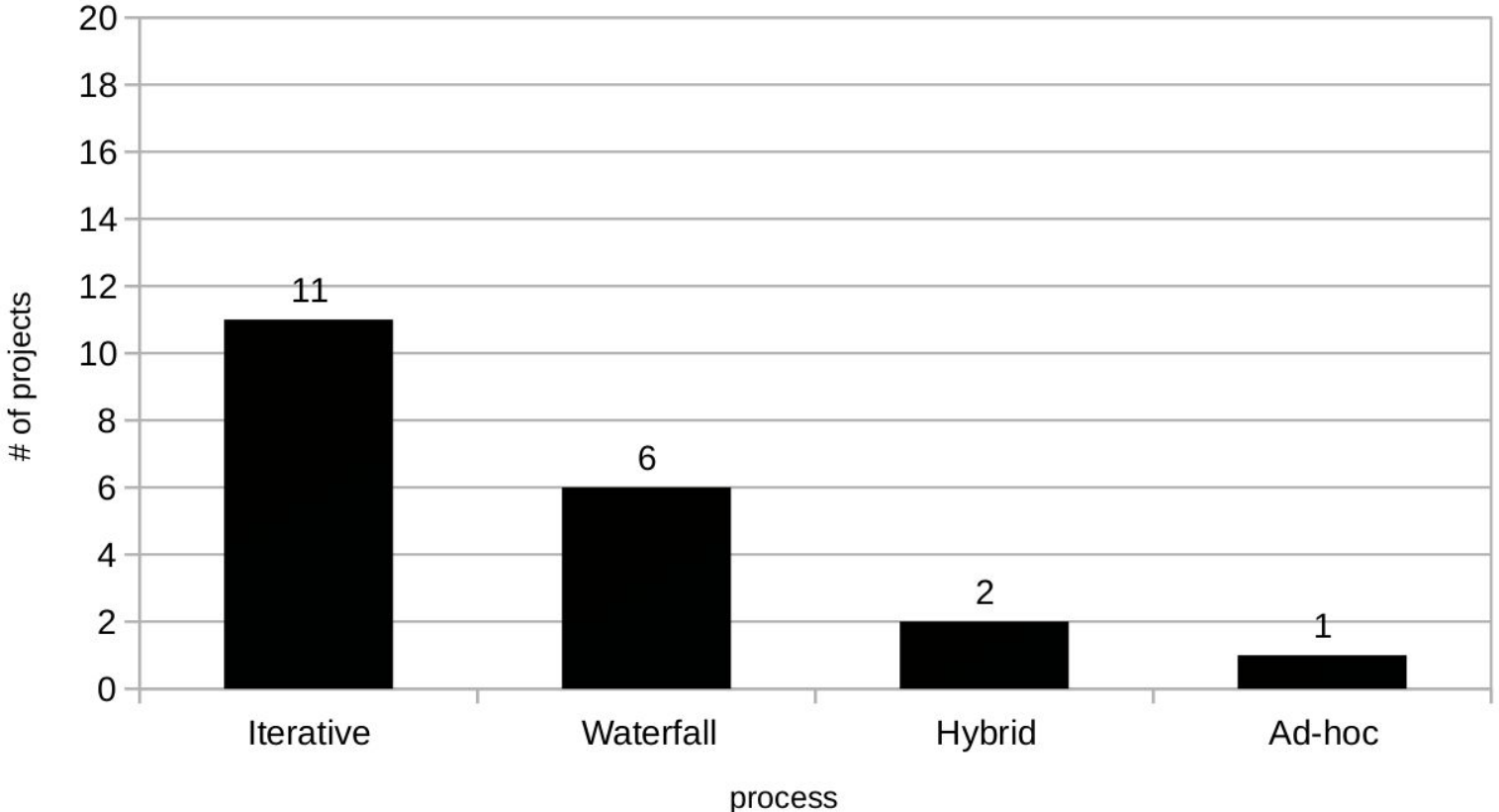
Ad-Hoc Process - Aaaa! *A Reckless Disregard for Gravity*



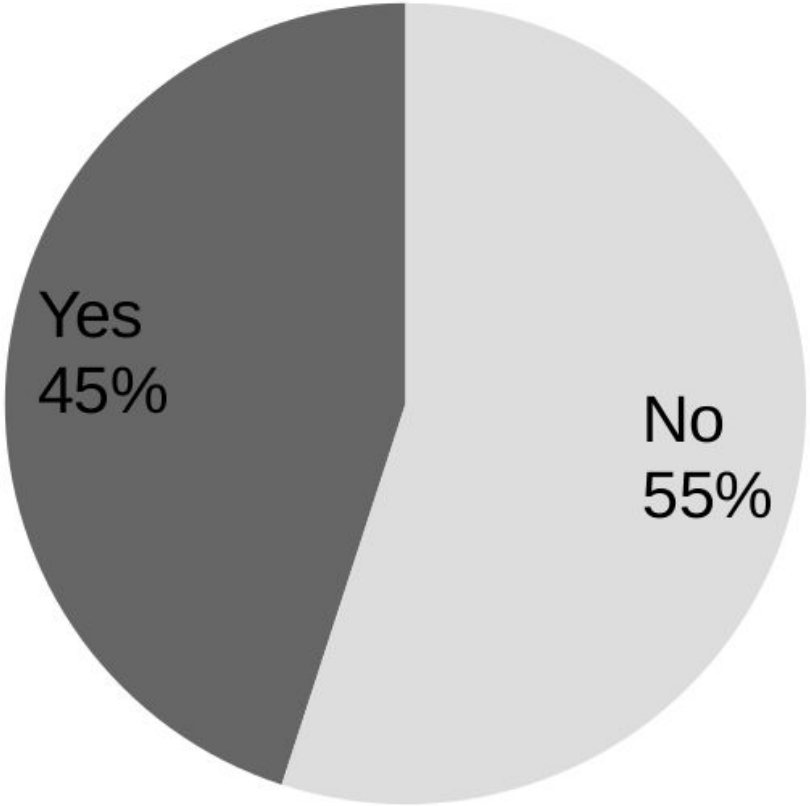
Waterfall Process - *Scooby-Doo*



Process occurrences by category



Agile practices in game projects



Conclusion 1
The “old days” are gone, but not completely at all.

Conclusion 2

Iterative process is currently mainstream in the game industry.

Conclusion 3

Agility are increasing in the last years.

Discussion (2016)

We believe that iterative process and agile practice **benefits** are yet **misunderstood** by some game developers, managers, producers, publishers, and educators.

.

2017/8

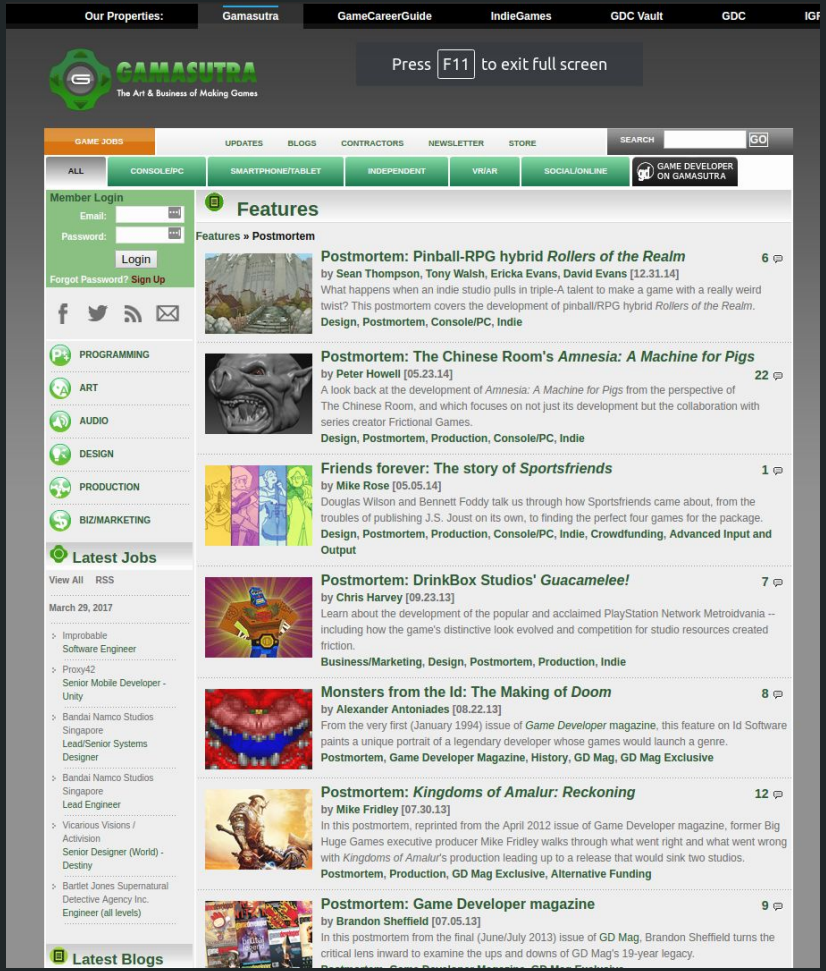
Learning from the Past:
a Process Recommendation System for Video Game Projects using Postmortems Experiences

Cristiano Politowski[1], Lisandra Fontoura[2], Fabio Petrillo[3], Yann-Gaël Guéhéneuc[1]
[1]Concordia University, Montreal - Canada
[2]Universidade Federal de Santa Maria - Brazil
[3]Université du Québec à Chicoutimi - Canada

IST Journal-First presentation on
12th International Symposium on Empirical Software Engineering and Measurement
October 11-12, 2018 - Oulu, Finland

234 gathered
100 analysed
55 used

from 1997 until 2016



Contributions

Video game processes database

database of game development processes from the analysis of 55 postmortems

Recommendation System

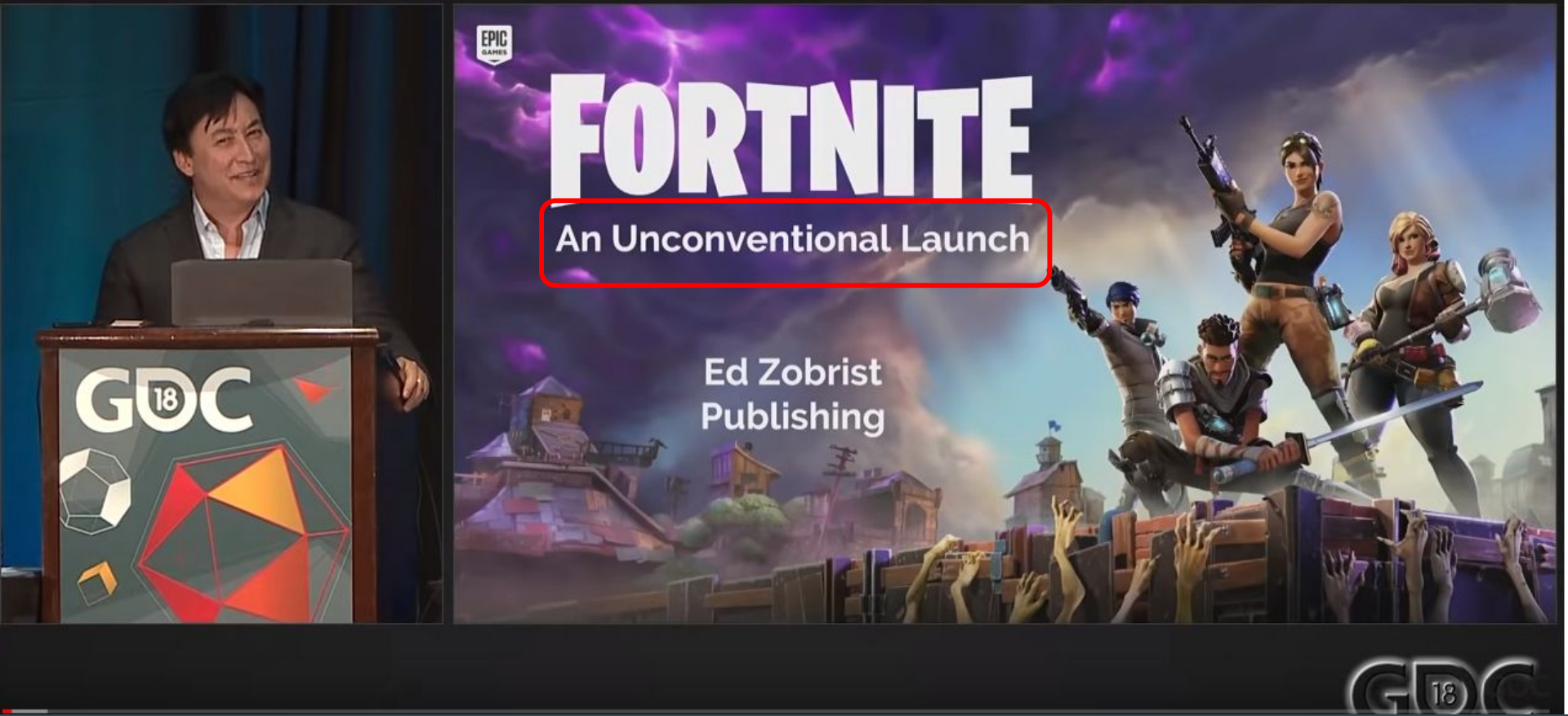
recommendation system capable of generating processes based on previous projects with similar contexts

2018


FORTNITE
BATTLE
ROYALE

How Epic Games keeps Fortnite online for millions of players

Fortnite has hit 125 million players – peaking at 3.2 million concurrent gamers. Keeping it online requires some serious web infrastructure


EPIC GAMES
FORTNITE
An Unconventional Launch
Ed Zobrist
Publishing
GDC 18
GDC 18

Fortnite: An Unconventional Launch

...AND THEN CAME BATTLE ROYALE

- 2 months development – Sep 26 launch
- Initially a mode within Save the World
- Shifted to a free model in final 2 weeks

BR launch 25 Million Bush C4 vid 50v50 Mobile

27:01 / 1:00:53

Role para ver detalhes

Why Fortnite: Battle Royale is a huge success???

Why Fortnite: Battle Royale is a huge success???
Because
the old days are gone!

Fortnite: Less is more...

- *“Their goal was to develop the Battle Royale mode quickly from the core "Save the World" mode,* ***putting off any complex features that weren't already in place*** *as to launch the new mode as* ***soon as possible****; while they explored such potential ideas, they held off inclusion until after the main mode was launched.”*
- **Less is more**
  - **Less weapons**
  - **a small subset of traps**

Why Fortnite: Battle Royale is a huge success?

- The game is **free-to-play**, supported by microtransactions
- The game is run as **seasons**, lasting about **10 weeks each**.
- **Epic's consistent updates for the game.**

Why Fortnite: Battle Royale is a huge success?

“It’s been weird, because from my perspective we’ve been **continuously interacting with players the entire time** – it’s just that we haven’t made a big deal about it with the press,” he says.

“There’s a significant amount of difference between the game two years ago and the game now, so we’ve just been **furiously iterating**.”

https://www.pcgamesn.com/fortnite/why-has-fortnite-taken-so-long

Why Fortnite: Battle Royale is a huge success?

*"**We value the ability of players to adapt to the game changing over time.** We also believe these **changes keep Fortnite fresh** for everyone including players, competitors and spectators..."*

https://www.pcgamer.com/epic-hopes-to-give-fortnite-pros-more-time-to-adjust-to-big-changes-in-2019

Fortnite: Battle Royale is changing the mindset of waterfall/stage gate process...

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Fortnite success **(finally)** opens opportunities to explore

agile practices

on AAA computer games' industry...

Fortnite success (finally) opens opportunities to explore agile practices on AAA computer games' industry...

more than 20 years of agile manifesto and 14 years we discuss that in academia!

Computer game industry has several challenges to SE researches because

- Conservative mindset
- “It is more art than engineering”
- “SE is for mortals; we need performance…”

Concordia/UBISOFT Montréal (MITACS)

- Intern at UBISOFT Montréal (Supervisor: Wahab Hamou-Lhadj)
- Game server logging analysis
- Anomaly detection models using supervised ML (Random Forest)
- I developed an approach for labeling anomalies from log aggregations
- **The model predicted anomalies in the system ~ 30 min. earlier than engineers!**

2020

New results are coming!!!

Dataset of Video Game Development Problems

MSR - Data Showcase

Who *Cristiano Politowski, Fabio Petrillo, Yann-Gaël Guéhéneuc, Gabriel Cavalheiro Ullmann, Josias De Andrade Werly*

Track MSR 2020 Data Showcase

When **Mon 29 Jun 2020 14:30 - 14:40 at MSR:Zoom** - Bugs & Issues Chair(s): Francisco Servant

Abstract Different from traditional software development, there is little information about the software-engineering process and techniques in video-game development. One popular way to share knowledge among the video-game developers' community is the publishing of postmortems, which are documents summarizing what happened during the video-game development project. However, these documents are written without formal structure and often providing disparate information. Through this paper, we provide developers and researchers with grounded dataset describing software-engineering problems in video-game development extracted from postmortems. We created the dataset using an iterative method through which we manually coded more than 200 postmortems spanning 20 years (1998 to 2018) and extracted 1,035 problems related to software engineering while maintaining traceability links to the postmortems. We grouped the problems in 20 different types. This dataset is useful to understand the problems faced by developers during video-game development, providing researchers and practitioners a starting point to study video-game development in the context of software engineering.

Cristiano Politowski Author
Concordia University, Canada
Canada

Fabio Petrillo Author
University of Quebec at Chicoutimi
Canada

Yann-Gaël Guéhéneuc Author
Concordia University and Polytechnique Montréal
Canada

Gabriel Cavalheiro Ullmann Author
UNIJUI - Universidade Regional do Noroeste do Estado do Rio Grande do Sul
Brazil

Josias De Andrade Werly Author

Game Industry Problems: an Extensive Analysis on the Gray Literature

Cristiano Politowski[a,*], Fabio Petrillo[b], Gabriel C. Ullmann[c] and Yann-Gaël Guéhéneuc[a]

[a] *Concordia University, Montreal, Quebec, Canada*
[b] *Université du Québec à Chicoutimi, Chicoutimi, Quebec, Canada*
[c] *Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Santa Rosa, Rio Grande do Sul, Brazil*

ARTICLE INFO

Keywords:
video-game
software
problems
postmortem

IST major revision!

Abstract

Context: Given its competitiveness, the video-game industry has a closed-source culture. Hence, little is known of the problems faced by game developers. However, game developers do share information about their games projects through postmortems, which describe informally what happened during the projects.

Objective: The software-engineering research community and game developers would benefit from a state of the problems of video game development, in particular the problems faced by game developers, their evolution in time, and their root causes. This state of the practice would allow researchers and practitioners to work towards solving these problems.

Method: We analyzed more than 200 postmortems, comprising 927 problems divided in 20 types from 1997 to 2019. Through our analysis, we describe the overall landscape of game industry problems in the past 23 years and how these problems evolved over the years. We also give details on the most common problems, their root causes, and possible solutions. We finally provide recommendations for future projects.

Results: We observe that (1) the game industry suffer from management and production problems in the same proportion; (2) management problems decreased over the years giving space to business problems, while production problems remained constant; (3a) technical and game design problems are decreasing over the years, the later only after the last decade; (3b) problems related to the team increase over the last decade; (3c) marketing problems are the ones that had the biggest increase over the 23 years compared to other problem types; (4) finally, the majority of the main root causes are related to people, not technologies.

Conclusions: In this paper we provide a state of the practice for researchers to understand and study video-game development problems. We also offer recommendations to help practitioners to avoid the most common problems in future projects.

Are Game Engines Software Frameworks? A Three-perspective Study*

Cristiano Politowski[a,*], Fabio Petrillo[b], João Eduardo Montandon[c], Marco Tulio Valente[c] and Yann-Gaël Guéhéneuc[a]

[a] *Concordia University, Montreal, Quebec, Canada*
[b] *Université du Québec à Chicoutimi, Chicoutimi, Quebec, Canada*
[c] *Universidade Federal de Minas Gerais, Belo Horizonte, Brazil*

ARTICLE INFO

Keywords:
Game-engine
Framework
Video-game
Mining
Open-source

JSS Minor revision!

Abstract

Game engines help developers create video games and avoid duplication of code and effort, like frameworks for traditional software systems. In this paper, we explore open-source game engines along three perspectives: literature, code, and human. First, we explore and summarise the academic literature on game engines. Second, we compare the characteristics of the 282 most popular engines and the 282 most popular frameworks in GitHub. Finally, we survey 124 engine developers about their experience with the development of their engines. We report that: (1) Game engines are not well-studied in software-engineering research with few studies having engines as object of research. (2) Open-source game engines are slightly larger in terms of size and complexity and less popular and engaging than traditional frameworks. Their programming languages differ greatly from frameworks. Engine projects have shorter histories with less releases. (3) Developers perceive game engines as different from traditional frameworks. Generally, they build game engines to (a) better control the environment and source code, (b) learn about game engines, and (c) develop specific games. We conclude that open-source game engines have differences compared to traditional open-source frameworks although this differences do not demand special treatments.

1. Introduction

> *"It's hard enough to make a game (...). It's really hard to make a game where you have to fight your own tool set all the time."*
> — Schreier [1] quoting a game developer on the

tended from open-source ones, or bought off the shelves. They are essential to game development but misunderstood and misrepresented by the media [11] and developers due to lacks of clear definitions, architectural references [12], and academic studies. They are also the source of problems, especially between design and technical teams [13, 14].

SE for Computer Games has tons of **research opportunities**!!!


So, let's work **together**....

Main research projects

Game testing

- Game testing is an intensive, **manual human labor**
- Build models and new techniques for computer game testing
- Applying automatic transformations and targets the early detection of **regression bugs**.
- Machine learning techniques and new approaches to automatically test large number of scenarios to **reduce the costs of manual testing**.

Game engine architectures

- Extension of our JSS paper to explore **architectural aspects** of game engines
- Reverse engineering
- Architectural recovering challenges

Computer game debugging

- New techniques for computer game debugging
- Using Swarm Debugging for CG
- Crowd approaches to address debugging challenges in software development
- The effort to debug CGs using a collaborative approach is a research opportunity to explore.

Build empirical theories and quality models for computer games

- A deep comprehension of testing and debugging phenomena for CGs opens an opportunity to explore **new theories and quality models for CG**
- Practices and human factors in SEGA (creativity vs. technical aspects);
- Data-driven and machine learning SE to improve software quality practices in CGs;
- Building software **quality models** to support CG.

Game as a service

- Game server technologies
  - Agones - https://github.com/GoogleCloudPlatform/agones
- Cloud computing and serverless
- Scaling, load balancing and resource optimization
- CI/CD challenges

Vision for the next 5/10 years

Enhance SE for Computer Games theories and practices to improve engineers' life!

Vision for the next 5/10 years

Become the **research leader** on SE for Computer Games in Canada!

Highlights

Don’t worry, a research idea **takes time** to become mainstream (decades)...

Research ideas pop up **everywhere**... So, listen carefully your colleagues, clients, users, etc...

Computer games are a great **sandbox and playground** to state-of-art in SE and Computer Science.

Mindset changes are in progress in Computer Games industry to **embrace the agility** and improve their process....

Scope management is the most important problem in Computer Games yet.

Tons of research opportunities, specially in terms of architecture, testing, reliability, CI/CD, and observability.

So, let's work together....

Some references and resources

- AWS re:Invent 2018: Epic Games Uses AWS to Deliver Fortnite to 200 Million Players at https://aws.amazon.com/solutions/case-studies/EPICGames/
- Newzoo 2018 Global games market report, Trends, insights, and projections towards 2021
- https://www.svg.com/108467/untold-truth-fortnite/?utm_campaign=clip
- Fortnite: An Unconventional Launch at https://youtu.be/j5rO7dKKhMo
- What game developers can learn from Fortnite's rise at https://appdevelopermagazine.com/what-game-developers-can-learn-from-fortnite's-rise/
- Fortnite may improve our development cycles, July 11, 2018 by frost front games, posted in blog - https://frostfrontpodcast.com/2018/07/11/fortnite-may-improve-our-development-cycles/
- The Secrets Behind Fortnite's Massive 40 Million Player Success at https://kotaku.com/the-secrets-behind-fortnites-massive-40-million-player-1822119730

Challenges and Opportunities on Software Engineering for Computer Games

Fabio Petrillo
fabio.petrillo@uqac.ca
fabio@petrillo.com
@drfabiopetrillo (twitter)