GDC | March 20-24, 2023
San Francisco, CA

# Testing with Building Blocks

## A Modular Approach to Quality Assurance

Chris Rios (he/him) – Test Lead @ Bungie

#GDC23

# Who am I?

# Who am I?

- Raids and Dungeons Test Lead at Bungie

# Who am I?

- Raids and Dungeons Test Lead at Bungie
- Worked on our raids and dungeons for the past 3 years

# Who am I?

- Raids and Dungeons Test Lead at Bungie
- Worked on our raids and dungeons for the past 3 years
- Involved in activities and player rewards testing

# Common pain points of testing Raids and Dungeons

# Common pain points of testing Raids and Dungeons

- Systems are often complex.

# Common pain points of testing Raids and Dungeons

- Systems are often complex.
- Difficult to cost bespoke systems early on.

# Common pain points of testing Raids and Dungeons

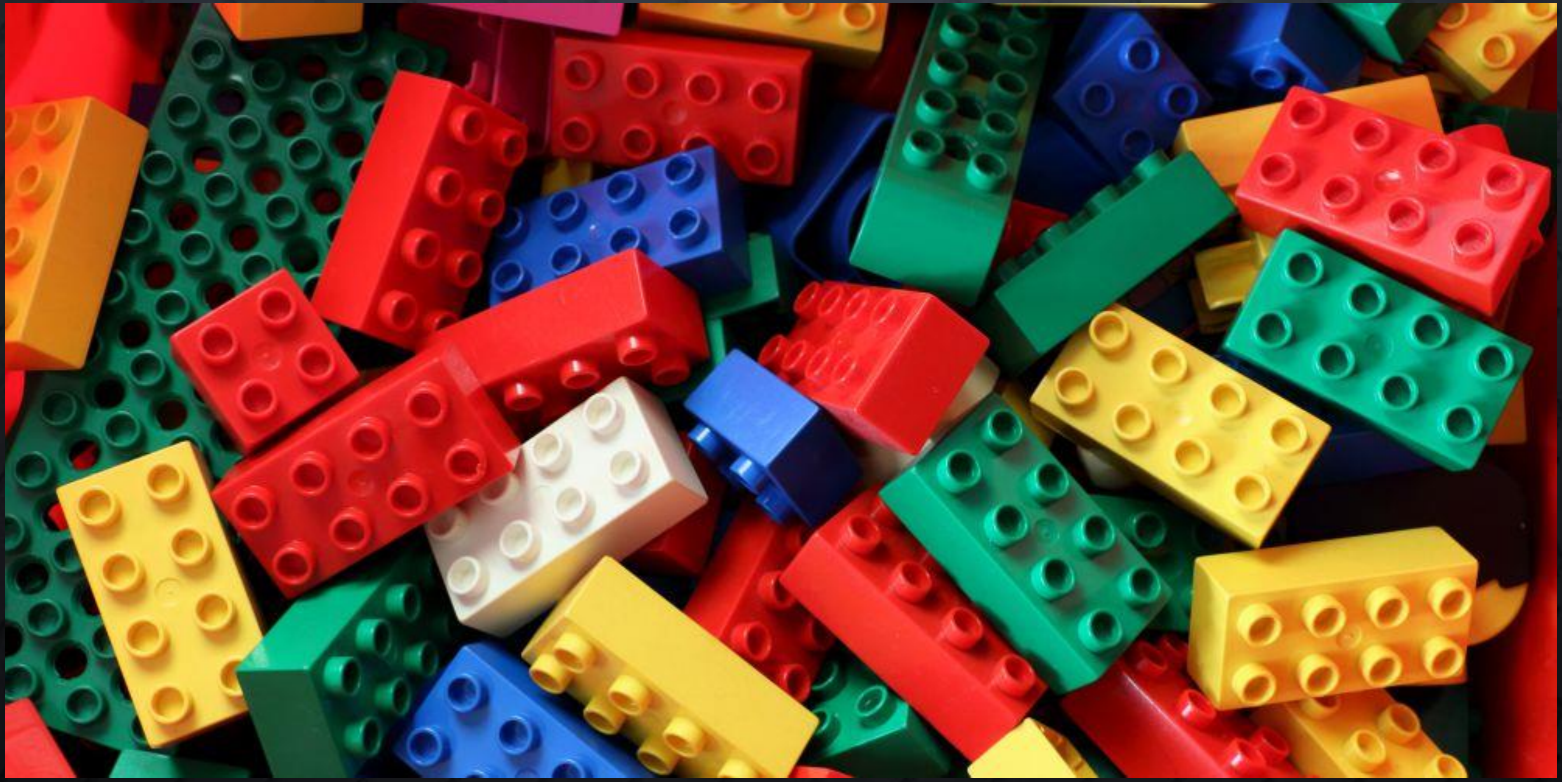
- Systems are often complex.
- Difficult to cost bespoke systems early on.
- Features are always changing.

# Common pain points of testing Raids and Dungeons

- Systems are often complex.
- Difficult to cost bespoke systems early on.
- Features are always changing.
- Undocumented knowledge can be difficult to share.

# So, what can we do to address these pain points?

# Let’s make it fun and impactful – with Building Blocks!



# What is the Concept of "Testing with Building Blocks?"

# What is the Concept of "Testing with Building Blocks?"

- A creative mindset to ***discover***

# What is the Concept of "Testing with Building Blocks?"

- A creative mindset to ***discover***
- A way to ***interact*** with systems

# What is the Concept of "Testing with Building Blocks?"

- A creative mindset to ***discover***
- A way to ***interact*** with systems
- A way to ***create*** use cases

# What is the Concept of "Testing with Building Blocks?"

- A creative mindset to ***discover***
- A way to ***interact*** with systems
- A way to ***create*** use cases
- …all the things we use building blocks for!

# How to Use Building Blocks for Testing

# How to Use Building Blocks for Testing

1

**Identify** buckets to store building blocks in.

# How to Use Building Blocks for Testing

1

**Identify** buckets to store building blocks in.

2

**Create** building block pieces designed for buckets.

# How to Use Building Blocks for Testing

1

**Identify** buckets to store building blocks in.

2

**Create** building block pieces designed for buckets.

3

**Combine** to create modular test plans.

# Implementing a Building Blocks Testing Approach

**Step 1 - IDENTIFY** Your Buckets

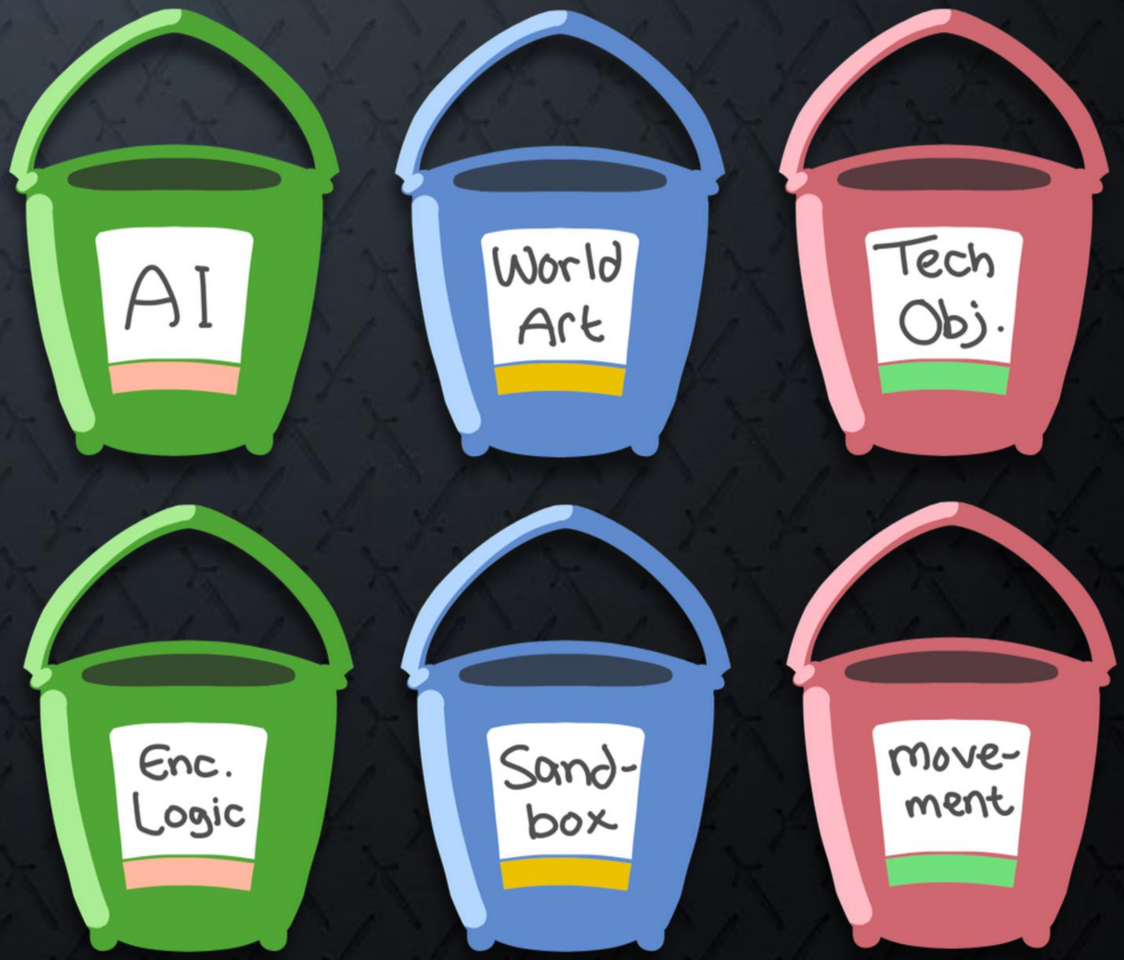
# **Step 1 – Identify** your buckets

# **Step 1 – Identify** your buckets

- Macro level systems act as our **buckets**.

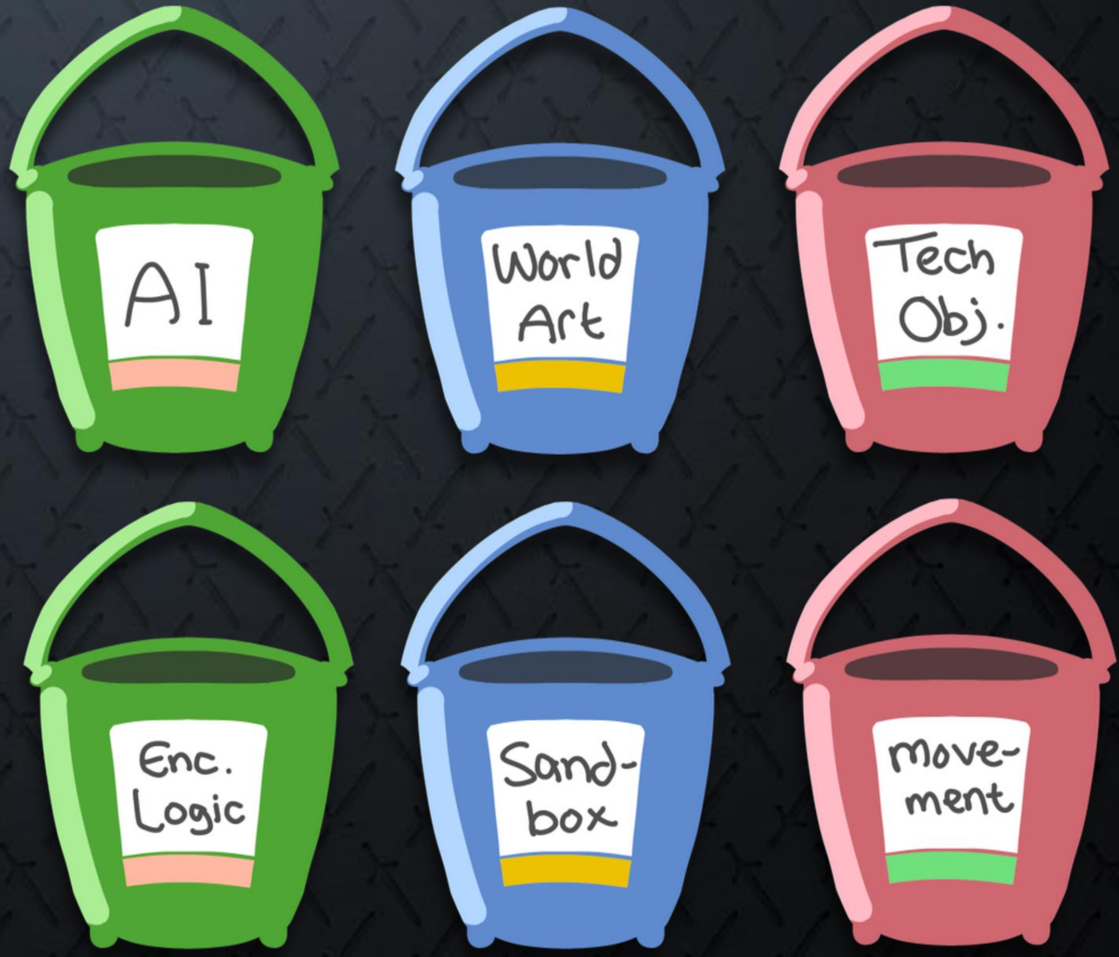
# **Step 1 – Identify** your buckets

- Macro level systems act as our **buckets.**



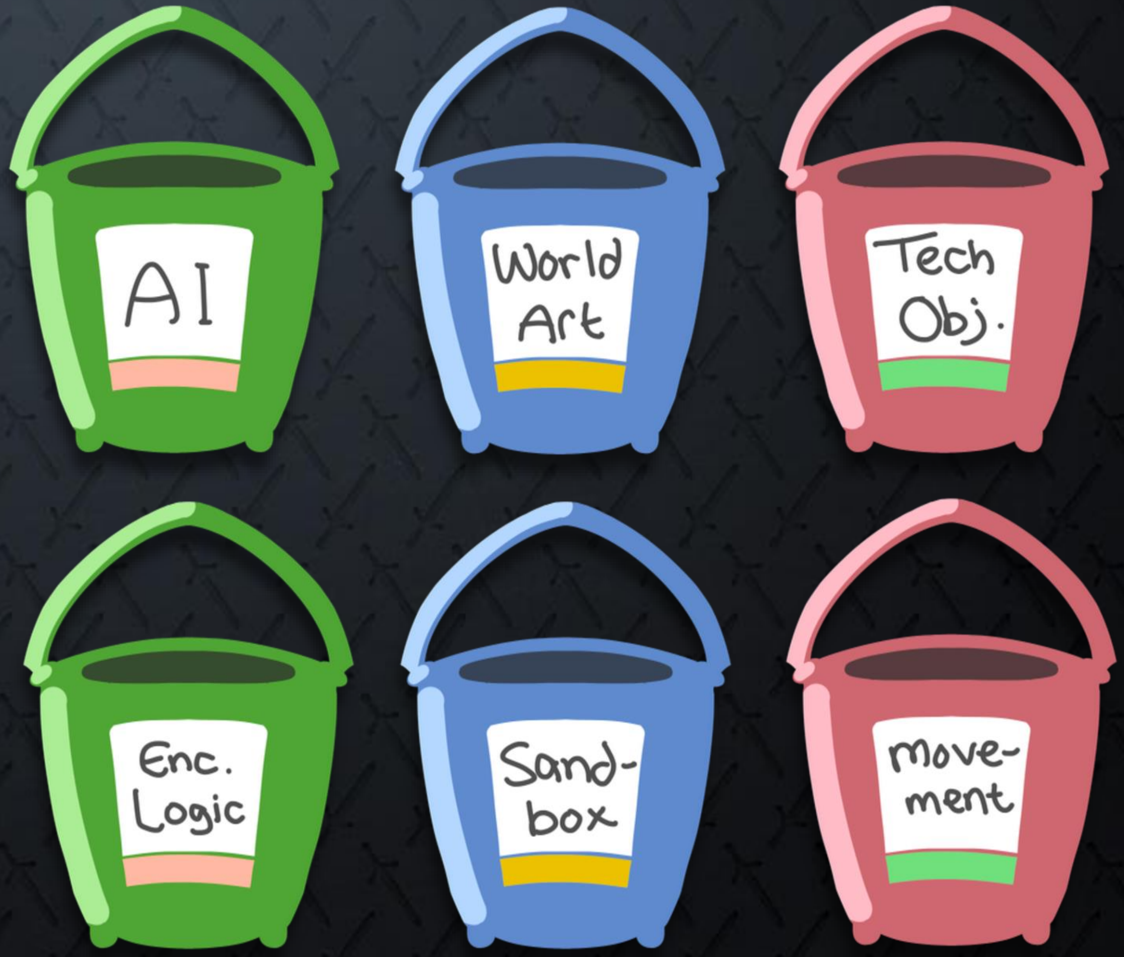
# Step 1 – **Identify** your buckets

- Macro level systems act as our **buckets.**
- Prioritize frequently engaged systems over others.



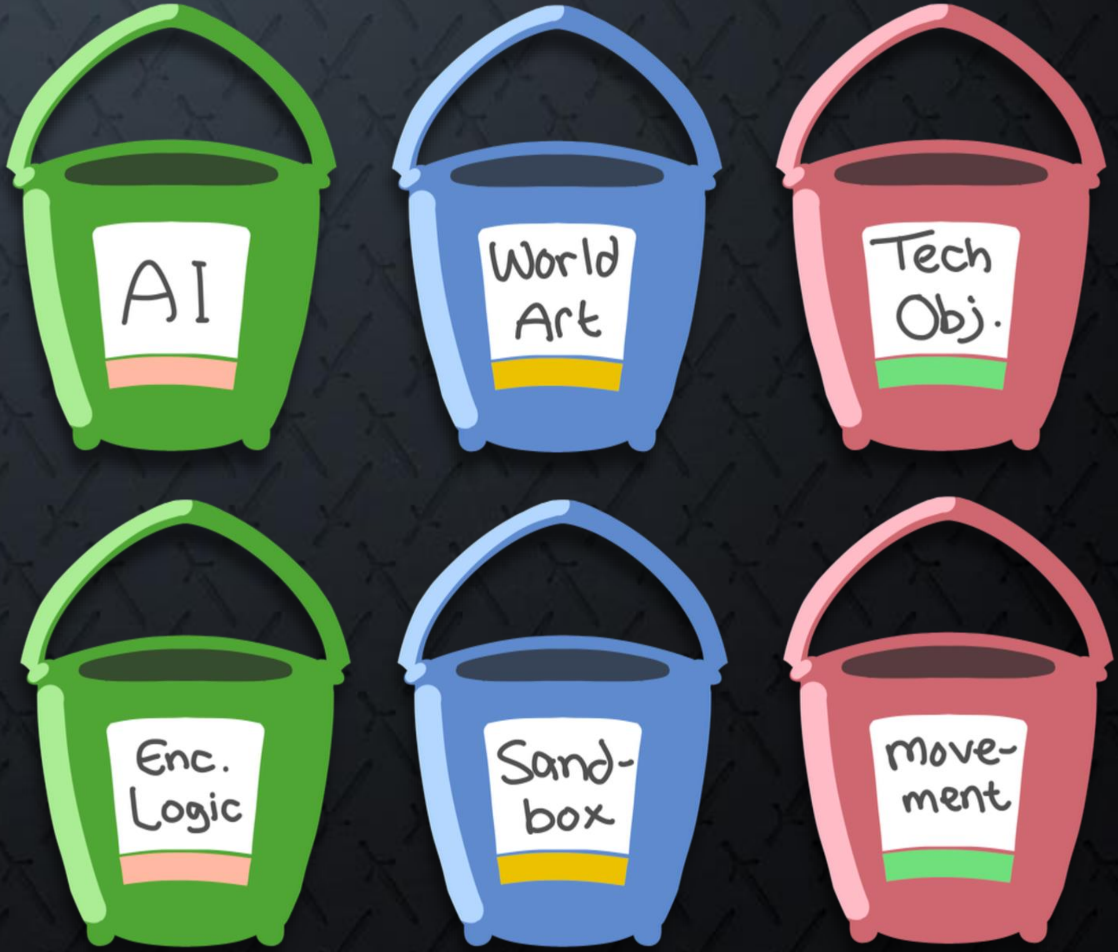
# Step 1 – **Identify** your buckets

- Macro level systems act as our **buckets.**
- Prioritize frequently engaged systems over others.
  - We created 6-12 buckets to reduce information overload



# **Step 1 – Identify** your buckets

- Macro level systems act as our **buckets.**
- Prioritize frequently engaged systems over others.
    - We created 6-12 buckets to reduce information overload
- Buckets are reusable



BUNGIE

☑

**Identify** buckets to store building blocks in.

2

**Create** building block pieces designed for buckets.

3

**Combine** to create modular test plans.

# Implementing a Building Blocks Testing Approach

**Step 2 - CREATE** Your Building Blocks

# **Step 2.A - Create** Your Building Blocks: Components

# **Step 2.A - Create** Your Building Blocks: Components

- Think of verbiage and traits that impact player experience.

# **Step 2.A - Create** Your Building Blocks: Components

- Think of verbiage and traits that impact player experience.
    - e.g. **Movement bucket:** sprinting, sliding, jumping, teleporting…

# **Step 2.A - Create** Your Building Blocks: Components

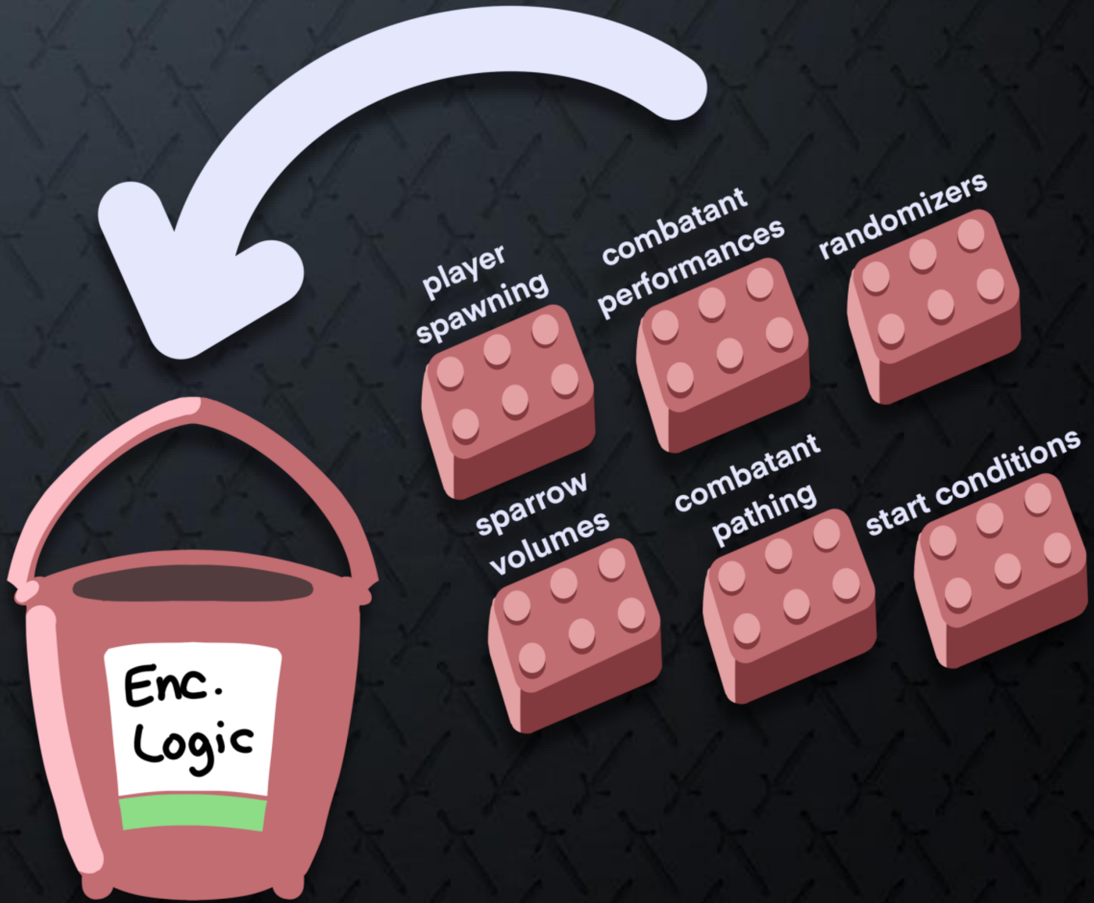
- Think of verbiage and traits that impact player experience.
  - e.g. **Movement bucket:** sprinting, sliding, jumping, teleporting…
- Analyze bug database and trends

# **Step 2.A - Create** Your Building Blocks: Components

- Think of verbiage and traits that impact player experience.
  - e.g. **Movement bucket:** sprinting, sliding, jumping, teleporting…
- Analyze bug database and trends
- Components are reusable

# Step 2.A - Create Your Building Blocks: Components

- Think of verbiage and traits that impact player experience.
  - e.g. **Movement bucket:** sprinting, sliding, jumping, teleporting…
- Analyze bug database and trends
- Components are reusable



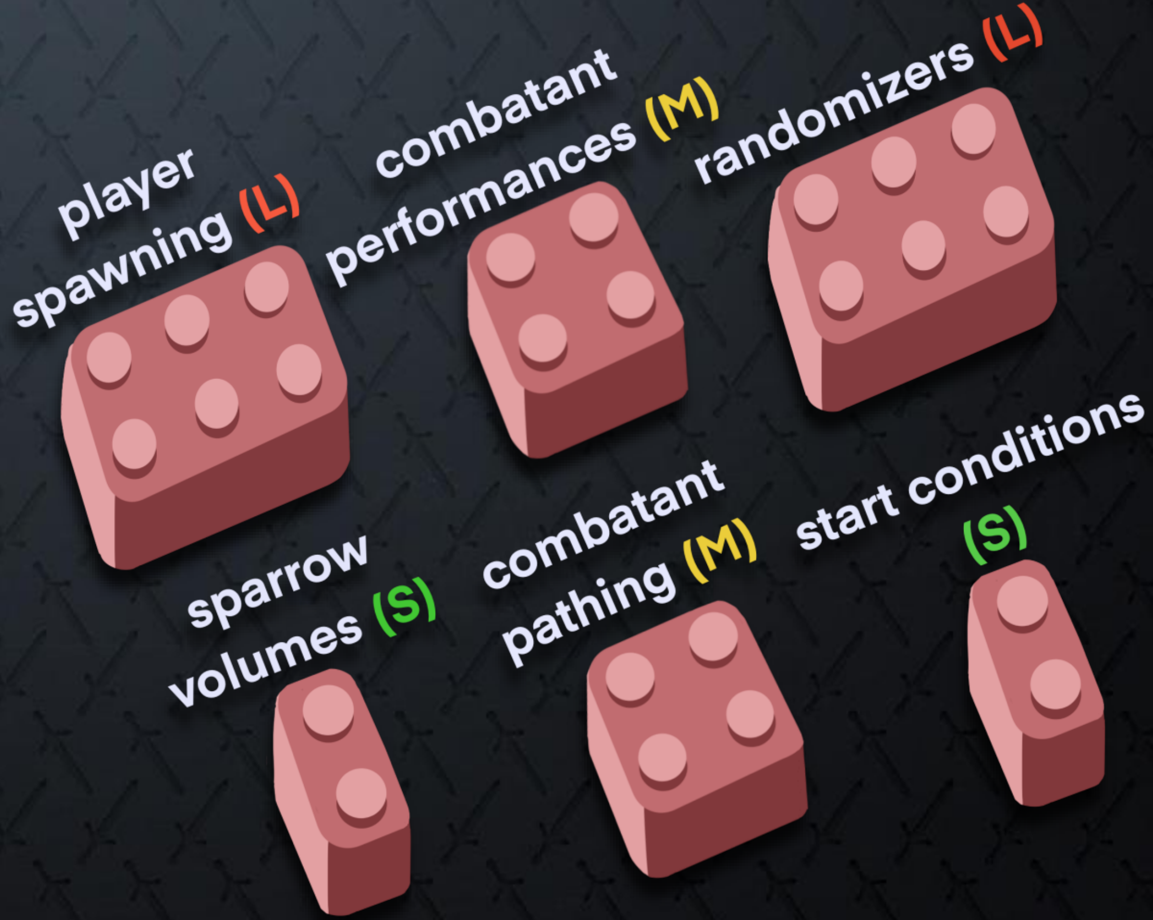
# **Step 2.B - Create** Your Building Blocks: Sizing

# **Step 2.B - Create** Your Building Blocks: Sizing

- Sizes of building blocks: small, medium, and large

# **Step 2.B - Create** Your Building Blocks: Sizing

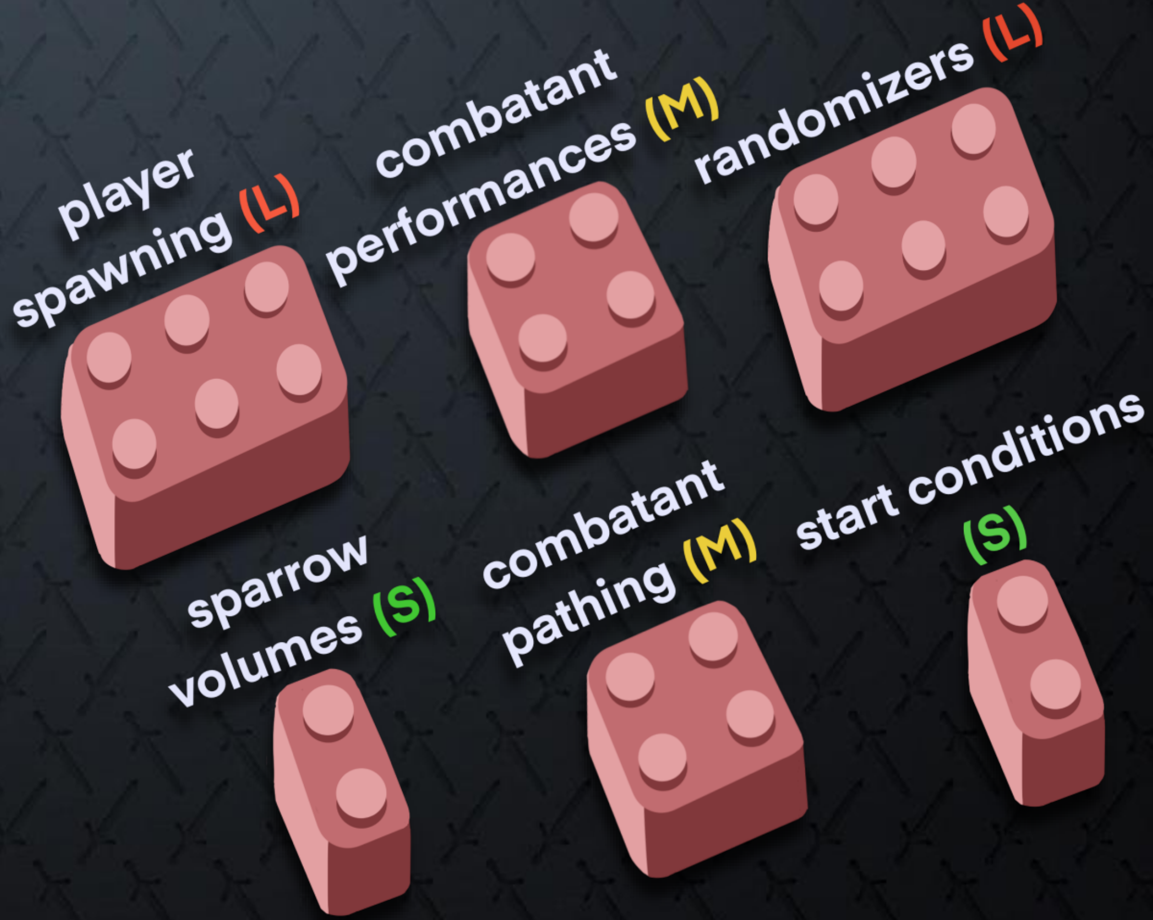
- Sizes of building blocks: small, medium, and large
- Size correlates to bug **vulnerability** and **intended user experience**



BUNGiE

# Step 2.B - Create Your Building Blocks: Sizing

- Sizes of building blocks: small, medium, and large
- Size correlates to bug **vulnerability** and **intended user experience**
- How likely are you to find bugs, and how dangerous is it?



☑

**Identify** buckets to store building blocks in.

☑

**Create** building block pieces designed for buckets.

3

**Combine** to create modular test plans.

# Implementing a Building Blocks Testing Approach

**Step 3 - COMBINE** the Building Blocks

# **Step 3 - Combine** Building Blocks

Three primary ways to combine building blocks:

# Step 3 - **Combine** Building Blocks

Three primary ways to combine building blocks:

1. Goal & Value Driven Testing (Directed Ad-Hoc)

# Step 3 - **Combine** Building Blocks

Three primary ways to combine building blocks:

1. Goal & Value Driven Testing (Directed Ad-Hoc)
2. Planning and Costing

# Step 3 - **Combine** Building Blocks

Three primary ways to combine building blocks:

1. Goal & Value Driven Testing (Directed Ad-Hoc)
2. Planning and Costing
3. Live Investigation

# Combine Building Blocks: Directed Ad-Hoc

# **Step 3.A - Combine** Building Blocks: Directed Ad-Hoc

# **Step 3.A - Combine** Building Blocks: Directed Ad-Hoc

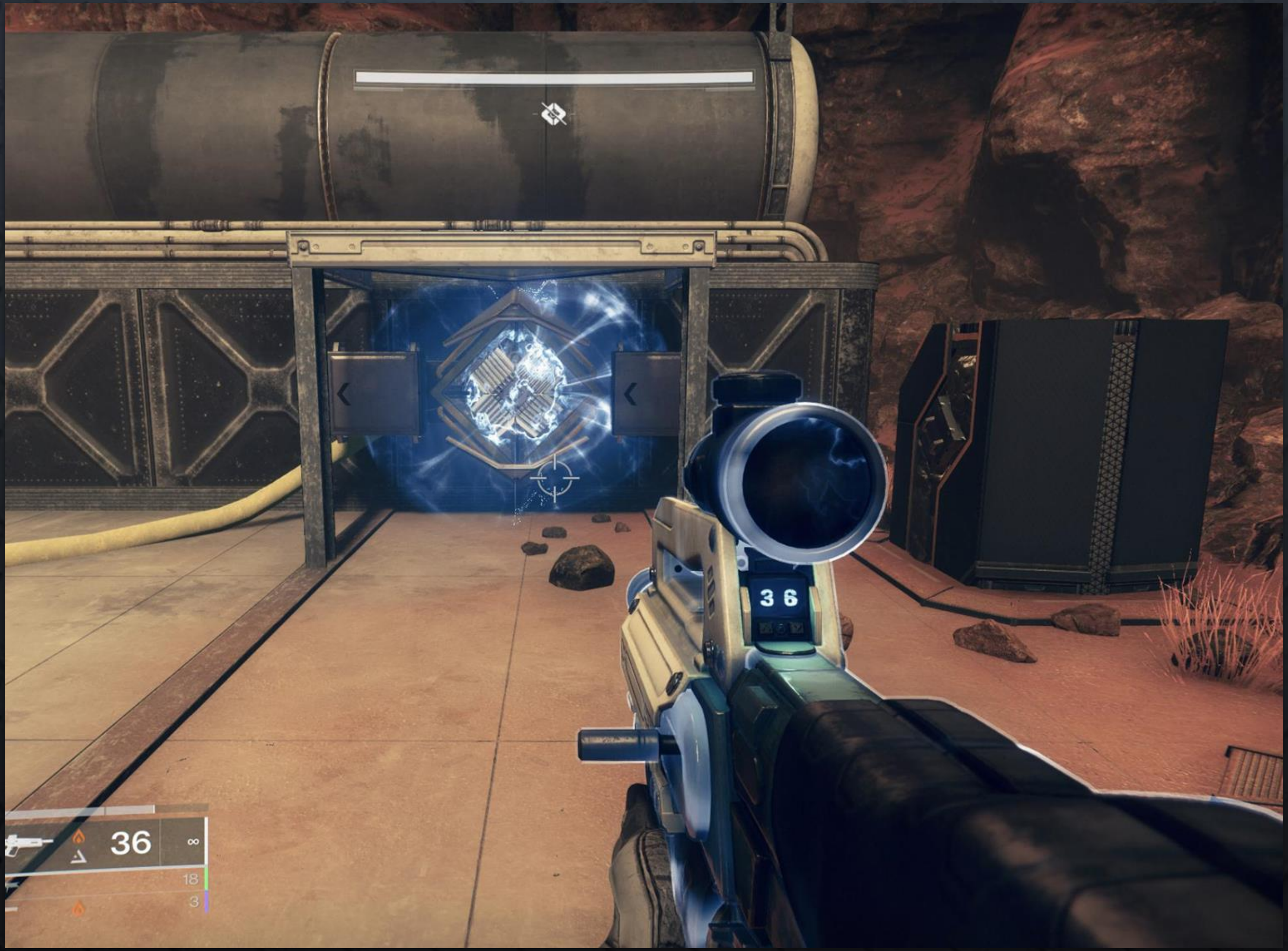
- Create “what happens when…” question

# **Step 3.A - Combine** Building Blocks: Directed Ad-Hoc

- Create “what happens when…” question
- Utilize 2-3 buckets you’ve made

# Step 3.A - Combine Building Blocks: Directed Ad-Hoc

- Create “what happens when…” question
- Utilize 2-3 buckets you’ve made
- Use conditional statements to string the pieces together
  - and, not, if, while…

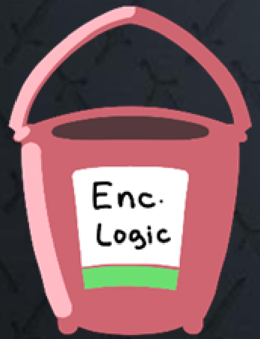


***Let's use an example from the Spire of the Watcher (Destiny 2 – The Witch Queen)!***

Simply put: These are ***objects*** you can ***shoot*** to change their ***state***

# Use Case: Directed Ad-Hoc

# Use Case: Directed Ad-Hoc



# Use Case: Directed Ad-Hoc

| | |
|---|---|
| | Randomizers, Player Spawning , Race Conditions |
| | Combatant Performances, Combatant Pathing |
| | Sparrow volumes, start conditions |

# Use Case: Directed Ad-Hoc

| | |
|---|---|
| 🟥 | Randomizers, Player Spawning , Race Conditions |
| 🟥 | Combatant Performances, Combatant Pathing |
| 🟥 | Sparrow volumes, start conditions |

| | |
|---|---|
| 🟦 | Latency, States (Inactive, Semi-Active, Active, Enabled, Disabled) |
| 🟦 | Interaction Scope (Client vs. Network Authoratative), Buff Activation, Interaction timer |
| 🟦 | Object Health Pools |

# Use Case: Directed Ad-Hoc

| Enc. Logic | Tech Obj. | Actor Dmg. |
|---|---|---|
| Randomizers, Player Spawning , Race Conditions | Latency, States (Inactive, Semi-Active, Active, Enabled, Disabled) | Weapon Procs, Damage Multipliers, Perk Activation/Deactivation |
| Combatant Performances, Combatant Pathing | Interaction Scope (Client vs. Network Authoratative), Buff Activation, Interaction timer | Damage Over Time, Explosive Damage |
| Sparrow volumes, start conditions | Object Health Pools | Automatic fire |

# Use Case: Directed Ad-Hoc

1. What happens when…

# Use Case: Directed Ad-Hoc

1. What happens when…

**Player deals damage over time**

# Use Case: Directed Ad-Hoc

1. What happens when…

Player deals damage over time

while an

Object transitions states

# Use Case: Directed Ad-Hoc

1. What happens when…

Player deals damage over time

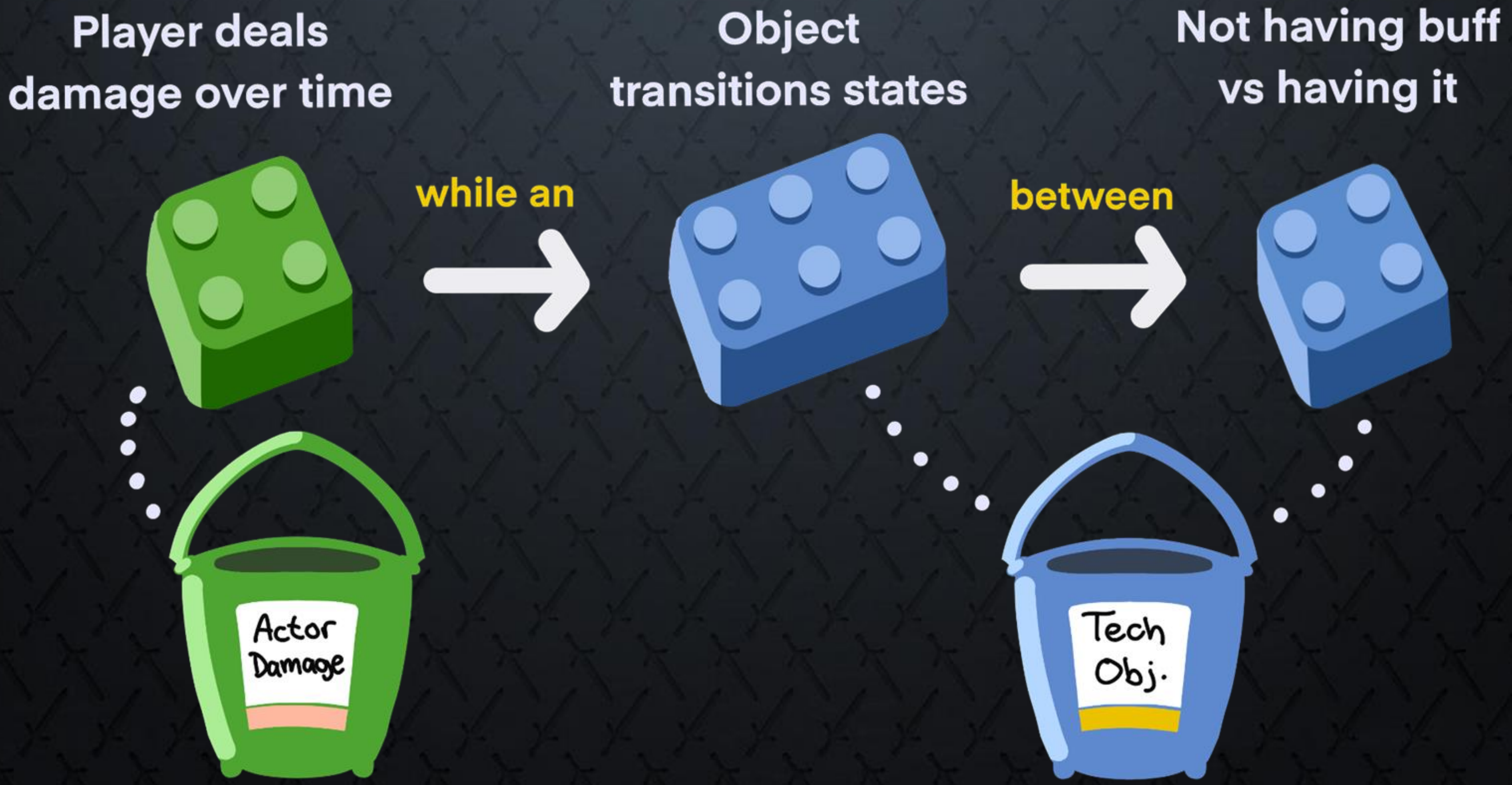
while an

Object transitions states

between

Not having buff vs having it

# Use Case: Directed Ad-Hoc



Player deals damage over time

while an

Object transitions states

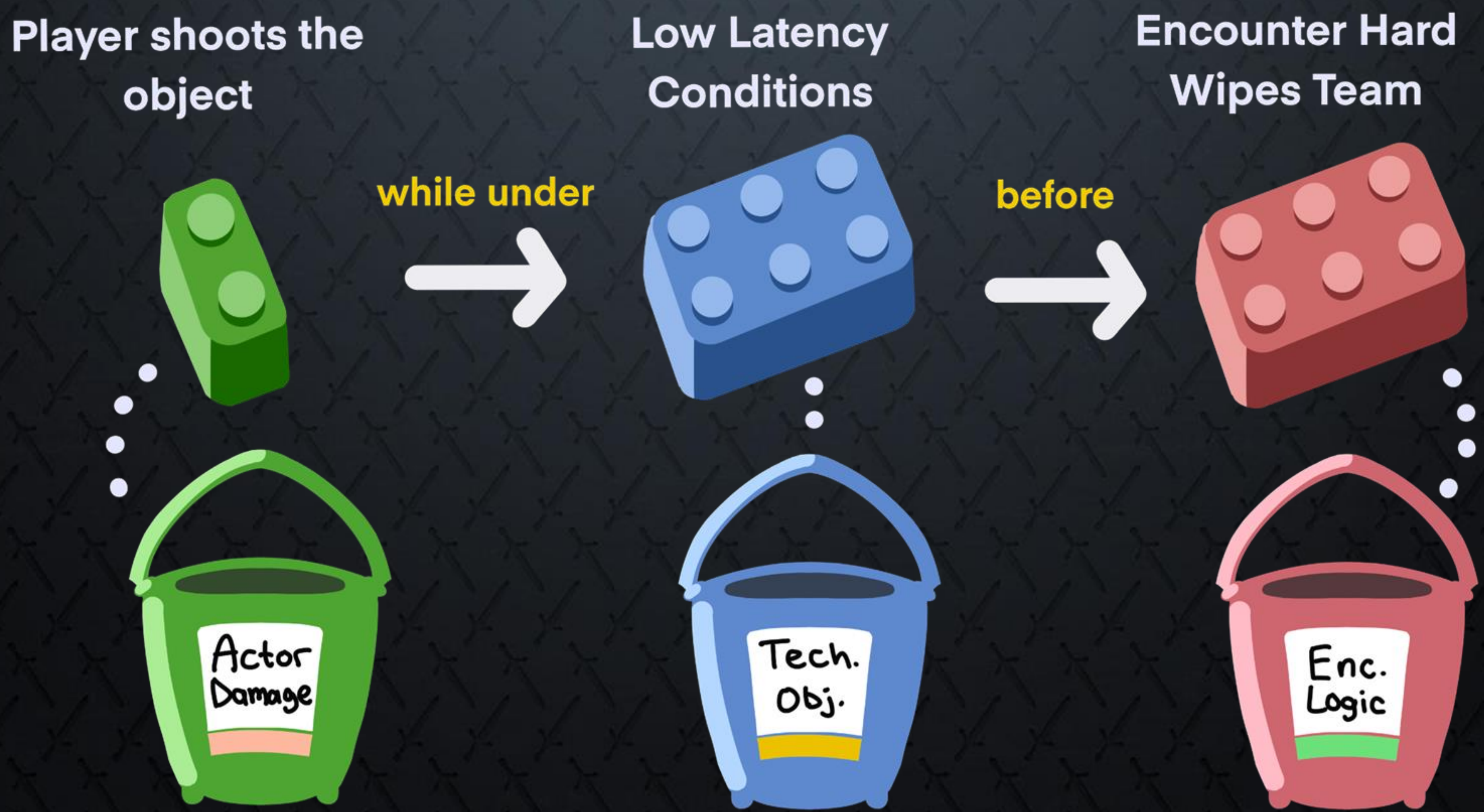
between

Not having buff vs having it

# Use Case: Directed Ad-Hoc

2. What happens when…

# Use Case: Directed Ad-Hoc



Player shoots the object

while under

Low Latency Conditions

before

Encounter Hard Wipes Team

# Takeaways:
# Directed Ad-Hoc Summary

- Ad hoc leads **discovery**.
- Building blocks represent a guide to that discovery and value.
- Sometimes a bug will appear, and sometimes it won’t.

# Combine Building Blocks: Costing & Planning

# * Quick Recap *

- We’ve made our reusable buckets and building blocks

- We’ve individually combined the building blocks for discovery (ad-hoc)

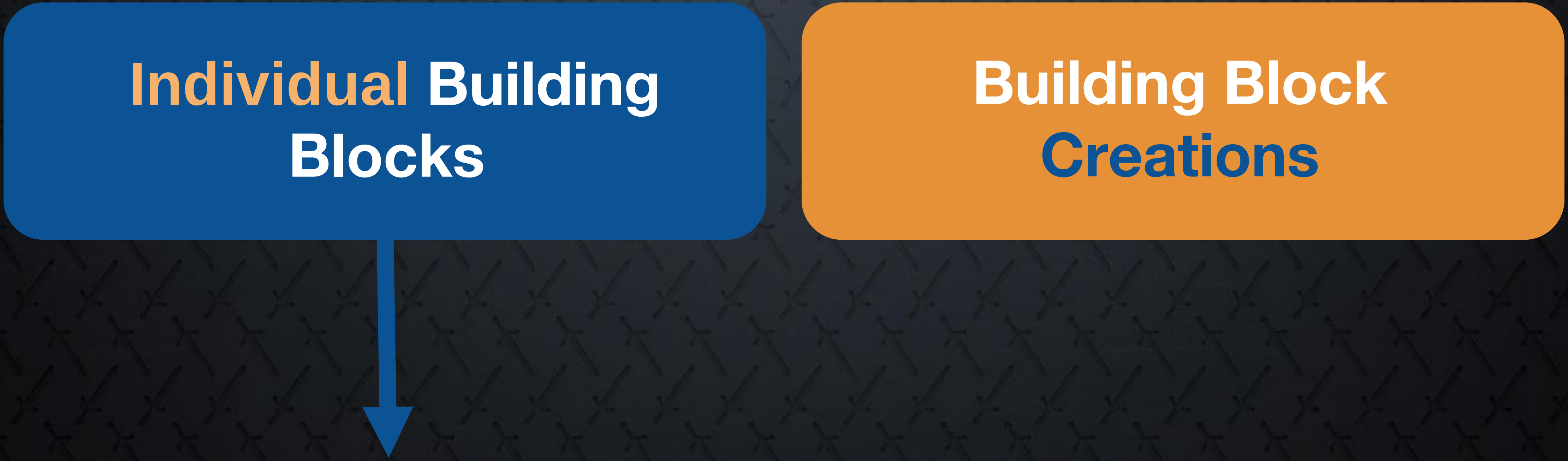
- …now it’s time to make larger creations!

# Building Blocks vs Creations

Individual Building Blocks

Building Block Creations

# Building Blocks vs Creations

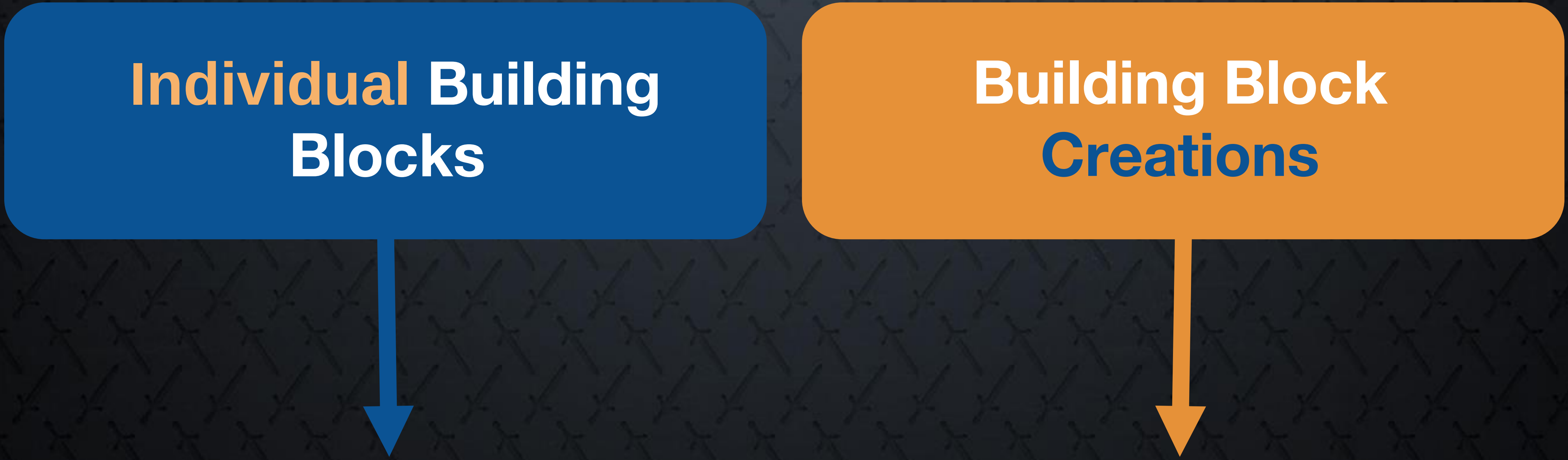


Individual Building Blocks

Building Block Creations

Focuses on the vulnerability of individual components

# Building Blocks vs Creations



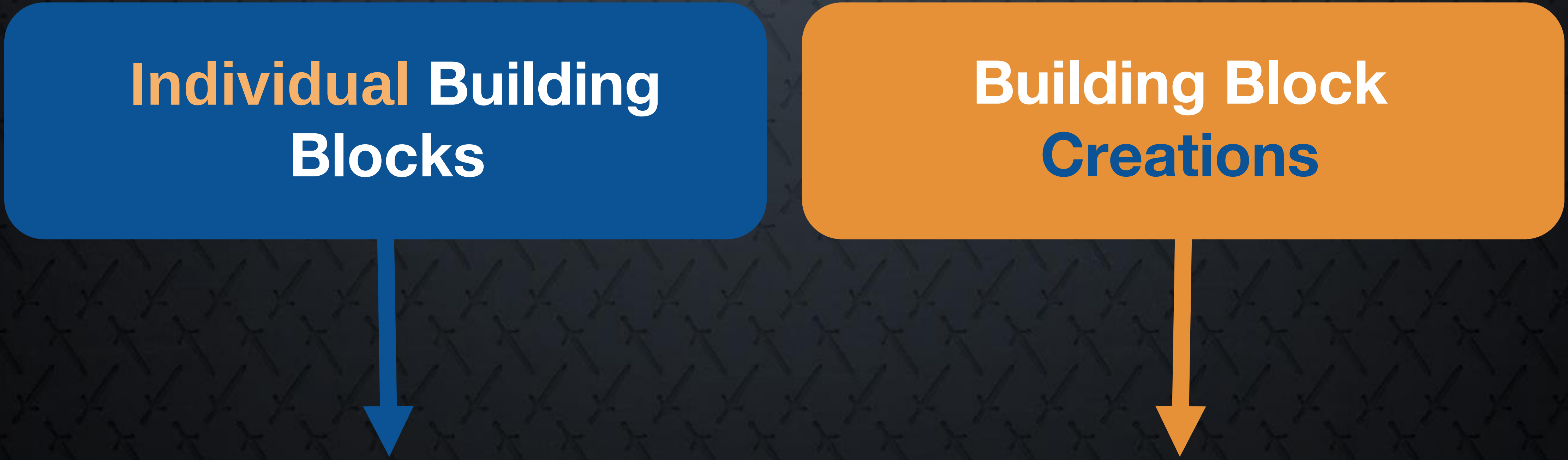
## Individual Building Blocks

Focuses on the vulnerability of individual components

## Building Block Creations

Focuses on the combined vulnerability of a system/feature

# Building Blocks vs Creations



Individual Building Blocks

Building Block Creations

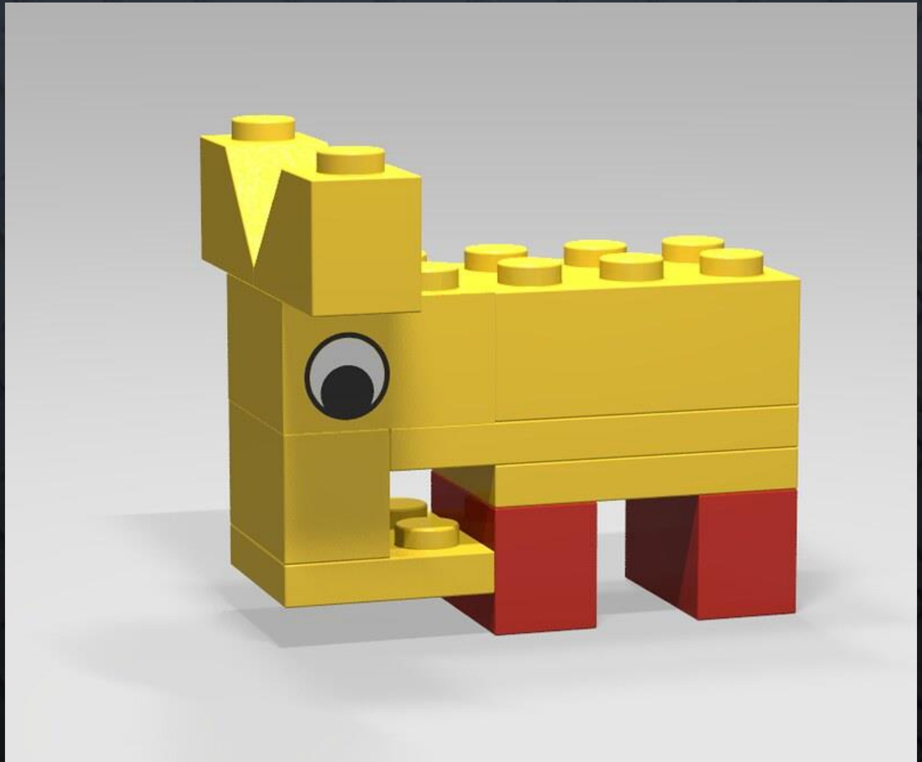
Focuses on the vulnerability of individual components

Focuses on the combined vulnerability of a system/feature

🤝

# Save. Your. Creations!

(We use User Stories and Documentation)



# Step 3.B - Combine pieces: Planning & Costing

# **Step 3.B - Combine** pieces: Planning & Costing

**Exercises for QA/Production:**

- Collaborate with feature leads
- Identify bug database trends
- Analyze the types of bugs that occur and strategize bug prevention

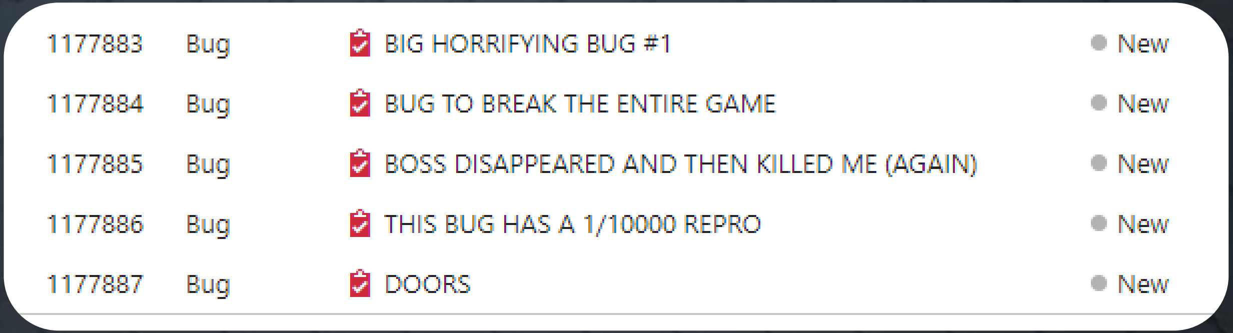
# Step 3.B - Combine pieces: Planning & Costing

**Exercises for QA/Production:**

- Collaborate with feature leads
- Identify bug database trends
- Analyze the types of bugs that occur and strategize bug prevention

**Building the Creation:**

- Identify commonly built features (e.g. Destiny bosses)
- Combine building blocks to determine your vulnerability and value scale
- Reuse creations for rapid iteration



| | | | |
|---|---|---|---|
| 1177883 | Bug | BIG HORRIFYING BUG #1 | ● New |
| 1177884 | Bug | BUG TO BREAK THE ENTIRE GAME | ● New |
| 1177885 | Bug | BOSS DISAPPEARED AND THEN KILLED ME (AGAIN) | ● New |
| 1177886 | Bug | THIS BUG HAS A 1/10000 REPRO | ● New |
| 1177887 | Bug | DOORS | ● New |

We think of a base creation...

We think of a base creation...

And then use it as our foundation to add on top of it for newer creations!

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

| **User Story:** [REDACTED] Testing Final Raid Boss Encounter | |
|---|---|
| | |
| | |

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

| | |
|---|---|
| **User Story:** [REDACTED] Testing Final Raid Boss Encounter | |
| **Description:** We're building a new boss. But this boss has a hat. We've never built a hat before.<br>We believe the following vulnerabilities will be newly introduced to this:<br>● [NIGHTMARE ISSUE #1]<br>● [THING TO CAUSE BUG TO REACTIVATE 6 TIMES] | |
| | |

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

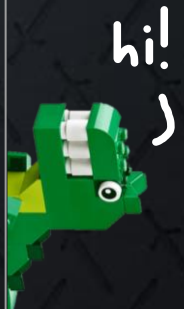
| | |
|---|---|
| **User Story:** [REDACTED] Testing Final Raid Boss Encounter | |
| **Description:** We're building a new boss. But this boss has a hat. We've never built a hat before.<br>We believe the following vulnerabilities will be newly introduced to this:<br>● [NIGHTMARE ISSUE #1]<br>● [THING TO CAUSE BUG TO REACTIVATE 6 TIMES] | |
| | **Linked User Stories/Test Pass/Playtest:**<br>• Generic Bosses: Vulnerability Creation<br>This is your prior building block creation acting as a baseline cost! |

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

| | |
|---|---|
| **User Story:** [REDACTED] Testing Final Raid Boss Encounter | |
| **Description:** We're building a new boss. But this boss has a hat. We've never built a hat before.<br>We believe the following vulnerabilities will be newly introduced to this:<br>● [NIGHTMARE ISSUE #1]<br>● [THING TO CAUSE BUG TO REACTIVATE 6 TIMES] | |
| | **Linked User Stories/Test Pass/Playtest:**<br>• Generic Bosses: Vulnerability Creation<br>This is your prior building block creation acting as a baseline cost! |

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

<table>
<tr><td colspan="2"><b>User Story:</b> [REDACTED] Testing Final Raid Boss Encounter</td></tr>
<tr><td rowspan="2"><b>Description:</b> We're building a new boss. But this boss has a hat. We've never built a hat before.<br>We believe the following vulnerabilities will be newly introduced to this:<br>● [NIGHTMARE ISSUE #1]<br>● [THING TO CAUSE BUG TO REACTIVATE 6 TIMES]</td><td>Vulnerability: VERY HIGH</td></tr>
<tr><td><b>Linked User Stories/Test Pass/Playtest:</b><br>• Generic Bosses: Vulnerability Creation<br>This is your prior building block creation acting as a baseline cost!</td></tr>
</table>



# Step 3.B - Combine pieces: Planning & Costing (Example Template)

**User Story:** [REDACTED] Testing Final Raid Boss Encounter

**Description:** We're building a new boss. But this boss has a hat. We've never built a hat before.

We believe the following vulnerabilities will be newly introduced to this:

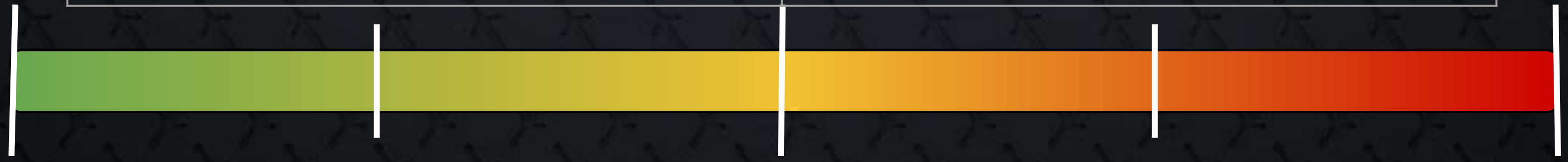
- [NIGHTMARE ISSUE #1]
- [THING TO CAUSE BUG TO REACTIVATE 6 TIMES]

Vulnerability: VERY HIGH

**Linked User Stories/Test Pass/Playtest:**

- Generic Bosses: Vulnerability Creation

This is your prior building block creation acting as a baseline cost!



**Very Low Vulnerability**
**< 5 Bugs**

**Low Vulnerability**
**>= 5 Bugs**

**Medium Vulnerability**
**>= 20 Bugs**

**High Vulnerability**
**>= 35 Bugs**

**Very High Vulnerability**
**>= 50 Bugs**

***Bug numbers are scaled according to your project!***

# Step 3.B - Combine pieces: Planning & Costing (Example Template)

| | |
|---|---|
| **User Story:** [REDACTED] Testing Final Raid Boss Encounter | |
| **Description:** We're building a new boss. But this boss has a hat. We've never built a hat before.<br>We believe the following vulnerabilities will be newly introduced to this:<br>● [NIGHTMARE ISSUE #1]<br>● [THING TO CAUSE BUG TO REACTIVATE 6 TIMES] | Vulnerability: VERY HIGH |
| | **Linked User Stories/Test Pass/Playtest:**<br>• Generic Bosses: Vulnerability Creation<br>This is your prior building block creation acting as a baseline cost! |

**Very Low Vulnerability**
**Mostly small sized building blocks**

**Medium Vulnerability**
**Even split in building block sizes**

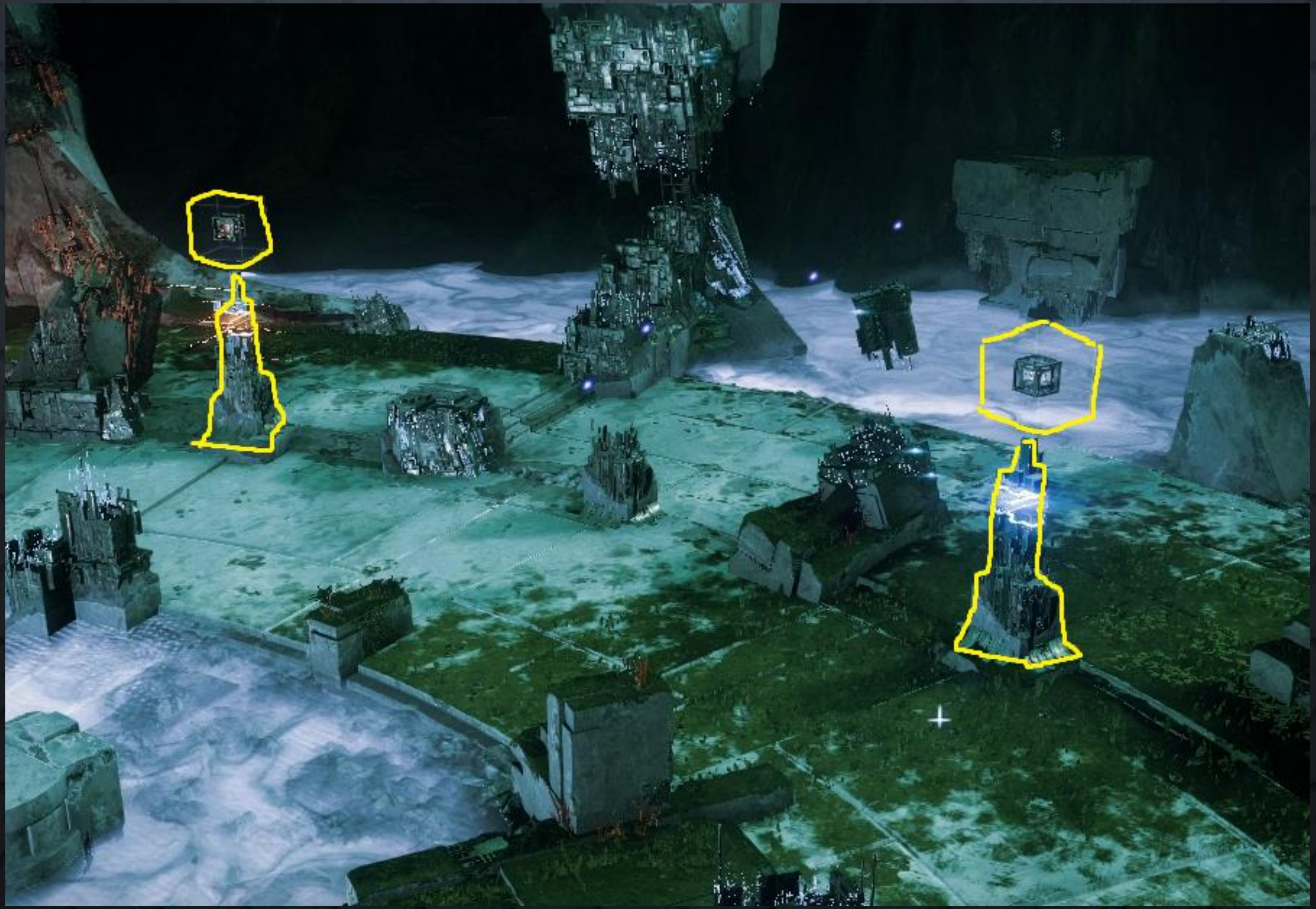
**Very High Vulnerability**
**Mostly large sized building blocks**

***Scale by identifying value over pure bug counts***

# Takeaways:
# Planning & Costing Summary

- Streamline costing and planning by building creations.
- Focus scaling cost by vulnerability and player value.
- Use building block creations as a knowledge base

# Combine Building Blocks: Live Investigation

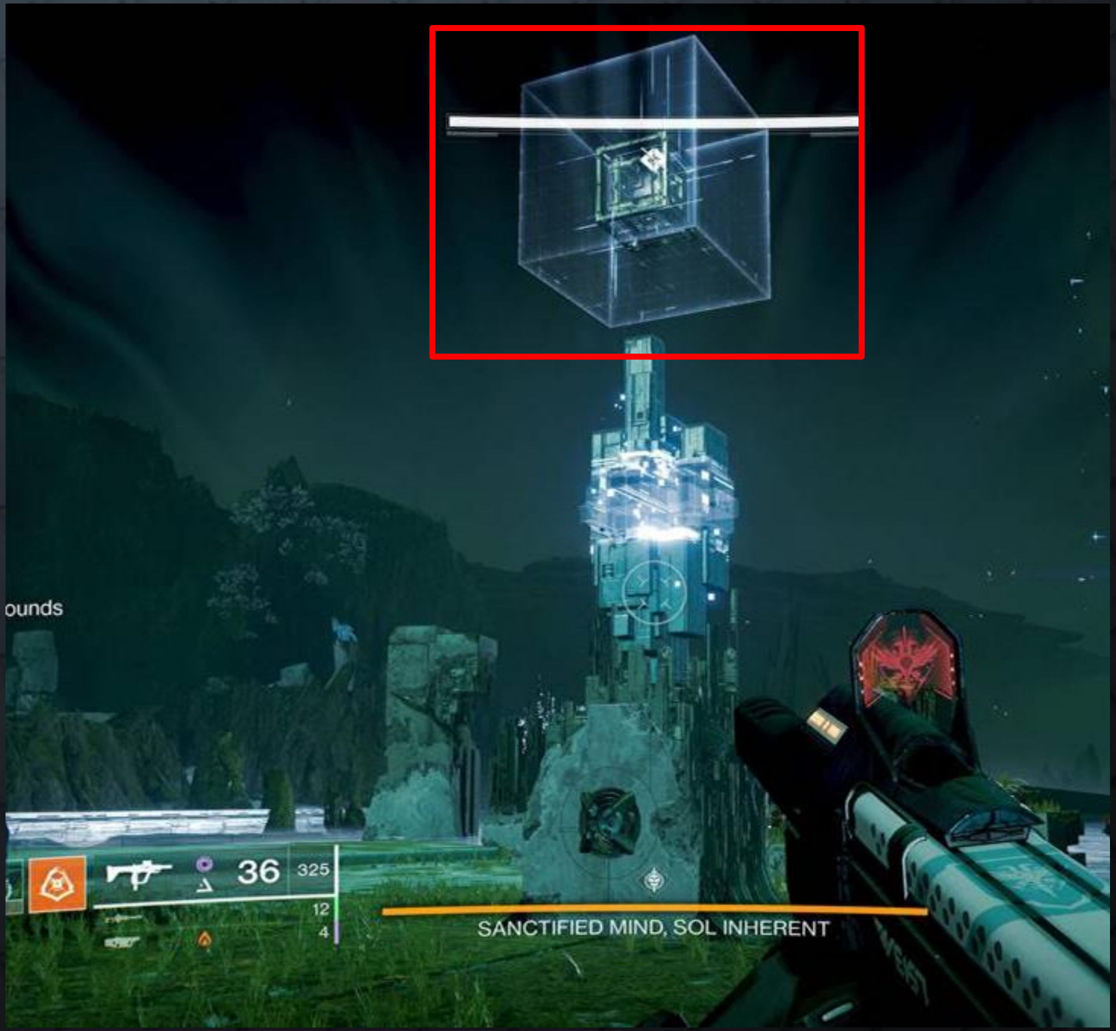


*Example featuring Garden of Salvation (Destiny 2 – Shadowkeep)*

# Use Case – Live Bug Investigation

**Live report:** Sometimes players say shooting the box returns “Immune” when they expect it to be damaged.

Repro steps are unknown.



# Step 1: Isolating the Problem

# Step 1: Isolating the Problem

- We focused solely on the box and its behaviors

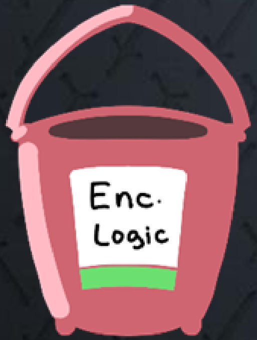
# Step 1: Isolating the Problem

- We focused solely on the box and its behaviors
- We knew that the issue involves *shooting* it.

# Step 1: Isolating the Problem

- We focused solely on the box and its behaviors
- We knew that the issue involves *shooting* it.
- We knew that the box goes through different states.

# Step 2: Identify Buckets and Building Blocks



# Step 2: Identify Buckets and Building Blocks

| Enc. Logic | Tech Obj. | Actor Dmg. |
|---|---|---|
| Randomizers, Combatant Blockers , Race Conditions | Low Latency, Buff Application, Object Loops | Weapon Procs, Damage Multipliers, Perk Activation/Deactivation |
| Multiple prefab placements, Encounter Loops | Box State (On/Off toggle), Line of sight, Tower State (Uncharged/Charged), | Damage Over Time, Explosive Damage |
| Sparrow volumes, start conditions | Object Health Pools | Automatic Fire |

# Step 2: Identify Buckets and Building Blocks

| | |
|---|---|
| 🧱 | Randomizers, Combatant Blockers , Race Conditions |
| 🧱 | Multiple prefab placements, Encounter Loops |
| 🧱 | Sparrow volumes, start conditions |

| | |
|---|---|
| 🧱 | **Low Latency,** Buff Application, Object Loops |
| 🧱 | Box State (On/Off toggle), Line of sight, Tower State (Uncharged/Charged), |
| 🧱 | Object Health Pools |

| | |
|---|---|
| 🧱 | Weapon Procs, Damage Multipliers, Perk Activation/Deactivation |
| 🧱 | Damage Over Time, Explosive Damage |
| 🧱 | **Automatic Fire** |

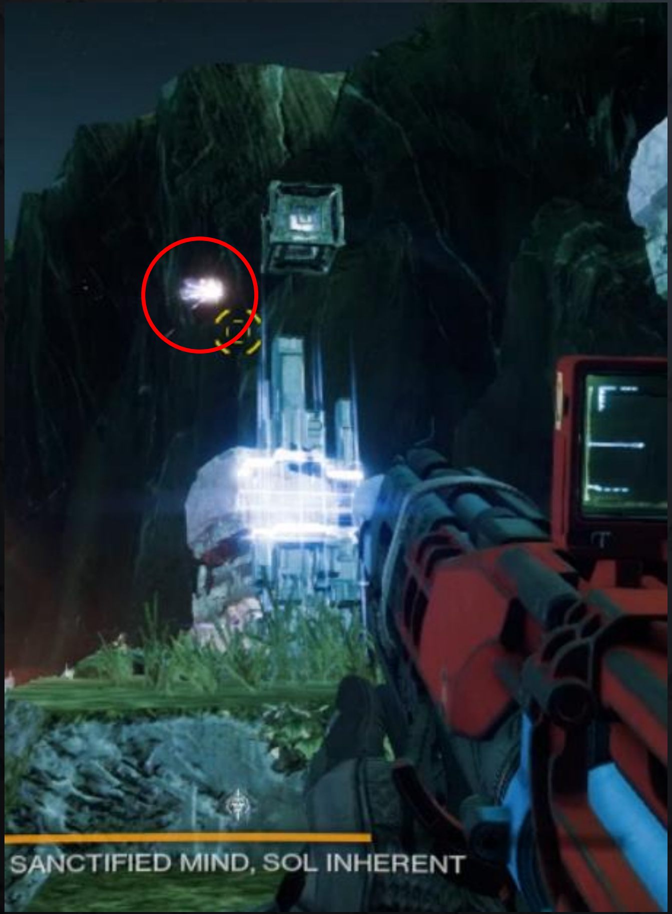
# Step 2: Identify Buckets and Building Blocks

| | | |
|---|---|---|
| Randomizers, Combatant Blockers , Race Conditions | **Low Latency,** Buff Application, Object Loops | Weapon Procs, Damage Multipliers, Perk Activation/Deactivation |
| **Multiple prefab placements,** Encounter Loops | Box State (On/Off toggle), Line of sight, Tower State (Uncharged/Charged), | Damage Over Time, **Explosive Damage** |
| Sparrow volumes, start conditions | Object Health Pools | Automatic Fire |

BUNGIE

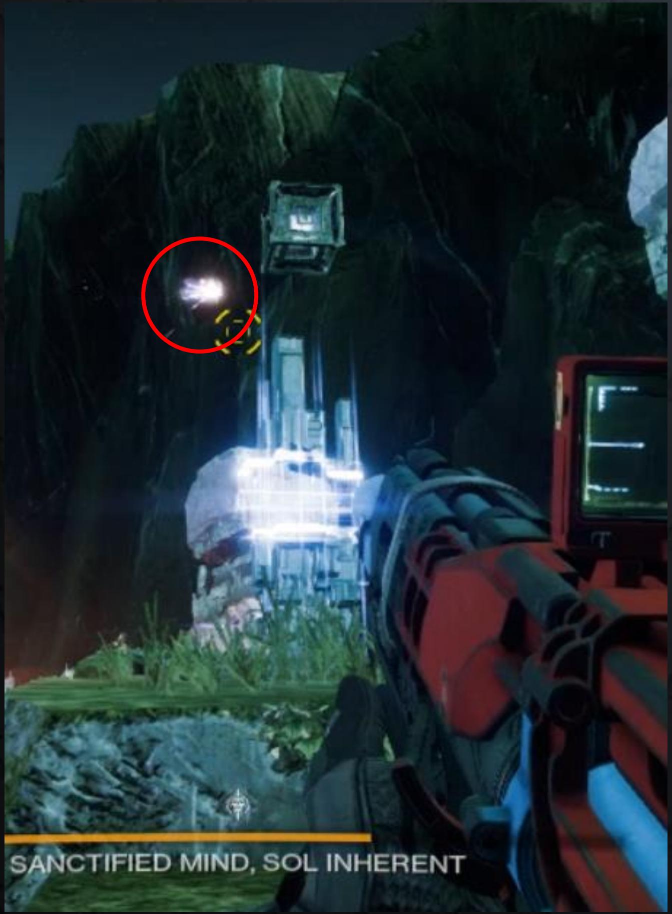
# Step 2: Identify Buckets and Building Blocks

*What's the bullet hitting?*

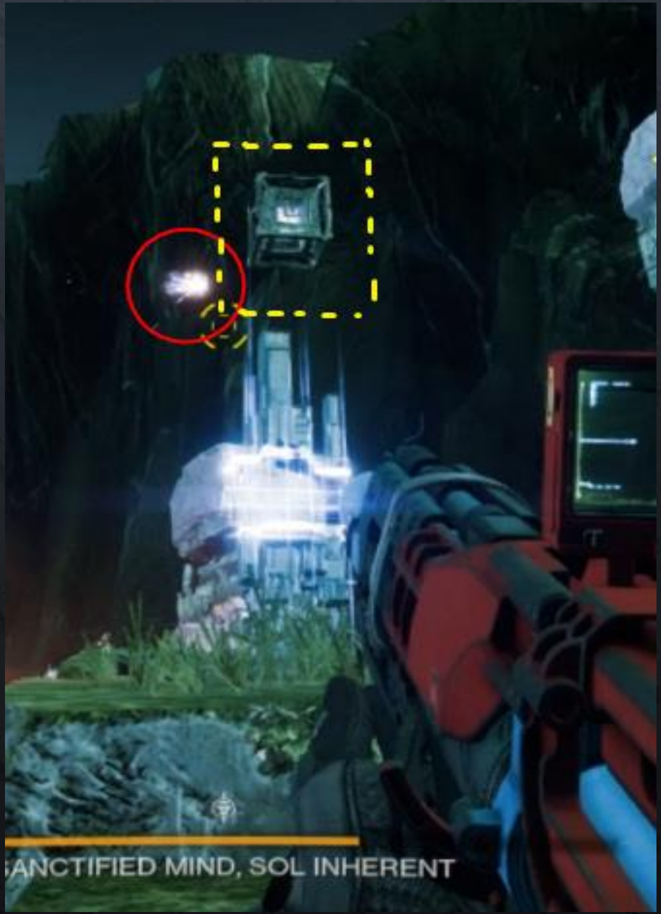


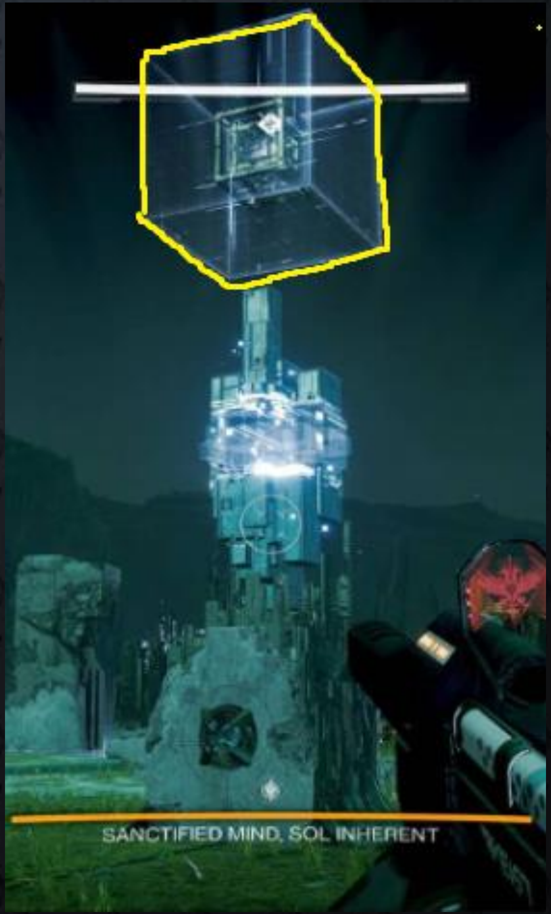
# Step 2: Identify Buckets and Building Blocks

*What’s the bullet hitting?*



*...ah!!*



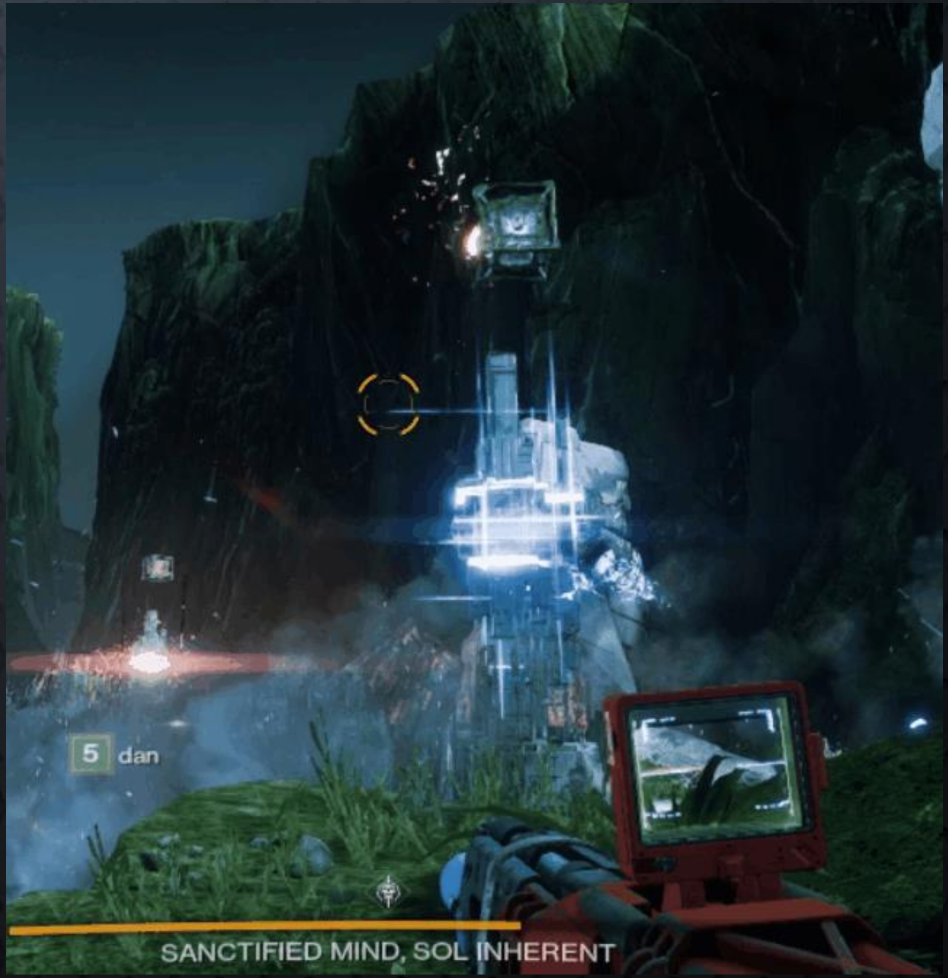


# Step 2: Identify Buckets and Building Blocks

| | | |
|---|---|---|
| Randomizers, Combatant Blockers , Race Conditions | Low Latency, Buff Application, Object Loops | Weapon Procs, Damage Multipliers, Perk Activation/Deactivation |
| Multiple prefab placements, **Encounter Loops** | **Box State (On/Off toggle),** Line of sight, Tower State (Uncharged/Charged), | Damage Over Time, **Explosive Damage** |
| Sparrow volumes, start conditions | Object Health Pools | Automatic Fire |

# Bam! Issue Reproduced!

- Discovered by combination of building blocks
- Invisible geometry was formed around cube
- Encounter wasn't getting rid of the invisible geometry



# Takeaway: Live Ingestigation Summary

- Isolate problems with building blocks
- Visual checklist of repro attempts
- Lead us to future bug prevention!

# Takeaways from Presentation

We've learned...

# Takeaways from Presentation

We've learned...

- How to Identify > Create > Combine!

# Takeaways from Presentation

We've learned...

- How to Identify > Create > Combine!
- Guidelines to explore your own game's buckets and building blocks.

# Takeaways from Presentation

We've learned...

- How to Identify > Create > Combine!
- Guidelines to explore your own game's buckets and building blocks.
- Modularity of building blocks and how they retain value throughout projects.

# Takeaways from Presentation

We've learned...

- How to Identify > Create > Combine!
- Guidelines to explore your own game's buckets and building blocks.
- Modularity of building blocks and how they retain value throughout projects.
- Boxes are a pain.

# Get to Building!

**Twitter:** @ChrisLRios | **LinkedIn:** chrislrios | **Email:** crios@bungie.com