



UBISOFT®

---

On the way to good

GAME DESIGN

# About me



Andrii Dotsenko

Game Designer at Ubisoft Kiev

11 years in game design

Far Cry 4, Far Cry: Primal



# From theory to practice

We love to talk about design **theory**

There are many design talks about system design, UX, narrative, or new monetization methods. But we often missed the path from design to implementation when you as a designer have to interact with other people in order to implement your ideas.

But you need others to **execute** your ideas

# Why these games are better?



# What's so special about them?



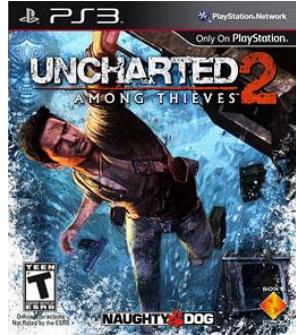
Different genres

Different platforms

Different audience and market

One similarity: **quality of experience**

# What is the key?



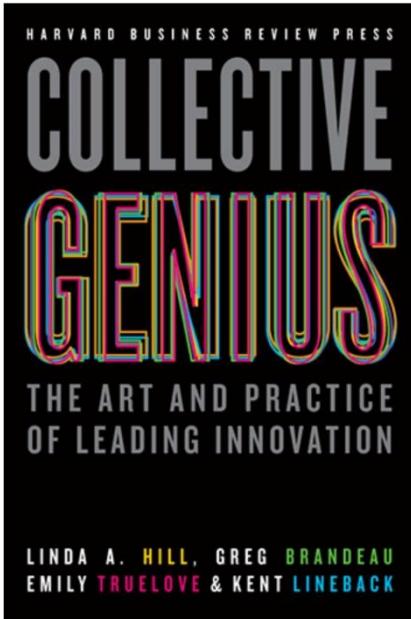
Ideas?

Talents?

Marketing?

Production culture is the key?

# Collective Genius



It's not just about **talent**

There are some very successful innovative companies (systemically successful, like Pixar). And again, the key is always production culture.

But about talent in proper **context**

# Successful innovation principles

## Creative Abrasion

Ability to create a market of ideas through discussions. Diversity + Conflict.

## Creative Agility

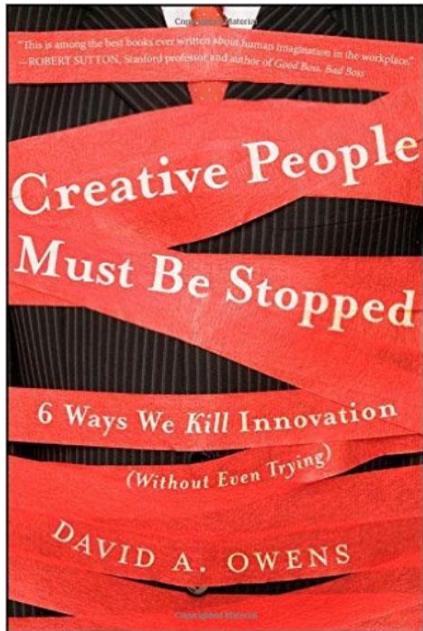
Ability to test and refine ideas through the discovery-driven learning process. Scientific Method + Artistic Process.

## Creative Resolution

Ability to make decisions in a way that can combine different ideas and approaches to produce a new solution.

\*TED talk: Linda Hill, ‘How to manage for collective creativity’

# Creative people must be stopped



Proper principles are **not enough**

There are many obstacles on the innovation's way. Society always acts like creative people must be stopped.

You should learn how to overcome **constraints**

\*Coursera: Leading Strategic Innovation in Organizations

# Can we use it?

Design can be **different**

Perhaps, we can use these principles? Yes, our design and games are almost always new, but all “human factors” are very similar from project to project.

Can we learn how they work and then learn how we can overcome them?

But people are the **same!**

# Game Design Framework

*Stage 1*

**DESIGN**

How to create design

*Stage 2*

**COMMUNICATION**

How to communicate design

*Stage 3*

**IMPLEMENTATION**

How to implement design

*Stage 1*

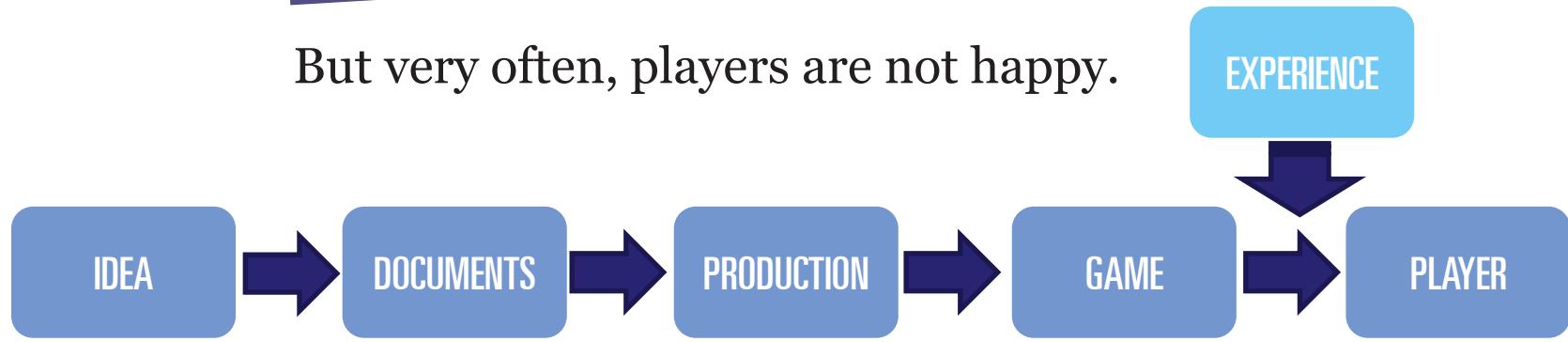
---

# DESIGN PRINCIPLES

# What's wrong?

Everything seems right?

But very often, players are not happy.



Something missed here. What is it?

# It's all about player experience



... that comes from the **interaction**

It doesn't matter how good are your ideas, documents and code. The game appears **only** after the interaction.

The player experience is what we **really** sell

# From experience to design



It all starts from the **experience idea**

Every game starts from the experience idea, from the core fantasy that you want to bring to the player.

You need to turn it into **design**

# Macro Design

# Answers to question WHAT?

The high-level vision of the feature that describes the player **experience**, should be created **before** the implementation, and has **very few changes** during the implementation.

**Has an extremely high cost of the mistake.**

# Macro Design format

# Macro Design should be **short**

Good Macro Design is clear, short and has a high level of abstraction.

Formats can be different: Creative Brief, 1-pager, table of levels, etc.

This is your answer “Where do we go?”

# And with proper abstraction level



# Micro Design



## Answers to question HOW?

A detailed description of the features that takes in account all technical restrictions. Should be created **during** the implementation.

Might have **many iterative changes**.

# Micro Design format



Micro Design should be **short** and **specific**

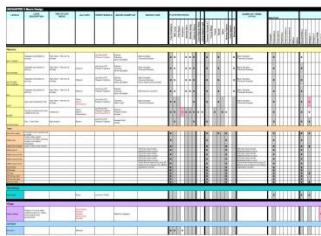
Micro Design should be as specific as possible, and take in account details and edge cases.

Formats: tables, wikis, to-do lists, mock-ups.

Just enough to **keep tracking**.

# Macro Design vs. Micro Design

80% of Design is an iterative **LIVE** process



20%



80%

20% of WHAT (macro design) and 80% of HOW (micro design).

80% of design made during the implementation!

Game Design is about **execution!**

# ► Obstacles on your way

---



# Bad Macro Design



Wrong abstraction level

Unclear or not polished

Misinterpreted

Can lead to **grave problems** with the game scope

# Solution: Clarity

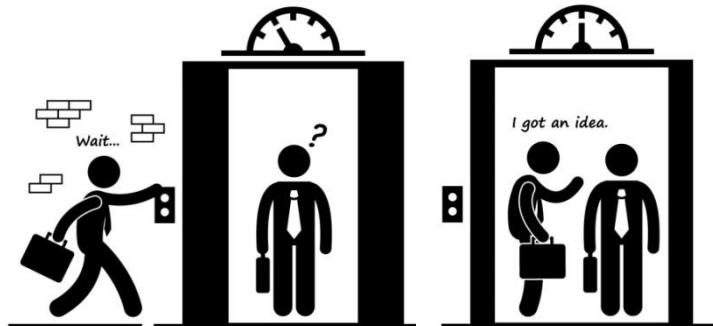
# **CLARITY** is the key

Invest more efforts to reach clear  
Macro Design.

Support your text by visuals to avoid misinterpretation, texts can be deceiving.

# Clarity can be developed!

# Clarity exercise



Use 'elevator pitch' technique

Write your idea in four short sentences. **Or less.** Imagine that you have ~30 sec to tell them.

If you cannot write it clear enough in four sentences or less – your idea is bad.



Support your idea by **visuals**

# Macro Design reduces risks



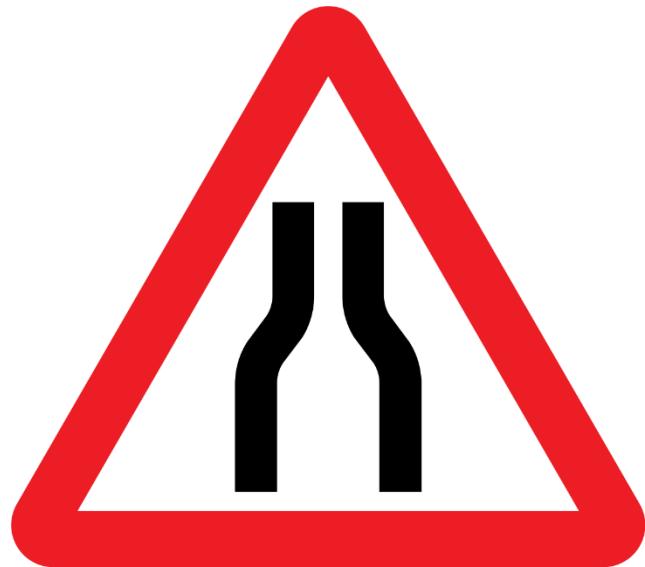
It might be not enough to make a **document**

You build an interactive experience,  
and words and pictures are **not**  
**interactive**.

You may need to test your Macro  
Design to **reduce risks**.

You may need **the prototype**

# Bad Micro Design



Not ready

Don't take details into account

Too abstract

Can lead to design **bottleneck**

# ► Solution: Timing & Clarity



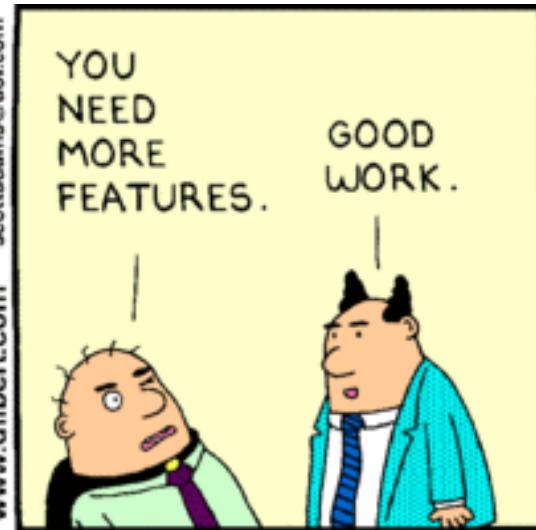
Develop **CLARITY**

Clarity and visuals are also important for Micro Design.

But there's another main point: Micro Design is iterative and constantly changing, so **timing** is very important.

Be just **far enough** ahead of the team

# Feature Creep



# Feature Creep



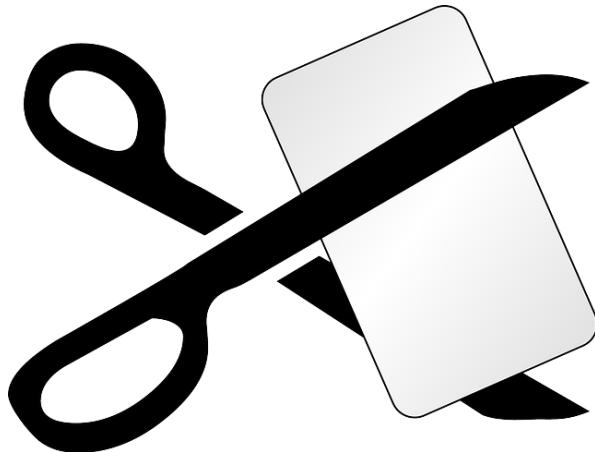
'Design by committee'

Some features are more equal than others

May increase complexity exponentially

The **main source** of scope problems

# ► Solution: Feature Cut



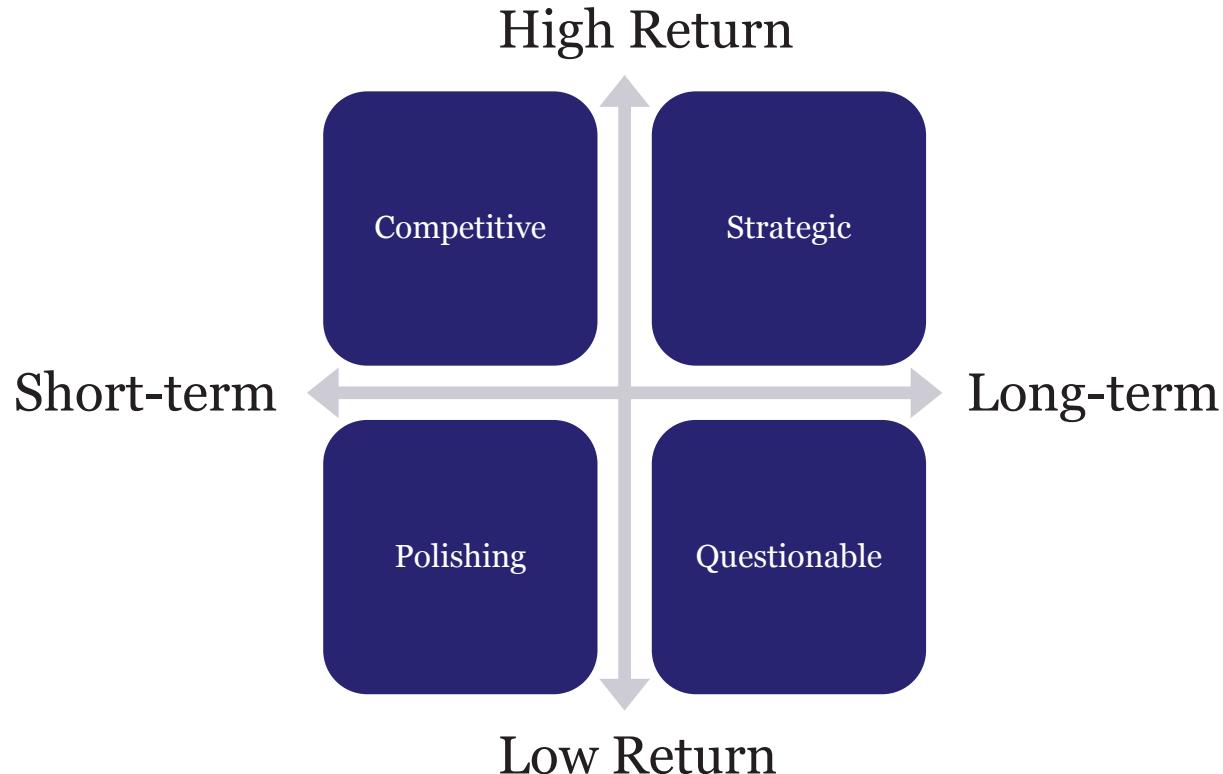
Less is more.

It's better to make fewer features but with higher quality.

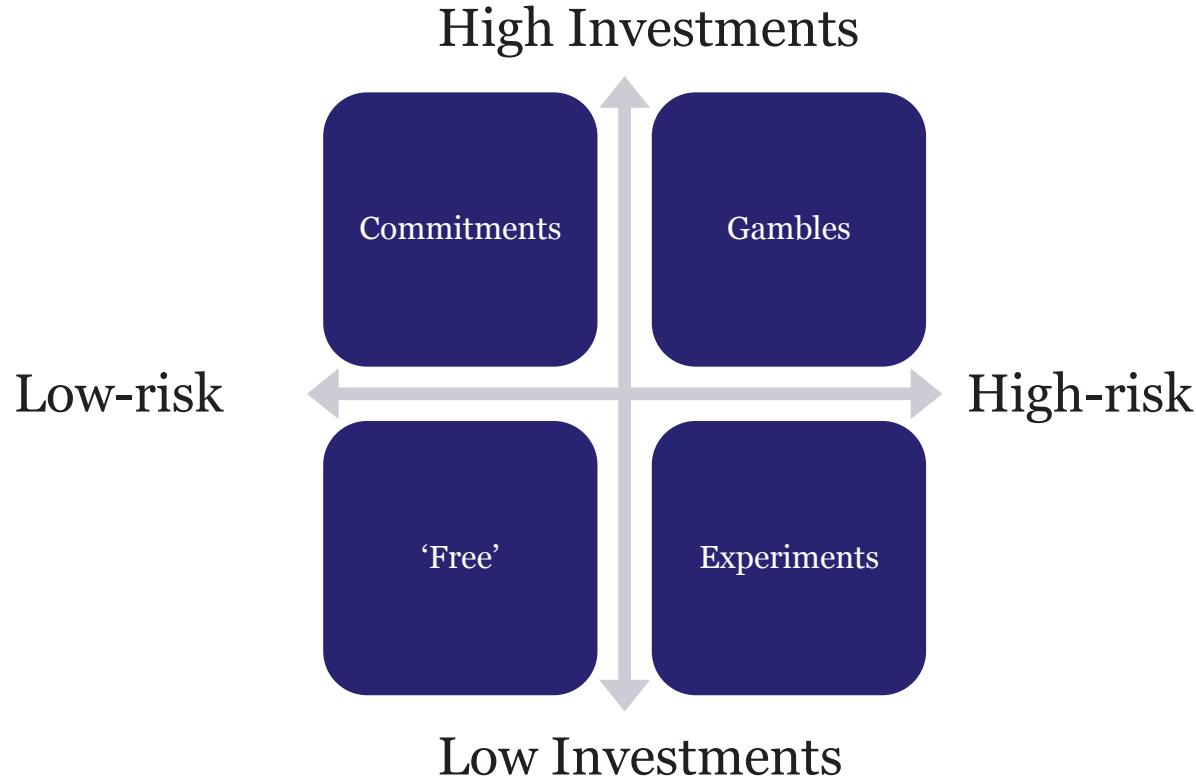
Develop the “composition” sense and don't be afraid to cut bad features.

Design is good when you have **nothing to cut**

# Feature ROI Matrix



# Feature risk Matrix



# ► Feature value

Value	ROI	Risk	Classification
V1	R1 (Competitive)	Ro ('Free')	Competitive advantage
V2	R1 (Competitive)	R1 (Commitment)	
V3	R2 (Strategic)	Ro ('Free')	Strategic advantage
V4	R2 (Strategic)	R1 (Commitment)	
V5	R3 (Polishing)	Ro ('Free')	Polishing/Quality
V6	R3 (Polishing)	R1 (Commitment)	
V7	R1 (Competitive)	R2 (Experiment)	Experiments
V8	R2 (Strategic)	R2 (Experiment)	
V9	R3 (Polishing)	R2 (Experiment)	
V10+	Too risky		

# Unconscious incompetence



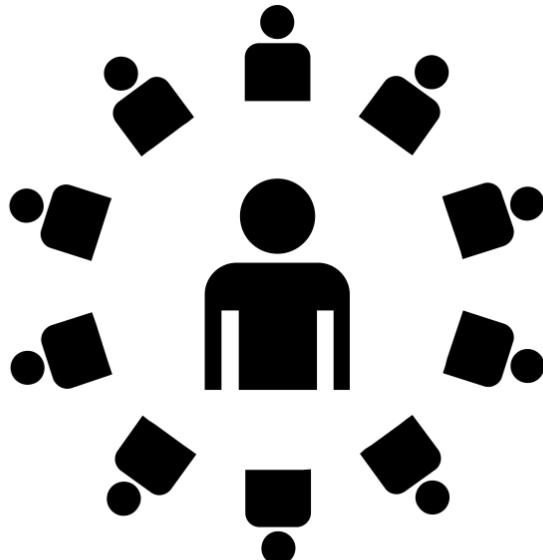
Wrong risk assessment

Outdated knowledge

Previous experience bias

Decrease **quality** of decisions

# ► Solution: Team feedback



Ask the designer!

Give feedback on the design as early as possible.

The more feedback, the better.

Demand **clarity**, until you get it

# Cloning



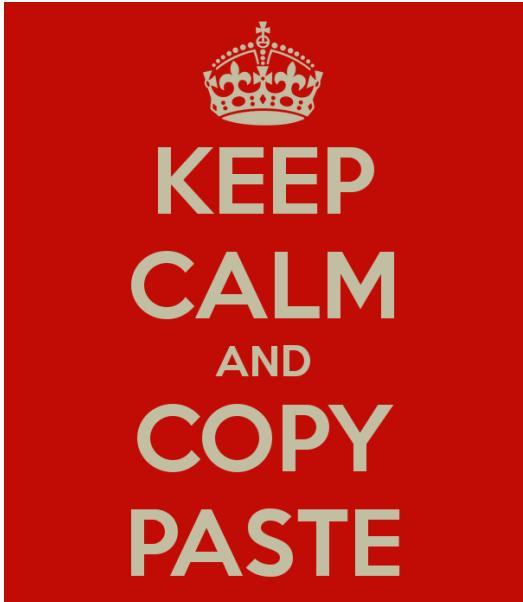
Competitors constraints are unknown

Might not work for your product

'Lazy design'

Decrease **quality** of decisions

# ► Solution: De-construction



**De-construct to clone properly**

Cloning isn't necessarily bad.  
The fewer obstacles on the player's way, the better; standard solutions are great for that (3C/UI/etc.)  
Features are your tools to create an experience.

**Fundamental knowledge will help you**

# Design

## Experience

It's all about player experience, remember it in any design decision.

## Clarity

Know what design you do (Macro or Micro), choose proper abstraction level and aim for CLARITY.

## Execution

Game Design is about execution, control your feature scope and aim for high quality of decisions.

*Stage 2*

---

## COMMUNICATION PRINCIPLES

# → Errare humanum est



People are **imperfect**

People make mistakes, they're biased and usually bad listeners.

The worst: you have no other people.

You **can't change** them

# ► Designer is also imperfect

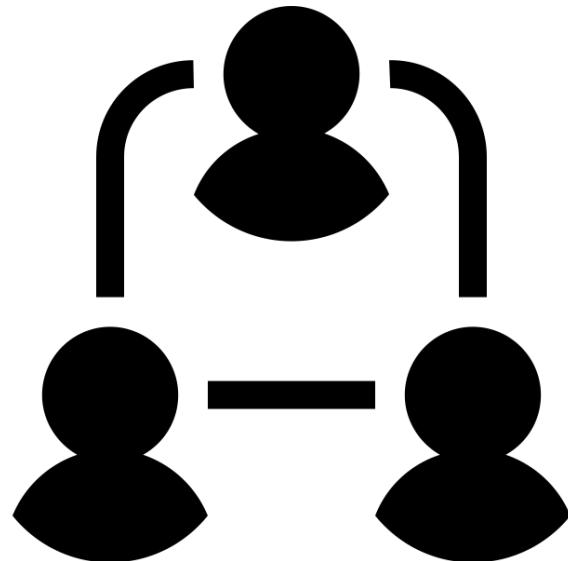


You as a designer is also **imperfect**

You cannot know everything and can't plan in advance the full and detailed vision of the game.

You vision is **limited**

# ► Shared design



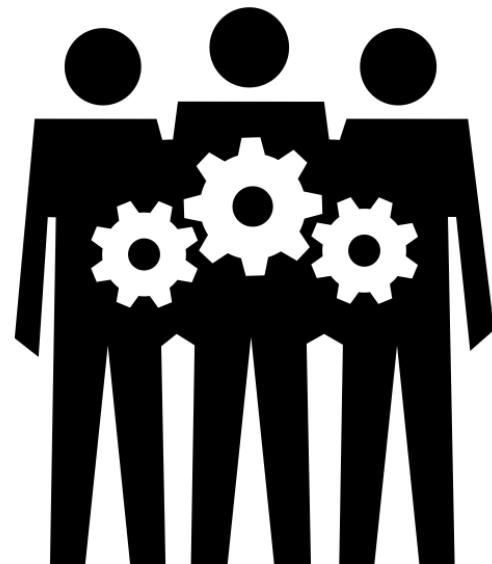
**Share design with your team!**

People on the implementation level know more details and can help you to reach the much higher quality of design decisions.

Give freedom to artisans!

They will become **personally invested**

# ► Cross-disciplinary collaboration



**Gather people from different disciplines**

Include people to design discussions who will be implementing your feature, they know each ‘part of the puzzle’ and can significantly increase the quality of the design.

**Such collaboration can create *magic***

# ► Conflicts



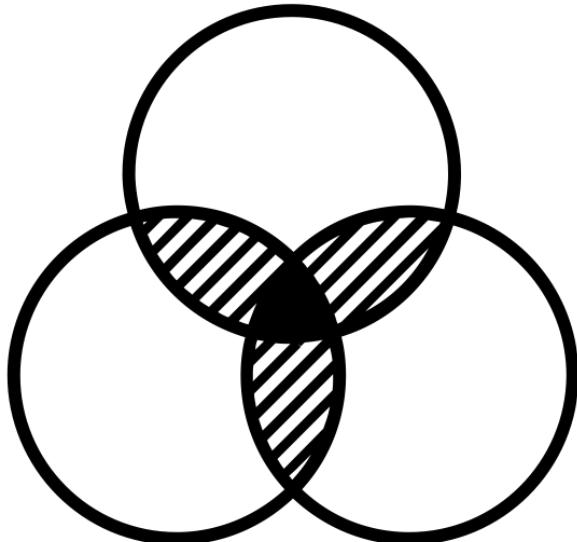
**Task conflicts can be good for the project**

If you have a lot of small and medium task conflicts, it's a good sign that team is alive and care about the project.

Also, try to solve all process conflicts on the early stage.

**BUT: Never get personal!**

# Integrated Decisions



Look for **Integrated Decisions**

Combine different ideas and approaches of your team to produce the best solution.

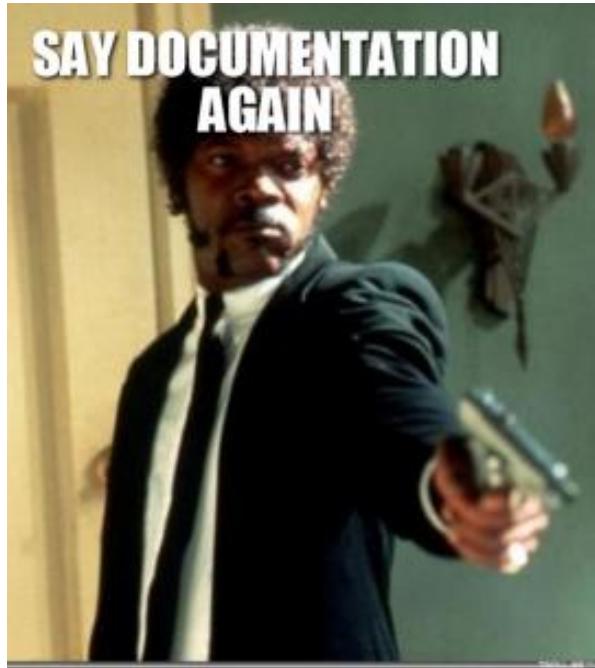
Include **team's creativity** into them

# ► Obstacles on your way

---



# Documents



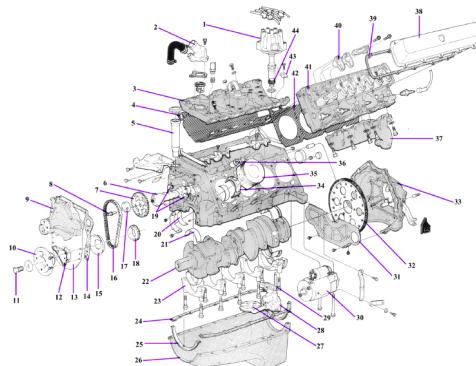
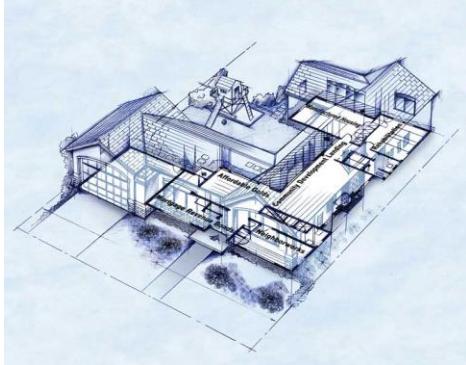
Become outdated fast

Hard to maintain in up-to-date state

People don't read (and don't want to)

Source of **bottlenecks** on the project

# What to do with the documents?



Make your documents **digestible**

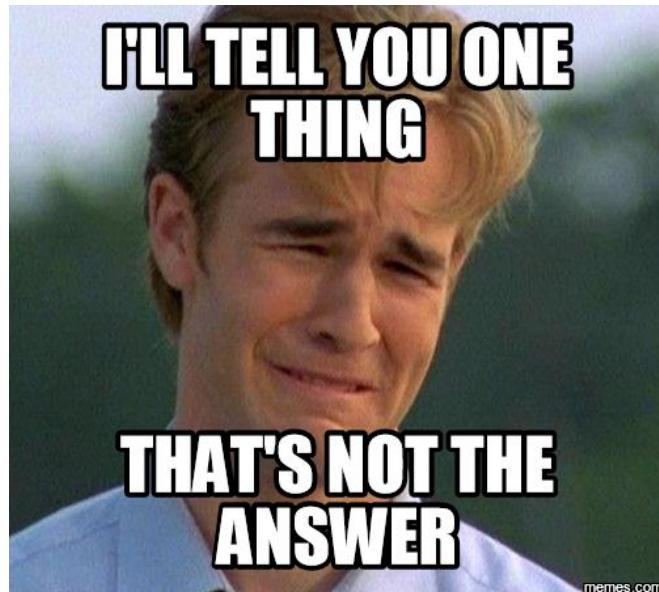
Usually, people don't read more than 1 page at a time. So, don't make document longer than 1 page! Or divide them into 'digestible' blocks.

Remember, that words can be deceiving.

Support them with **visuals**

\*Excel tables + Mock-ups

# ► I have bad news for you...



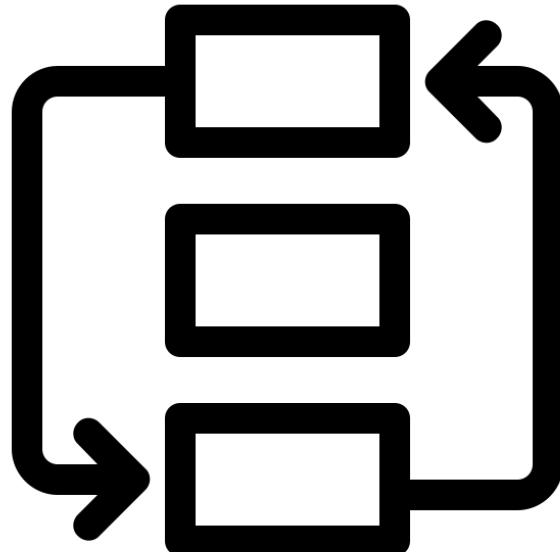
You'll **never** have 100% correct documents

People don't read even the best documents

There are many other communication channels!

You'll **always** be criticized for 'outdated docs'

# ► Solution: It is LIVE!



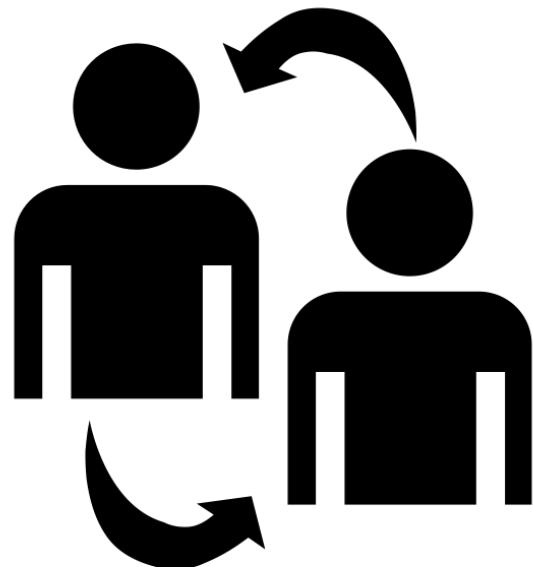
Documents are not **made of marble**

We still often live by the old stereotype that there's possible to have 'final' GDD.

Try to treat documents in a similar way as code: like something LIVE.

Documents are **LIVE** and may have **bugs**

# ► Magic tool for the designer



Talk **directly**

Game Design is a **LIVE** process that is based on constant iterations and feedback gathering.

Feedback from direct communication is **MUCH** faster than any other.

This is **the fastest** way

# ► If you work remotely

**BRACE YOURSELVES**



**Use 3<sup>rd</sup> clarification rule**

If the question not answered in three e-mails, you must call.

Otherwise, your mail thread might be almost infinite.

**Get direct feedback if mail is not working**

# Expert arrogance



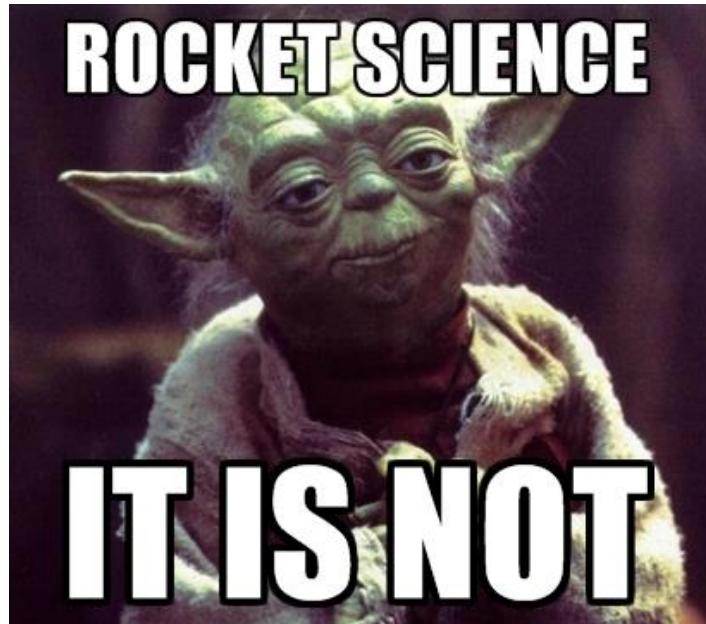
Design can be blocked by an 'expert opinion'

It might be fear of responsibility or just laziness

It might be personal or political

Can **unreasonably** block some features

# ► It's not a rocket science



Demand **CLARITY** from the 'expert'

In the most cases, the 'expert' is afraid of potential risk (personal or the project) and overvalues the complexity of the feature.

Get more information, ask questions and separate facts from the 'expert opinion'.

Focus on **WHAT** you want, not **HOW**

# ‘Not invented here’



People love to **create** something new

Which leads to conflict with business goals

Big problem, if you have demanding customer

Feedback perceived as an **attack on ownership**

# ► It's all about the player



There is a **common ground**

All the desires of ‘wheel invention’ should be evaluated how valuable are they for the player.

It can be a common ground for the team and the customer.

The player is a **King**

# ► Stakeholders feedback



Inadequate

Stakeholders contradict each other

Force you to make more than was negotiated

Can **ruin** your scope (and **you** will be blamed)

# ► Stakeholders are people



You are reviewed by **imperfect people**

Stakeholders are also imperfect, they have limited vision and judge based on that vision, not necessarily because they're bad.

**Talk to them, explain your decisions**

# ► Hidden Stakeholders



There are people who can say you **NO**

There might be some people on the project who are not officially stakeholders, but can block your design for some reasons (processes, technical, etc.).

Find them on the **early stage**

# Communication

People

People are imperfect and you can't change that. But you can learn how to overcome it.

Share

Share the design and push cross-discipline collaboration. It can create magic.

Simplify

Simplify documents, add visuals to avoid misinterpretation. Talk directly for the fastest feedback.

*Stage 3*

---

## IMPLEMENTATION PRINCIPLES

# ► Truth about players



“Creature who can fill the Louvre with all the shits he doesn’t give”

**It doesn't matter how many problems you have**

Your problems with scope, resources, deadlines, whatever else just don't matter for the player.

The only thing that is valuable for players is their personal experience.

**The player is like Honey Badger**

\*Doesn't give a shit

# Iterate, iterate, iterate...

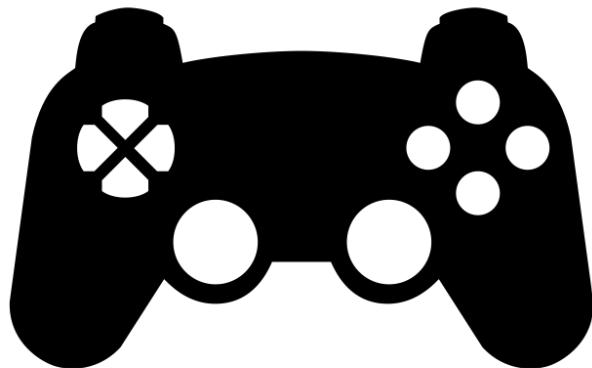


Almost always, it's **innovation**

Game Design is very similar to innovation adoption process, and impossible without many iterations. Be ready to fail (and learn from it, of course), it's a natural process.

Implementation is **discovery-driven** learning

# ► Playtests



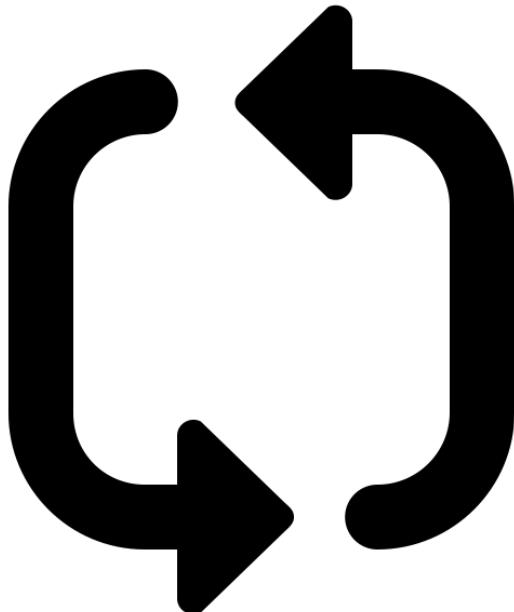
You create an **interactive** experience

There is only one opportunity to see your **real** game: playtest.

In Game Design, you can believe only in things that you can interact with.

Playtest is a part of **discovery-driven** learning

# Feedback



**Do not trust too much to what people say**

Your playtesters are also imperfect, and might have very weird ideas how to improve your game.

Look for the **player's insight**.

**See what they DO in your game**

# ► Do not afraid of design changes



You design can **fail**

The design of your feature can fail on the implementation stage for many reasons (in many cases, not because it's bad).

**Re-design** or just **cut** this feature

# Fake it



The player **does not see** your back-end

In many cases, you just don't need a sophisticated technology to create the experience for the player.

If you can't implement something, **fake it!**

# ► Obstacles on your way

---



# ► Processes cargo cult



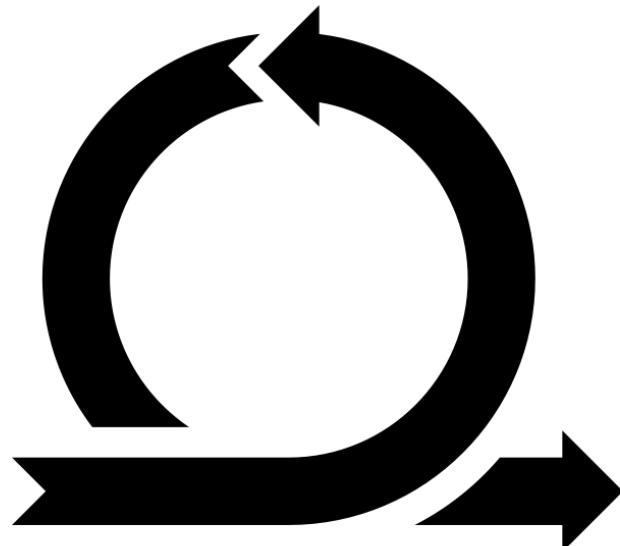
Processes are more important than the game

Treat people as 'resources', not persons

Overinflated importance of process decisions

Can **slow down** or even **stop** the design process

# ► Do we REALLY follow Agile?



The solution is simple . . .

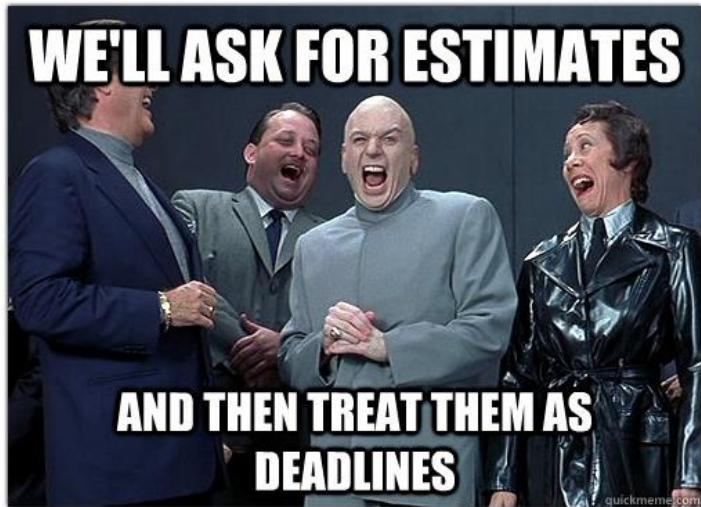
Individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Responding to change over following a plan.

. . . but very hard to follow

# Estimates



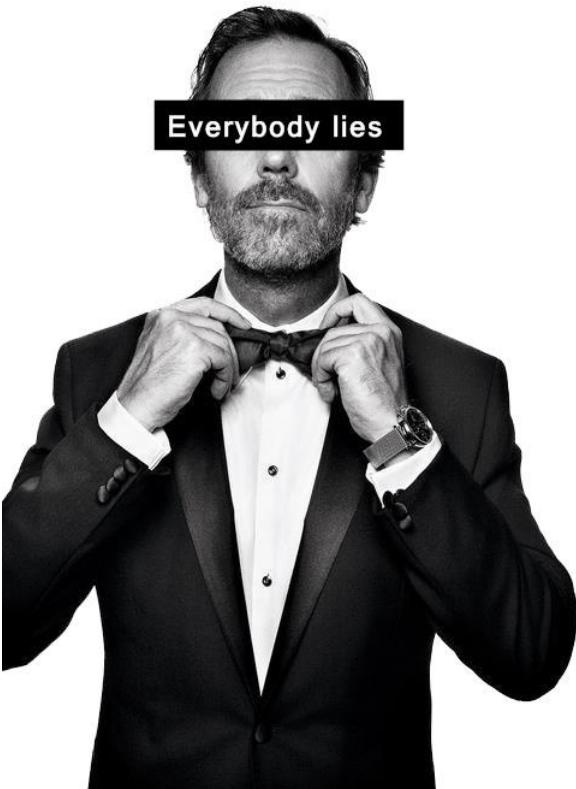
ETA for 'cool' features is shorter

ETA for 'boring' features is longer

ETA misinterpretation

Source of **scope** problems

# Everybody lies



**Do not trust estimates**

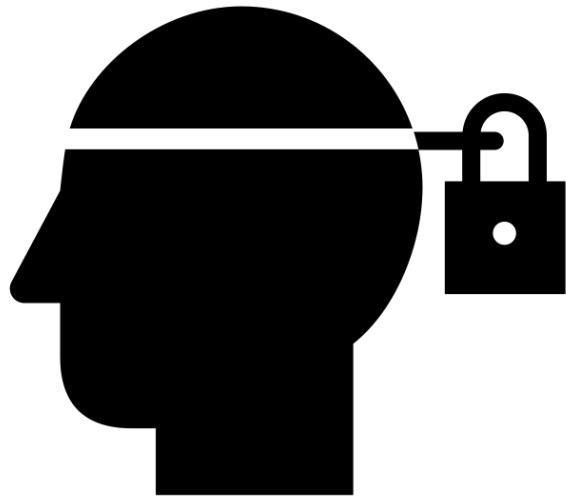
Know your team and what features and tasks they like.

Do not trust long estimates especially.

**Feature functional =! Feature complete**

**Use Batching for similar features**

# Low flexibility



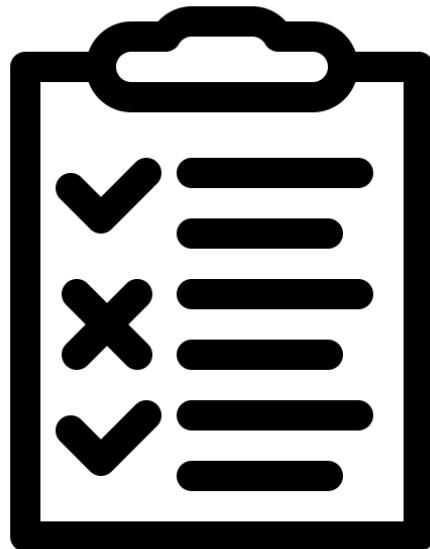
Long iteration time

Bad tools

Complex approval process

Can **stop** design and polishing process

# Fast to test



**Fast iterations** is a key to good design

Game Design is a way of discovery-driven learning.

If you want good design, all your processes and tools should be focused on making iteration time as less as possible.

The faster you go to **play**, the better

# ► Start polishing early



Quality comes from **polishing**

Final player experience appears on the **polishing** stage, after you tested how the player **interacts** with your game.

Shorter iteration time will allow you to reach the polishing stage faster.

Start it **early**

# Implementation

## Player

Players don't give a shit about your problems. They only care about their personal experience.

## Interaction

You can believe only in things that you can interact with. Iterate, playtest, gather feedback.

## Processes

Even the best processes and plans can fail because of people imperfection.

# To summarize

It's all about the player experience

Simplify documents and talk directly

Aim for CLARITY

The player doesn't give a shit

Game Design is about execution

Believe only in things you can interact with

People are imperfect

Even the best processes can fail

Share design with your team

Game Design is about people

# ► Sources

---

- (D.I.C.E. 2002) Mark Cerny, THE METHOD
- (TED) Linda Hill, ‘How to manage for collective creativity’
- (Coursera) Leading Strategic Innovation in Organizations
- (D.I.C.E. 2010) Naughty Dog Presentation
- (GDC 2010) Among Friends – An Uncharted 2: Among Thieves post-mortem
- (GDC 2012) The Last 10: Going From Good To Awesome
- (GDC 2015) Hearthstone: How to Create an Immersive User Interface
- (GDC 2014) Hearthstone: 10 Bits of Design Wisdom
- (GDC 2013) Through the Grinder: Refining Diablo III’s Game Systems
- (GDC 2010) Blizzard Design Philosophies

\*any questions?  
thank you very much