SUMMER SCHOOL REANIMATE

2024 Summer School on Retro Gaming History, Critic, and Development

Challenges and Opportunities on Software Engineering for Computer Games

Fabio Petrillo

École de Technologie Supérieure
Université du Québec - Canada

June 2024

ÉTS
Le génie pour l'industrie

ÉTS
Le génie pour l'industrie

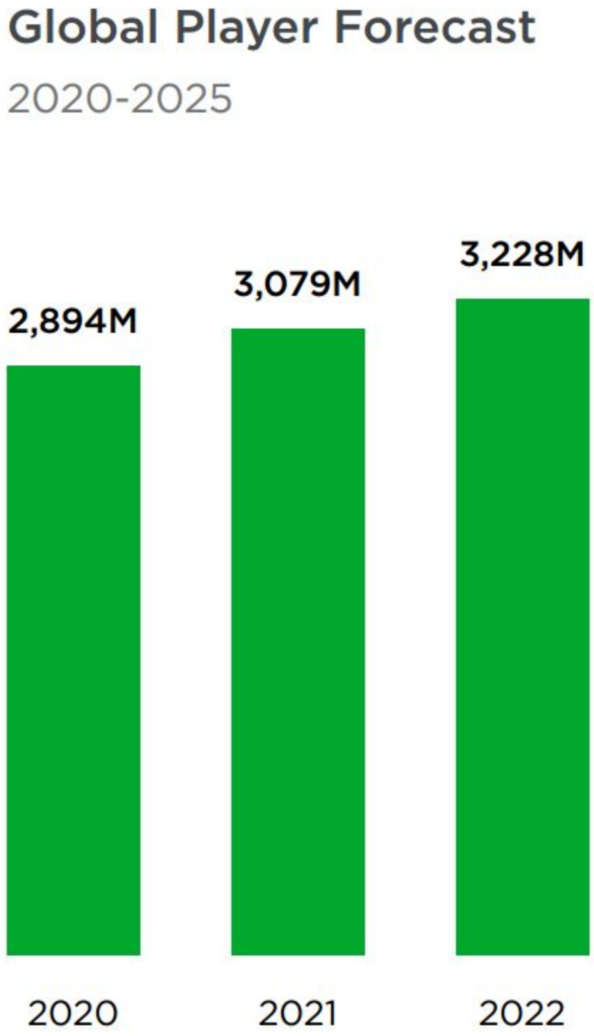
Fabio PETRILLO

- **Associate Professor at LOGTI - ÉTS (2022)**
  - **SE for Computer Games, Software Quality and Architecture**
- Master and Ph.D. in Computer Science (UFRGS/Brazil)
  - **Agile methods for computer game**
  - Analysis, comprehension et visualisation of software
  - **Swarm Debugging**
- Professeur à l'UQAC (2018 - 2022)
- Lecturer at Polytechnique Montréal
- Postdoctoral Fellow at Concordia University (Montréal)
  - **UBISOFT** Montréal (MITACS) - Logging analysis and anomaly detection
- > 20 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?

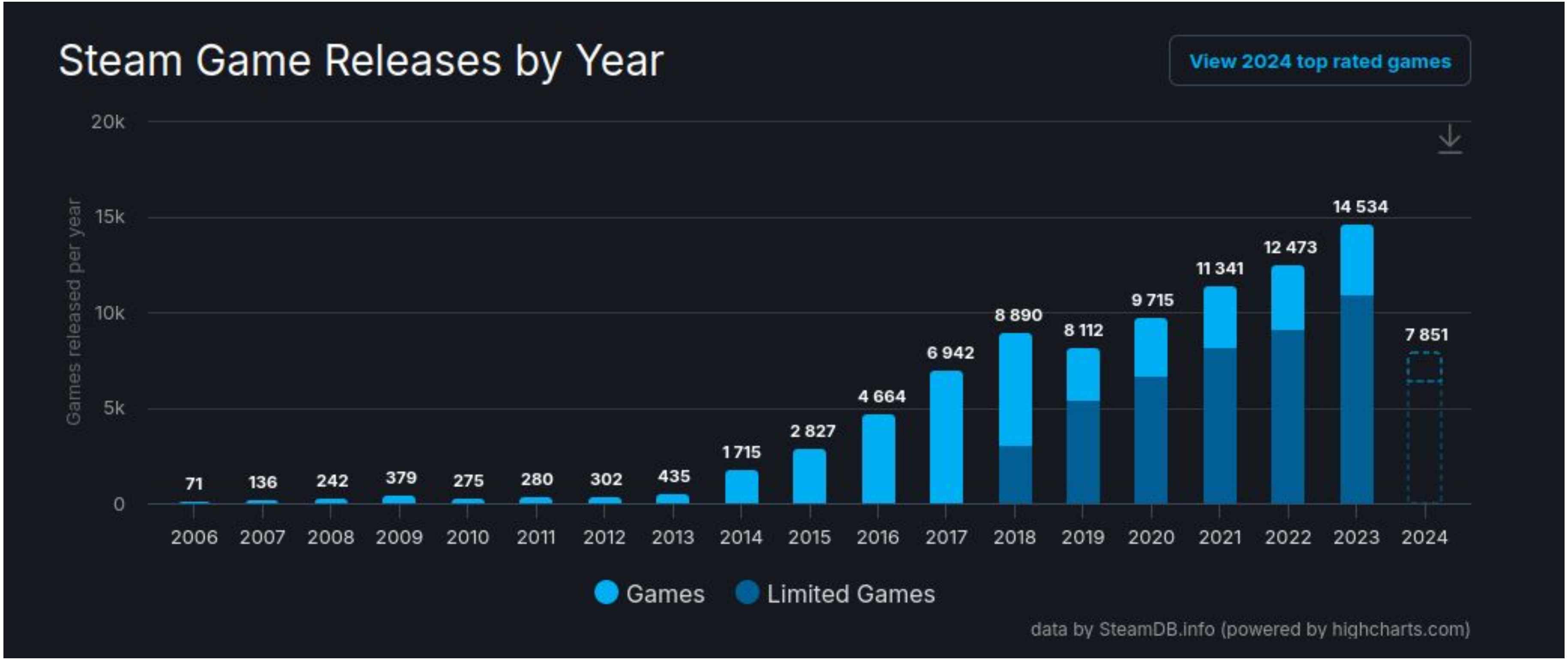
Game industry is billionaire, greater than cinema and music **together**

(NEWZOO,2016)



3.2 Billion of players in 2022!

Games Released on Steam by Year (June 2024)



https://steamdb.info/stats/releases/

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

Game Industry 2023

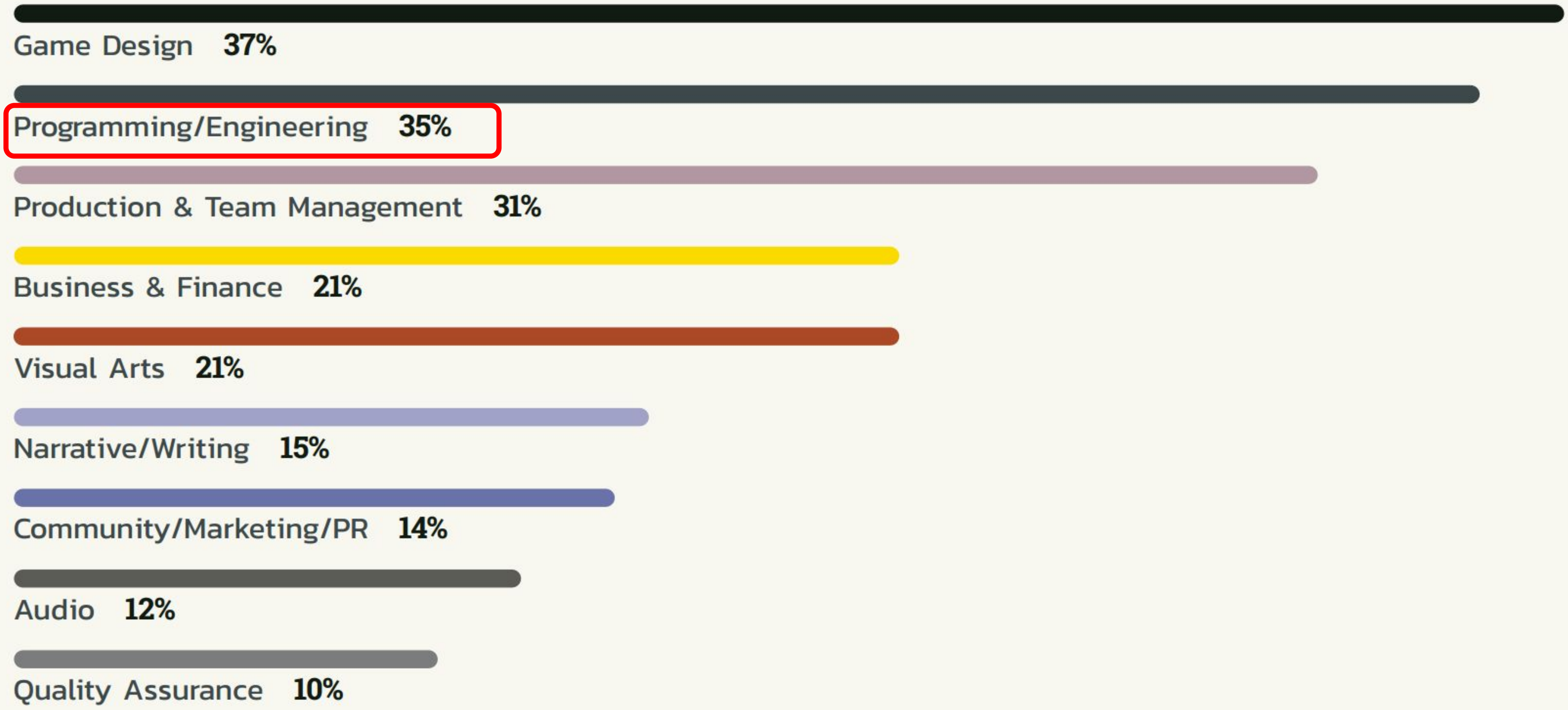
Presented by GDC Game Developer

This past year has been a time of change and opportunity. Studios and companies are working towards a new normal, while developers have their eye on how to reshape the game industry and their role within it. The metaverse has become

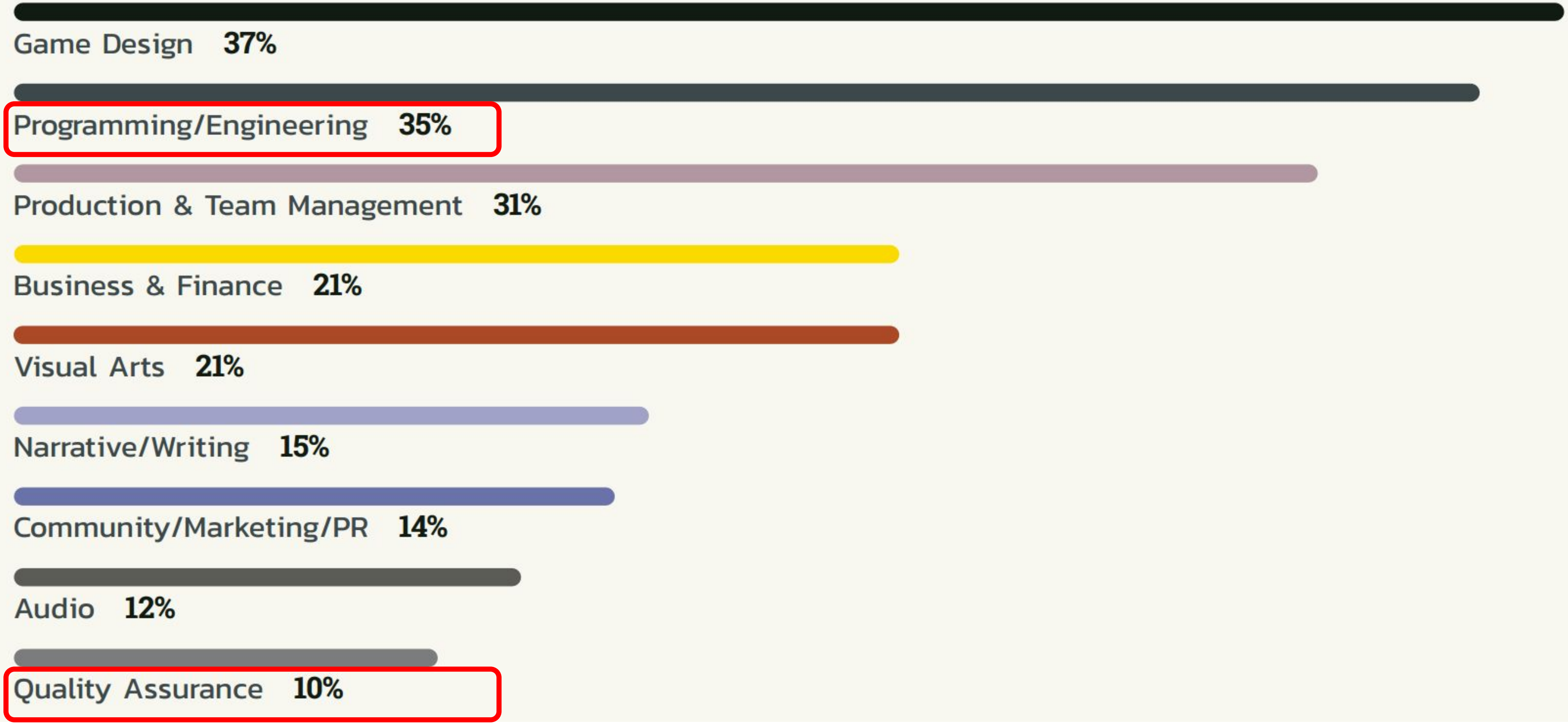
GDC Game Industry 2023

A survey of **2,300 video game developers** about their work and the industry.

What best describes your job role? (Choose all that apply)



What best describes your job role? (Choose all that apply)



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]

Computer Games and Software Engineering

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

Software Engineering for Computer Games (SEGA)

Process

The Old Day are Gone? Software Engineering Processes in Video Game Industry
GAS 2016

Process recommendation system for video game projects
IST 2018

Video Game Project Management Anti-patterns
GAS 2022

What Makes a Game High-rated? Towards Factors of Video Game Success
GAS 2022

Problems and Techniques

What went wrong? A survey of problems in game development
SAC 2008/Computers in Entertainment 2009

Dataset of Video Game Development Problems
MSR 2020

Game industry problems: An extensive analysis of the gray literature
IST 2021

Are Game Engines Software Frameworks?
IST 2021

Solutions

Is agility out there?: agile practices in game development
SIGDOC 2010

A Survey of Video Game Testing
AST 2021

Towards Automated Video Game Testing: Still a Long Way to Go
GAS 2022

Assessing Video Game Balance using Autonomous Agents
GAS 2023

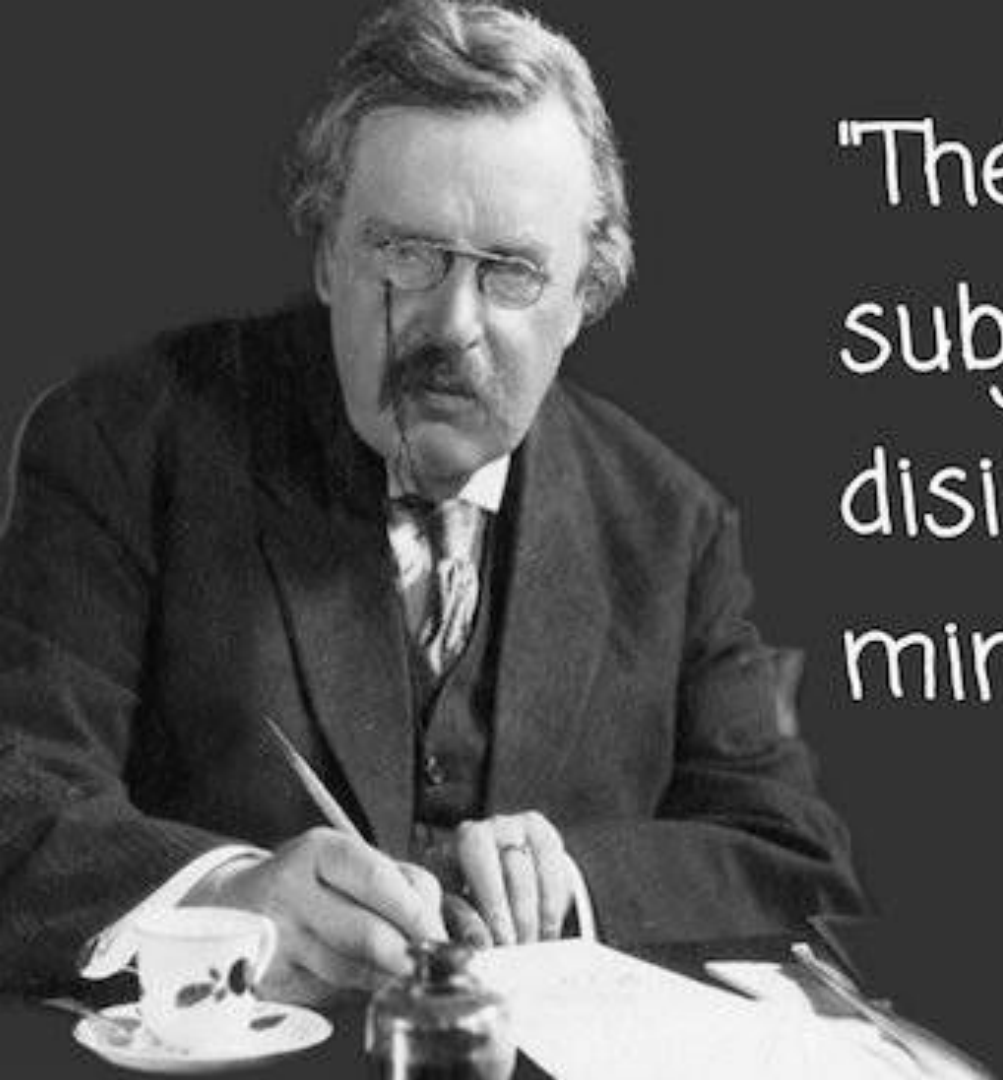
I have worked on SEGA since 2007 ...

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

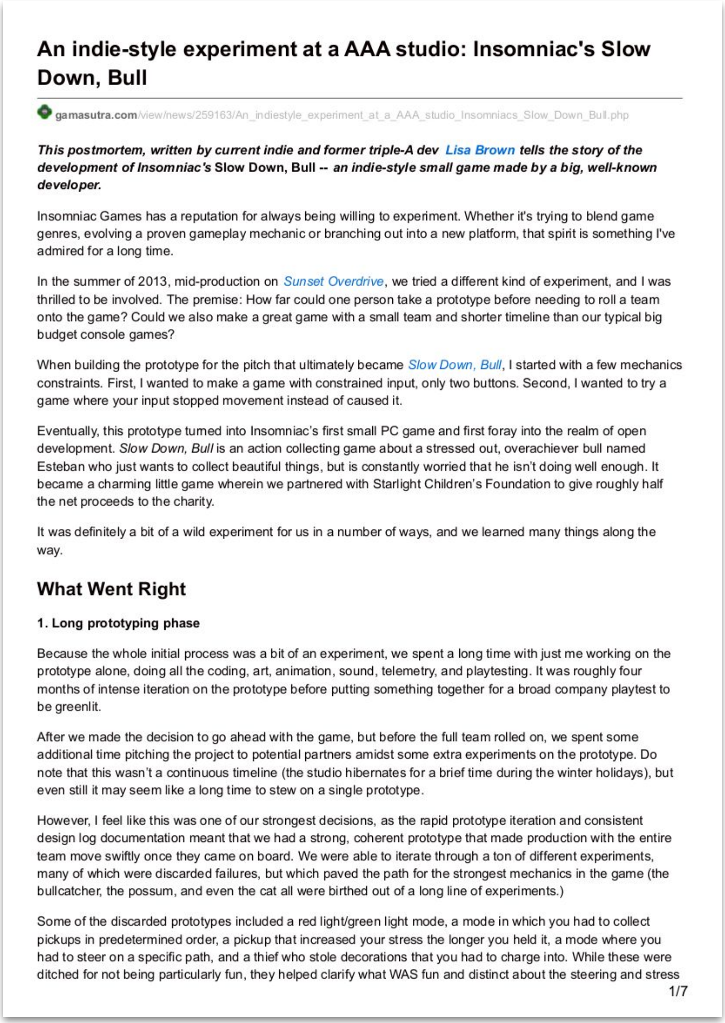
Does game industry have the same **problems** that "traditional" software industry?

20 game industry postmortems

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

1/7

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.

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?

Are the Old Days Gone? A Survey on Actual Software Engineering Processes in Video Game Industry

Cristiano Politowski, Lisandra Fontoura
Federal University of Santa Maria
Santa Maria, Brazil
{cpolitowski,lisandra}@inf.ufsm.br

Fabio Petrillo, Yann-Gaël Guéhéneuc
École Polytechnique de Montréal
Montréal, Canada
fabio@petrillo.com,
yann-gael.gueheneuc@polymtl.ca

ABSTRACT

In the past 10 years, several researches studied video game development process who proposed approaches to improve the way how games are developed. These approaches usually adopt agile methodologies because of claims that traditional practices and the waterfall process are gone. However, are the "old days" really gone in the game industry?

In this paper, we present a survey of software engineering processes in video game industry from postmortem project analyses. We analyzed 20 postmortems from Gamasutra Portal. We extracted their processes and modelled them through using the Business Process Model and Notation (BPMN).

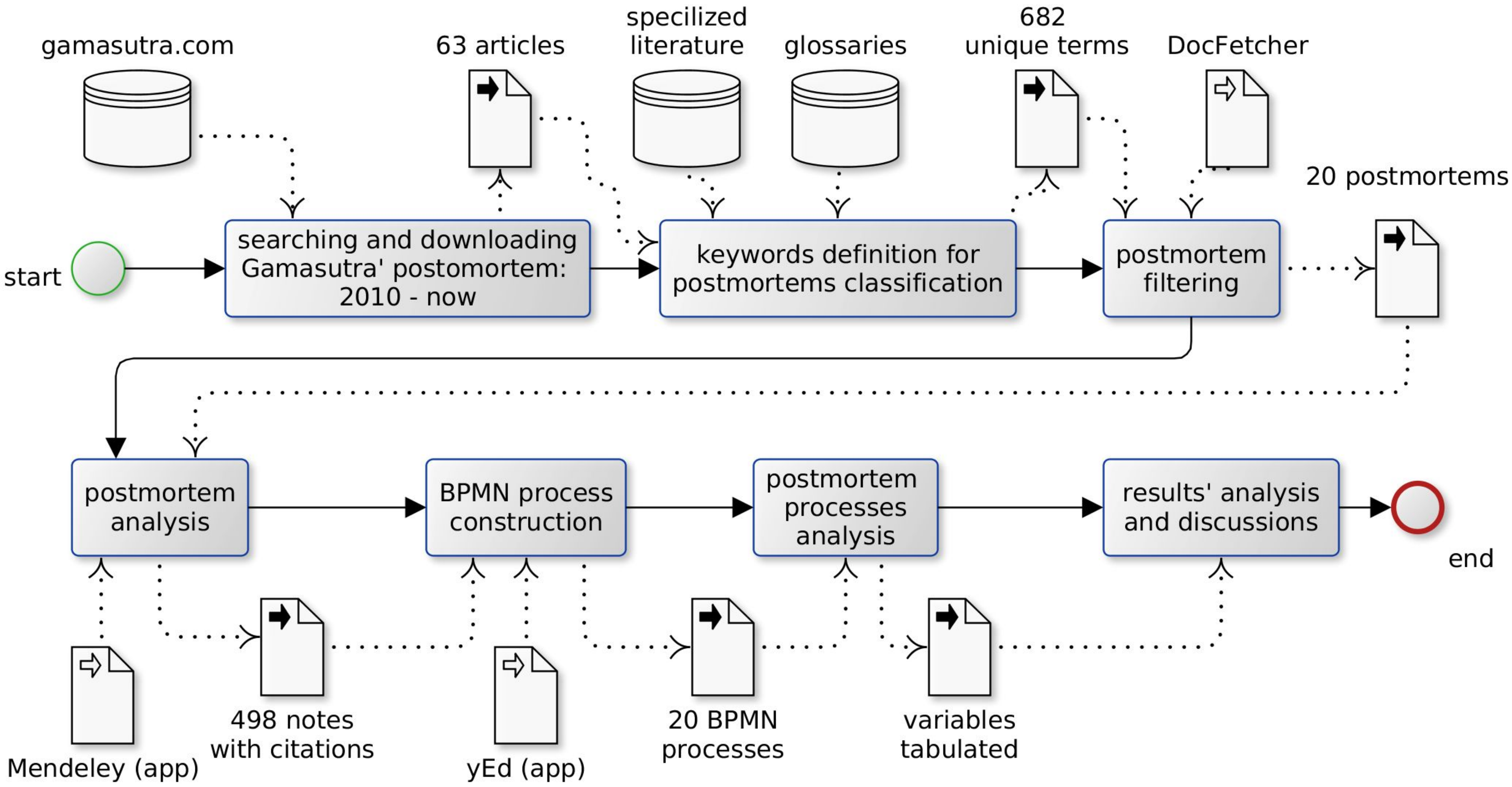
This work presents three main contributions. First, a

stantially fewer industrial studies about game development processes and claims that agile processes are appropriate when innovation and speed to market are vital in game development. In the same direction, a developer gave us some interesting observations in a recent postmortem about game development [9]:

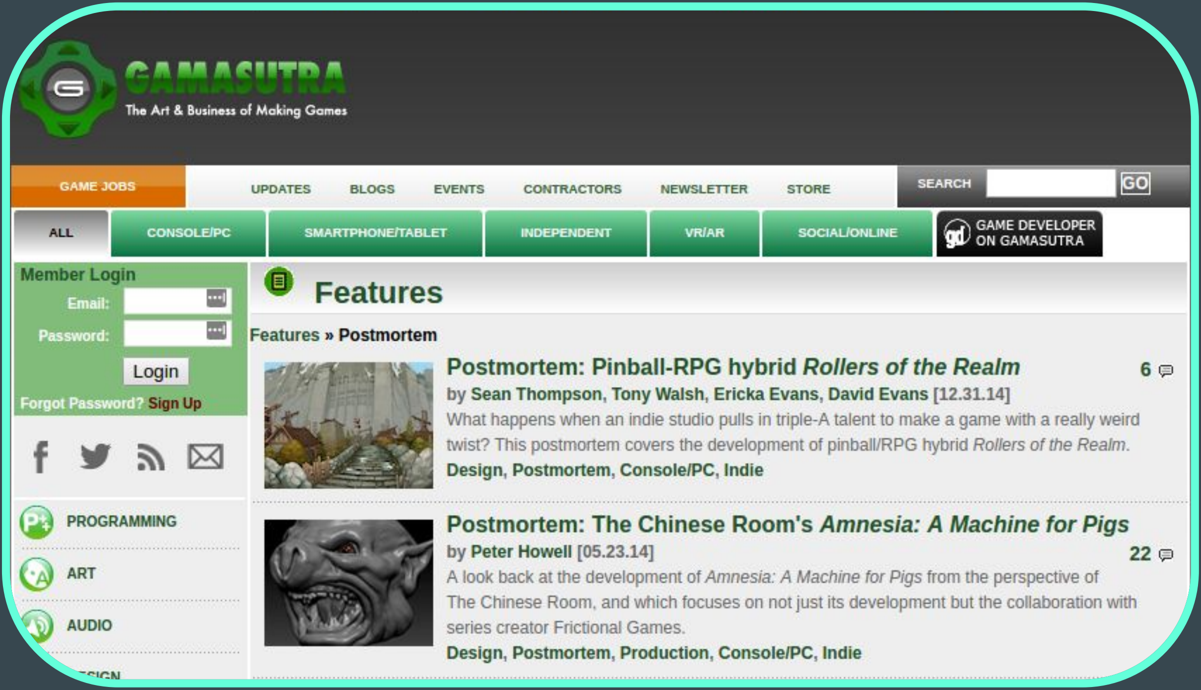
> *"**The old days are gone**. You can't expect producers or leads to come up with a huge waterfall of everything they thought would get done over the next three years. In the game development business, it's insane to think you have any insight into what your team will be doing one year from now. You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

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

Methodology


gamasutra.com
63 articles
specilized literature
glossaries
682 unique terms
DocFetcher
start
searching and downloading Gamasutra' postomortem: 2010 - now
keywords definition for postmortems classification
postmortem filtering
20 postmortems
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)



Features

Features » Postmortem

Postmortem: Pinball-RPG hybrid *Rollers of the Realm* 6

by Sean Thompson, Tony Walsh, Ericka Evans, David Evans [12.31.14]

What happens when an indie studio pulls in triple-A talent to make a game with a really weird twist? This postmortem covers the development of pinball/RPG hybrid *Rollers of the Realm*.

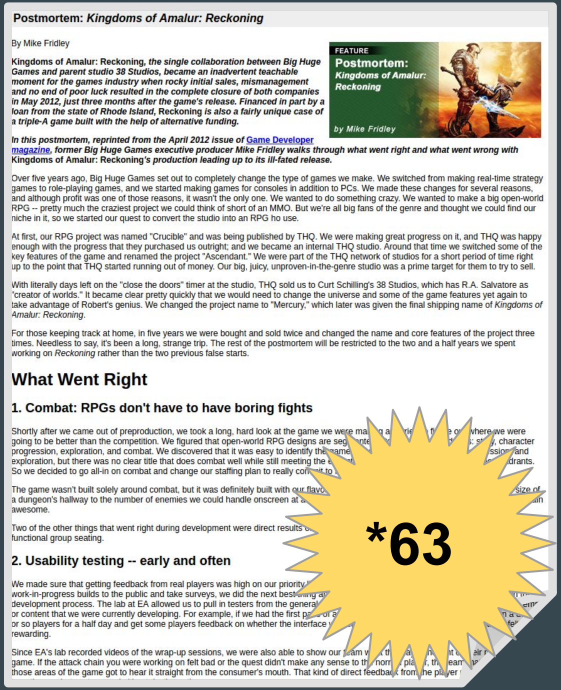
Design, Postmortem, Console/PC, Indie

Postmortem: The Chinese Room's *Amnesia: A Machine for Pigs* 22

by Peter Howell [05.23.14]

A look back at the development of *Amnesia: A Machine for Pigs* from the perspective of The Chinese Room, and which focuses on not just its development but the collaboration with series creator Frictional Games.

Design, Postmortem, Production, Console/PC, Indie



Postmortem: *Kingdoms of Amalur: Reckoning*

By Mike Fridley

***Kingdoms of Amalur: Reckoning**, the single collaboration between Big Huge Games and parent studio 38 Studios, became an inadvertent teachable moment for the games industry when rocky initial sales, mismanagement and no end of poor luck resulted in the complete closure of both companies in May 2012, just three months after the game's release. Financed in part by a loan from the state of Rhode Island, **Reckoning** is also a fairly unique case of a triple-A game built with the help of alternative funding.*

*In this postmortem, reprinted from the April 2012 issue of Game Developer magazine, former Big Huge Games executive producer Mike Fridley walks through what went right and what went wrong with **Kingdoms of Amalur: Reckoning**'s production leading up to its ill-fated release.*

Over five years ago, Big Huge Games set out to completely change the type of games we make. We switched from making real-time strategy games to role-playing games, and we started making games for consoles in addition to PCs. We made these changes for several reasons, and although profit was one of those reasons, it wasn't the only one. We wanted to do something crazy. We wanted to make a big open-world RPG -- pretty much the craziest project we could think of short of an MMO. But we're all big fans of the genre and thought we could find our niche in it, so we started our quest to convert the studio into an RPG ho use.

At first, our RPG project was named "Crucible" and was being published by THQ. We were making great progress on it, and THQ was happy enough with the progress that they purchased us outright; and we became an internal THQ studio. Around that time we switched some of the key features of the game and renamed the project "Ascendant." We were part of the THQ network of studios for a short period of time right up to the point that THQ started running out of money. Our big, juicy, unproven-in-the-genre studio was a prime target for them to try to sell.

With literally days left on the "close the doors" timer at the studio, THQ sold us to Curt Schilling's 38 Studios, which has R.A. Salvatore as "creator of worlds." It became clear pretty quickly that we would need to change the universe and some of the game features yet again to take advantage of Robert's genius. We changed the project name to "Mercury," which later was given the final shipping name of *Kingdoms of Amalur: Reckoning*.

For those keeping track at home, in five years we were bought and sold twice and changed the name and core features of the project three times. Needless to say, it's been a long, strange trip. The rest of the postmortem will be restricted to the two and a half years we spent working on *Reckoning* rather than the two previous false starts.

What Went Right

1. Combat: RPGs don't have to have boring fights

Shortly after we came out of preproduction, we took a long, hard look at the game we were ma[illegible] where we were going to be better than the competition. We figured that open-world RPG designs are seg[illegible] character progression, exploration, and combat. We discovered that it was easy to identify the game[illegible] and exploration, but there was no clear title that does combat well while still meeting the [illegible] drants. So we decided to go all-in on combat and change our staffing plan to really com[illegible]

The game wasn't built solely around combat, but it was definitely built with our flavo[illegible] size of a dungeon's hallway to the number of enemies we could handle onscreen at [illegible] awesome.

Two of the other things that went right during development were direct results [illegible] functional group seating.

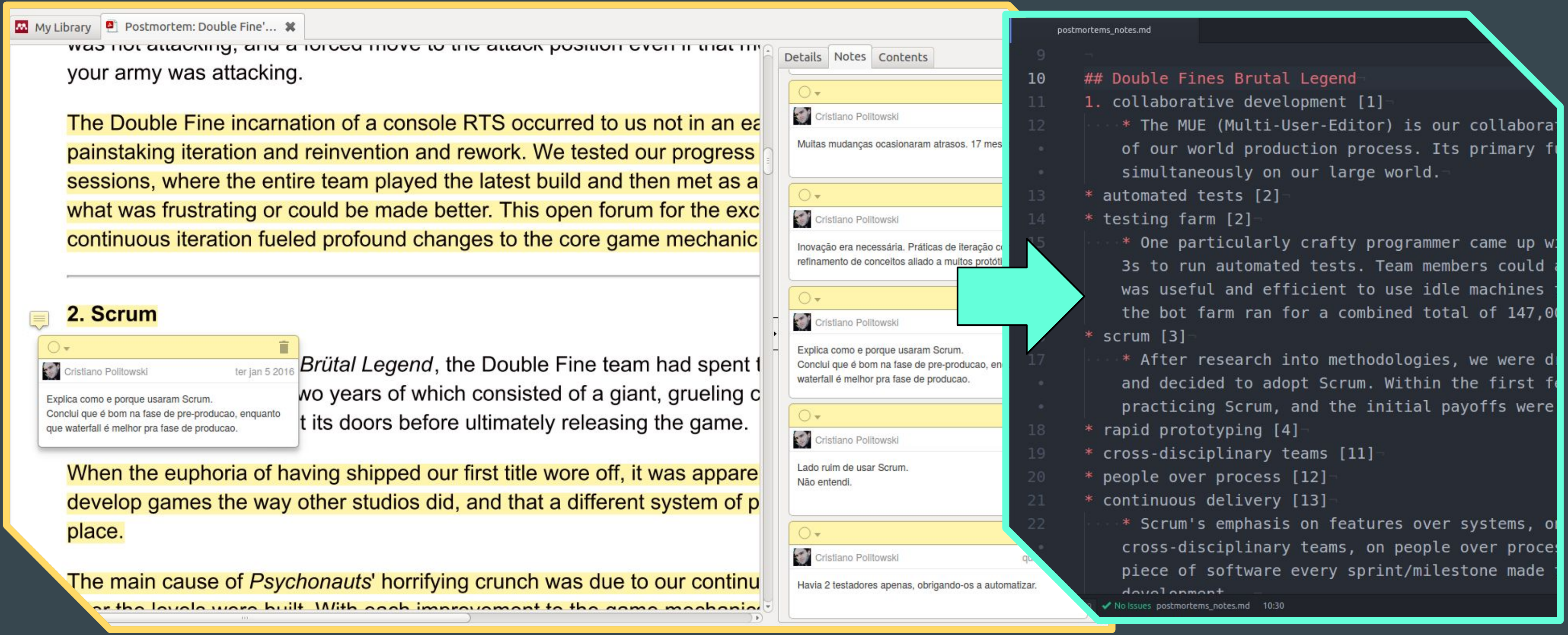
***63**

2. Usability testing -- early and often

We made sure that getting feedback from real players was high on our priority [illegible] work-in-progress builds to the public and take surveys, we did the next best [illegible] development process. The lab at EA allowed us to pull in testers from the general [illegible] or content that we were currently developing. For example, if we had the first pa[illegible] or so players for a half day and get some players feedback on whether the interface [illegible] rewarding.

Since EA's lab recorded videos of the wrap-up sessions, we were also able to show our team [illegible] game. If the attack chain you were working on felt bad or the quest didn't make any sense to th[illegible] those areas of the game got to hear it straight from the consumer's mouth. That kind of direct feedback from the player [illegible]

Postmortem Analysis (20 articles)



was not attacking, and a forced move to the attack position even if that m
your army was attacking.

The Double Fine incarnation of a console RTS occurred to us not in an ea
painstaking iteration and reinvention and rework. We tested our progress
sessions, where the entire team played the latest build and then met as a
what was frustrating or could be made better. This open forum for the exc
continuous iteration fueled profound changes to the core game mechanic

2. Scrum

Cristiano Politowski ter jan 5 2016

Explica como e porque usaram Scrum.
Conclui que é bom na fase de pre-producao, enquanto que waterfall é melhor pra fase de producao.

Brütal Legend, the Double Fine team had spent t
wo years of which consisted of a giant, grueling c
t its doors before ultimately releasing the game.

When the euphoria of having shipped our first title wore off, it was appare
develop games the way other studios did, and that a different system of p
place.

The main cause of *Psychonauts*' horrifying crunch was due to our continu
er the levels were built. With each improvement to the game mechanic

Cristiano Politowski

Muitas mudanças ocasionaram atrasos. 17 mes

Cristiano Politowski

Inovação era necessária. Práticas de iteração c
refinamento de conceitos aliado a muitos protóti

Cristiano Politowski

Explica como e porque usaram Scrum.
Conclui que é bom na fase de pre-producao, en
waterfall é melhor pra fase de producao.

Cristiano Politowski

Lado ruim de usar Scrum.
Não entendi.

Cristiano Politowski qu

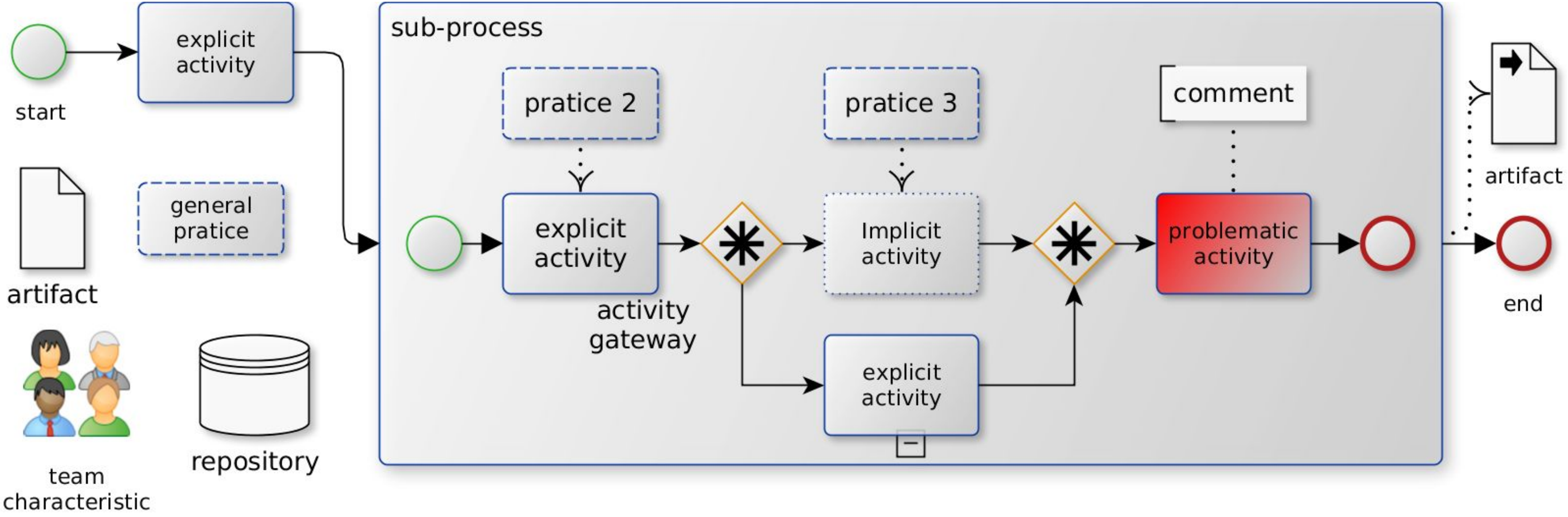
Havia 2 testadores apenas, obrigando-os a automatizar.

postmortems_notes.md

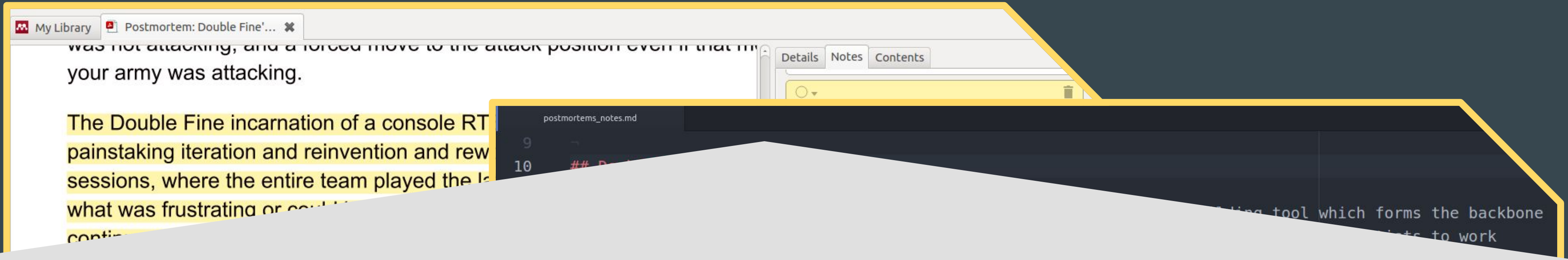
```
## Double Fines Brutal Legend
1. collaborative development [1]
    * The MUE (Multi-User-Editor) is our collabora
      of our world production process. Its primary fu
      simultaneously on our large world.
* automated tests [2]
* testing farm [2]
    * One particularly crafty programmer came up wi
      3s to run automated tests. Team members could a
      was useful and efficient to use idle machines t
      the bot farm ran for a combined total of 147,00
* scrum [3]
    * After research into methodologies, we were di
      and decided to adopt Scrum. Within the first fe
      practicing Scrum, and the initial payoffs were
* rapid prototyping [4]
* cross-disciplinary teams [11]
* people over process [12]
* continuous delivery [13]
    * Scrum's emphasis on features over systems, on
      cross-disciplinary teams, on people over proces
      piece of software every sprint/milestone made t
      development
```

✓ No Issues postmortems_notes.md 10:30

Process Metamodel



Postmortem Analysis (20 articles)

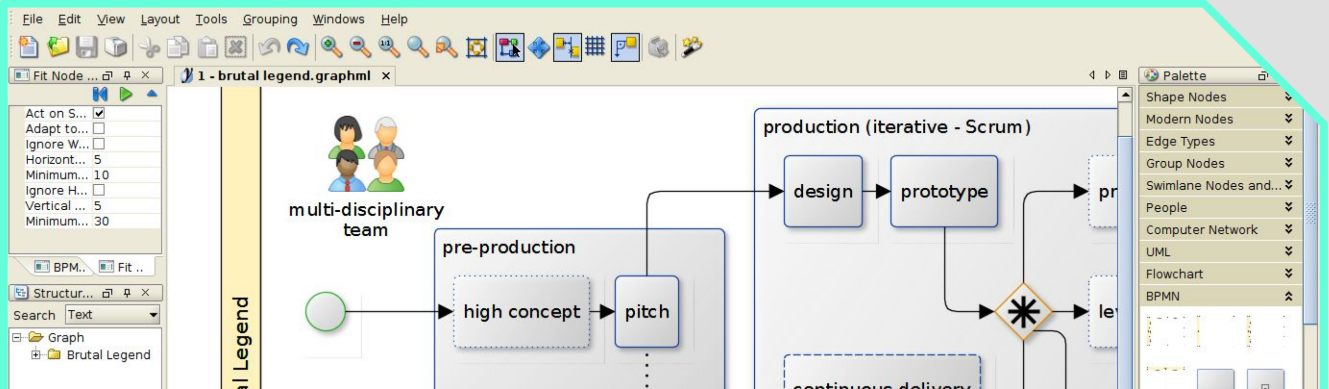


was not attacking, and a forced move to the attack position even if that m
your army was attacking.

The Double Fine incarnation of a console RT
painstaking iteration and reinvention and rew
sessions, where the entire team played the l
what was frustrating or cou
contin

```
## D
ling tool which forms the backbone
ists to work
```

BPMN Process Construction

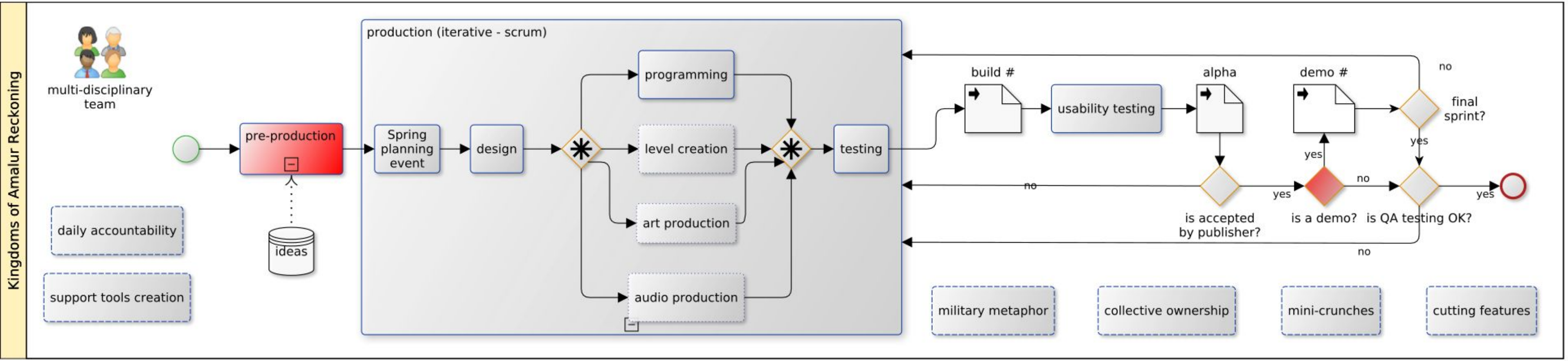


Act on S... ☑
Adapt to... ☐
Ignore W... ☐
Horizont... 5
Minimum... 10
Ignore H... ☐
Vertical ... 5
Minimum... 30

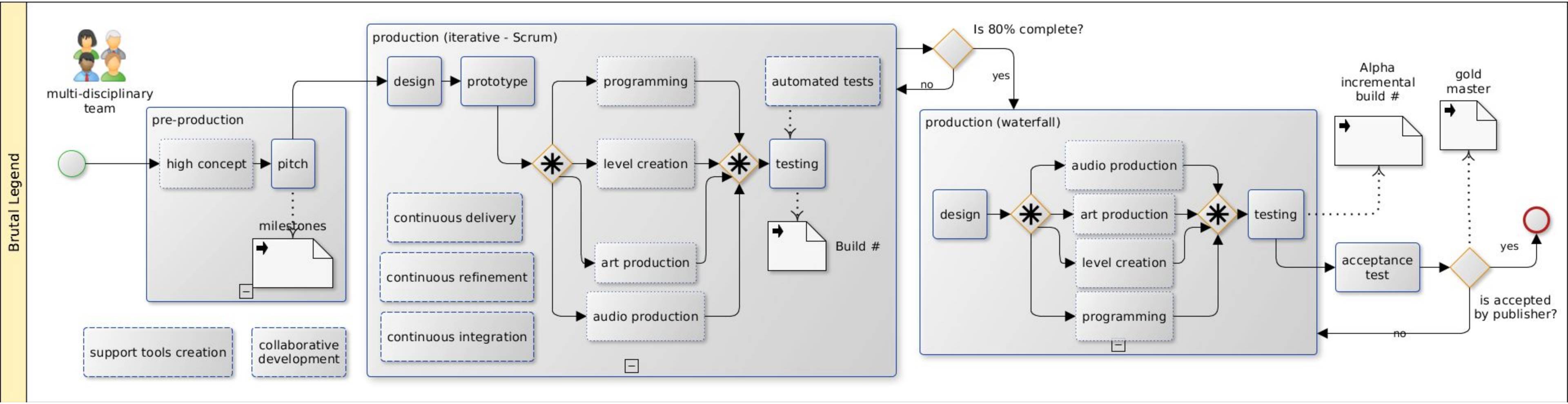
Search Text

20 process models (BPMN)

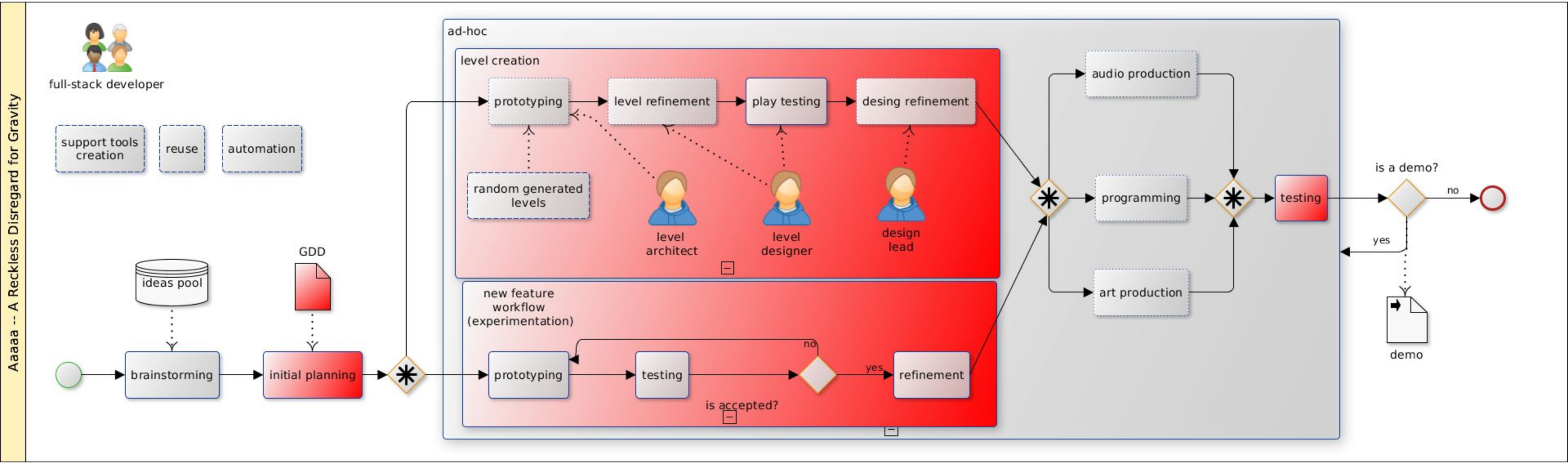
Iterative Process - *Kingdoms of Amalur: Reckoning*



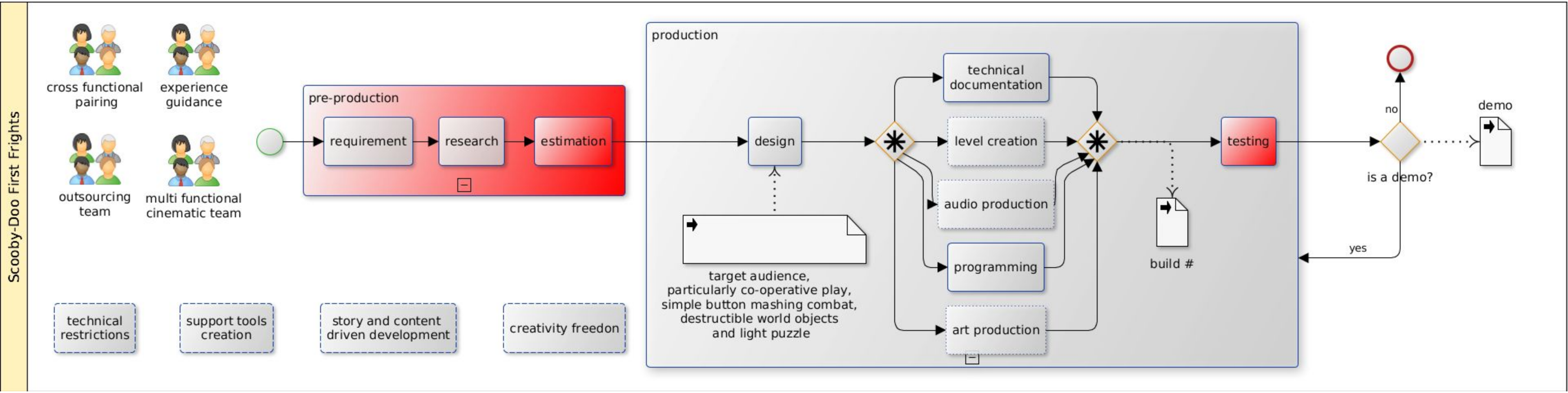
Hybrid Process - *Brutal Legend*



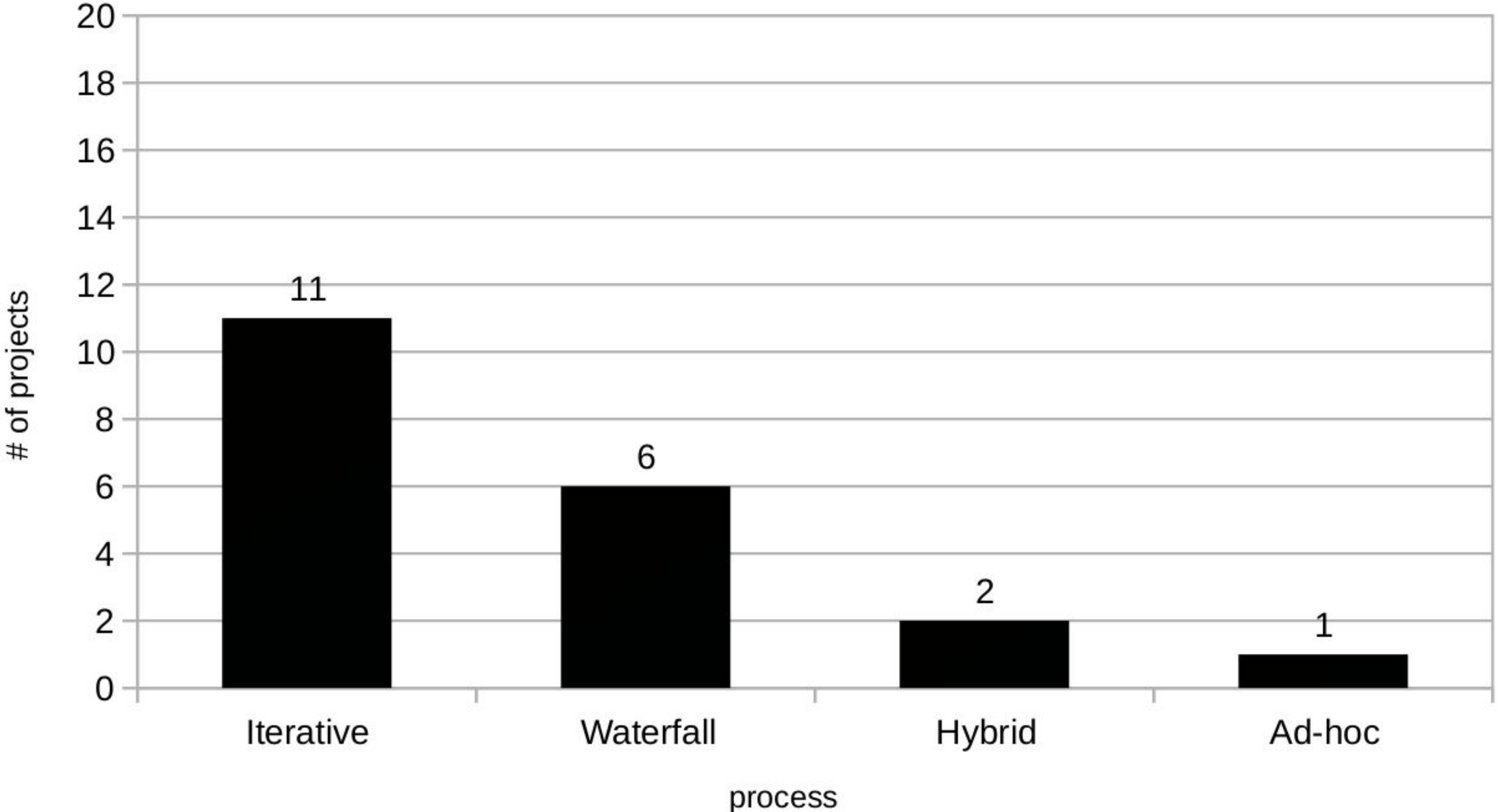
Ad-Hoc Process - Aaaa! *A Reckless Disregard for Gravity*



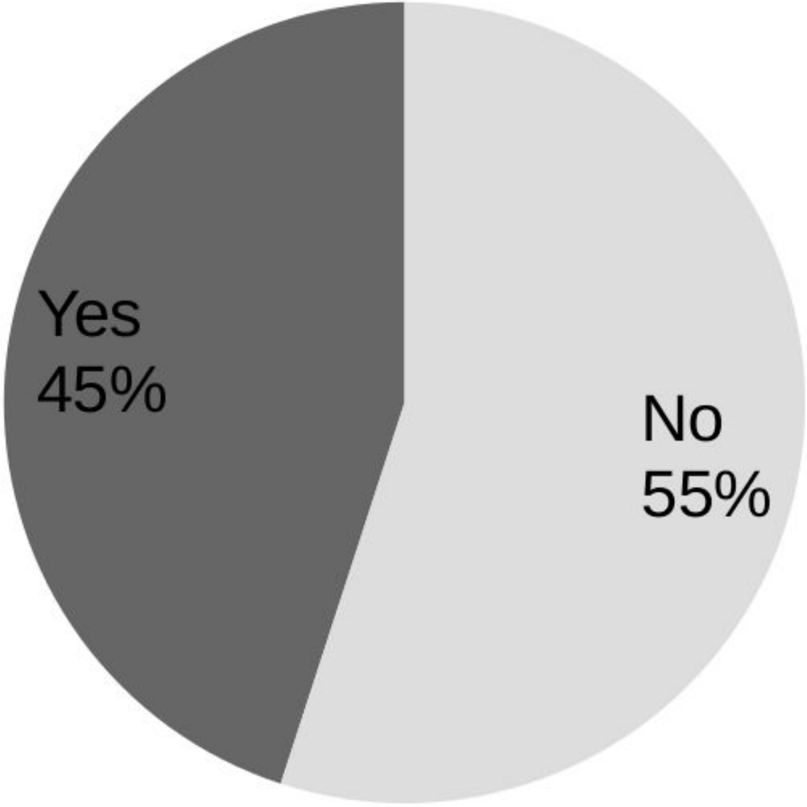
Waterfall Process - *Scooby-Doo*



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.

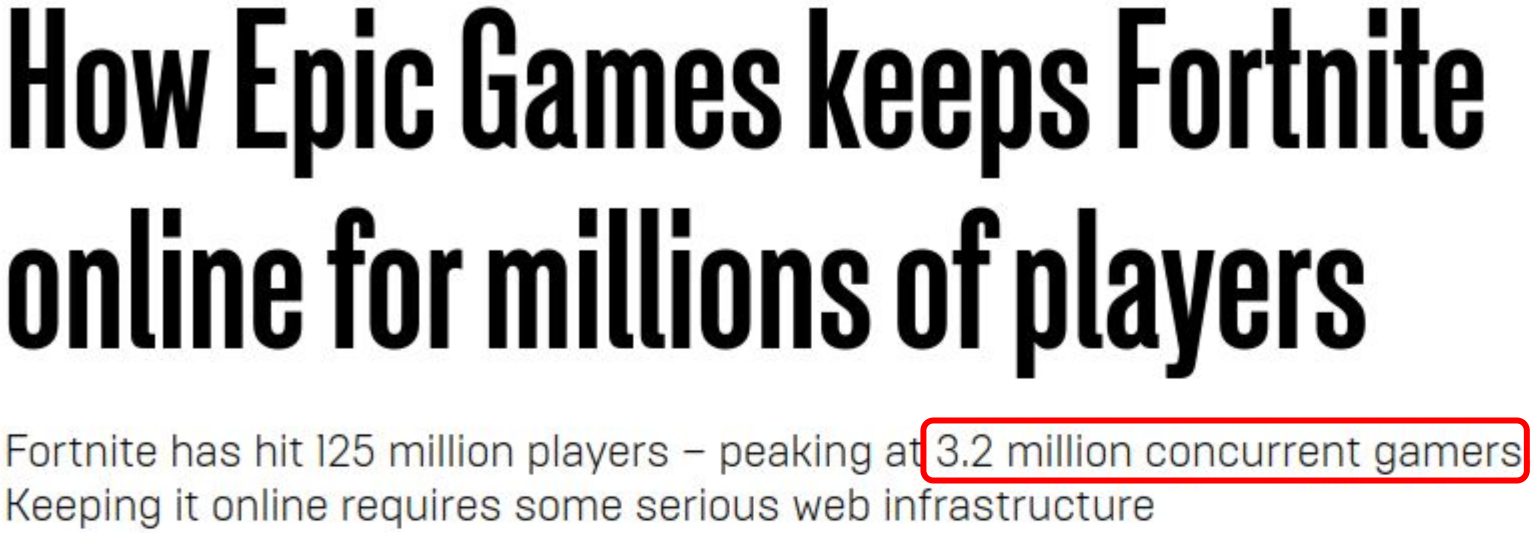
Discussion (2016)

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

.

2018

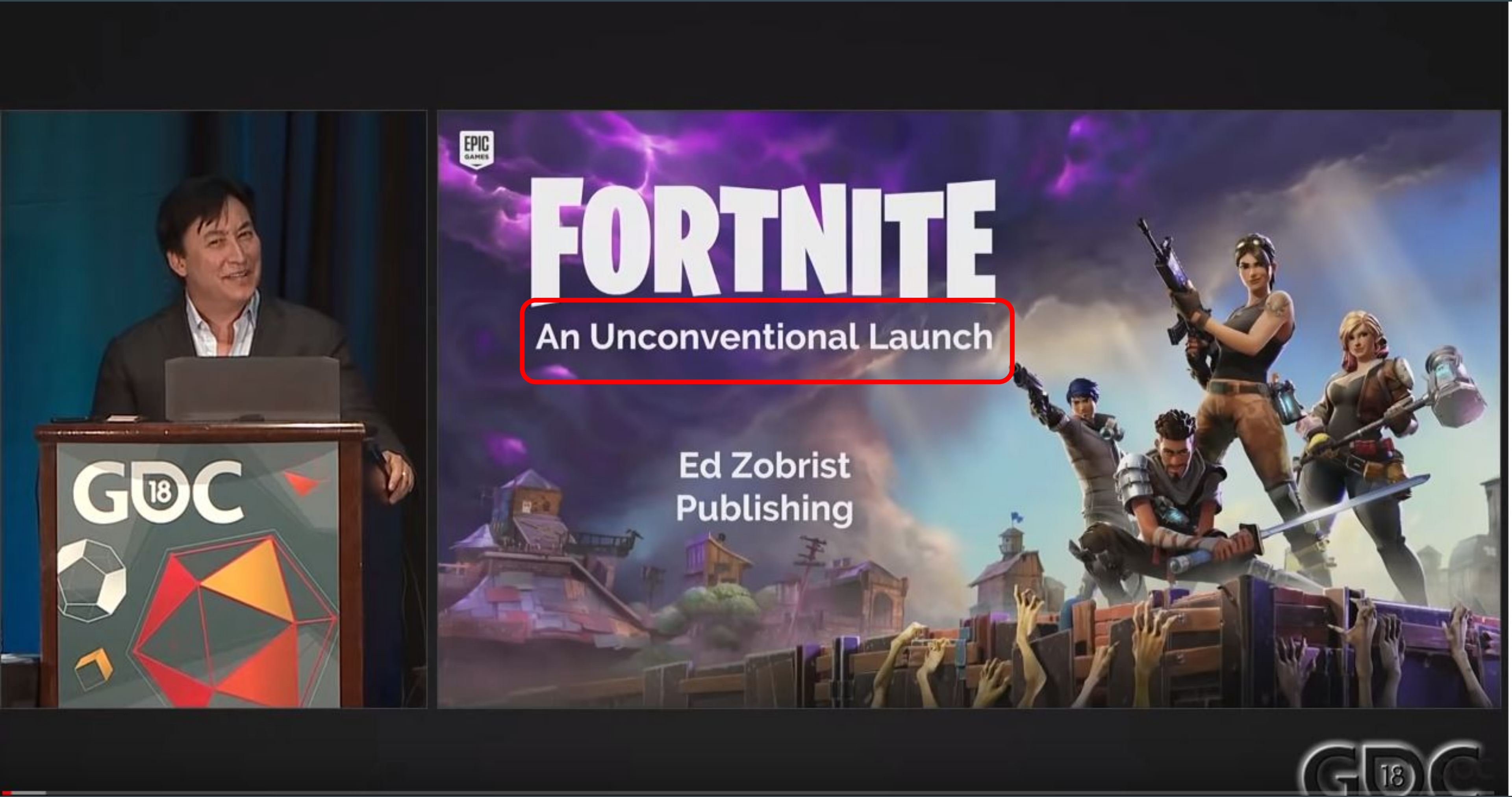
FORTNITE
BATTLE
ROYALE



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

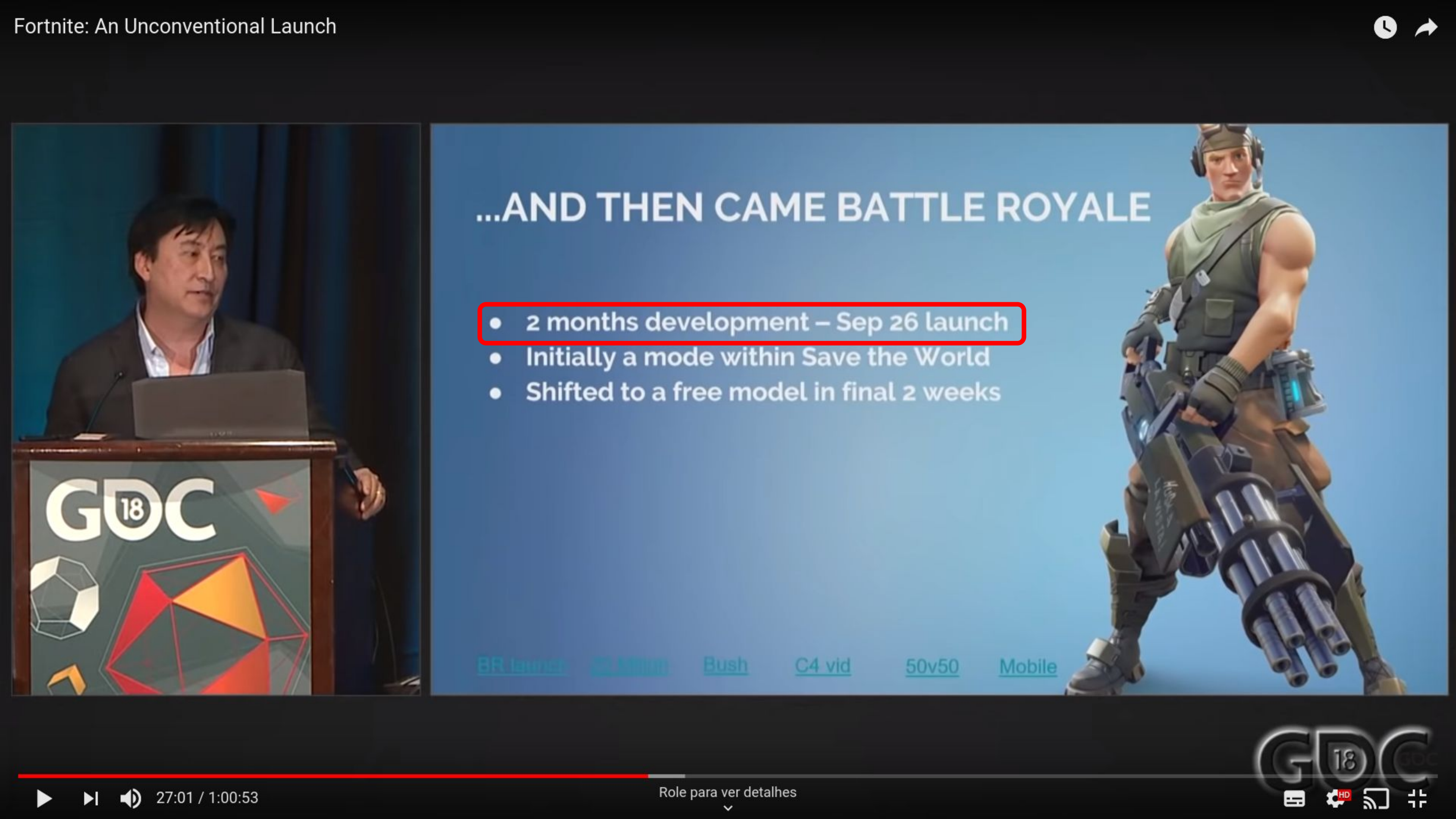
Why Fortnite: Battle Royale is a huge success???



FORTNITE

An Unconventional Launch

Ed Zobrist
Publishing



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   20 Million   Bush   C4 vid   50v50   Mobile

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.**

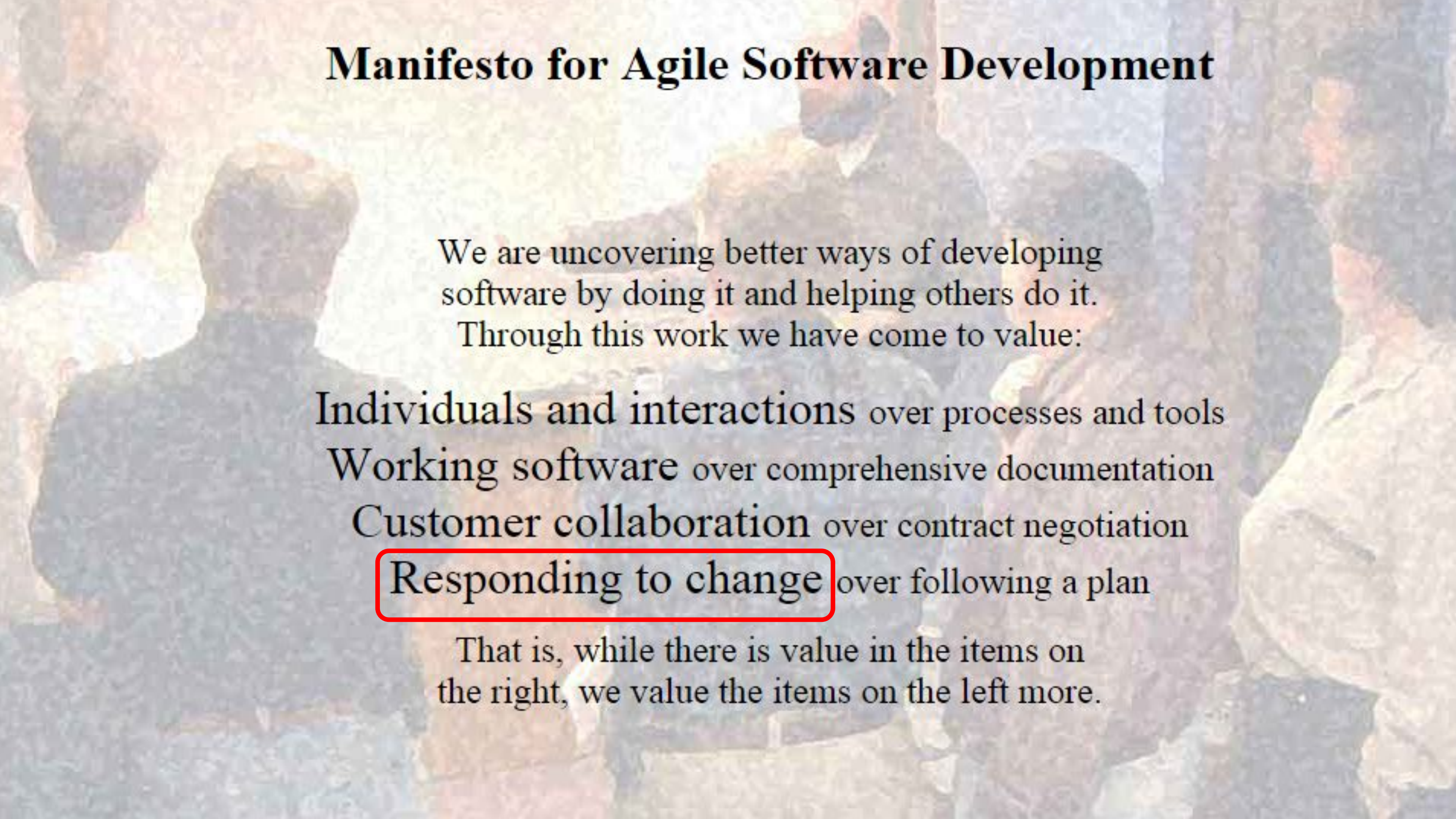
Why Fortnite: Battle Royale is a huge success?

“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

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.

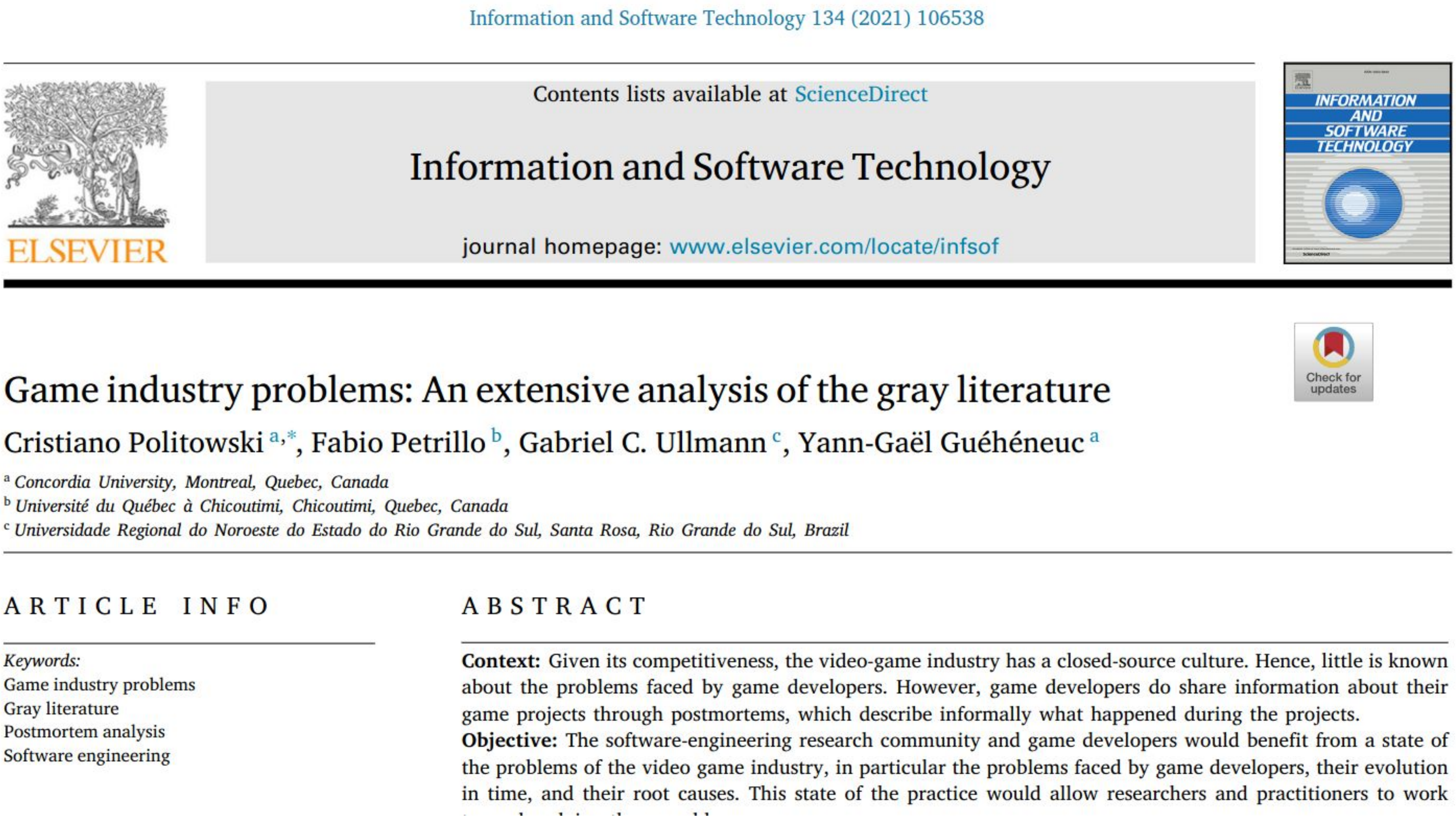
Fortnite success (finally) opens opportunities to explore agile practices on AAA computer games' industry...

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

New games follow similar strategies, such as …

GENSHIN
IMPACT

2020-23



Information and Software Technology 134 (2021) 106538

Contents lists available at ScienceDirect

Information and Software Technology

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

Game industry problems: An extensive analysis of the gray literature

Cristiano Politowski [a,*], Fabio Petrillo [b], Gabriel C. Ullmann [c], 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:
Game industry problems
Gray literature
Postmortem analysis
Software engineering

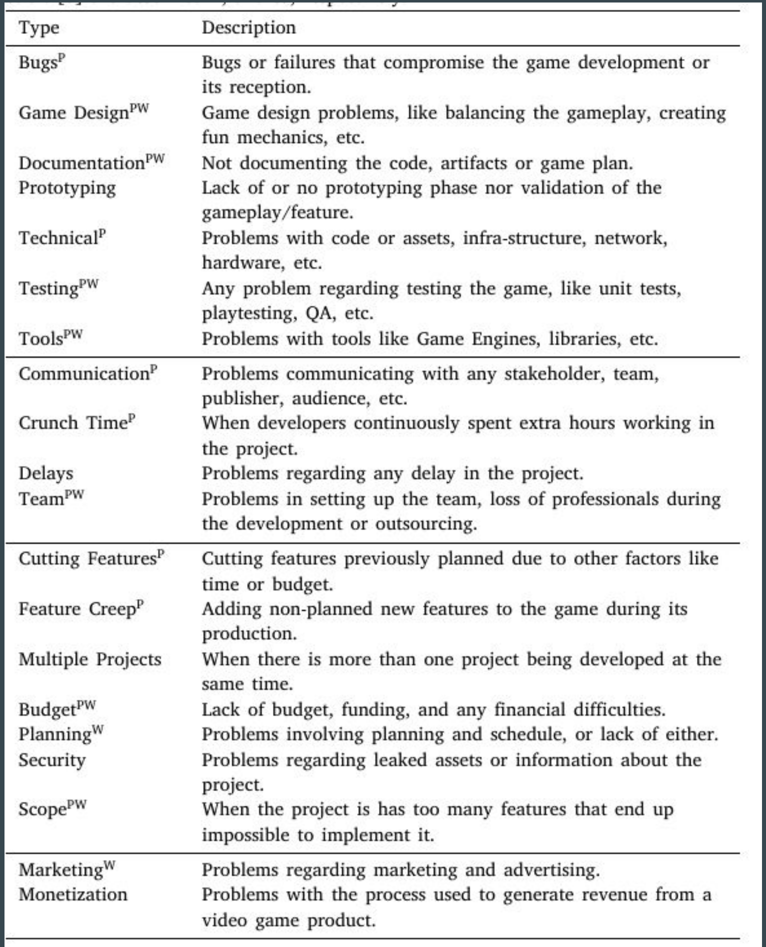
ABSTRACT

Context: Given its competitiveness, the video-game industry has a closed-source culture. Hence, little is known about the problems faced by game developers. However, game developers do share information about their game 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 the video game industry, 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

Game industry problems: An extensive analysis of the gray literature

- **New version of my paper 2009 journal paper (10 years later)**
- We analyzed more than 200 postmortems from 1997 to 2019
- 927 problems, divided in 20 types
- **The main root causes are related to human aspects, not technical ones**



Type	Description
Bugs^{P}	Bugs or failures that compromise the game development or its reception.
Game Design^{PW}	Game design problems, like balancing the gameplay, creating fun mechanics, etc.
Documentation^{PW}	Not documenting the code, artifacts or game plan.
Prototyping	Lack of or no prototyping phase nor validation of the gameplay/feature.
Technical^{P}	Problems with code or assets, infra-structure, network, hardware, etc.
Testing^{PW}	Any problem regarding testing the game, like unit tests, playtesting, QA, etc.
Tools^{PW}	Problems with tools like Game Engines, libraries, etc.
Communication^{P}	Problems communicating with any stakeholder, team, publisher, audience, etc.
Crunch Time^{P}	When developers continuously spent extra hours working in the project.
Delays	Problems regarding any delay in the project.
Team^{PW}	Problems in setting up the team, loss of professionals during the development or outsourcing.
Cutting Features^{P}	Cutting features previously planned due to other factors like time or budget.
Feature Creep^{P}	Adding non-planned new features to the game during its production.
Multiple Projects	When there is more than one project being developed at the same time.
Budget^{PW}	Lack of budget, funding, and any financial difficulties.
Planning^{W}	Problems involving planning and schedule, or lack of either.
Security	Problems regarding leaked assets or information about the project.
Scope^{PW}	When the project is has too many features that end up impossible to implement it.
Marketing^{W}	Problems regarding marketing and advertising.
Monetization	Problems with the process used to generate revenue from a video game product.

Highlights

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…”

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
- Cloud computing and serverless
- Scaling, load balancing and resource optimization
- CI/CD challenges

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.

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

Games is an amazing tech industry to create new opportunities and jobs using **open source platforms**!

Let’s work together....

Challenges and Opportunities on Software Engineering for Computer Games

Thanks a lot!

Fabio Petrillo
fabio.petrillo@etsmtl.ca
fabio@petrillo.com
Twitter: @drfabiopetrillo