

# KLAMUK!

G A M E S

## TEAM-HANDBUCH

**ALEXANDER STOLOVNIK**

**ERIC-IV STAICI**

**JOHANN TEßMER**

**LINDA ADELMANN**

**LUCA MERTEN**

24.05.2026 - HAW Hamburg MA Games





***Dieses Handbuch soll unserem Team als Grundlage dienen, um sich mit dem Konzept des Iterativen Designs auseinanderzusetzen und dabei zugleich die gemeinsame Arbeitsweise im Team zu reflektieren und weiterzuentwickeln. Im Vordergrund steht die Erarbeitung klarer Regeln und Verhaltensweisen für die Zusammenarbeit, mit dem Ziel, effizient und qualitativ hochwertige Ergebnisse zu erzielen.***



# INHALTSVERZEICHNIS

5	Iteratives Design	31	Eigene Erfahrungen
7	Team		Mini-Prototypen-System
		33	Planungs- und Arbeitsweisen
11	Bestehende Konzepte	34	Meinungsverschiedenheiten
12	Acht-Filter Modell		
13	Schleifenregel		
	Risikoanalyse	35	Unser Konzept
16	MDA-Modell		
		36	Prototyping
17	Bestehende Methoden	37	Play-Testing
		39	Risikominimierung
	Scrum	41	Arbeitskultur
20	Papier- und Tabletop Prototypen	44	Erfolgskriterien
	Greyboxing / Whiteboxing	45	Noch Fragen?
21	UI-Prototypen	47	Zusammenfassung
22	Digital Core-Loop Prototyp		
24	Fail Faster		
	Wasserfallmodell	49	Quellen
25	Minimum Viable Product (MVP)		
	Kanban		
28	Vertical Slicing		
	Playtesting-Methoden		
29	Regel: Wenig Tester, viel Iteration		

# TEAM



## ALEXANDER STOLOVNIK

Komponist

Sound Designer

// Ich möchte

"Das wichtigste ist Spaß und das Sammeln von Erfahrungen. Ich will auf dem Weg viel lernen und Freunde machen, aber auch meine Grenzen kennenlernen."

ALEX



## JOHANN TEßMER

Programmierer

QA

Lighting Artist

Gameplay Designer

// Ich möchte

"Ich möchte vor allem ein Projekt, welches mich für coole Herausforderungen stellt. Ich arbeite gerne an interessanten Systemen, welche nicht gleich eine eindeutige Lösung haben. Wichtig ist mir vor allem Systeme zu entwickeln, auf die ich selbst stolz sein kann."

JOHANN



## ERIC-IV STAICI

Technical Art

Environment Art

Art Support

VFX

// Ich möchte

"... erstes voll durchpoliertes Projekt abliefern; Skills ausbauen und neue dazulernen; Professionelle Workflows mit Teammitgliedern aus anderen Bereichen iterieren und durchsetzen; Ein Ergebnis, das mich audiovisuell und spielerisch glücklich macht"

ERIC

# TEAM



## LINDA ADELMANN

Concept Artist

UI Artist

3D Artist

### // Ich möchte

"... Fähigkeiten in 3D Bereich und Environment Design ausbauen; ein cooles Projekt umsetzen, dass ich alleine nicht schaffen würde"

LINDA



## LUCA MERTEN

Writer

Narrative-Designer

Game Designer

Management & Marketing

### // Ich möchte

"Ich erhoffe mir aus dem Projekt ein solides Spiel mit interessanter Story und Mechaniken, welche die Spieler beim Spielen positiv beeinflussen. Ich würde gerne den Prozess der Veröffentlichung/Gründung gehen. Ich möchte aus den verschiedenen Bereichen lernen und verstehen, was es bedeutet, ein großes Projekt in einem Team zu erstellen."

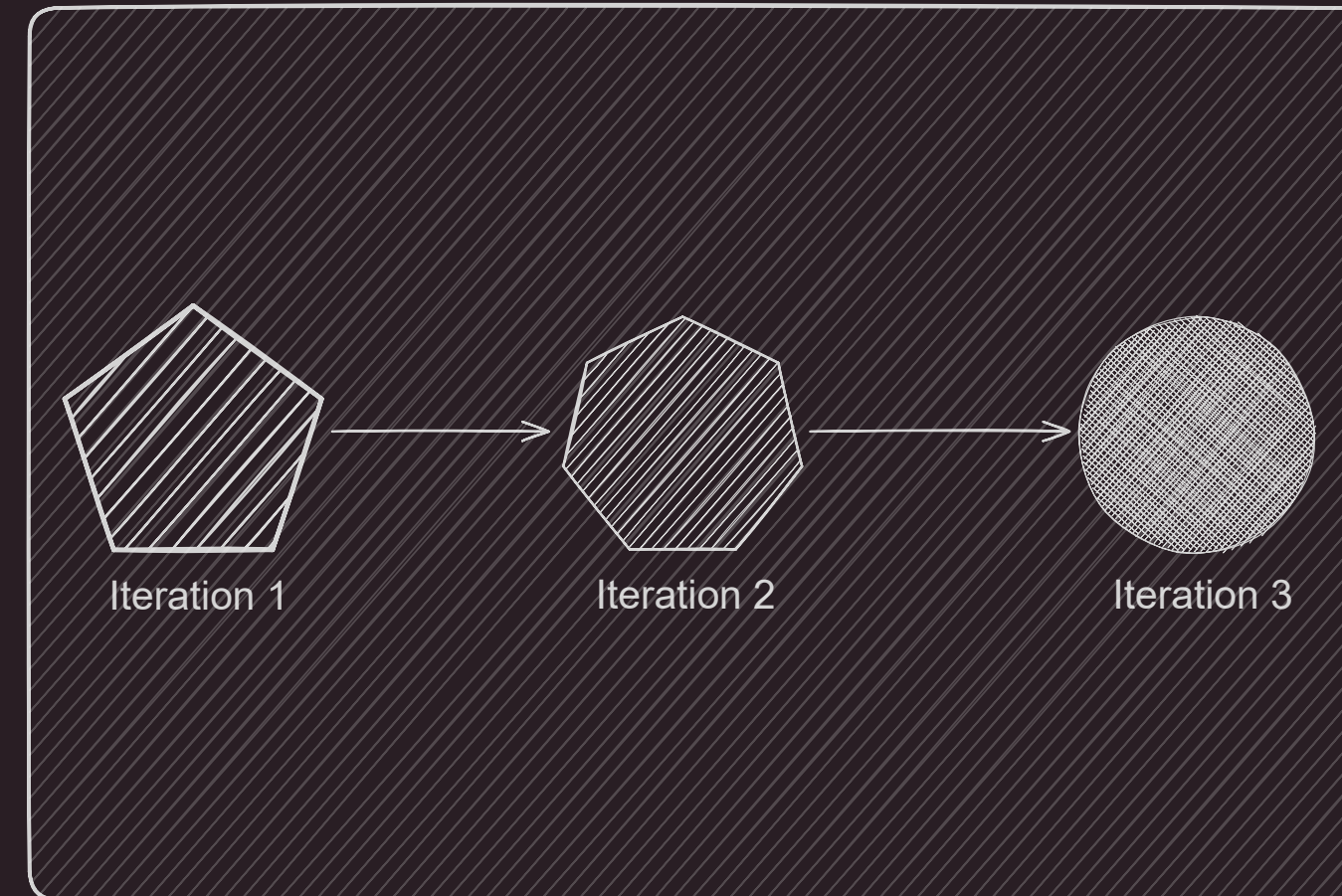
LUCA



# ITERATIVES DESIGN

Dieses Handbuch befasst sich zunächst mit der Frage, was Iteratives Design als Konzept bedeutet. Darauf aufbauend werfen wir gemeinsam einen Blick auf bestehende Konzepte und Methoden des iterativen Designs, die speziell für die Videospielbranche relevant sind und dort bereits erfolgreich Anwendung finden.

Ebenso umfasst das Handbuch mögliche Probleme und Risiken, die während der Zusammenarbeit oder des Iterationsprozesses auftreten könnten, und wie mit diesen umzugehen ist. Das dabei entstehende Konzept stützt sich sowohl auf etablierte Methoden als auch auf unsere eigenen praktischen Erfahrungen und verbindet beides zu einem Rahmen, der zu uns passt.

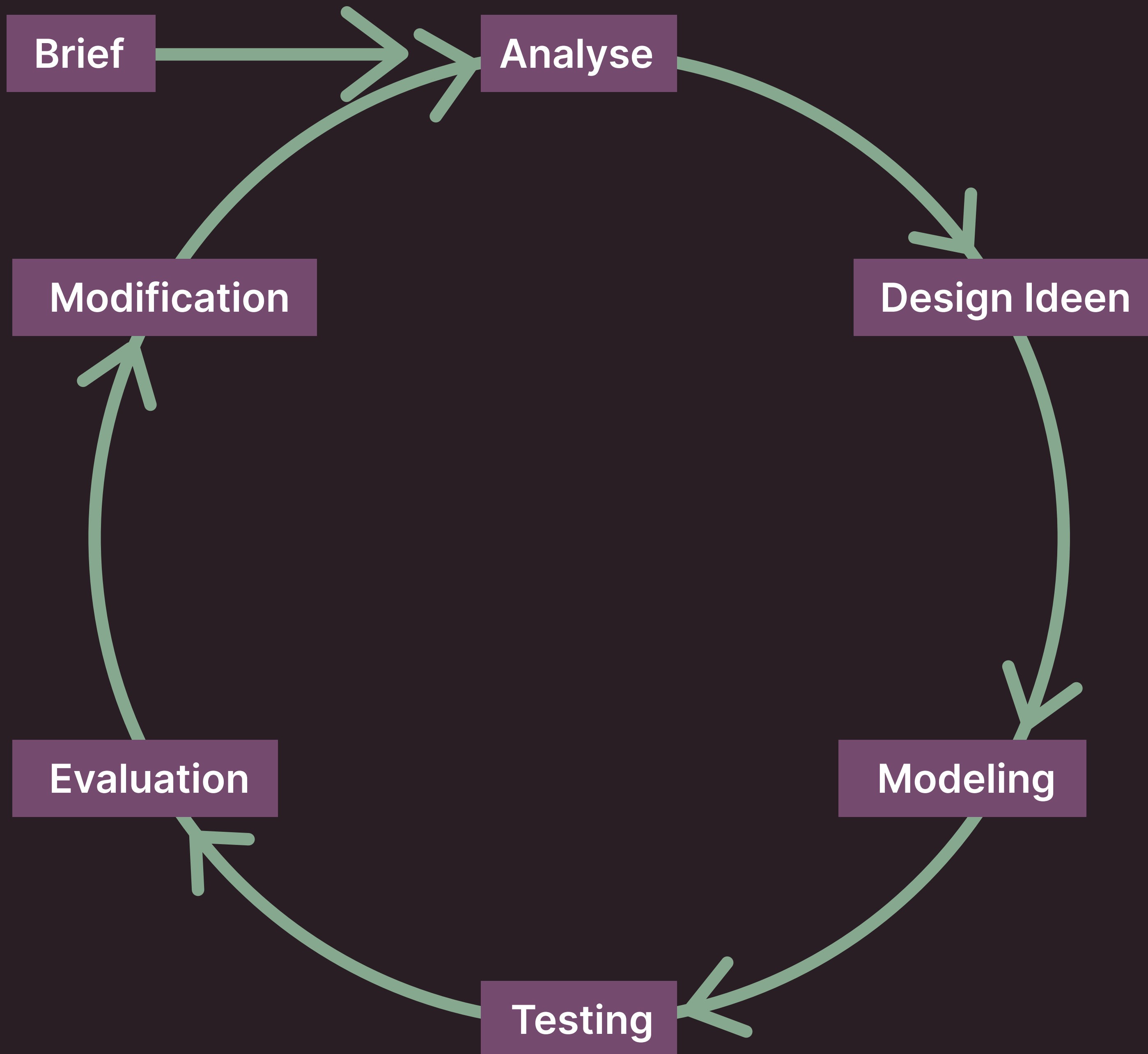


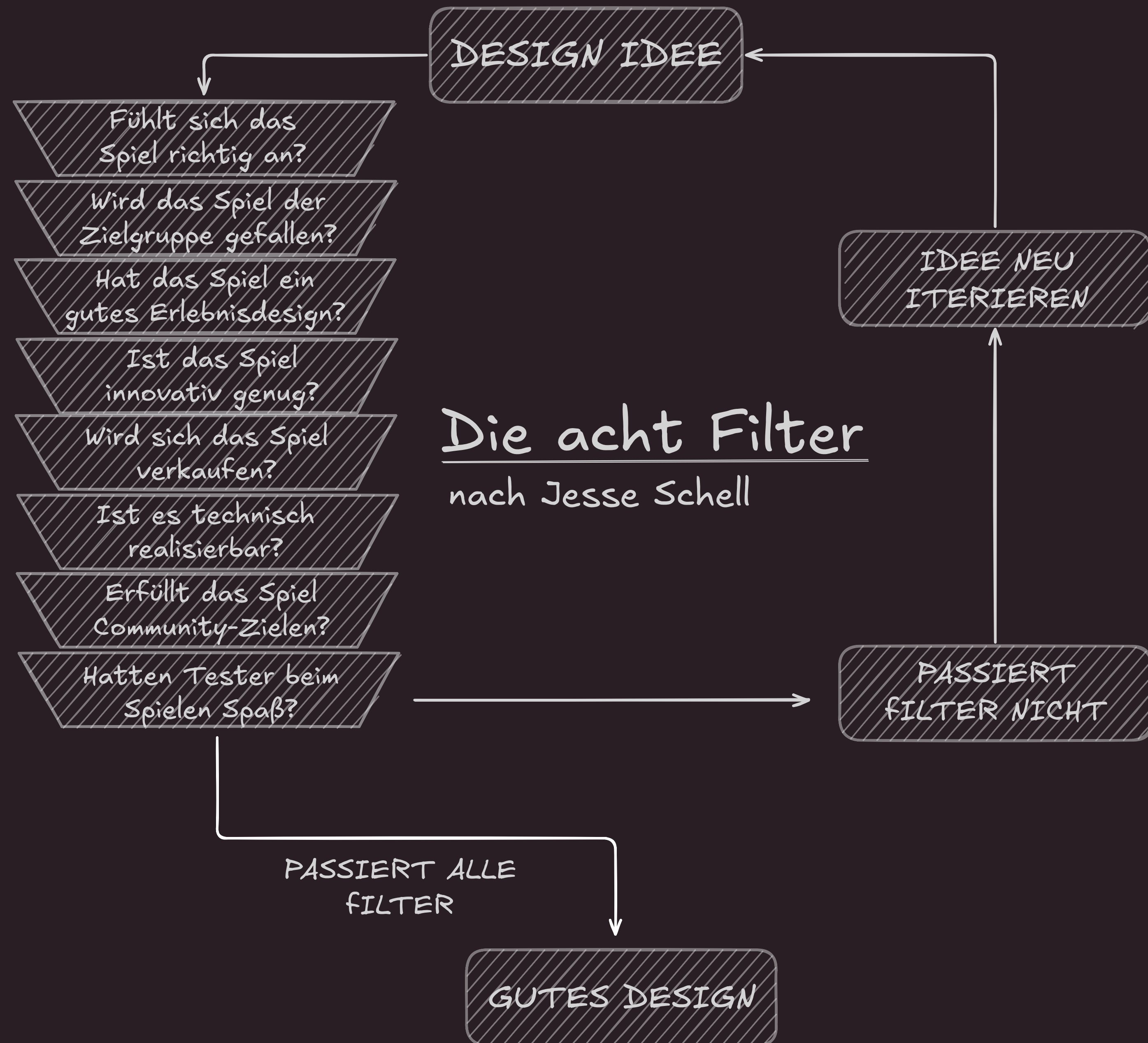
## Iteratives Design

Beim iterativen Design handelt es sich um einen Prozess, bei dem das zu erstellende Design in regelmäßigen Abständen geprüft und reflektiert wird. Das Ziel besteht darin, das Design zu verbessern und zu verfeinern, um das bestmögliche Ergebnis zu erzielen.

Beim Iterativen Prozess wird auf der vorherigen Iteration aufgebaut und verbessert

Konkreter Ablauf eines iterativen Prozesses





# BESTEHENDE KONZEPTE

Eine der größten Herausforderungen im Game Design ist nicht die Ideenfindung selbst. Diese liegt in der Entscheidung für eine konkrete Idee. Viele Ideen machen Entscheidungen schwer. Noch schwerer ist es, eine falsche Entscheidung später zu korrigieren. Eine zentrale Frage ist daher, anhand welcher Kriterien sich gute von schlechten Ideen unterscheiden lassen. Hierfür existieren verschiedene etablierte Konzepte.

## Acht-Filter-Modell

Das erste Konzept ist das Modell der acht Filter von Jesse Schell.

Ein gutes Design muss diese acht Filter durchlaufen und bestehen. Wenn ein Design auch nur an einem der Filter scheitert, muss es angepasst werden. Nach der Überarbeitung muss das Design erneut alle Filter durchlaufen. Änderungen könnten neue Probleme an anderer Stelle verursachen.

In dem Konzept von Jesse Schell geht vieles über die sogenannten Lupen. Bei den Lupen handelt es sich um verschiedene Perspektiven. Demnach bietet jede Perspektive eine Möglichkeit das Design sehr genau „unter die Lupe“ zu nehmen.



Visuelle Repräsentation des Acht-Filter-Modells



## BESTEHENDE KONZEPTE

### Schleifenregel

Ein weiteres Konzept von Jesse Schell ist die Schleifenregel. Sie ist weniger ein Modell als eine grundlegende Regel, die vielen iterativen Entwicklungsmethoden zugrunde liegt. Die Regel besagt, dass Designs und Ideen in der Spielentwicklung so lange getestet und überarbeitet werden müssen, bis sie ausgereift sind.

Schell formuliert dies treffend: „Je öfter Sie Ihr Design testen und verbessern, desto besser wird Ihr Spiel werden.“ Da jeder Schleifendurchlauf in der Praxis mit zusätzlichem Aufwand und Kosten verbunden ist, stellen sich zwei wesentliche Frage. Wie wird jeder Schleifendurchlauf lohnend? Und wie lassen sich die Schleifendurchläufe schnellstmöglich bewerkstelligen?

Eine Darstellung die zeigt, wie anhand der Schleifenregel iteriert wird

Ablauf einer Risikoanalyse. Darstellung von weclapp.com <sup>[1]</sup>



### Risikoanalyse

Die Risikoanalyse greift eine ähnliche Fragestellung auf, betrachtet sie jedoch aus einer anderen Perspektive.

Was kann im Verlauf des Projekts schiefgehen? Dazu wird das Projekt in möglichst viele Einzelteile zerlegt und geprüft, ob diese in ihrer Grundfunktion sinnvoll und realisierbar sind. Das Ergebnis ist eine Risikoliste, die potenzielle Gefährdungen des Projekts frühzeitig benennt. Ziel ist es, Risiken entweder präventiv zu umgehen oder schnell auf sie reagieren zu können.



## Ablauf einer Risikoanalyse



Risiko-  
identifikation



Risiko-  
analyse

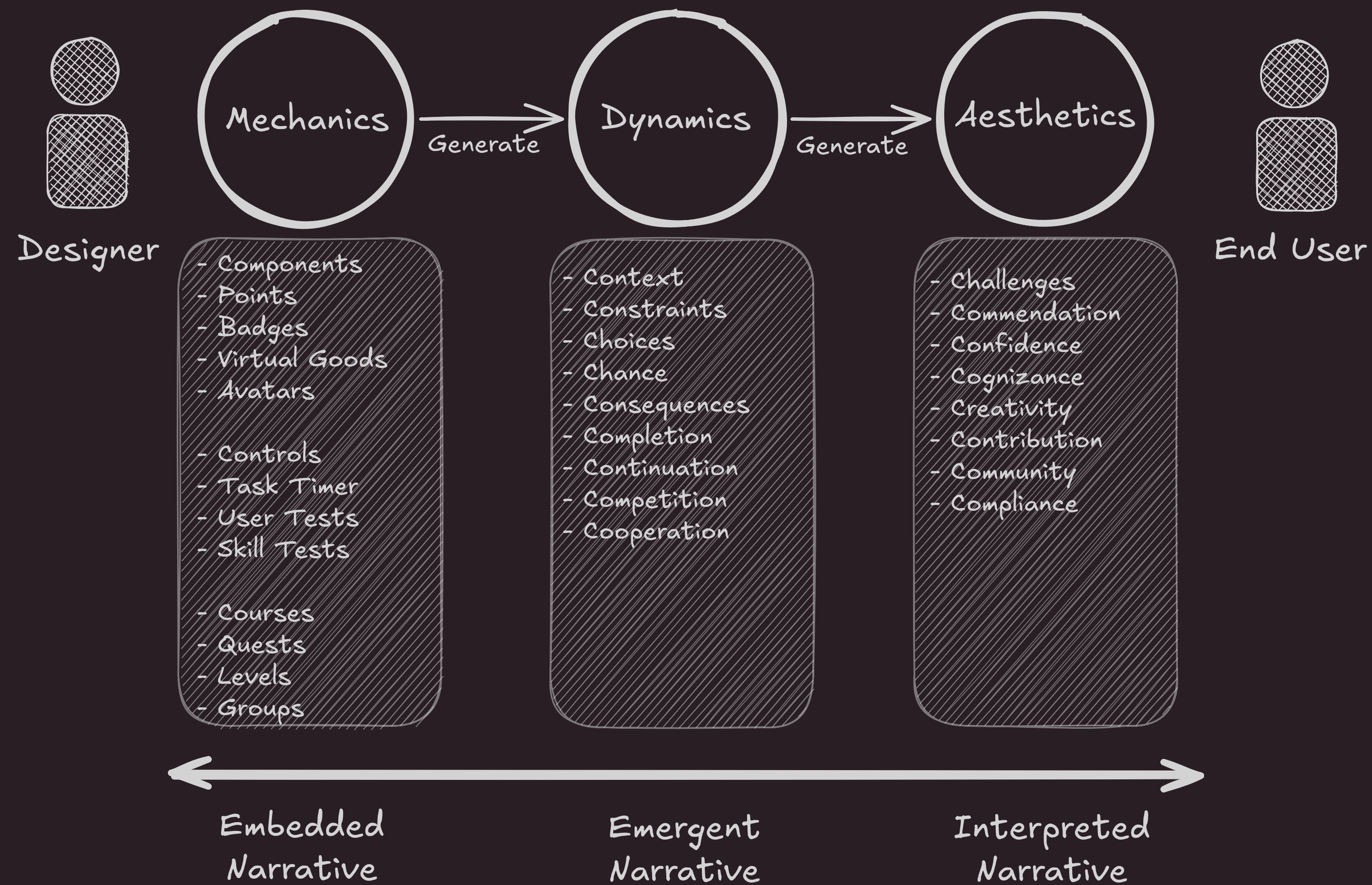


Risiko-  
bewertung



Risiko-  
behandlung

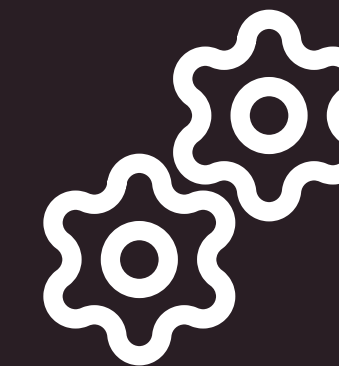




## BESTEHENDE KONZEPTE

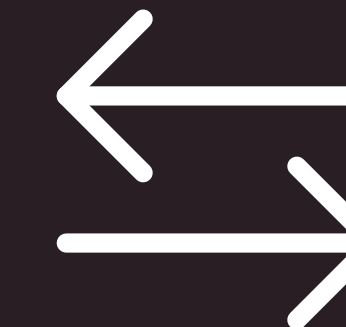
### MDA-Modell

Ebenfalls verbreitet in der Spielentwicklung ist das MDA-Modell von Robin Hunicke, Marc LeBlanc und Robert Zubek. Es bietet einen strukturierten Rahmen zur Analyse von Spielen. Dabei werden diese in drei Komponenten unterteilt: Mechanics, Dynamics und Aesthetics.



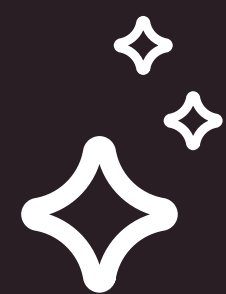
#### Mechanics

Mechanics bezeichnen die grundlegenden Regeln und Aktionen, die einem Spieler zur Verfügung stehen.



#### Dynamics

Dynamics beschreiben das Verhalten, das aus der Interaktion des Spielers mit diesen Mechaniken entsteht.



#### Aesthetics

Aesthetics umfassen schließlich die emotionalen Reaktionen, die beim Spieler während des Spielens hervorgerufen werden.

Darüber hinaus thematisiert das Modell die Wechselwirkungen zwischen diesen drei Komponenten, also inwiefern jede Ebene die jeweils anderen beeinflusst.

↖ Bestandteile des MDA-Modells. Inspiriert von einer Abbildung von [marcabraham.com](http://marcabraham.com) <sup>[2]</sup>



# BESTEHENDE METHODEN

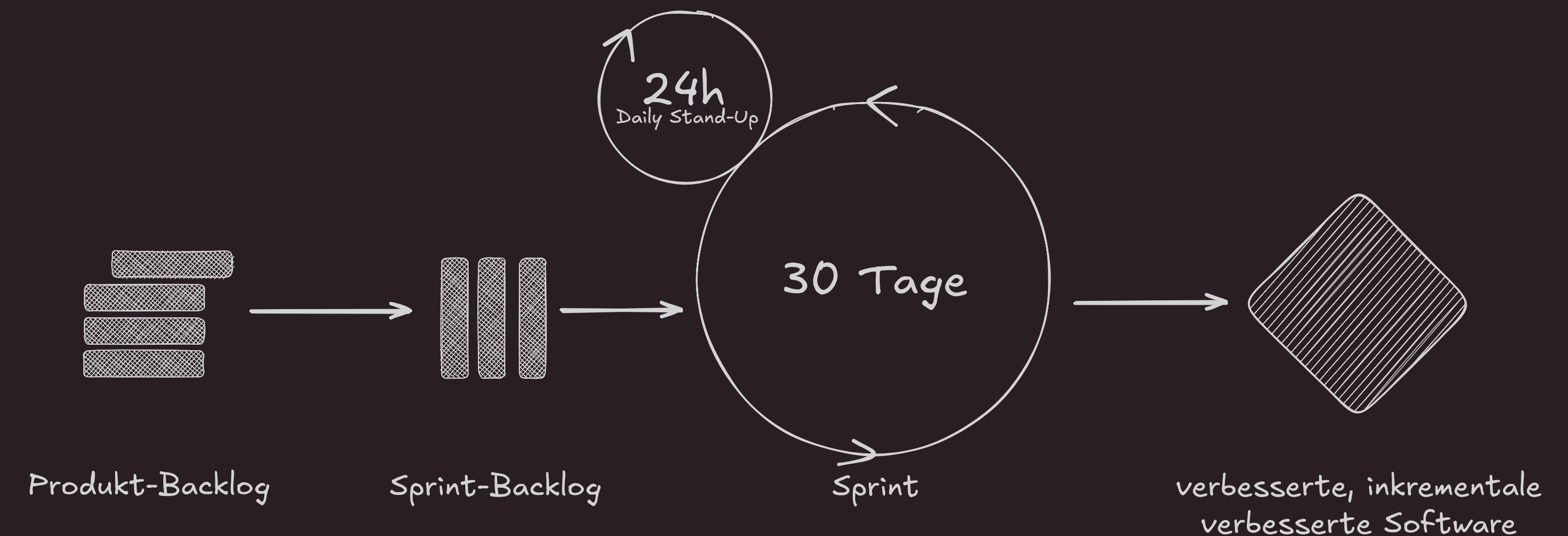
Neben den zuvor beschriebenen Konzepten gibt es mehrere Methoden, die in der praktischen Spieleentwicklung helfen, Ideen in überprüfbare Arbeitsschritte zu überführen. Sie sollen nicht als starre Regeln verstanden werden, sondern als Werkzeuge, die je nach Projektgröße, Teamstruktur und Entwicklungsphase angepasst werden können.

## Scrum

Scrum ist ein agiles Rahmenwerk, bei dem die Entwicklung in kurze, wiederholbare Arbeitsabschnitte unterteilt wird. Diese Abschnitte werden Sprints genannt und dienen dazu, regelmäßig überprüfbare Ergebnisse zu erzeugen. Innerhalb eines Sprints finden unter anderem Sprint Planning, Daily Scrum, Sprint Review und Sprint Retrospective statt.

Für unser Projekt ist Scrum nicht in seiner vollständigen Form sinnvoll, da das gesamte Framework für unsere Teamgröße zu viel organisatorischen Aufwand erzeugen würde. Dennoch sind einzelne Elemente hilfreich. Besonders der Sprint-Rhythmus und die Retrospektive können genutzt werden, um regelmäßig zu prüfen, was funktioniert hat, welche Probleme aufgetreten sind und was im nächsten Zyklus verbessert werden soll.

Scrum als wiederholbarer Entwicklungszyklus. Abbildung inspiriert von Wikimedia <sup>[3]</sup>



Scrum





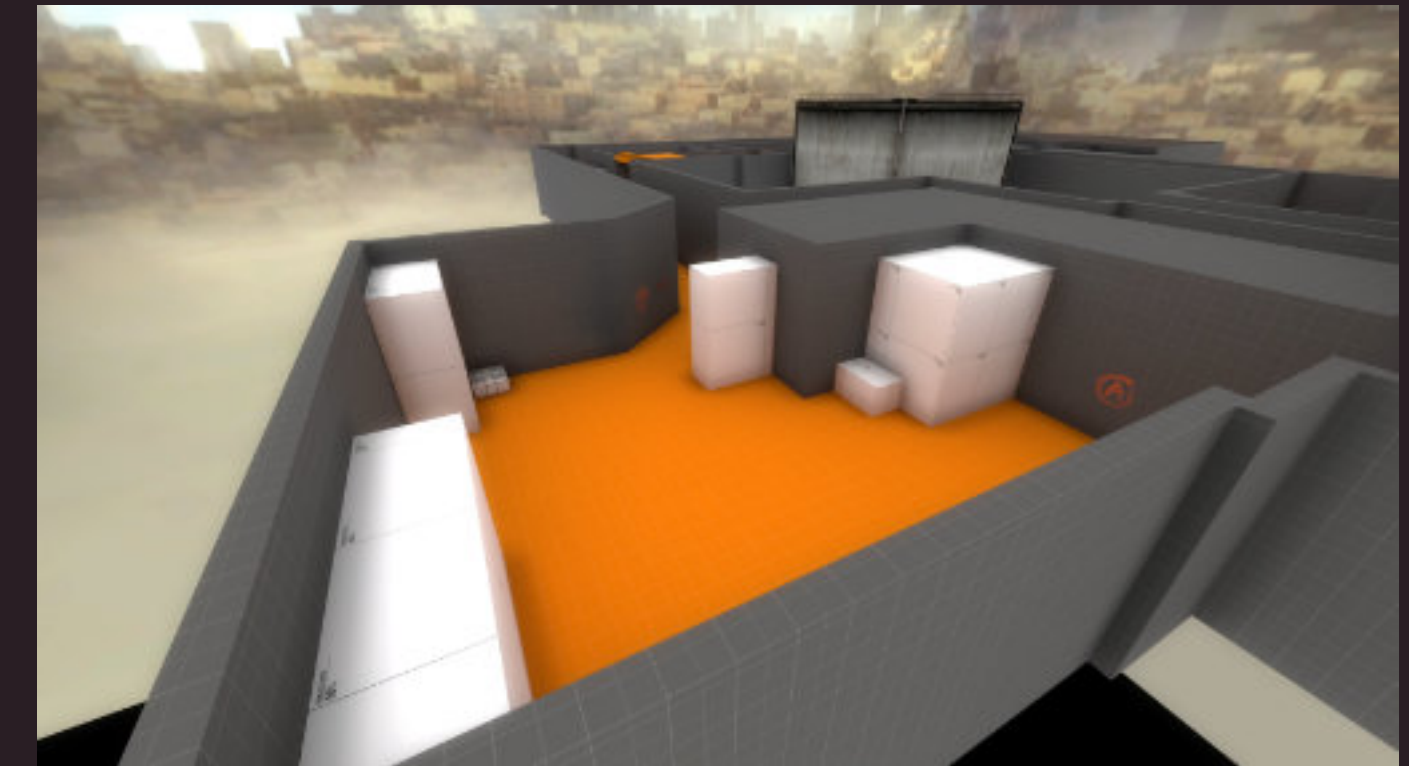
## BESTEHENDE METHODEN

### Papier- und Tabletop-Prototypen

Papier- und Tabletop-Prototypen sind eine sehr einfache Form des Prototypings. Mechaniken, Regeln, Karten, Progression oder Levelstrukturen können mit Papier, Stiften, Würfeln, Karten oder anderen einfachen Materialien getestet werden. Gerade am Anfang ist diese Methode sinnvoll, weil noch keine Programmierung oder finale Gestaltung notwendig ist.

Der Vorteil dieser Methode liegt darin, dass Anpassungen sehr schnell möglich sind. Wenn eine Mechanik nicht funktioniert, kann sie direkt angepasst oder verworfen werden, ohne dass bereits viel Entwicklungszeit investiert wurde. Dadurch eignet sich diese Methode besonders gut, um grundlegende Annahmen früh zu prüfen.

- Ein frühes Spiellevel im Greyboxing Zustand
- Papierprototypen ermöglichen frühe Tests ohne technischen Aufwand. Bild von Wikimedia <sup>[4]</sup>



### Greyboxing / Whiteboxing

Beim Greyboxing oder Whiteboxing werden Level zunächst nur mit einfachen Formen gebaut. Statt finaler Texturen, Lichtstimmung oder detaillierter Assets werden graue Blöcke und einfache Geometrien verwendet. Ziel ist es, früh zu prüfen, ob Wegeführung, Größe, Bewegung, Sichtachsen und Pacing funktionieren.

Diese Methode verhindert, dass zu früh Zeit in finale Gestaltung investiert wird. Erst wenn der Aufbau spielerisch sinnvoll ist, lohnt es sich, den Bereich mit finaler Art, Beleuchtung und Details auszuarbeiten.

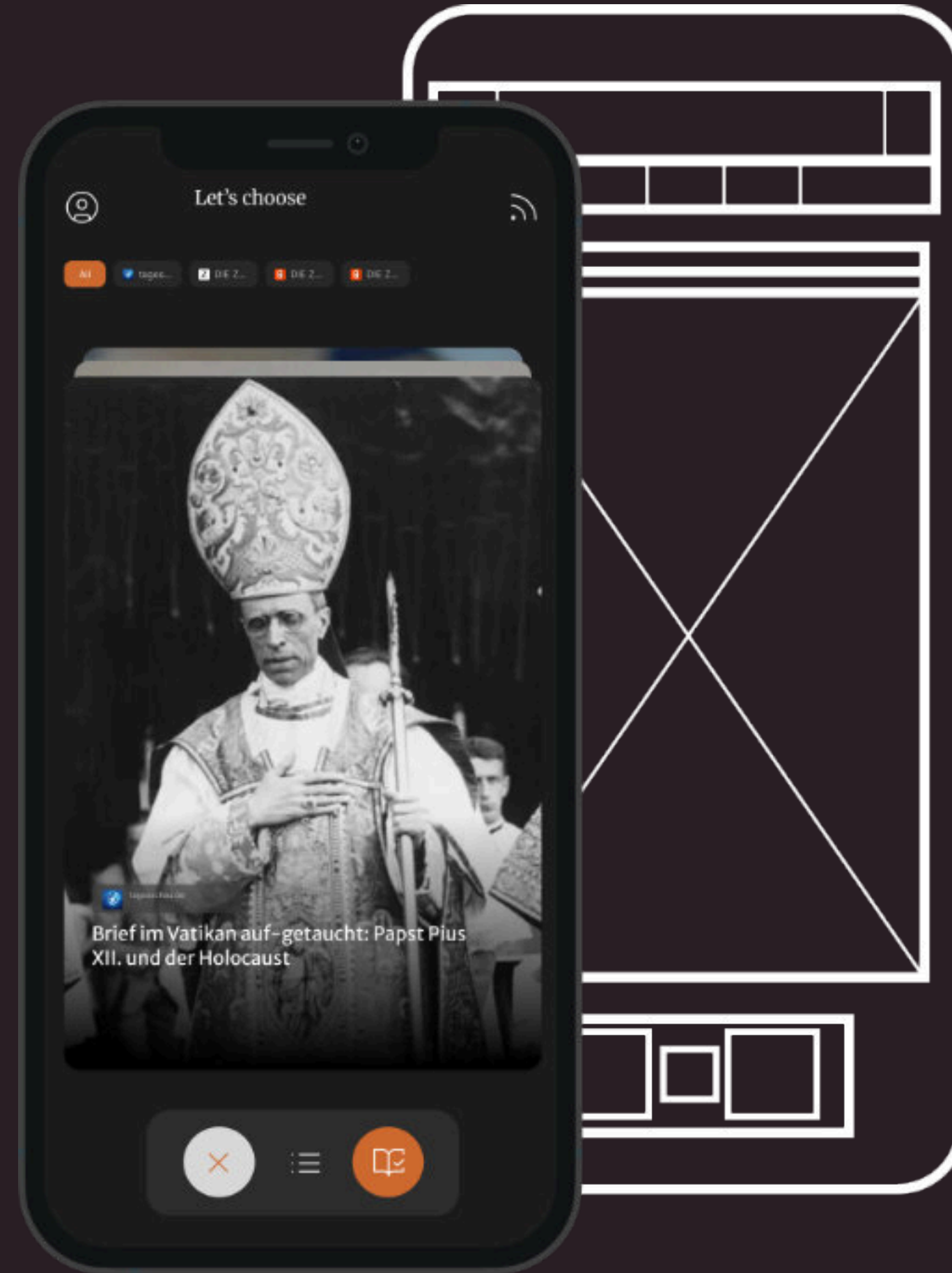


## BESTEHENDE METHODEN

### UI-Prototypen

Klickbare Mockups werden genutzt, um Menüs, HUDs und Interface-Abläufe zu testen, bevor diese in der Engine umgesetzt werden.

Diese Methode ist besonders sinnvoll, wenn noch unklar ist, ob ein Interface verständlich aufgebaut ist. Probleme in Menüführung, Beschriftung oder Spielerorientierung werden dadurch früh sichtbar. Da solche Anpassungen in einem Mockup deutlich schneller sind als in einer fertigen Engine-Implementierung, spart diese Methode Entwicklungszeit.



Ein UI-Mockup, zusammen mit seiner vervollständigten Version



### Digitaler Core-Loop-Prototyp

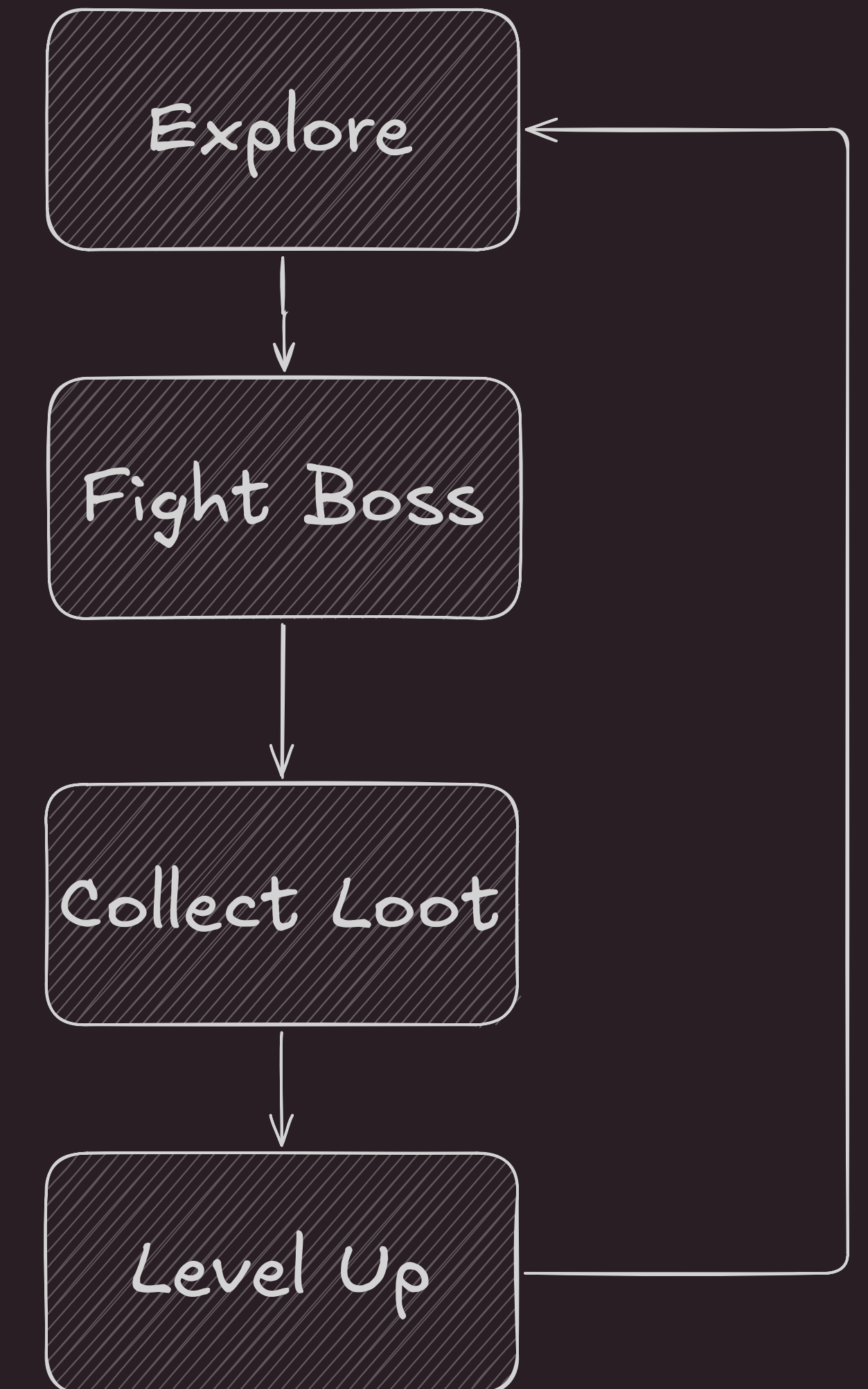
Ein digitaler Core-Loop-Prototyp konzentriert sich ausschließlich auf die zentrale Spielhandlung. Dabei wird nur der Kern gebaut, der sich im späteren Spiel immer wiederholen soll. Zusätzlicher Content, Story, Polishing oder umfangreiche Art werden zunächst weggelassen.

Für die Entwicklung ist diese Methode wichtig, weil sie die wichtigste Frage zuerst überprüft: Macht der grundlegende Spielablauf überhaupt Spaß? Wenn der Core Loop nicht funktioniert, helfen mehr Content oder bessere Grafik nur begrenzt. Deshalb sollte der Kern möglichst früh spielbar gemacht und getestet werden.

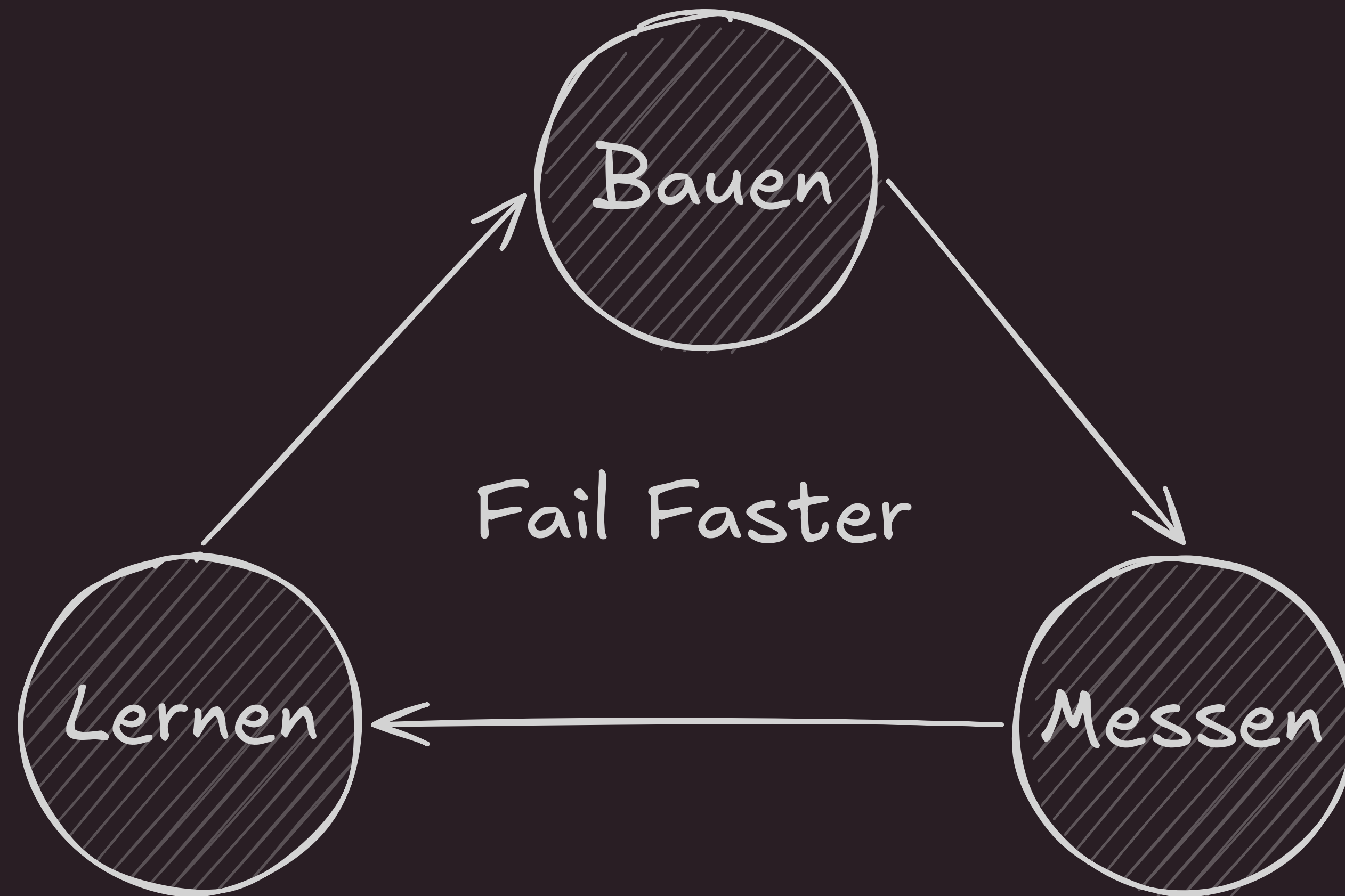


Der Core Loop zeigt die wiederkehrenden Kernaktionen des Spielers.

### Core-Game-Loop







Fehler früher sichtbar machen!!

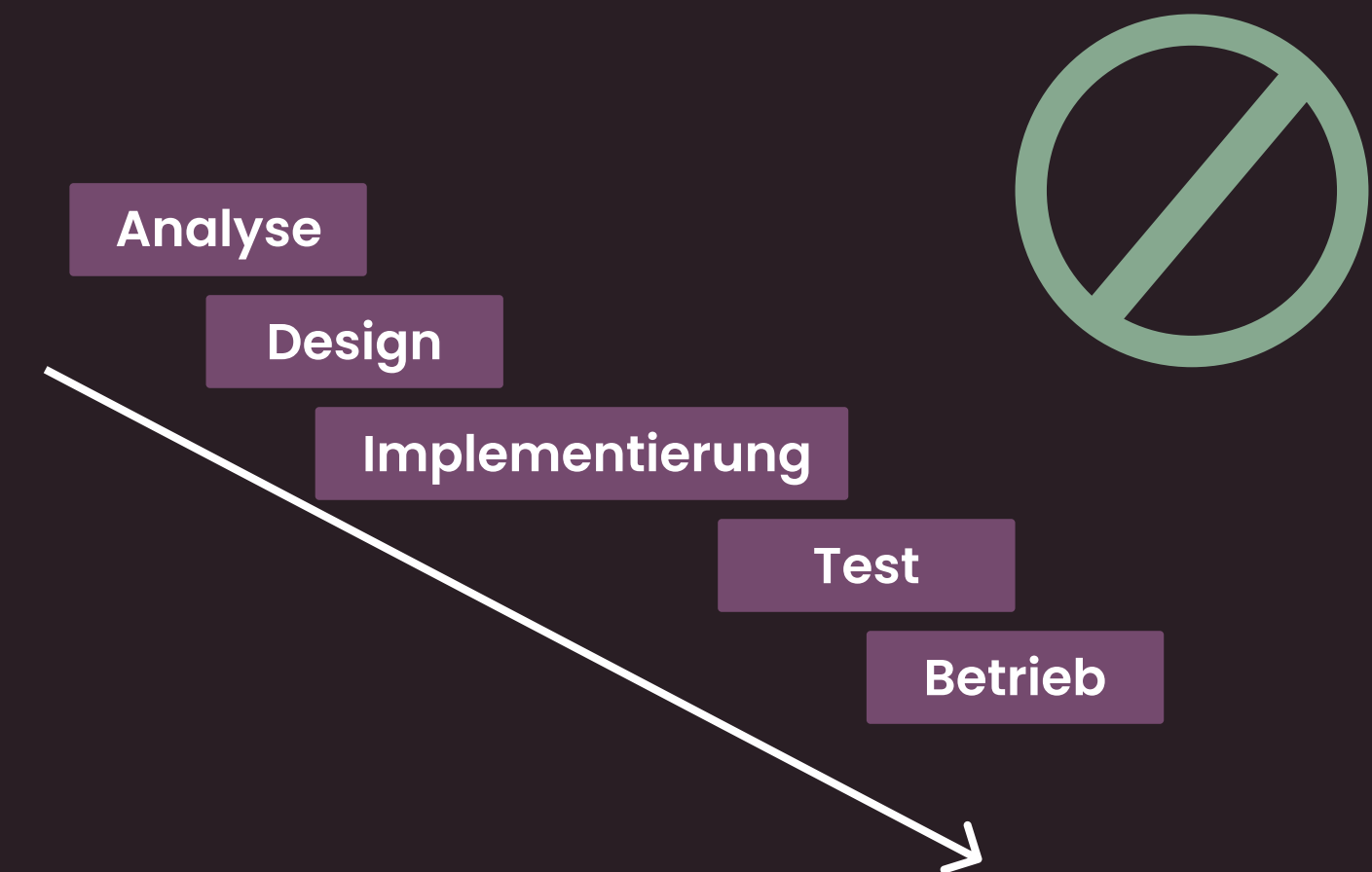
## BESTEHENDE METHODEN

### Fail Faster

Fail Faster bedeutet nicht, absichtlich Fehler zu machen. Gemeint ist, Fehler möglichst früh sichtbar zu machen, damit nicht über lange Zeit in eine falsche Richtung gearbeitet wird. Der Gedanke passt zum Build-Measure-Learn-Prinzip: Es wird etwas Kleines gebaut, getestet beziehungsweise gemessen, daraus gelernt und anschließend angepasst.

Für unser Projekt heißt das, dass wir möglichst kurze Testzyklen anstreben sollten. Wenn ein Problem nach wenigen Stunden oder Tagen sichtbar wird, ist der Verlust deutlich geringer als nach mehreren Wochen Arbeit. Fail Faster reduziert also nicht die Qualität, sondern hilft dabei, schlechte Annahmen schneller zu erkennen und bessere Entscheidungen zu treffen.

- Die Schritte des Wasserfallmodells. Ungeeignet für die Spielentwicklung.
- ➡ Kurze Feedbackzyklen machen Fehler früher sichtbar. Adaptiert nach Wikimedia <sup>[5]</sup>



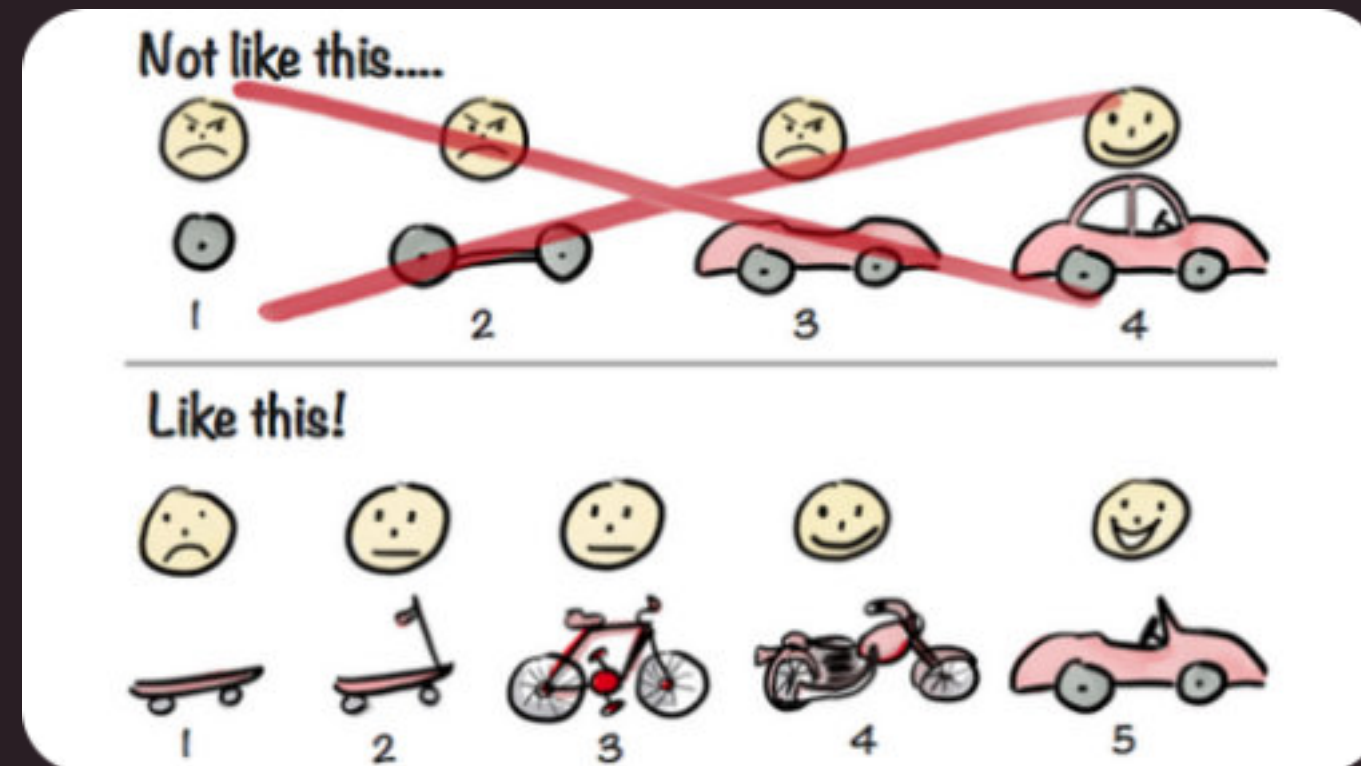
### Wasserfall

Das Wasserfallmodell beschreibt einen linearen Entwicklungsprozess, bei dem Phasen nacheinander abgeschlossen werden. Typische Schritte sind Anforderungen, Design, Implementierung, Test und Wartung.

Für Game Development ist dieses Modell problematisch, weil Spiele erst durch das Spielen wirklich bewertet werden können. Wenn Tests erst sehr spät stattfinden, werden Probleme mit Spielgefühl, Verständlichkeit oder Spaß oft zu spät erkannt. Anpassungen sind dann deutlich aufwendiger.



# BESTEHENDE METHODEN



## Minimum Viable Product (MVP)

Ein Minimum Viable Product ist nicht einfach eine kleinere Version des fertigen Spiels. Es ist die kleinste umsetzbare Version, mit der die wichtigste Frage des Projekts überprüft werden kann. Der bekannte Vergleich von Henrik Kniberg zeigt, dass ein MVP möglichst früh nutzbar sein muss, damit echtes Feedback entstehen kann.

Auf Games übertragen bedeutet das, dass ein MVP nur die Elemente enthalten sollte, die für die zentrale Testfrage notwendig sind. Wichtig ist nicht, dass das Spiel vollständig wirkt, sondern dass der Kern überprüfbar wird.

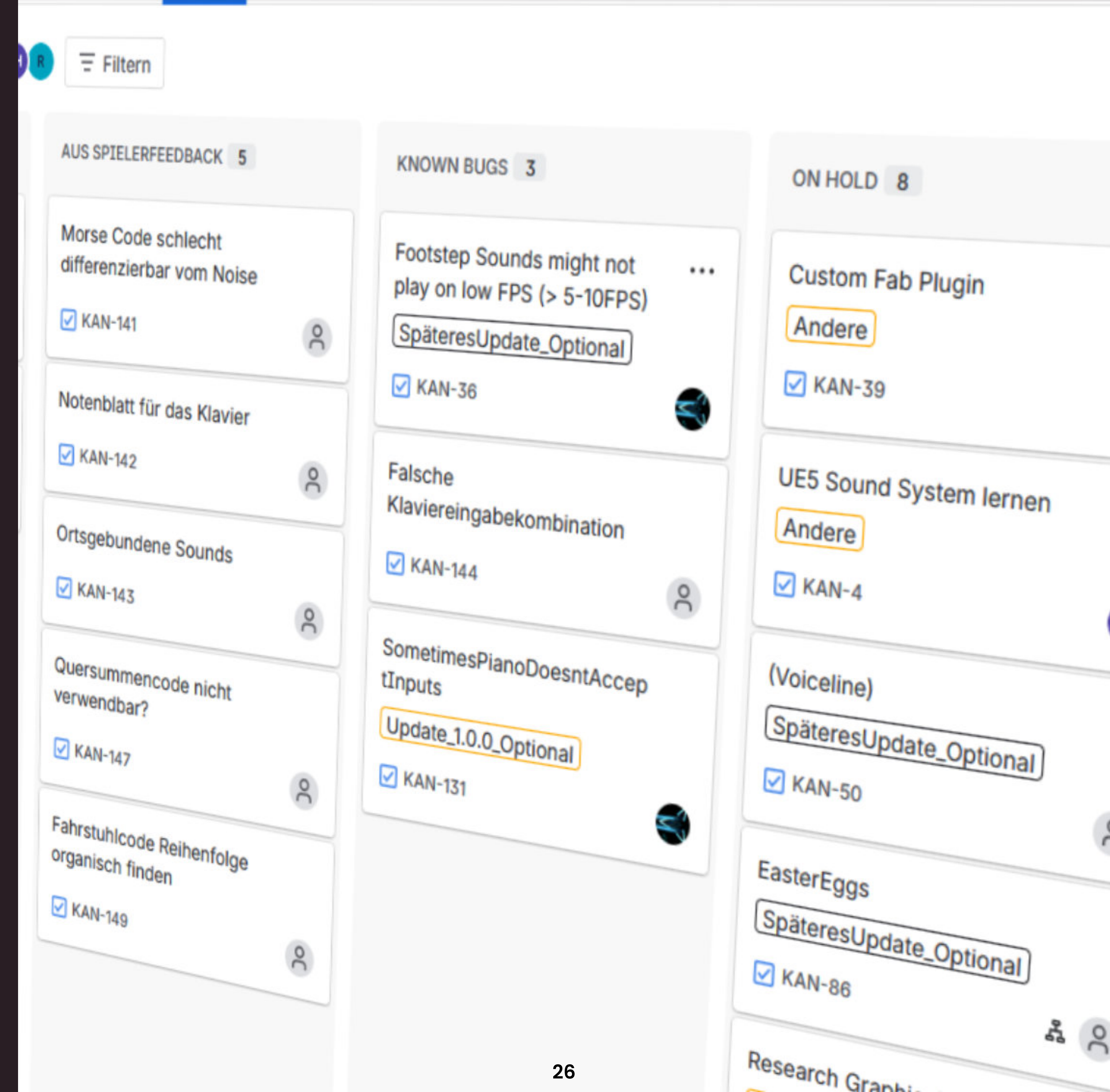
## Kanban

Kanban ist eine Methode, um Arbeit sichtbar zu machen und den Arbeitsfluss zu steuern. Typisch ist ein Board mit Spalten wie „To Do“, „Doing“ und „Done“. Aufgaben werden als Karten dargestellt und wandern von links nach rechts durch den Prozess.

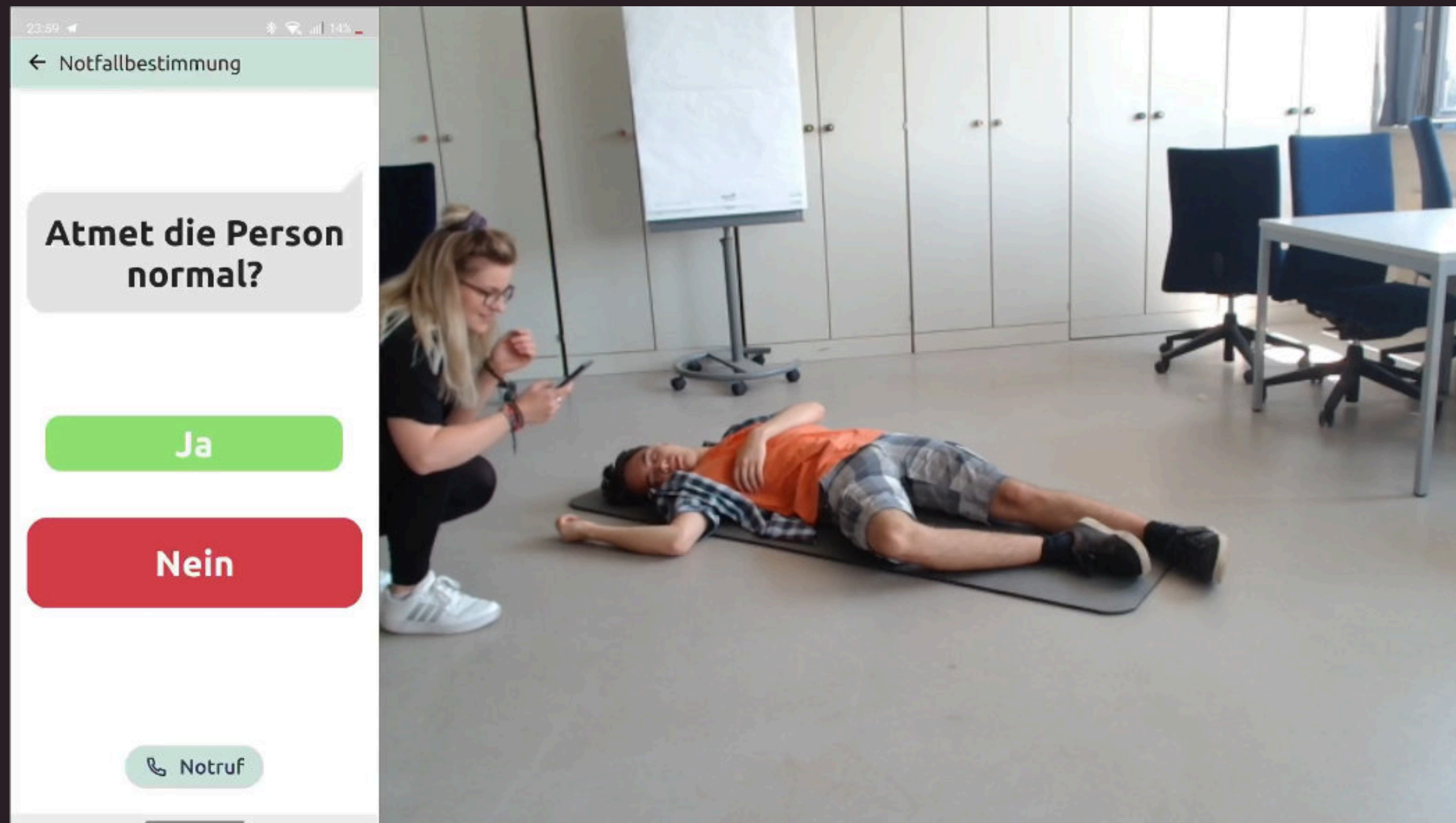
Besonders wichtig ist dabei die WIP-Begrenzung. Sie legt fest, wie viele Aufgaben gleichzeitig in einer Spalte liegen dürfen. Dadurch wird verhindert, dass zu viele Dinge parallel begonnen, aber zu wenige abgeschlossen werden. Für kleine Teams ist Kanban gut geeignet, weil es wenig organisatorischen Overhead erzeugt und trotzdem sichtbar macht, woran gerade gearbeitet wird.

Ein MVP soll früh nutzbares Feedback ermöglichen, nicht Einzelteile aufzeigen. Bild von Henrik Kniberg <sup>[6]</sup>

Ein Kanban Board in Jira für ein Spielprojekt. Macht Aufgabenstatus und Engpässe sichtbar.



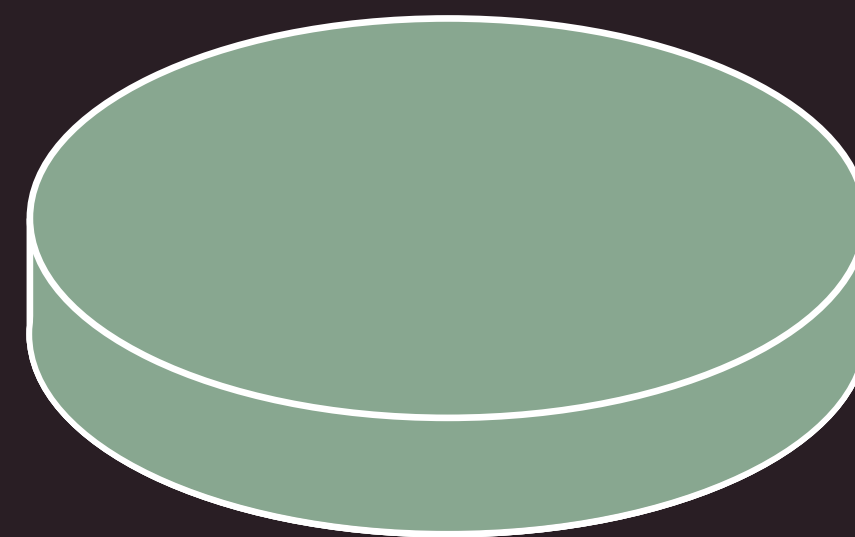




Full Project



Vertical Slice



Horizontal Slice

## BESTEHENDE METHODEN

### Vertical Slicing

Beim Vertical Slicing wird ein kleiner Ausschnitt des Spiels in möglichst finaler Qualität umgesetzt. Dieser Ausschnitt enthält nicht nur eine einzelne Disziplin, sondern alle wichtigen Ebenen: Gameplay, Art, Sound, UI, Technik und gegebenenfalls Story. Ein Vertical Slice ist deshalb kein einfacher Prototyp, sondern ein Produktionsnachweis.

Diese Methode ist besonders relevant für Pitches, Festivals oder Förderanträge. Ein Vertical Slice zeigt nicht nur eine Idee, sondern eine konkrete Vorstellung davon, wie sich das fertige Spiel anfühlen soll. Gleichzeitig werden technische, gestalterische und organisatorische Probleme früh sichtbar, weil alle Disziplinen einmal gemeinsam durch den Produktionsprozess laufen.

### Playtesting-Methoden

Playtesting umfasst verschiedene Methoden, um Spieler beim Spielen zu beobachten und daraus Rückschlüsse für das Design zu ziehen. Bei stiller Beobachtung spielt eine Person, während das Team Verhalten, Probleme und Reaktionen notiert. Beim Think-Aloud-Verfahren spricht der Tester während des Spielens aus, was er denkt.

Eine weitere Variante ist Play-Aloud. Diese Methode orientiert sich stärker an Streaming- oder Kommentarkultur. Spieler beschreiben dabei nicht nur ihre Gedanken, sondern kommentieren ihr Spielerlebnis natürlicher, ähnlich wie bei einem Let's Play. Dadurch können emotionale Reaktionen und Verständnisprobleme sichtbarer werden.



Ein Playtest/User Test für eine erste Hilfe App



Vergleich: Vertical Slice vs. Horizontal Slice



## BESTEHENDE METHODEN

### Faustregel: Wenig Tester, viele Iterationen

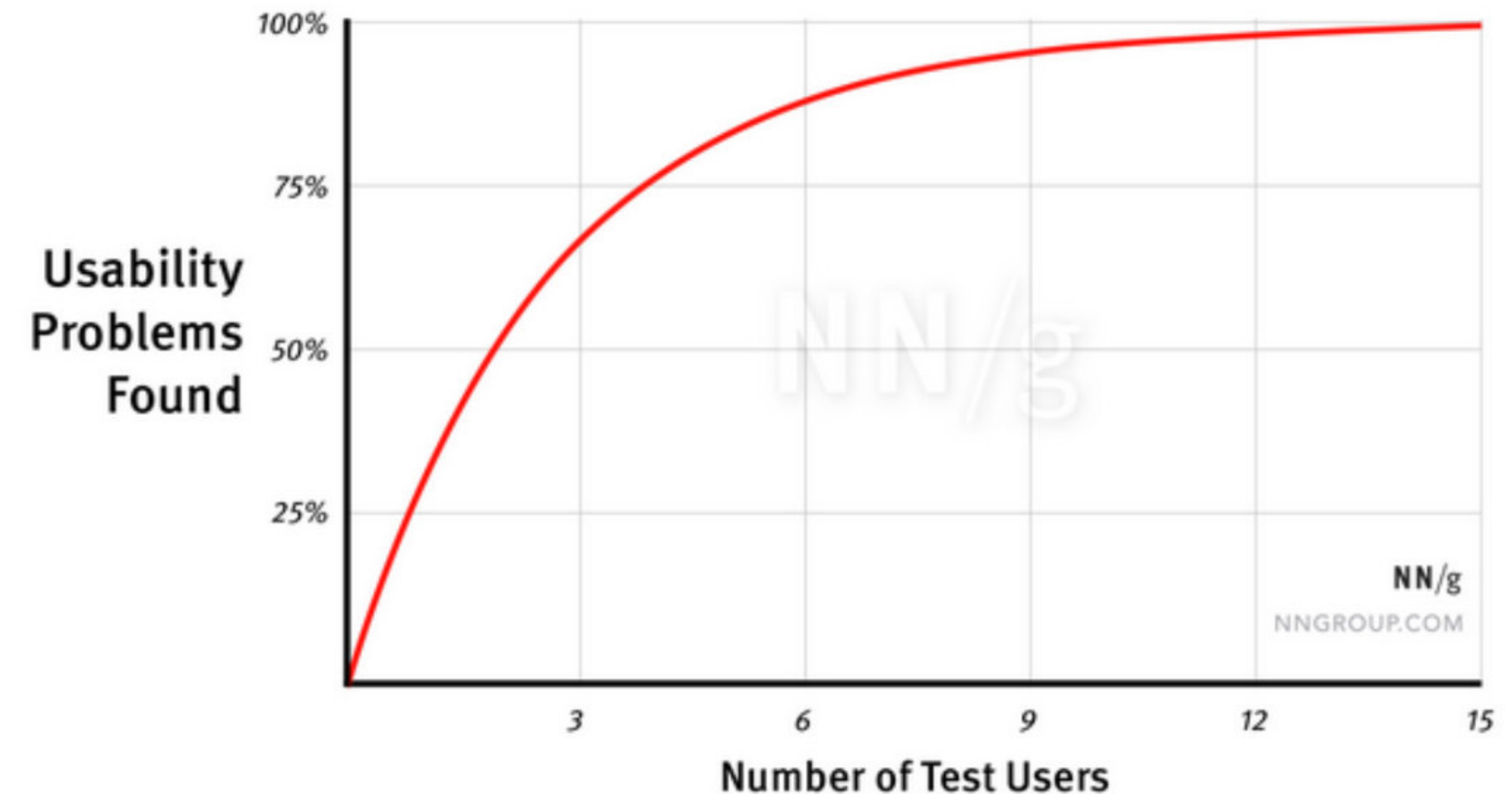
Eine wichtige Faustregel im Playtesting lautet, lieber mit wenigen Testern mehrfach zu testen als einmalig mit sehr vielen. Der Erkenntnisgewinn steigt zu Beginn stark an, flacht aber nach wenigen Testern deutlich ab. Dadurch ist es oft sinnvoller, mehrere kleine Testrunden durchzuführen, anstatt eine große Testphase erst sehr spät im Projekt einzuplanen.

Für unsere Entwicklung bedeutet das, dass drei bis fünf Tester pro Iteration meistens ausreichend sind, um die auffälligsten Probleme zu finden. Danach sollten die Ergebnisse ausgewertet, Anpassungen umgesetzt und erneut getestet werden. So entstehen mehrere Lernzyklen, statt eine große Testphase erst am Ende des Projekts durchzuführen.

Verhältnis von Testern zu gefundenen Problemen. Schematische Darstellung nach der Nielsen Norman Group <sup>[7]</sup>



### 5 Users: The Optimal Sample Size for Qualitative Usability Studies





# EIGENE ERFAHRUNGEN

## Mini-Prototypen-System

Wir entwickeln unsere Prototypen mit kleinstmöglichem Inhalt, zum Beispiel zum Testen einer einzelnen Mechanik, zum Aufbau einer Szene oder zum Testen, ob SFX zu einem Charakter passen.

Durch die Minimierung des Umfangs können diese Mini-Prototypen schneller erstellt und parallel zueinander entwickelt werden. Teammitglieder können so auch eigenständig Ideen ausprobieren, ohne auf andere Aspekte warten oder sich zunächst in einer Abstimmung durchsetzen zu müssen. Auch kann so der Zeitaufwand zur Implementierung eines Features gut anhand der Erstellungszeit des Mini-Prototypen abgeschätzt werden.

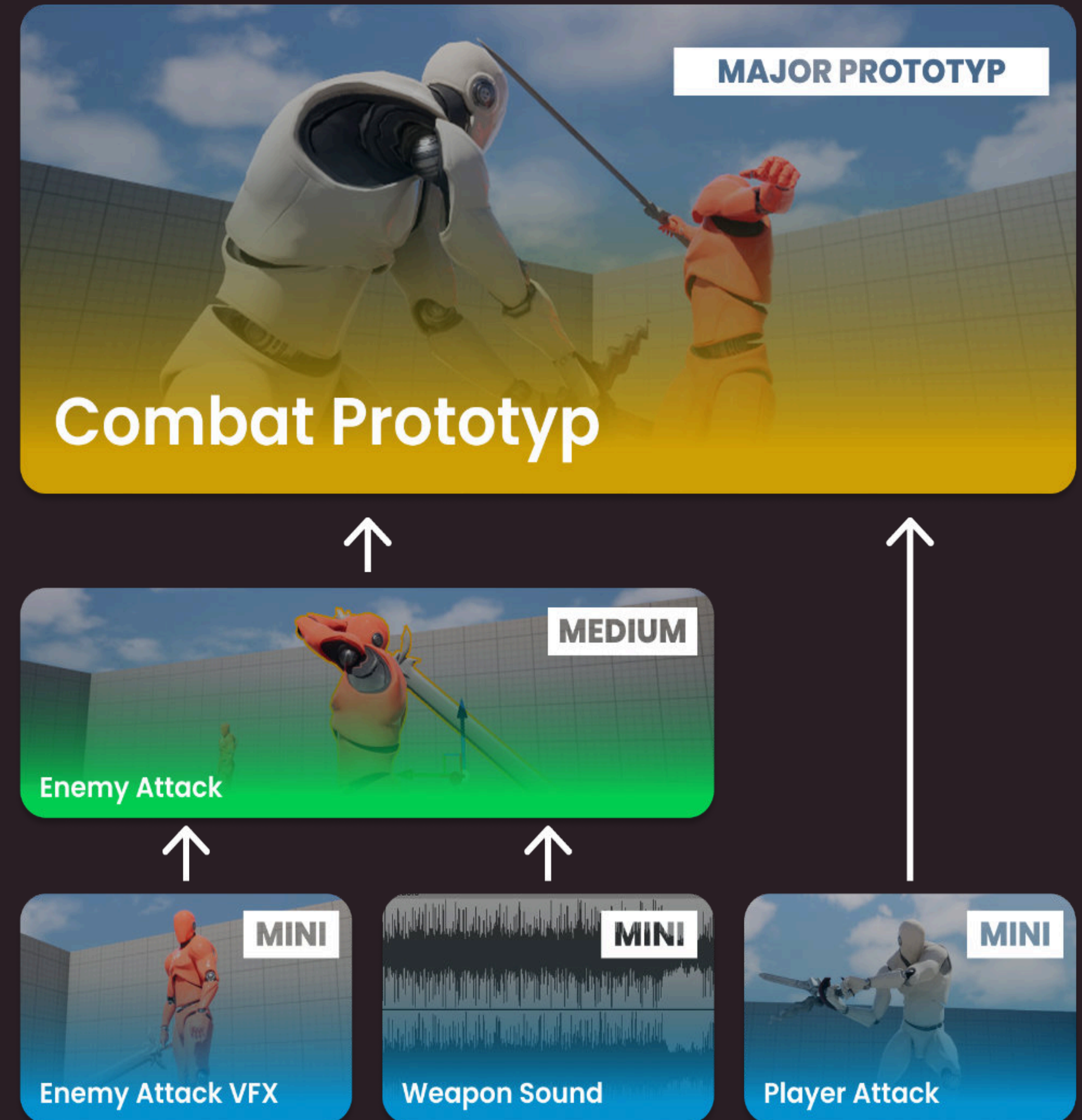
Zusammenstellung eines Major Prototypen aus mehreren kleineren Prototypen nach unserem Mini-Prototypen-System



Die erstellten Mini-Prototypen werden daraufhin entweder intern getestet oder direkt an Spieler weitergegeben. Somit kann schnellstmöglich Feedback gewonnen und auf den Ergebnissen iteriert werden.

Wenn ein solcher Mini-Prototyp erfolgreich beim Testen abschneidet, kann er mit anderen Mini-Prototypen kombiniert werden. Der daraus entstehende größere Prototyp muss natürlich erneut getestet werden, da nicht garantiert ist, dass zwei Features gut zusammen funktionieren.

Durch dieses Prinzip der Mini-Prototypen und der Kombinationen dieser können wir schnell viele Eindrücke gewinnen, Features und Ideen testen und gegebenenfalls Aspekte schneller verwerfen.





## Planung- und Arbeitsweisen

Was die Zusammenarbeit im Team betrifft, so haben wir uns auch hier bereits auf einige Regeln geeinigt, welche die Planungs- und Arbeitsweisen im Team festlegen.

### Planung

Planungsgespräche sollten immer im direkten Gespräch passieren, möglichst auch in Präsenz. Je wichtiger das zu besprechende Thema, desto wichtig ist auch ein Präsenzmeeting für uns. Planungsaspekte sollten möglichst nicht über Direktnachrichten oder Absprechung mit nur einem Anteil des Teams erfolgen. Dies dient auch dem Ziel, die Visionen des Teams einheitlich zu halten und Misskommunikationen zu verhindern.



Planung sind für uns in Präsenz effektiver. Abbildung von [www.embis.de](http://www.embis.de) <sup>[8]</sup>



Diskussionen sind nichts schlechtes. Es soll offen und respektvoll diskutiert werden können. Bild von Adobe Stock <sup>[9]</sup>



### Arbeitsweisen

Sobald im Team alle Aufgaben klar verteilt wurden und bis aufs weitere keine Planungsaspekte mehr anstehen, so sind alle Teammitglieder frei darin, ihre Arbeitsweise selbst zu gestalten.

Dies soll zu einer produktiven Arbeitsweise verhelfen und auch mal eine Auszeit ermöglichen. So wird Teammotivation hochgehalten, Reibungspunkte im Team vermieden und Meilensteine werden schneller erreicht.

## EIGENE ERFAHRUNGEN

### Meinungsverschiedenheiten

In bisherigen Planungsgesprächen und Diskussionen taten sich oft im Team verschiedene Vorgehensweisen, Definitionen oder Meinungen auf. Dies erschwerte zunächst unweigerlich die Zusammenarbeit im Team.

Wir gehen davon aus, dass diese Situation uns noch oft über den Zeitraum unserer Zusammenarbeit begleiten wird, also ist es wichtig, für solche Situationen vorbereitet zu sein und einige Regeln aufzustellen.

So soll oft durch regelmäßige Planungsgespräche sichergestellt werden, dass das Team noch dieselbe Vision vor Augen hat und denselben Definitionen folgt. Diskussionen sollen immer auf professioneller Ebene erfolgen und sich nie direkt gegen ein Teammitglied richten. Ein respektvoller Umgang ist dabei entscheidend.

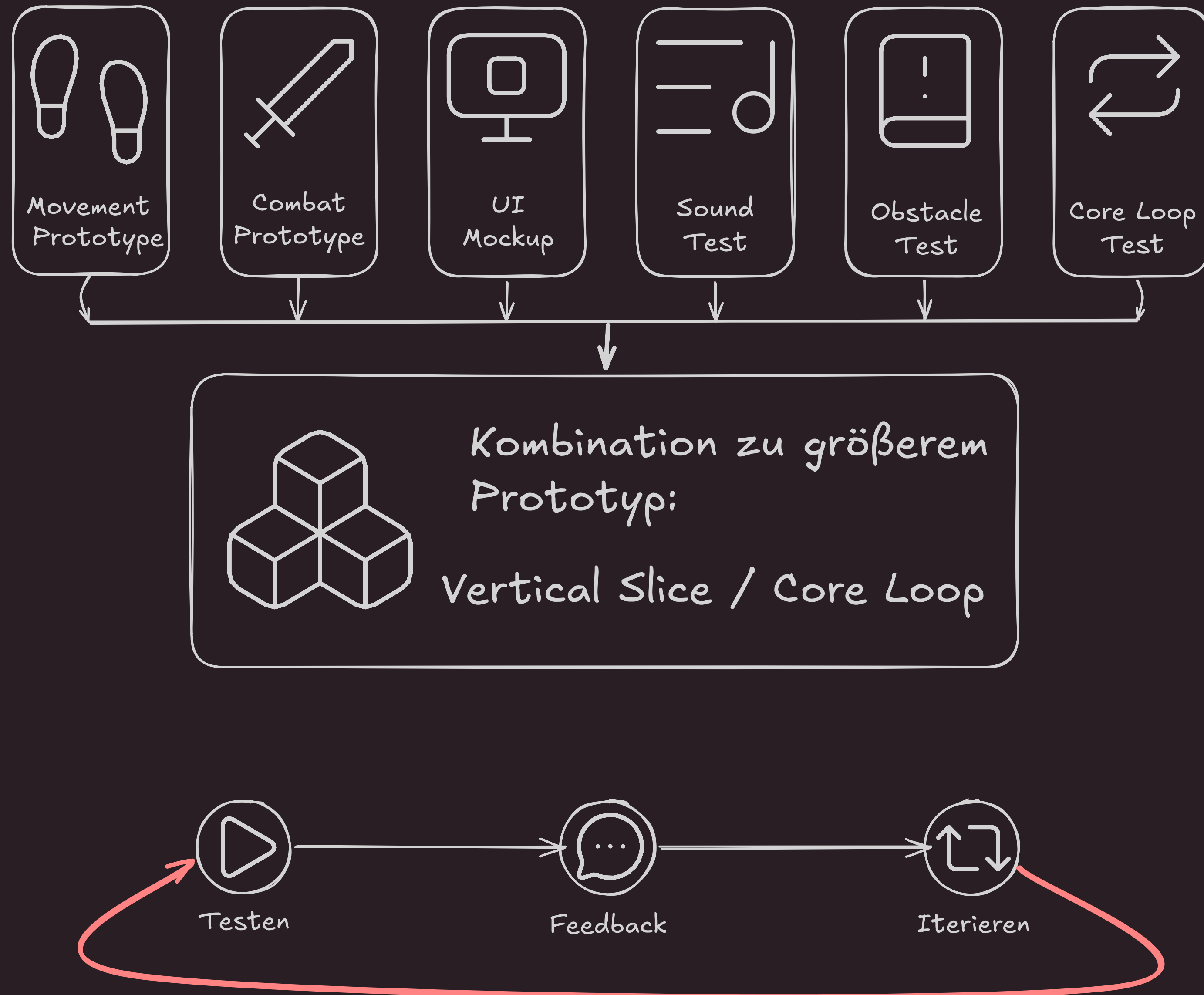


Zusätzlich sollte bei einer Auseinandersetzung zwischen zwei Mitgliedern eine schlichtende Person Einfluss nehmen können, darunter zum Beispiel ein Teammitglied oder eine Person von außen. Somit soll ein "Festdiskutieren" verhindert werden.

Wir sind auch der Meinung, dass dieses Problem mit der Zeit einen geringen Umfang einnehmen wird, sobald die Mitglieder des Teams besser miteinander vertraut sind.



## Mini-Prototypen-System



# UNSER KONZEPT

Das Teamkonzept setzt sich aus den vorher besprochenen Methoden, Konzepten und Erfahrungen zusammen. Es dient als Leitwerk für die Arbeit im Team und gibt Antwort auf mögliche Probleme und Risiken.

Die Inhalte dieses Konzepts basieren auf den uns aktuell verfügbaren Informationen und können sich im Verlauf der Zusammenarbeit ändern. Wie erfolgreich die verschiedenen Leitregeln sind, soll an entsprechenden Erfolgskriterien erkannt werden.

← Aufbau unseres Konzepts, basierend auf dem Mini-Prototypen System und den damit möglichen schnellen Feedback-Loops

## Prototyping

Wir haben uns dazu entschieden, unser bisher erfolgreiches Prototypensystem zu verwenden. Dieses sieht, wie zuvor beschrieben, die Erstellung von Miniprototypen sowie die Zusammenfassung dieser zu größeren Prototypen vor. Die Verwendung dieses Iterations-Systems erlaubt es uns auch viele der von uns positiv eingeschätzten Methoden und Konzepte umzusetzen. Darunter:

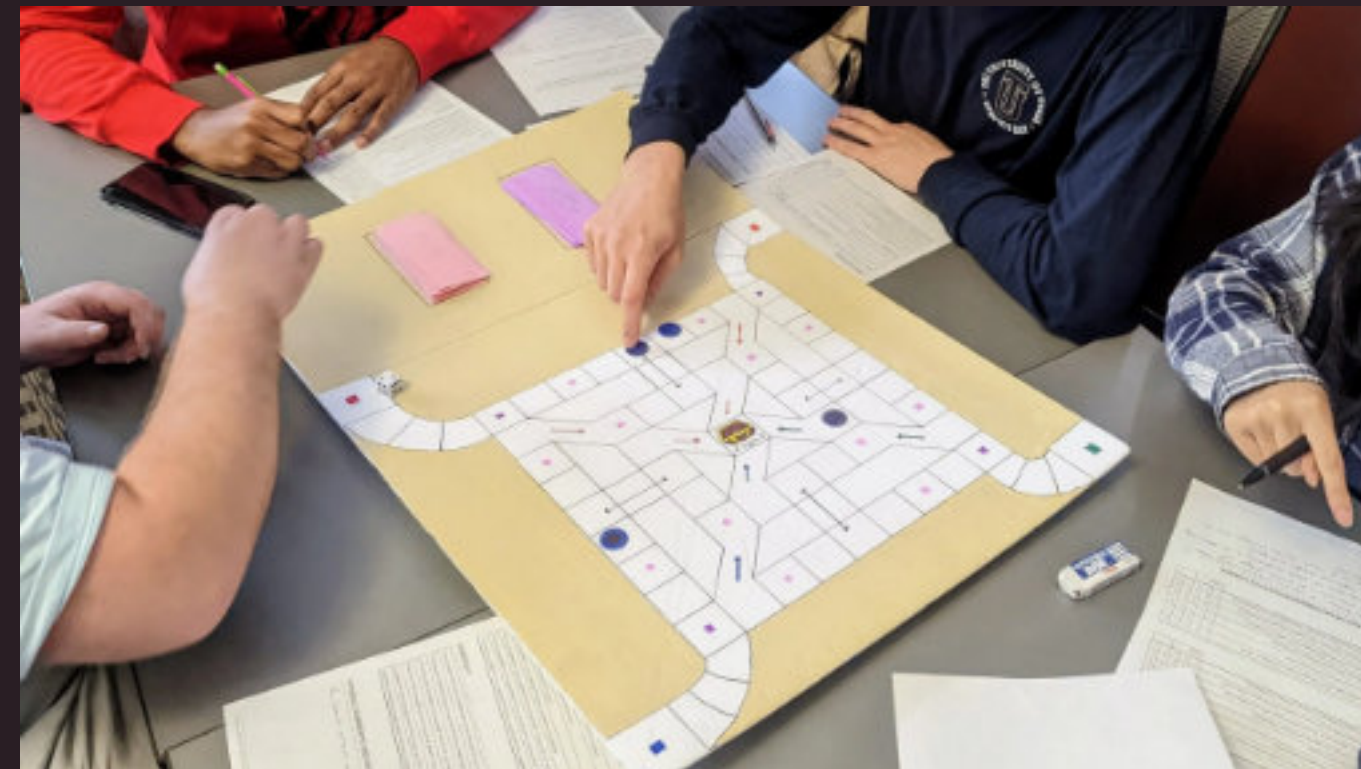
- **Fail Faster**
- **Die Schleifenregel**
- **Grey Boxing**
- **Mockups/ UI-Prototypen**
- **Digital Core-Loop-Prototyp (zusammengebaut aus Mini-Prototypen)**
- **Minimum Viable Product**
- **Vertical Slicing**
- **Schleifenregel**



# UNSER KONZEPT

## Play-Testing

Testing und Feedback stellen für uns den zweiten großen Grundstein dar. Durch unsere Methodik der Prototypenerstellung ist es uns möglich, die Mini-Prototypen sofort an Tester zu geben und Feedback zu erhalten. Play-Testing gibt uns verschiedene Perspektiven und deckt Probleme auf, die sonst nicht erkannt werden würden.



Die von uns zuvor beschriebenen Play-Testing-Methoden sollen hier Anwendung finden. Die passenden Methoden sind jeweils von der Art des Tests abhängig, somit kann beim Testen frei unter den Methoden gewählt werden.

Play-Testing erfolgt am besten bereits so früh wie möglich.

Bild von [getcreatvetoday.com](#) <sup>[10]</sup>

Die wichtigsten Risiken müssen identifiziert und regelmäßig überprüft werden



Auch die Faustregel: "Wenig Tester, viel Iteration" soll bei unseren Nutzertests Anwendung finden. Wir würden allerdings die Anzahl der Tester anhand der Wichtigkeit skalieren.

## Risikominierung

Um Risiken erfolgreich vorzubeugen, müssen wir diese zuerst identifizieren und entsprechende Maßnahmen festlegen. Wir haben im Team die wichtigsten Risiken analysiert und haben Präventionsregeln sowie entsprechende Reaktionen bei Eintritt des Worst-Cases auf den nächsten Seiten formuliert.

## Risiko-Level (gesamt)

Aktuelles Gesamtrisiko unseres Projekts.



## Risiken & Worst-Case-Szenarios

Wichtige Risiken im Blick behalten (jede 2 Meetings).



### Teammitglied-Ausfall

Verlust oder temporäre Personalausfälle.

MITTEL



### Fehlende Skills

Benötigtes Können nicht im Team.

MITTEL



### Schlechte Kommunikation

Fehlende oder unklare Absprachen.

HOCH



### Scope-Creep

Anforderungen wachsen über den Plan.

HOCH



### Burnout

Überlastung einzelner Mitglieder.

MITTEL



### Entwicklungsstagnation

Schwindende Motivation im Team.

MITTEL



Risiko	Minimierungsmaßnahme	Maßnahmen im Worst Case
Teammitglied verlässt Team oder temporäre Personalausfälle	<ul style="list-style-type: none"><li>Keine kritischen Aufgaben an nur eine Person verteilen</li><li>Wissen dokumentieren</li><li>Wöchentliche Meetings</li></ul>	<ul style="list-style-type: none"><li>Neuverteilung der Aufgaben</li><li>Evtl. neu onboarden</li><li>Scope anpassen</li><li>Mehrheit nimmt Projekt mit</li></ul>
Fehlende Skills	<ul style="list-style-type: none"><li>Planung basierend auf bestehenden Skills oder Motivation</li><li>Suchen von externer Hilfe</li><li>Lernen, wenn die Zeit es erlaubt</li></ul>	<ul style="list-style-type: none"><li>Nur Grundlegendes implementieren</li><li>Verwerfen von Features</li><li>Outsourcen</li></ul>
Fehlende / schlechte Kommunikation	<ul style="list-style-type: none"><li>Wöchentliche Meetings</li><li>Teammitglieder können Ideen in Mini-Prototypen umsetzen, ohne dass sie direkt verworfen werden</li><li>Bei Meetings: 1 gute Sache, 1 schlechte Sache nennen</li><li>Klare Planung u. Zuweisung von Rollen (Kanban-Tickets m. Name)</li><li>Klar definierte Vision</li></ul>	<ul style="list-style-type: none"><li>Notfall-Meeting</li><li>Hinzuziehen von neutralen Vertrauenspersonen / Koordinatoren</li><li>Wenn alles andere versagt: Aufteilung des Teams — absoluter Worst-Case, Scheitern des Projektes</li></ul>
Scope-Creep	<ul style="list-style-type: none"><li>MVP</li><li>Wöchentliche Reflexion</li><li>MoSCoW-Methode</li><li>Rücksprachen mit Koordinatoren</li></ul>	<ul style="list-style-type: none"><li>Verwerfen von Features</li><li>Reduzieren aufs Minimum</li></ul>
Unterschiedliche Workflows	<ul style="list-style-type: none"><li>Absprache</li><li>Konsens finden (Mehrheit)</li></ul>	<ul style="list-style-type: none"><li>Getrennte Arbeit</li><li>Verspätetes Zusammenführen</li></ul>

Risiko	Minimierungsmaßnahme	Maßnahmen im Worst Case
Vertrauensmangel	<ul style="list-style-type: none"><li>Transparenz zwischen Teammitgliedern</li><li>Gute Kommunikation</li><li>Besseres Kennenlernen</li></ul>	<ul style="list-style-type: none"><li>Notfall-Meeting</li><li>Hinzuziehen von neutralen Vertrauenspersonen / Koordinatoren</li></ul>
Kreative Konflikte	<ul style="list-style-type: none"><li>Mini-Prototypen</li><li>Klare Rollendefinition</li><li>Bei kritischen Entscheidungen: Rest des Teams hinzuziehen</li><li>Ansonsten hat Verantwortlicher das Wort</li></ul>	<ul style="list-style-type: none"><li>Hinzuziehen von neutralen Vertrauenspersonen / Koordinatoren</li></ul>
Entscheidungs lähmung (mehrere Mitglieder haben ganz unterschiedliche Meinungen)	<ul style="list-style-type: none"><li>Entscheidungsregeln festlegen (z. B. zeitliche Begrenzung)</li><li>Verantwortlicher hat das Wort</li></ul>	<ul style="list-style-type: none"><li>Teammehrheit (oder Koordinator) entscheiden lassen</li><li>Hilfe von außerhalb</li></ul>
Burnout	<ul style="list-style-type: none"><li>In Meetings auf Über-/Unterforderung fragen</li><li>Bei Verdacht auf Burnout: ansprechen, präventiv Pausen verhängen, positiv bestätigen, Kommunikation mit Vertrauensperson</li></ul>	<ul style="list-style-type: none"><li>Kommunikation mit Vertrauensperson</li><li>Notfall-Meeting</li><li>Scope-Anpassung (auf Personenunfall vorbereiten)</li><li>Aufgabenverteilung</li><li>Teammitglied Pause und Hilfe verschaffen</li></ul>
Entwicklungsstagnation (schwindende Motivation)	<ul style="list-style-type: none"><li>50/50 Must-do und Spaßarbeit</li><li>Fokus neu setzen</li><li>Klare Meilensteine setzen</li><li>Rewarding</li></ul>	<ul style="list-style-type: none"><li>Scope-Anpassung</li><li>Zeitraumen verändern — Pause vom Projekt</li><li>Hilfe von außen</li><li>Outsourcen von nicht spaßigen Aspekten</li></ul>



Auch für den alltäglichen Arbeitsrahmen würden wir gerne einige Regeln formulieren.

§1

Nachdem alle notwendigen Planungsaspekte stehen, können die Teammitglieder selbst ihre präferierte Arbeitsweise festlegen, so zum Beispiel, ob sie Hause oder in Präsenz arbeiten.

§2

Planungsgespräche sollten, wenn möglich, in Präsenz geführt werden, wobei alle notwendigen Teammitglieder anwesend sein müssen. Sehr wichtige Gespräche bedürfen aller Mitglieder.

§5

Die Faustregel: "Wenig Tester, viel Iteration" soll wenn bei uns getestet wird Anwendung finden. Die Anzahl der Tester darf allerdings anhand der Wichtigkeit des Tests skalieren.

§6

Jede Woche soll jedes Teammitglied einen positiven sowie einen negativen Aspekt vorbringen, damit Kommunikation offen und ehrlich bleibt. So soll auch Scope-Creep verhindert, Team-Vision geeinigt und Rollenüberschreitungen verhindert werden.

§3

Es kommen im Alltag immer wieder Diskussionen auf. Diese sollten stets offen und respektvoll geführt werden. Es wird erwartet, sich professionell zu verhalten und zu keinem Zeitpunkt persönlich zu werden. Arten Diskussionen aus, so soll mit einer dritten Person eine Schlichtung versucht werden.

§4

Zusätzlich zu den Planungsgesprächen sollen wöchentlich Teammeetings gehalten werden. In diesen soll kurz über Fortschritte berichtet sowie Ausprache über negative Aspekte ermöglicht werden.

§7

Für verschiedene Themenbereiche der Entwicklung werden im Team verantwortliche Personen ernannt. Diese dienen als interne und externe Ansprechpartner für den zuständigen Bereich im Team. Auch steht diesen Personen ein besonderes Veto-Recht bei Entscheidungen zu. Dies soll dazu führen, dass Entscheidungen schneller getroffen werden können und Scope-Creep verhindert wird.

§8

Falls im Team eine Meinungsverschiedenheit auftritt, so hat die verantwortliche Person zunächst das letzte Wort. Handelt es sich um eine wichtige Entscheidung, so kann der Fall auf Teamebene eskaliert werden. Hier kann das ganze Team über das Problem diskutieren und zu einer Entscheidung kommen. Hierzu sollte es nur kommen, wenn es absolut notwendig ist.



# Meilensteine

Klare Ziele in Etappen erreichen.

1. Konzept	In Arbeit
2. Core-Loop-Prototyp	Offen
3. Vertical Slice	Offen
4. Alpha	Offen
5. Beta	Offen
6. Release Candidate	Offen

# Fortschritt (gesamt)

Wie weit sind wir insgesamt?



# UNSER KONZEPT

## Erfolgskriterien

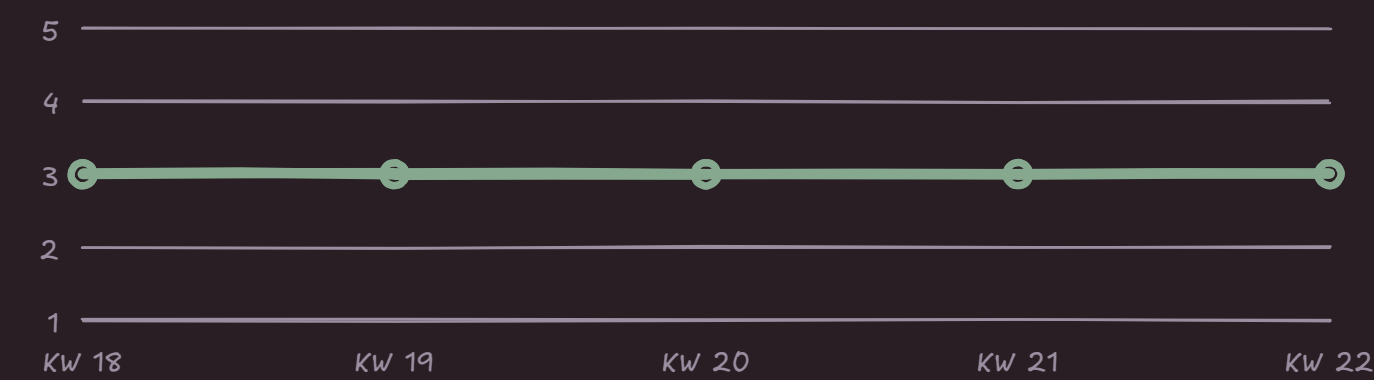
Um der schwindenden Motivation im Entwicklungsverlauf entgegenzuwirken, den Fortschritt zu bewerten und die in diesem Handbuch festgelegten Methoden auf die Probe zu stellen, legen wir zudem noch einige Möglichkeiten fest, unseren Erfolg zu ermitteln.

So sollen zum Beispiel konkrete Meilensteine aufgestellt werden. Anhand der Erfüllung dieser können Arbeitsgeschwindigkeit und Fortschritt bewertet werden. Zudem soll auch in den wöchentlichen Meetings auf Teamzufriedenheit und Konfliktnummern geachtet werden. Es soll festgestellt werden, wie effizient die Arbeit verlaufen ist.

- ← Meilensteine, dienen uns um Fortschritt abzuschätzen und Zeitziele zu setzen
- ➡ Ein konstanter Fortschritt und Teamzufriedenheit sind wichtig um Demotivation vorzubeugen

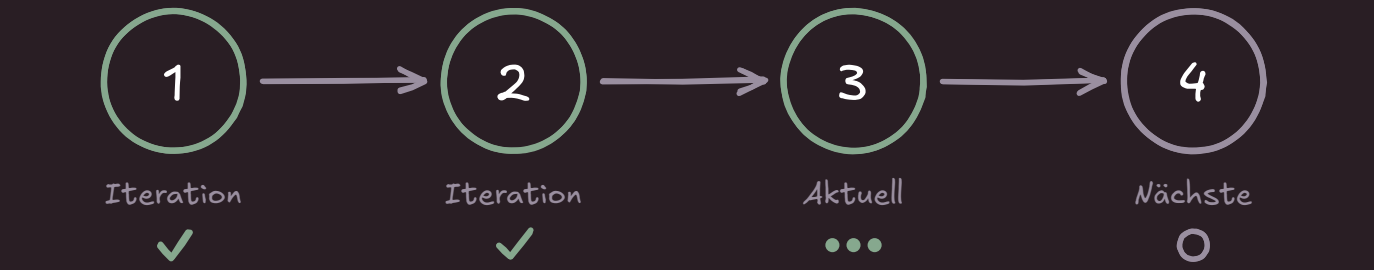
## Teamzufriedenheit

Regelmäßige Bewertung im Team (1-5).



## Iterationen & Reviews

Kontinuierlich prüfen & verbessern.



Wichtig ist außerdem, auf die zuvor definierte Liste an wichtigen Risiken zu achten. Es muss regelmäßig (jede 2 Meetings) bewertet werden, wie nah der Eintritt eines Worst-Case-Szenarios ist. Anhand dessen soll auch die Effektivität unserer Präventionsmaßnahmen ermittelt werden, damit diese notfalls angepasst werden können.



### Warum ein iteratives Design?

Unserem Konzept liegt ein interaktiver Gedanke zugrunde. Wir entscheiden uns für iteratives Arbeiten – basierend auf positiven Erfahrungen aus früheren Projekten und den Ergebnissen unserer Recherche. Durch Iterationen können wir...

- Fehler früher erkennen
- Nutzerfeedback direkt wieder ins Projekt einfließen lassen
- stetige Verbesserungen erzielen (gut für Teammoral)
- Neue Ideen direkt integrieren
- Entwicklungsrisiken niedrig halten, da flexibel reagiert werden kann

### Wann ist ein Prototyp erfolgreich?

Unser Konzept verwendet, über das Mini-Prototypen-System, viele Prototypen. Ein erfolgreicher Prototyp bereichert uns mit wertvollen Informationen für den weiteren Entwicklungsverlauf.

Selbst wenn ein Konzept nicht funktioniert oder keinen Spaß macht, sind dies wertvolle Erkenntnisse.

Darunter sind z.B Infos zu Umsetzungszeit oder Realisierbarkeit.

Im Gegensatz dazu liefert ein schlechter Prototyp keine relevanten oder bereits bekannte Informationen.

### Warum Play-Testen wir zusätzlich zu den Prototypen?

Ergänzend zu den objektiven Informationen, welche uns der Prototyp liefert, können in Play-Tests subjektive Erkenntnisse gewonnen werden. Sie erlauben uns mehr aus unseren Prototypen zu gewinnen.

Darunter sind z.B. Aspekte wie Spielspaß oder Spielerverhalten.

Anders als bei Prototypen sind hier wiederholende Informationen gewünscht, da so von subjektiven Eindrücken auf objektive Fakten geschlossen werden kann.

Playtesting soll bei uns somit nach Möglichkeit mit jedem Prototyp einhergehen.

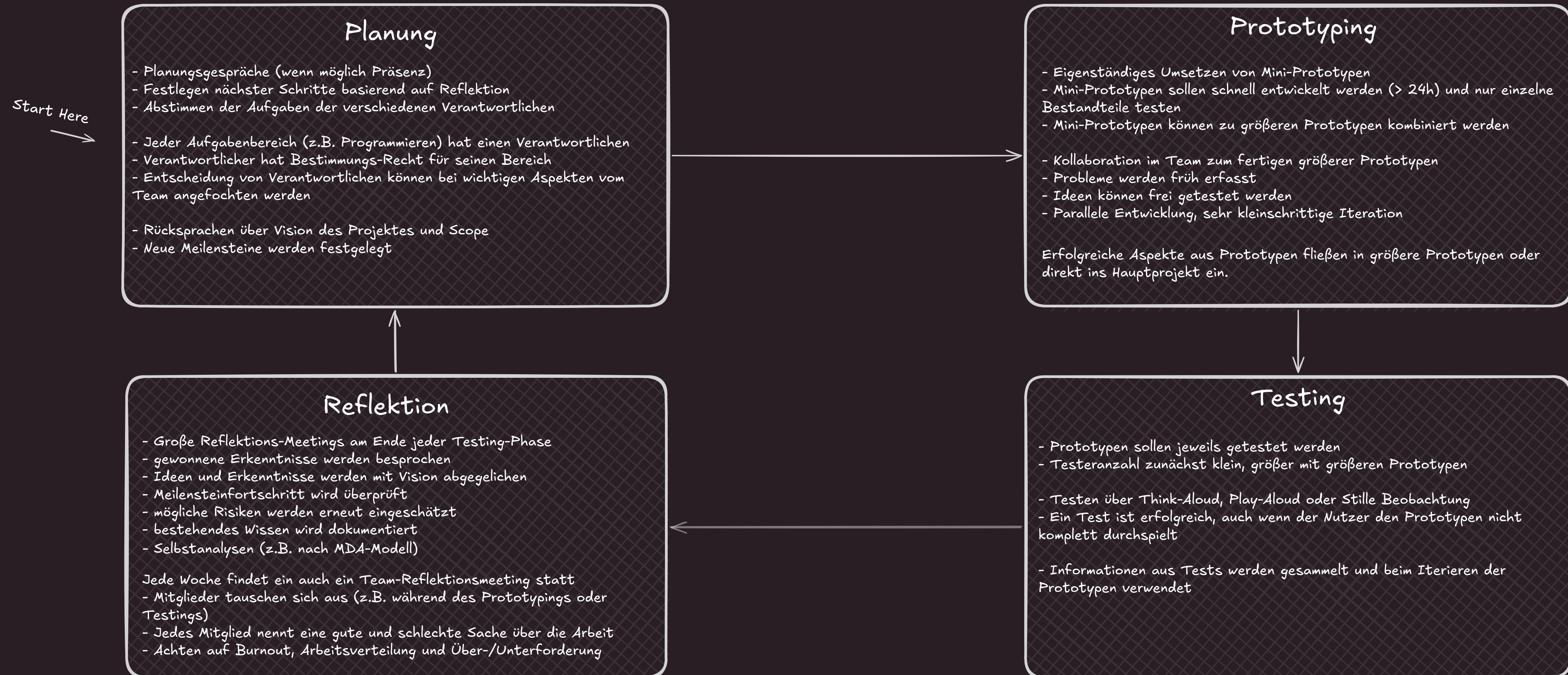
### Wie haben wir unser Konzept erarbeitet?

Dieses Konzept wurde von uns gemeinsam in mehreren Meetings geschaffen. Zunächst sammelten wir alle möglichen Konzepte und Methoden aus unseren bisherigen Erfahrungen und Literatur zusammen.

Dann in zwei großen Diskussions-Meetings einigten wir uns auf die jetzige Arbeitsweise. Dabei nutzten wir die Konzepte, welche gut zusammen passten und sich bewährt hatten.

Danach teilten wir Textformulierung auf und schrieben die Texte innerhalb einer Woche. Das Layout wurde dann in einer weiteren Woche fertiggestellt.







# QUELLEN

## Bilder

- [1]

s 13

<https://www.weclapp.com/de/lexikon/risikoanalyse/>  
*https://www.weclapp.com/de/wp-content/uploads/sites/4/2026/03/ablauf-risikoanalyse.png.webp*
- [2]

s 15

<https://marcabraham.com/2014/07/16/gamification-and-the-md-framework/>  
*https://marcabraham.com/wp-content/uploads/2014/07/the-md-framework-and-the-20-cs-of-meaningful-enterprise-gamification.png*
- [3]

s 17

[https://en.wikipedia.org/wiki/Scrum\\_\(project\\_management\)](https://en.wikipedia.org/wiki/Scrum_(project_management))  
*https://commons.wikimedia.org/wiki/File:Scrum1\_process-de.svg*
- [4]

s 19

[https://en.wikipedia.org/wiki/Paper\\_prototypingexperience/](https://en.wikipedia.org/wiki/Paper_prototypingexperience/)  
*https://commons.wikimedia.org/wiki/File:Diamond\_Trust\_of\_London\_-\_Paper\_Prototype\_zoom.jpg*
- [5]

s 23

<https://www.gamesindustry.biz/why-devs-need-to-fail-fast-and-how-to-do-so-gi-sprint>  
*https://www.weclapp.com/de/wp-content/uploads/sites/4/2026/03/ablauf-risikoanalyse.png.webp*
- [6]

s 25

<https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>  
*https://blog.crisp.se/wp-content/uploads/2016/01/Making-sense-of-MVP-.jpg*
- [7]

s 30

<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>  
*https://media.nngroup.com/media/editor/2023/08/07/why-you-only-need-to-test-5-users-1.jpg*
- [8]

s 33

<https://www.embis.de/campus/onlinemarketing-strategie/>  
*https://www.embis.de/app/uploads/2025/04/Wissen-Online-Marketing-Strategie-HeaderBeitragsbild-1024x768.jpg*
- [9]

s 34

*https://stock.adobe.com/de/images/fresh-ideas-group-of-young-business-people-discussing-something-and-smiling-while-sitting-at-the-office-table/289031631*
- [10]

s 37

<https://getcreativetoday.com/playtesting-measuring-player-experience/>  
*https://i0.wp.com/getcreativetoday.com/wp-content/uploads/2025/09/22fa-CSG1003-Playtest-scaled.jpg*

## Literatur

- Jesse Schell – The Art of Game Design
- Macklin & Sharp – „Games, Design and Play: A Detailed Approach to Iterative Game Design

*https://de.scribd.com/document/631010433/John-Sharp-Colleen-Macklin-John-Sharp-Games-Design-and-Play-A-Detailed-Approach-to-Iterative-G-pdf*

S. 215
- Tracy Fullerton – „Game Design Workshop: A Playcentric Approach to Creating Innovative Games

*https://books.google.de/books?id=taHNBQAAQBAJ*
- Salen & Zimmerman – „Rules of Play“

*https://gamifique.wordpress.com/wp-content/uploads/2011/11/1-rules-of-play-game-design-fundamentals.pdf*
- Game Design Is Iterative. What Does It Mean?

*https://www.thegamerspad.net/game-design-is-iterative-what-does-it-mean/*
- Understanding What Game Design is Iterative Means: Mastering Your Development Approach

*https://www.tunica.tech/blogs/understanding-what-game-design-is-iterative-means-mastering-your-development-approach*
- Minimal Coding, Iterative Prototyping, and Playtesting: A Novice Design Thinking Approach to Gamifying the User Experience

*https://journals.publishing.umich.edu/weaveux/article/id/140/*
- Towards an Iterative Design for Serious Games

*Viudes-Carbonell, S. J., Gallego-Durán, F. J., Llorens-Largo, F., & Molina-Carmona, R. (2021). Towards an Iterative Design for Serious Games. Sustainability, 13(6), 3290. https://doi.org/10.3390/sui3063290*
- Design strategies – Edexcell: Iterative design process

*https://www.bbc.co.uk/bitesize/guides/zjjkw6f/revision/4*



