



THE HITCHHIKER'S GUIDE TO LAUNCHING LIVE SERVICE GAMES

Phase 1



Table of Contents

Introduction	3
What is the purpose of this guide?	4
Who is this guide for?	4
The game development process: What does this guide cover?	5
To Live or Not to Live: The Pros and Cons of Creating a Live Game vs. a Non-Live Game	9
What is a live game?	10
Benefits of creating a live game	11
Drawbacks of creating a live game	13
Live games vs. non-live games: factors to consider	15
Live games vs. non-live games: key metrics	16
Overcoming challenges in developing a live game	17
To live or not to live?	19
Prototyping and Continuous Playtesting for Live Game Development	20
What is prototyping?	21
Best practices for prototyping	24
Continuous playtesting	26
Best practices for continuous playtesting	27
Common pitfalls when conducting playtests	29
The importance of prototyping and playtesting	31

The Hitchhiker's Guide to Launching Live Service Games

The gaming industry continues to grow as more creators release their games. But how does this process work? Don't panic! The Hitchhiker's Guide to Launching Live Service Games will take you through how to ship a live game from start to finish, simplifying the process for game developers so all you'll need to do is bring a towel and strap in for the ride.



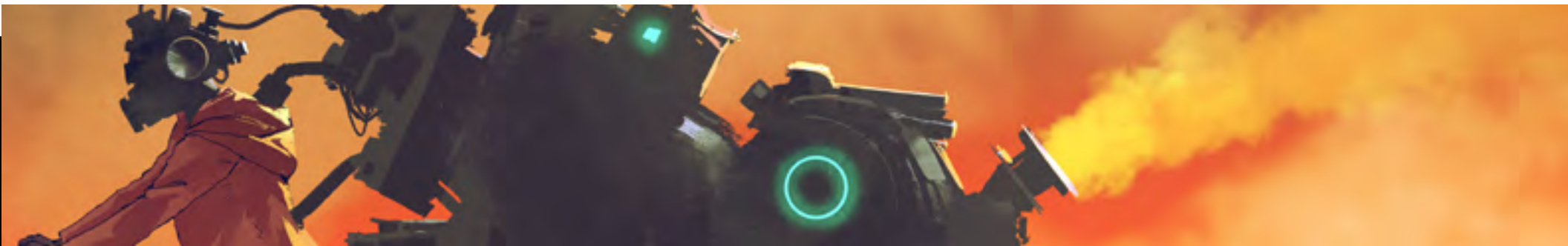
What is the purpose of this guide?

There are tons of game development-centric books on the market, but when it comes to shipping live games, these books lack information in two major areas: the general production process and the world of online gaming.

Who is this guide for?

Major game companies like Electronic Arts and Ubisoft have the game development process down to a tee, but this information is unavailable to the general public. Indie game developers, all the way through AAA game developers, are left to their own devices. Since no definitive manuals are available, this guide aims to teach how the process works.

Creators interested in building and shipping a live game will find value in this guide. Additionally, anybody who has yet to enter the game industry but is interested in doing so can use this guide to better understand the game development process. Whether you are a developer creating a game for the first time or just someone who wants to learn more about game production, this guide is for anyone interested in expanding their knowledge on how to ship games.



Game development process: What does this guide cover?

This comprehensive step-by-step guide provides a little more than the answer 42. It's dedicated to providing teams with the necessary information on creating an online game from start to finish. From conception to launch, *The Hitchhiker's Guide to Launching Live Service Games* will allow teams to learn the basics and take them through the video game production process by breaking down the cycle into five phases.

- **Phase 1:** Concept and Prototyping
- **Phase 2:** Pre-Production and Vertical Slice
- **Phase 3:** Production
- **Phase 4:** Road to Launch
- **Phase 5:** Live and Post-Launch

The guide will provide teams with the basic suggested framework for producing and launching a video game. Our phases are not definitive: We understand that each company may work at their own pace and choose to loop back to previous phases. This guide is meant to help you understand the general live game development process from start to finish.

Phase 1: Concept and Prototyping

The first phase focuses on the game's initial concept. Here, you will determine the basic idea of the game and what it consists of. This means answering all the questions surrounding the core idea and better understanding your characters, storylines, settings, themes, gameplay, and more. Before you start building, it is essential to answer questions such as:

- Who is the intended audience?
- What is the purpose of the game?
- Why do people want to play it?
- How long will this process take?

After the initial concept has been developed, the prototyping stage enables you and your team to build a preliminary version of the game to assess its marketability. If the team determines a demand for such a product, attention can then turn to obtaining seed funding to finance the game's production.

With seed funding in place, the game concept agreed upon, and sufficient information gathered from the prototyping stage, your team can proceed to phase two: pre-production.

Phase 2: Pre-Production and Vertical Slice

The pre-production phase can be the most prolonged phase during game development. Pre-production is precisely what it sounds like—preparing for production. This section focuses on finalizing the game's foundation, determining what needs to be done, and delegating these responsibilities among the teams. A vertical slice in game development is a representation of a finished feature or gaming experience in a game that is still under development. It is a brief, standalone segment of a game designed to preview how the overall experience will look. It enables the design team to evaluate the gameplay mechanics and gather input from interested parties like focus groups, publishers, and investors. Based on the information received from the vertical slice, developers can make informed decisions about the scope of production.

Ask yourself these questions:

- What platforms and tools need to be used?
- How will we measure the quality of the game?
- Do we plan to do everything ourselves? Or do we want to use a third-party company?
- What features and gameplay mechanics should be included in the vertical slice?
- How much time and resources should be allocated to creating a vertical slice?
- How can we ensure that the vertical slice accurately represents the overall gameplay experience?

Once the big questions are answered, you and your team will have a better understanding of what the final outcome will look like. Decisions made during pre-production give the team a more definitive idea of the end product. After there is a team consensus, pre-production transitions over to the production phase.

Phase 3: Production

During the production stage, efforts and resources are dedicated to creating the end product.

Game design often evolves continuously during the production process. The production phase involves coding, asset creation, level design, playtesting, and bug fixing. The primary goal of production is to create the game as planned in the pre-production phase, test it, refine it, and prepare it for release. However, it is not uncommon for developers to run into hurdles that shift the original scope of the project. Once production is in effect, there may be a constant battle between creating and descoping the plan as development progresses. The production process may test one's ability to adapt and adjust now that the pieces are falling further into place.

During the production phase, key areas covered in the guide will include testing and playtesting, user-generated content, game telemetry, analytics, and more.

Phase 4: Road to Launch

Production is complete, and it is nearly time to release the product. Now what? Phase four is the pre-launch stage and covers what needs to be done as you prepare to publish your game to your audience.

Numerous live games have faced failure during their launch phase. The success of a live game hinges on several factors, such as the quality of the game at its initial release, the development team's capacity to provide consistent updates and support, and the player community's willingness to persevere through the game's ebbs and flows.

One way to mitigate the risk of failure during the launch phase is by offering alpha and beta testing or early access to players, which can help identify and fix issues before the official release. Incorporating these testing phases into your pre-launch stage can help ensure the quality of the game at launch and boost player community trust in the game's long-term potential.

To prevent you from failing near the finish line, our guide will help you with your road to launch by focusing on topics such as load testing, console certifications, security, and more.

Phase 5: Live

Although your game may be live and complete, the work continues. The final phase focuses on ensuring that the game stays operational and relevant. This can include fixing bugs, following up with players, and tracking your customer base.

Once your game is live, it is important to continually work on the product to ensure it performs to standard. To ensure your game continues to excel during the post-launch phase, the guide will cover further information on areas such as how to track the lifetime value of players, how to respond to incidents, why you should avoid releasing content via title update, and more. It is important to continually improve your game post-launch since user segmentation and optimizing player spending can be leveraged to analyze player behavior and preferences, fine-tune the monetization strategy, and enhance the overall player experience for increased engagement and loyalty.

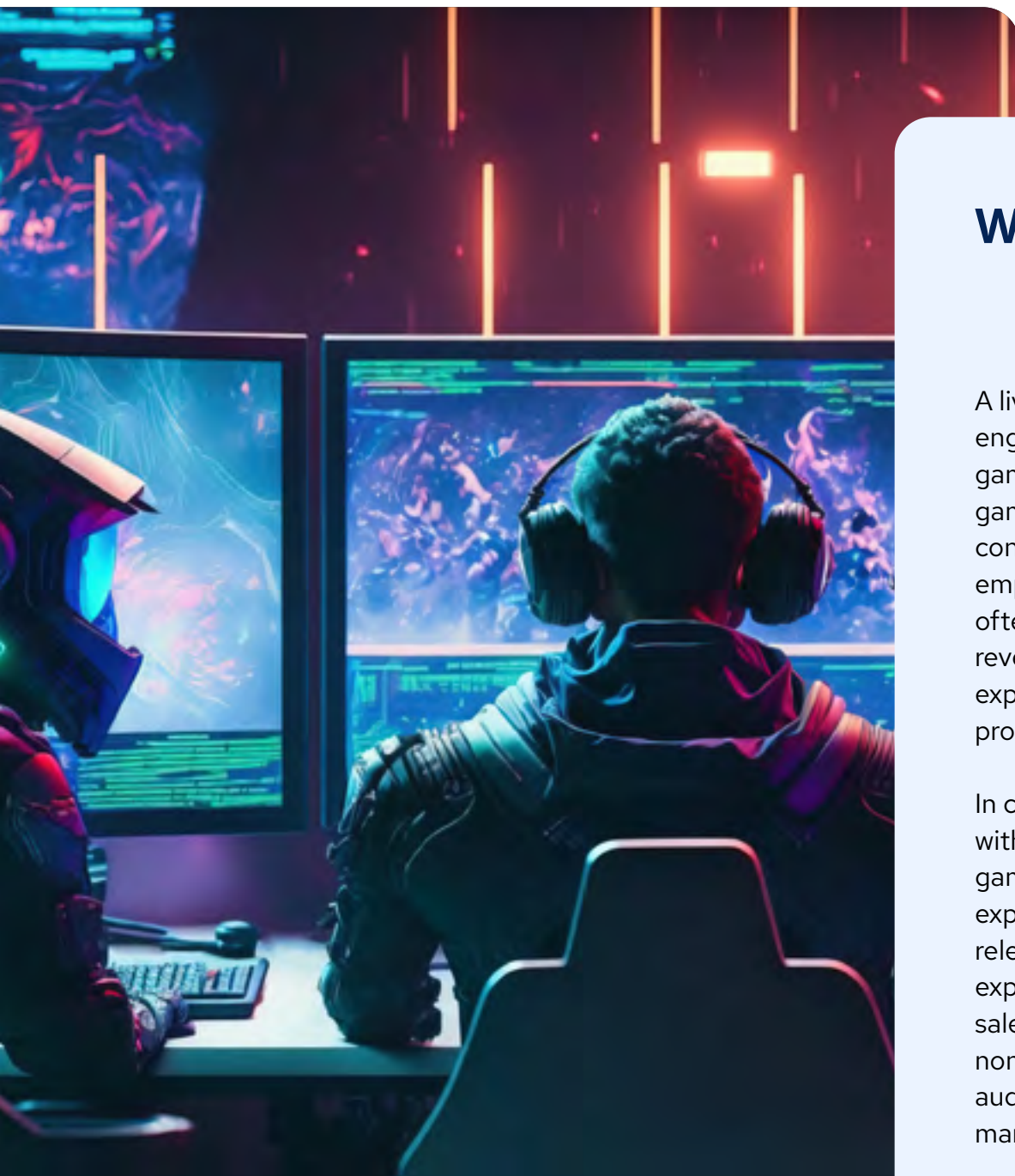


To Live or Not to Live: The Pros and Cons of Creating a Live Game vs. a Non-Live Game

In this chapter of *The Hitchhiker's Guide to Launching Live Service Games*, we explore a critical decision that confronts game developers: the dilemma of choosing between developing a live game or a non-live game.

Making an informed decision requires knowing the main distinctions between these two strategies. This article explores the intricacies of live and non-live games, their respective benefits and drawbacks, factors to consider when deciding between the two, and strategies for overcoming challenges in live game development.





What is a live game?

A live game is a video game that prioritizes continuous engagement and interaction with players. Unlike traditional games that have an end to their development cycle, live games provide ongoing updates, expansions, and new content over an extended period. Live games strongly emphasize player interaction and community building and often employ various monetization strategies to generate revenue. The objective is to develop a dynamic, engaging experience that keeps players interested in the game for a prolonged period.

In contrast, non-live games are standalone experiences without consistent updates and generally have a finite gameplay lifespan. Non-live games are standalone experiences without regular updates or expansions. Upon release, they offer a complete narrative or gameplay experience and primarily generate revenue through initial sales. Choosing between developing a live game versus a non-live game can be influenced by variables such as target audience, game type, the development team's skills, and market trends.

Benefits of Creating a Live Game

Live games provide a win-win situation for players and developers thanks to an immersive player experience, improved customer retention, and sustainable income creation.



Enhanced Player Experience

A live game offers unparalleled engagement for players, providing them with a more immersive and captivating experience. Unlike games with a definitive end, many live games may offer updates to keep the player further engaged.

For example, developers may release campaigns related to current events or popular topics. A game can create excitement and keep players interested if it delivers unique events or missions based on actual occurrences. Another example is celebrating in-game anniversaries, where creators plan special events or incentives to recognize significant turning points in the game's history.

As a result, players may feel more part of the game, which may motivate them to keep playing. Lastly, remixing the game's content for new and unique experiences can add freshness to the gameplay and make it more engaging.

This open-ended nature of gameplay keeps players motivated and challenged as they strive for personal improvement and mastery. With no limits to their progress, players can fully immerse themselves in the game's world, creating a deep sense of engagement and satisfaction.

Increased Customer Retention and Community Building

The continuous engagement offered by a live game fosters strong customer retention and cultivates a dedicated community of players. As players become deeply invested in the game, they develop a sense of loyalty and attachment. This sense of belonging to a vibrant gaming community further strengthens player retention as players actively participate in discussions, events, and competitions. The shared experiences and interactions within the community create a strong camaraderie, enhancing the overall gaming experience and keeping players coming back for more.

Sustainable Revenue Generation

Live gaming promotes excellent customer retention and builds a passionate player base. By continuously offering new features, items, and content packs, developers provide players with enticing opportunities for in-game purchases. This adds value to the player experience and generates revenue for the developers. The steady income stream enables the game's creators to invest in the game's continuous development and provide frequent updates, enhancements, and expansions. This revenue-driven model ensures the longevity and success of the live match, benefiting both players and developers.





Drawbacks of Creating a Live Game

On the contrary, there are a handful of downsides to producing a live game. From ongoing costs and financial burdens to the pressure of meeting player expectations and the risk of losing the player base, navigating these drawbacks requires careful consideration.

Ongoing Costs and Financial Burden

Ongoing costs associated with operating a live game can pose a significant drawback for developers. Unlike non-live games, live games demand continuous server maintenance, customer support, and infrastructure upkeep. These continuing costs can accumulate over time and strain the development team's financial resources. Additionally, as the player base grows, the costs associated with scaling up the game's infrastructure and server capacity also increase. Maintaining these ongoing costs can become challenging—especially for smaller studios with limited resources.

Pressure to Meet Player Expectations


Player expectations for ongoing content updates and expansions can pressure the development team immensely. In a live game, players anticipate regular updates, new features, and expansions to keep the game fresh and exciting. Meeting and exceeding customer expectations requires a dedicated team of developers who can consistently deliver high-quality content within tight timelines. The constant demand for new content can lead to burnout and compromise the quality of the updates. Failing to meet player expectations can result in player dissatisfaction and a drop in popularity.

Risk of Losing Player Base

Arguably one of the most significant risks associated with a live game is the potential failure to maintain its player base over time. Today, players have thousands of options to choose from. If a live game fails to produce engaging updates, address player concerns, or provide meaningful content, players may lose interest and move on to other games.


This can lead to a dwindling player base, making it challenging to sustain the game's ecosystem and revenue stream. There is little value to continuing live game production if a consistent and dedicated player base is lacking.





Live Games vs. Non-Live Games: Factors to Consider

Understanding the difference between the two types is just the beginning. Factors that developers should consider when deciding whether to create a live game or a non-live game vary depending on several key aspects.

- **Target audience preferences:** Determine if the intended player base is more inclined towards continuous engagement and long-term progression or a self-contained experience with a definitive end.
 - **Game genre:** Certain genres, such as MMOs or online multiplayer games, lend themselves more naturally to live game formats due to their social and interactive nature.
 - **Development team's capabilities:** Assess the team's expertise and experience in ongoing content updates, community management, and server infrastructure management, which are crucial for live game development.
 - **Financial resources:** Evaluate the economic viability of sustaining ongoing costs of operating a live game, such as server maintenance, customer support, and infrastructure upkeep.
 - **Market trends:** Analyze current trends in the gaming industry and determine if the target audience actively seeks live games or if there is still a demand for non-live games with focused narratives or single-player experiences.
- 

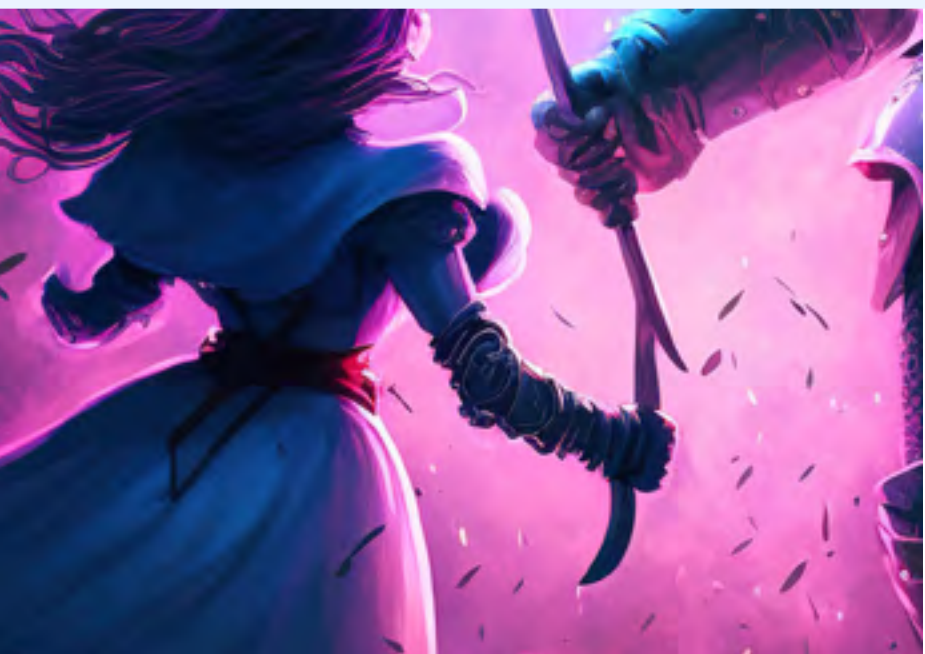


Live Games vs. Non-Live Games: Key Metrics

From player engagement and in-game monetization metrics to player progression and satisfaction indicators, understanding the key metrics is vital for optimizing game development. Dive into the details and unlock valuable insights to drive the success of your game.

Measuring Success in Live Games: Key Metrics

Standard metrics used to measure the success of a live game differ significantly from those used for a non-live game. In the context of live games, player engagement metrics take center stage. These include metrics such as:

- **Daily Active Users (DAU):** The number of unique users actively engaging with the game daily.
 - **Monthly Active Users (MAU):** The number of individual users actively engaging with the game within a month.
 - **Average Session Length:** A player's average time in a single gaming session.
 - **Retention Rate:** The percentage of players who continue to play or engage with the game over a specific period.
 - **Average Revenue per User (ARPU):** The average revenue generated per user.
 - **Lifetime Value (LTV):** The predicted total revenue a player is expected to create throughout their entire lifespan as a game customer.
- 

Overcoming Challenges in Developing a Live Game

Developing a live game comes with its fair share of challenges. From managing ongoing costs and retaining players to balancing content updates and maintenance tasks, navigating these obstacles requires careful planning and effective strategies.



Managing Ongoing Costs of Operating a Live Game

Ongoing costs can quickly deplete your resources if you and your team are unprepared. Analyze and optimize server infrastructure to minimize costs without compromising performance. Utilizing cloud-based solutions and scalable architecture can help scale resources based on player demand and, as a result, reduce unnecessary expenses. Additionally, exploring alternative monetization methods, such as in-game advertisements, can provide additional revenue streams to offset operating costs. Regular evaluation of operational expenses and identifying areas for cost optimization are crucial to maintaining financial sustainability in the long term.

Retaining Players and Preventing Churn in a Live Game

With no definitive end, how do developers ensure players stay interested? Building a solid community and fostering player engagement through social features, forums, and events encourages player retention. Implementing a robust player feedback system and promptly addressing player concerns and suggestions helps to enhance the player experience, foster loyalty, and reduce churn. Monitoring player engagement metrics, analyzing player behavior, and proactively addressing factors contributing to churn are vital for maintaining a thriving player base.

Balancing Content Updates with Maintenance Tasks

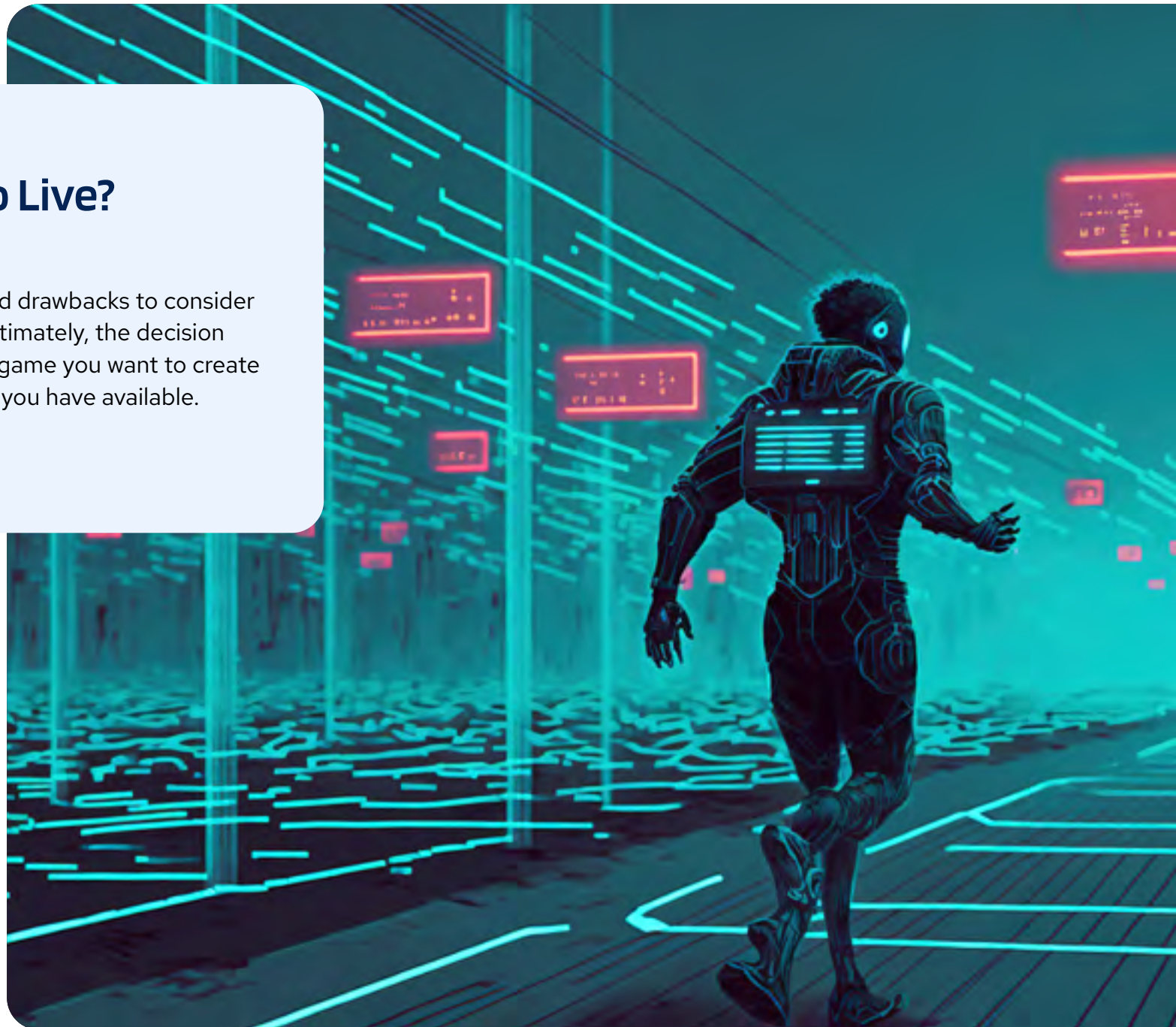
Developers must strike a balance between the requirement for continuing bug fixes and other maintenance duties in a live game. A stable and bug-free gaming environment may be maintained while still providing users with fresh and intriguing content. A fluid gameplay experience is guaranteed by prioritizing essential bug fixes, performance enhancements, and scheduled content updates.

Regular communication with the player community about impending upgrades, bug fixes, and maintenance plans is necessary to manage player expectations and uphold openness.



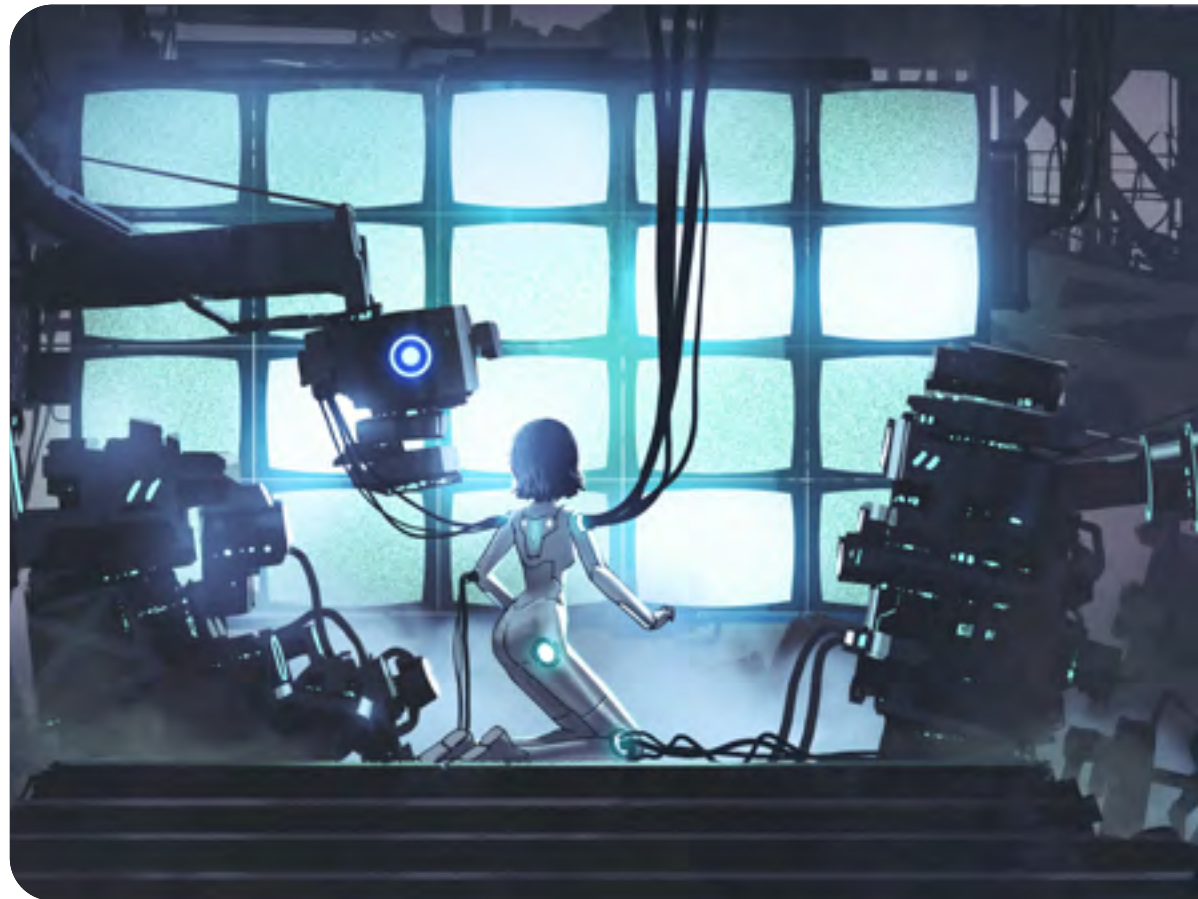
To Live or Not to Live?

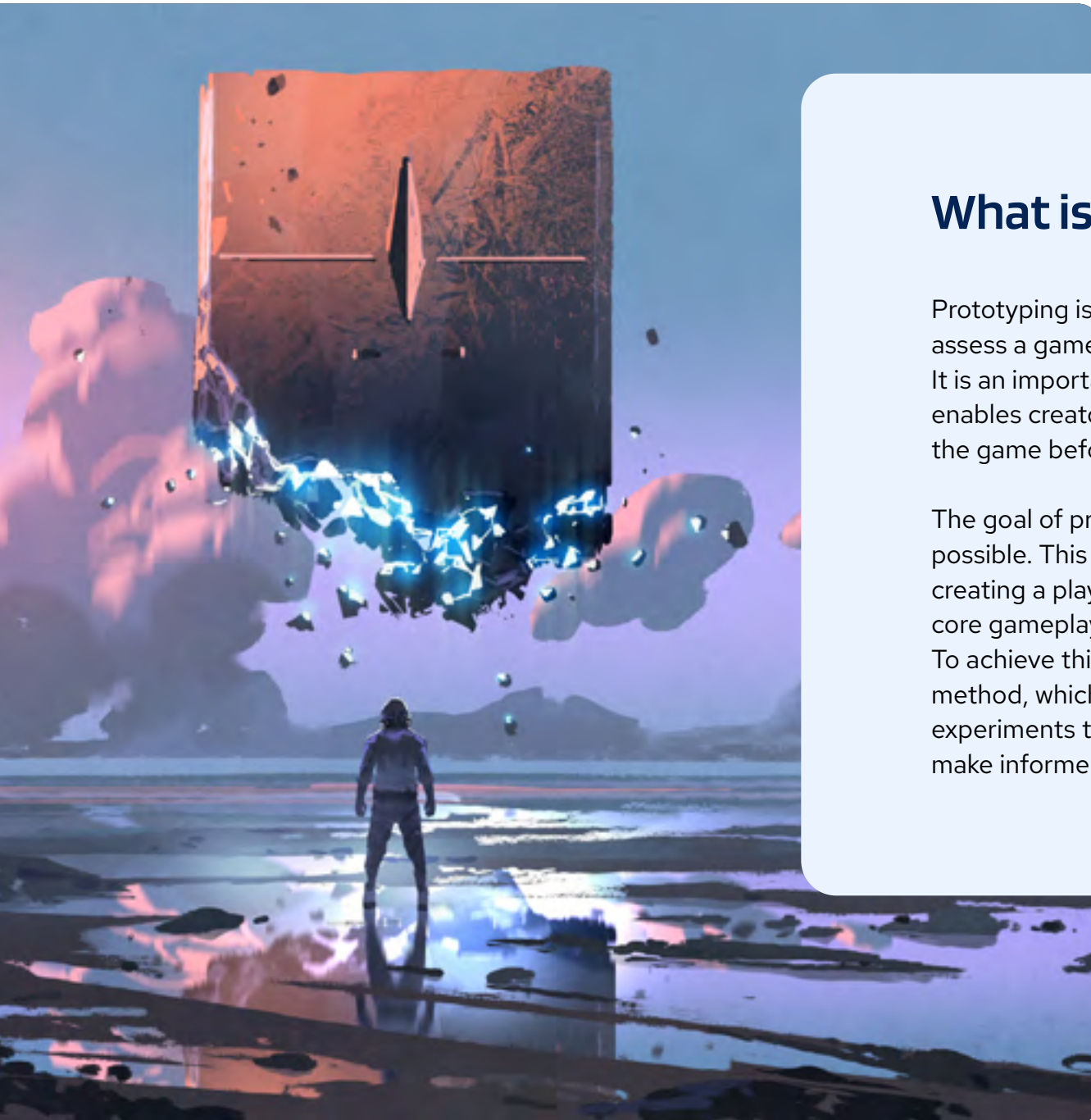
There are several benefits and drawbacks to consider when creating a live game. Ultimately, the decision comes down to what type of game you want to create and the amount of resources you have available.



Prototyping and Continuous Playtesting for Live Game Development

In this chapter of *The Hitchhiker's Guide to Launching Live Service Games*, we explore the critical domains of prototyping and continuous playtesting. We look at the vital role of iterative design, user feedback, and rapid experimentation in creating successful and engaging live service games.





What is prototyping?

Prototyping is a process that allows developers to test and assess a game's mechanics, playability, and overall concept. It is an important stage in game development since it enables creators to find potential challenges and improve the game before dedicating significant time and resources.

The goal of prototyping is to find the fun as quickly as possible. This means that developers must focus on creating a playable version of the game that showcases the core gameplay mechanics and the overall user experience. To achieve this, it's recommended to use the scientific method, which involves developing hypotheses, designing experiments to test them, and analyzing the results to make informed decisions.

Types of prototyping

There are different types of prototyping for live game development, such as vertical, horizontal, and rapid prototyping. These methods allow developers to refine specific features, test overall concepts, and experiment with different design elements.

- **Vertical prototyping** – allows for a more detailed version of a specific aspect of the game, like the combat system, to test its mechanics and ensure it's up to par. It gives a taste of what a small piece of the game will look like in its final form. Vertical prototyping tends to be as much about the visuals/experience as it is about the playable mechanics, and can also be done in a non-interactive way as well such as video.
- **Horizontal prototyping** – creates a basic version of the entire game to test its overall concept and user experience. It helps in quickly exploring and validating different features and interactions, gathering feedback early on to refine the design before full-scale development.
- **Rapid prototyping** – comes into play when developers want to try out different design elements and explore new gameplay mechanics quickly. It involves utilizing tools like game engines, asset libraries, and scripting languages to rapidly build and evaluate different gameplay elements before committing to full production.

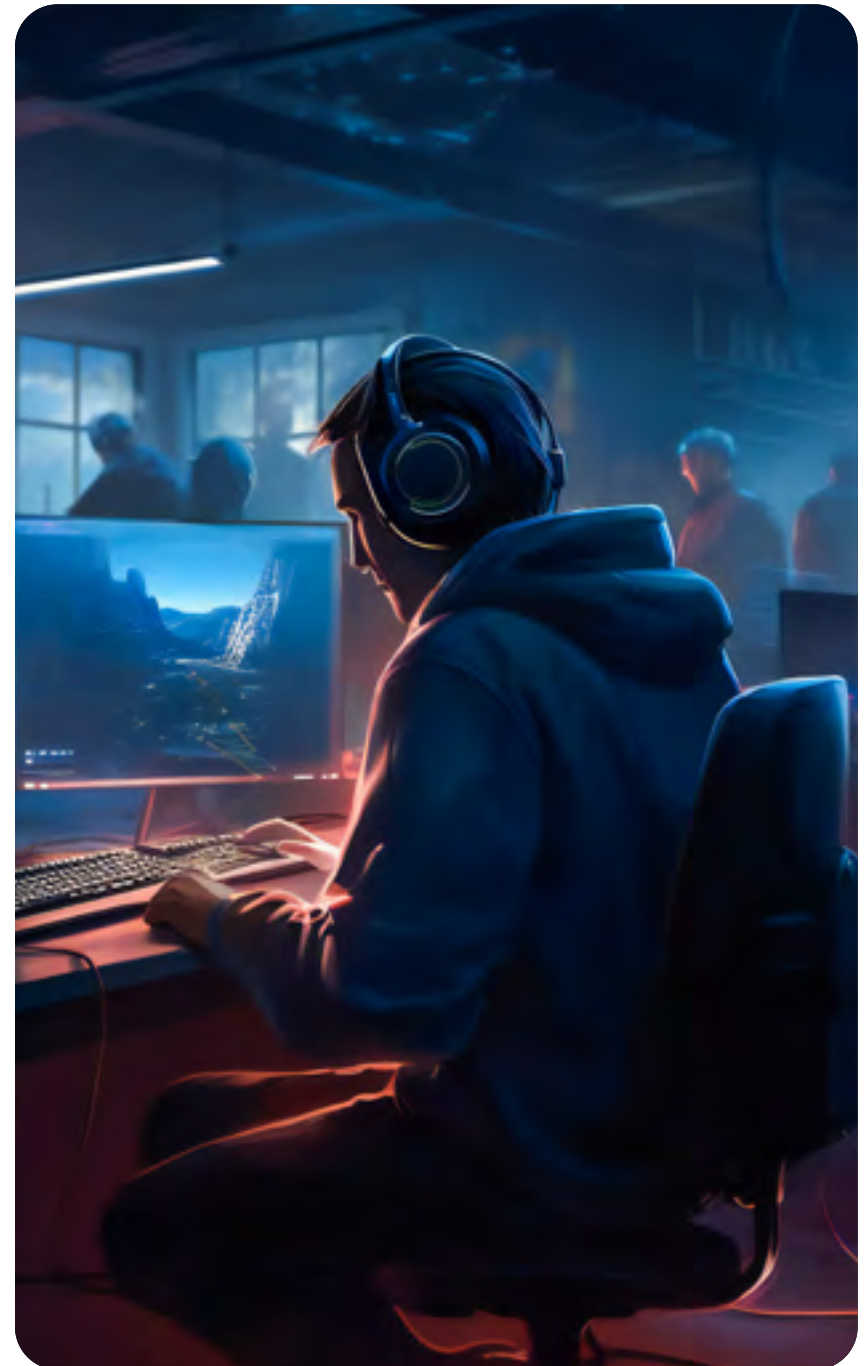
By using prototyping, game developers can save time and resources by identifying and addressing issues early in the development process, resulting in a better overall game.

How to create a prototype

1. **Define your concept:** Define your game concept, its target audience, and its unique selling point. This will give you a clear direction on what to include in your prototype.
2. **Identify your core experience:** Identify the fundamental gameplay experience that makes your game fun and engaging. Focus on the experience that will set your game apart from others in the same genre.
3. **Build a prototype:** Select the prototyping method that best suits your game's needs. Vertical prototyping is ideal for testing specific aspects of the game, while horizontal prototyping is best for testing the overall concept. Create an essential design document that includes details such as the game's story, mechanics, characters, and levels. It should be detailed enough to guide the prototyping process but not too detailed that it becomes a hindrance. Using your chosen prototyping method, create a playable version of your game that showcases the core mechanics and the overall user experience. Ensure the prototype is simple, easy to use, and has clear instructions.

4. **Test your prototype:** Test your prototype with a small group of people from your target audience. Get feedback on what works and what doesn't, and use this information to refine your prototype.
5. **Refine and iterate:** Use the feedback from testing to refine and iterate on your prototype. Keep testing and refining until you have a prototype that is engaging and fun to play.
6. **Decide when the prototype is complete:** Once you are satisfied that the prototype has answered your hypotheses then you're ready to move onto the next stage of the development process.

By following these steps, you can create a prototype that showcases your game's core mechanics, identifies any issues, and helps you refine and iterate until you have a game ready for release. Remember to test, refine, and repeat until you have a prototype players will love!



Best Practices for Prototyping

Developers must follow certain best practices to ensure that prototyping is effective and valuable. Following the best practices below can allow developers to maximize the value of prototyping and create better games in less time.

Reduce Your Scope

For several reasons, keeping the scope small when prototyping for live game development is crucial. By restricting your scope to its barebones idea, you can ensure that the prototype is achievable within a reasonable time frame and with the available resources. This helps avoid delays and burnout among the development team.

Additionally, limiting your scope allows you to focus on the game's core experience and ensure that it is engaging and enjoyable. If the prototype is too complex, it may be challenging to identify the key gameplay elements and the areas that need improvement. Focusing on a smaller scope allows you to refine the mechanics and iterate until you have sufficiently answered your original questions.



Define Clear Criteria for Success

A prototype should answer a specific question and have clear criteria for when it is considered successful (or equally unsuccessful). Having well-defined success criteria provides a tangible benchmark against which the prototypes and playtest results can be evaluated; otherwise, prototypes can end up meandering into pre-production and never ending.

It helps in assessing whether the prototypes meet the desired objectives and if the playtesting provides the desired outcomes. Clear success criteria also aid in making informed decisions on the next steps in development, such as refining or iterating the prototypes based on the playtest feedback. Ultimately, clear success criteria ensure that the prototyping and playtesting processes align with the overall goals and vision of the game, leading to a more focused and effective development cycle.

Test Early and Often

Speed and collaboration are vital during the prototyping stage. At this time, it is crucial to communicate with your team and test your game frequently. This allows you to quickly identify problems and improve gameplay mechanics as soon as possible. The earlier you test, the earlier you can catch and address any problems. Frequent tests save time and resources, as fixing issues early in the development process is much easier and cheaper.

Additionally, testing early and often helps you gather valuable feedback from players. This feedback can be used to refine the game mechanics and ensure that the game is fun and engaging. It can also help you identify any issues with the user interface, controls, or other aspects of the game that may be affecting the player's experience.

Keep the Team Small

Although feedback can be helpful during the early stages of development, too many cooks spoil the soup. Keeping the team small is important as it facilitates faster and more effective communication—which is crucial in the fast-paced and dynamic environment of game development. With fewer people involved, it is easier to coordinate tasks, share information, and make decisions quickly, thus saving time and avoiding miscommunication.

A small core team can help maintain focus and clarity on the project's goals and objectives. Involving too many people may elicit contradicting opinions that can be confusing if not everyone is perfectly aligned with one another. Keeping the team small can lead to a more cohesive and successful project. With fewer people involved, each team member has more opportunities to contribute their ideas and perspectives, which can lead to a more diverse and innovative approach to game development.



Continuous Playtesting

According to the [GDC 2021 State of the Game Industry Report](#), 56% of game developers said playtesting was the most effective way to improve the user experience.

Continuous playtesting is an iterative process of regularly testing a game throughout its development cycle to identify and address any issues or potential improvements. It involves collecting feedback from players and using it to refine the game mechanics and improve the overall user experience.

It helps developers identify and address any issues or bugs in the game as soon as possible. By continuously testing the game, developers can catch and fix issues before they become more complex and costly to address later in development. Playtesting also allows developers to gather valuable feedback from players, which can be used to refine the game mechanics and ensure that the game is fun and engaging. This feedback can help developers understand what players like and dislike about the game, what is working and what isn't, and how they can improve the user experience.



Best Practices for Continuous Playtesting

Developers must follow certain best practices to ensure that playtesting is practical and valuable. Consider the following best practices below.



Set Clear Objectives

To ensure that playtesting is effective, it is important to set clear objectives that align with the game's goals. Objectives could include improving player retention, balancing gameplay mechanics, or identifying and fixing bugs. In addition to setting objectives, gathering a diverse group of testers, including experienced players and newcomers, is crucial. This helps ensure that feedback represents the player base as a whole.

To make the most of playtesting, it is important to establish a clear feedback loop and act on the insights gained from testing. By continuously testing and iterating on the game, developers can ensure that they deliver a high-quality product that meets the needs and expectations of their

Focus on Your Target Audience

Know your audience and cater to them. Understanding your target market helps developers gain valuable feedback from those who are most likely to play and enjoy their game. By identifying and targeting the right audience, developers can ensure that their game meets the needs and expectations of their intended player base. This means that developers need to consider age, gender, gaming experience, and interests when selecting the right playtesters. Gathering feedback from a representative sample of the target audience can help developers identify areas of the game that need improvement and features that are working well.

Playtesting with the wrong audience can lead to misleading results and suggestions, as the feedback gathered may not accurately represent the preferences and expectations of the intended player base. For example, suppose a game is targeted at casual players, but playtesting is conducted only with hardcore gamers. In that case, the feedback received may be skewed towards more challenging and complex gameplay mechanics that casual players may not enjoy. This can result in developers making changes to the game that do not improve its appeal to the target audience, potentially leading to a product that fails to meet market expectations.

Associating playtesters to specific cohorts and attributing feedback accordingly is crucial because it allows developers to gather further insights and understand the varying perspectives within their target audience. By categorizing playtesters based on personas such as achievers, builders, and explorers, as well as demographics and gaming preferences, developers can identify patterns and trends in the feedback received. This approach enables them to delve deeper into specific cohorts and address their unique concerns, whether it's balancing difficulty levels for casual or hardcore gamers or tailoring gameplay mechanics to suit different playstyles. By drilling down into cohort-specific feedback, developers can make informed decisions that result in a more satisfying and engaging gaming experience for their intended player base.

Prioritize Feedback

Prioritizing feedback is critical when conducting continuous playtesting for live games, as it provides developers with the information needed to improve the game and create a better experience for players. To gather feedback effectively, developers can use a variety of techniques, such as:

- **Surveys** - One of the most popular ways to collect feedback from players, surveys provide both qualitative and quantitative data making them an excellent tool for gauging interest levels.
- **Direct observation** - Watching players through a one-way mirror or recorded sessions can gather qualitative feedback by understanding how players interact with the game in real time.
- **Focus groups** - Facilitating group discussions and identifying common themes or issues through focus groups can help gather further information.
- **Collect objective data** - Collecting data as players are playing the game can also provide valuable insights into player behavior and preferences. This includes information such as where a player dies or gets stuck.

Ultimately, prioritizing feedback and using a variety of techniques to gather it can help developers create a game that better meets the needs and expectations of their target audience.

Common Pitfalls When Conducting Playtests

Avoid making these crucial and common errors to ensure playtesting is as efficient and effective as possible.



Playtesting for the Sake of Playtesting

Developers should always have a clear objective or plan for how the feedback gathered will be used to improve the game. Without a clear purpose, playtesting can become a wasted effort that doesn't yield useful information for developers.

To avoid this, developers must have a hypothesis in mind before conducting playtesting and a specific question that they want to answer through playtesting. This ensures that the feedback gathered is focused and can provide useful insights into the game's strengths and weaknesses. By having a clear purpose for playtesting, developers can ensure that they gather the right information to make improvements to the game and create a better experience for players.

However, ad hoc playtesting, a form of informal and spontaneous playtesting conducted during game development, can be valuable for obtaining early user feedback. It involves gathering individuals or a small group of testers to play the game in an impromptu or unscheduled manner, without a pre-planned structure or formalized testing protocol. While it may lack the structure of formalized playtesting, ad hoc playtesting offers a more flexible and responsive approach to gathering real-time insights, which can aid in the iterative refinement of the game during development.



Not Accounting for Bias


Remain open to new ideas and perspectives throughout the playtesting process, as this allows developers to gather feedback that may challenge their assumptions and preconceptions. Being married to an idea or approach can prevent developers from seeing the flaws or shortcomings in their game and can lead to missed opportunities for improvement.

Additionally, developers must avoid their own biases and be open to feedback that may not align with their personal preferences or opinions. This requires a willingness to listen carefully to what players are saying and to be very open to new ideas and suggestions. By remaining open-minded throughout the playtesting process, developers can create a game that better meets the needs and expectations of their target audience and ultimately achieve greater success in the marketplace.

Involving Too Many People

While gathering feedback from a diverse range of players is important, inviting conflicting feedback makes it difficult to sort through and act on. Having too many opinions can make it challenging for developers to decide how to improve the game. Additionally, involving too many people can take up extra time and resources.

While involving too many people in the early stages of playtesting can be overwhelming and counterproductive, the team should involve more people as the process progresses. As the game approaches completion, involving a wider range of stakeholders and players can be beneficial. Developers should avoid involving too many people too early in the process and only involve stakeholders once the game is complete and good enough to be released.



The Importance of Prototyping and Playtesting

During the concept phase of the live game development process, it is crucial to work alongside a core team, define the project's scope, and begin prototyping to better understand your team's vision. Conducting playtests can also provide you with information from your audience that can further help you during the prototyping process. You must be deliberate with the playtests to receive information that answers any questions or concerns you may have. Resolving these issues at this stage is important to prevent you from dedicating additional time and energy later in the production process.

If you are venturing into the realm of live game development, be sure to follow along to explore the intricacies of the game development process and the next phases.



Curious about how Accelbyte can assist you with your backend requirements? Reach out to our team for a demo and explore the possibilities.

[REQUEST A DEMO](#)