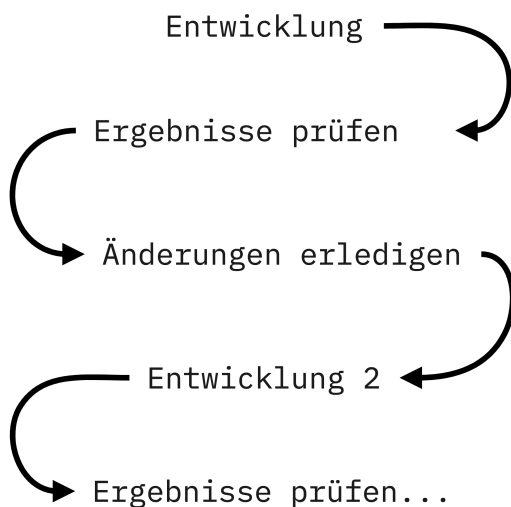




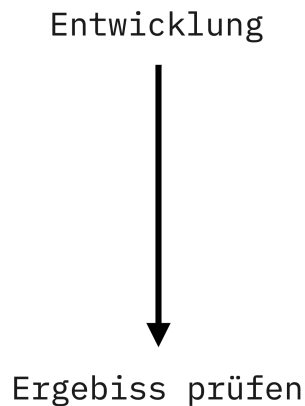
Refinement

Das Refinement, welches sicher einige aus SCRUM kennen, ist das zusammenkommen und beschreiben der Tickets um sie bearbeit-bar zu machen. Im Fokus vom Refinement steht die Kommunikation und der Konsent darüber, was getan werden muss, und **warum**.

▼ Warum macht man überhaupt ein Refinement?



Egal was das Produkt ist, es gibt immer **Anforderungen** die geklärt werden müssen. Diese zu klären, ist Ziel vom Refinement. Um zu vermeiden, dass man von der Entwicklung in eine Review bzw Ergebnis Überprüfung geht und diese Rückmeldung eine erneute Entwicklung auslöst, ist das Refinement dafür da, alle notwendigen Informationen bereits vorab zu klären;- noch bevor die Entwicklung startet.



Am Ende sitzt jeder im Vorteil, denn je genauer die Klärung der Aufgabe, desto weniger hin und her gibt es. Die Entwicklung kann klar stattfinden und beinhaltet alle notwendigen Informationen um mit Fokus an der Aufgabe zu arbeiten.

▼ Fragen die in einem Refinement geklärt werden

- Wer nimmt die Arbeit am Ende ab?
- Mit wem muss die Qualität besprochen werden? Was ist möglich / nicht möglich?
- Wie können wir beweisen, dass es funktioniert?
- Welches Bedürfnis soll erfüllt werden?
- Wer ist der Kunde? Wie viele?
- Wer ist der Nutzer? Wie viele?
- Welche Ansprüche müssen erfüllt werden?
- Wer arbeitet an dem Thema?
- Was wurde bereits erledigt?
- Was sind die Vorbedingungen?
- Was sind die Abhängigkeiten?
- Woher kommt die Aufgabe?
- Wer hat welche Informationen? Welche sind noch notwendig?
- Was ist wirklich gewollt in der Aufgabe?

- Wer benötigt was genau?
- Wer erledigt was?
- Wie viel Zeit bleibt uns? Wie viel Zeit brauchen wir?
- Wann genau sind wir wirklich fertig? Was beinhaltet "alles"?

Eine typische Liste aller notwendigen Fragen die beantwortet werden müssen, wenn man an etwas arbeiten will oder muss. LIFe erarbeitet schrittweise durch User Story Mapping und Refinements auf der Meta und der Ticket Ebene, um nach und nach alle Fragen bereits beantwortet zu haben, wenn die Arbeit an einer Aufgabe losgeht. Somit kann ein voller Fokus auf die Aufgabe gewährleistet werden, welche die Qualität des Produktes stark erhöht.

Es gibt in LIFe drei große Refinement's:

1. **Initiativen**
2. **Slices**
3. **(Job) Tickets**

In diesen drei Bereichen werden *verschiedene Fragen vom Refinement* geklärt und **Dokumentiert**. In jeder der Hierarchien abschnitte bis zur Ticket Ebene, können somit Fragen abgeschlossen werden und müssen nach dem gefundenen Konsent nicht kontinuierlich jedes Mal neu diskutiert werden.

▼ **Template für Initiativen**

1. Impact:

[Warum gibt es diese Initiative? Welches Thema soll hier geklärt werden?]

-

2. Definition of Ready (DoR):

[Welche notwendigen Vorbedingungen oder Vorbereitungen sind notwendig, um den Start der Initiative einzuleiten]

-

3. Definition of Done (DoD):

[Was ist der Finale "Shippable Value", also wann ist die Initiative fertig?]

-

4. ACC:

[Was sind die Akzeptanzkriterien für die Initiative (Requirements)]

-

5. Persona:

[Wer sind die Persona / Wer ist der Kunde / Wer ist der Nutzer ?]

-

6. Involved:

[Welche Kompetenzen sind wahrscheinlich notwendig um die Initiative zu bearbeiten?]

-

7. Stakeholders:

[Wer sind die Anfoderer an die Initiative? Wer setzt die Qualitätsanforderungen an die Initiative?]

-

Das Refinement für Initiativen bezieht sich auf die höchste Ebene für Refinement's und soll das Finale Ergebnis vom Thema der Initiative festziehen. Um die Aufwandklärung für das Thema zu betreiben durch das User Story Mapping, muss zuerst das Refinement der Initiative abgeschlossen sein. Hier werden die ersten *warum* fragen beantwortet.

▼ Template für Slice

1. User Story:

Als <Nutzer>, will ich <Funktionalität>, so dass <Vorteil>

<User Story>

2. Impact:

[Was ist die Lösung für die sich entschieden wurde? Welchem Impact hat dieser?]

-

3. Definition of Ready (DoR):

[Was sind die Vorbedingungen um mit der Arbeit vom Slice zu beginnen]

-

4. Definition of Done (DoD):

[Wann ist der Slice fertig?]

-

5. ACC:

[Was sind die Akzeptanzkriterien vom Slice?]

-

6. Review

[Wer nimmt den finalen Slice ab?]

-

Das Refinement für den Slice zieht den **Wert für den Release** fest, auch *Inkrement* genannt. Ein Slice besteht aus allen notwendigen Aufwänden aus der **Ticket Ebene**, welche zum größten groben Teil aus den verschiedenen User Story Map's generiert wurde. Um festzustellen, **welche Tickets** alles benötigt werden, muss der Slice mit seinem **Wert festgelegt werden**-dieses findet im Refinement vom Slice statt.

▼ Template für Job Ticket

1. Job Story:

Wenn folgende <Situation>, habe ich <Motivation>, so dass <erwartetes outcome>

<Job Story>

2. Definition of Ready (DoR):

[Was sind die Vorbedingungen oder notwendigen Vorbereitungen um mit der Arbeit vom Slice zu beginnen]

-

3. Definition of Done (DoD):

[Wann ist das Ticket fertig? ("Shippable Value")]

-

4. ACC:

[Was sind die Akzeptanzkriterien vom Ticket? ("Requirements")]

-

5. TestCase:

[How can you check that the ACC and DoD have been fulfilled]

-

Das Refinement von der Ticket Ebene findet zum Schluss statt. Hier geht es darum, nur noch das einzelne Ticket mit seinem **Aufwand für die Kompetenz festzuziehen**, wenn nötig das Ticket zu schneiden und die Sub-tasks aufzuschreiben, die das Ticket erfüllen. Job Tickets werden auf den Kanban Board's durch den Prozess gezogen. Alle Tickets erfüllt, wäre der Slice abgeschlossen, dem sie zugeordnet sind.

Sub-Task

Ein weiterer Teil vom Refinement des Job Tickets ist die Job Story,- eine Ableitung von der bekannten User Story:

User Story & Job Story

▼ Template für Defect/Bug Ticket

Abschnitt 1

[Auszufüllen beim finden vom Defect oder Bug]

1.1 Steps to reproduce:

[Wie kann der Defect/bug generiert werden?]

-

1.2. Expected result:

[Was sollte eigentlich passieren?]

-

1.3. Actual result:

[Was passiert stattdessen?]

-
-

Abschnitt 2

[Auszufüllen während der Reparatur bzw. Behebung]

2.1 Rootcause:

[Was lief schief? und wo?]

-

2.2 Proposal to fix:

[Wie könnte man ihn reparieren? Verschiedene Optionen aufschreiben und besprechen, sowie auch den Entschluss für welche Lösung sich entschieden wurde begründen]

-

2.3 TestCase:

[Wie wird überprüft das der Defect / Bug behoben ist?]

-

Defect oder Bug Tickets sind mit einem spezifischen Refinement gekennzeichnet, da sie andere notwendige Daten als ein normales Job Ticket benötigen. Mit Defect/Bug Tickets geht man um wie mit normalen Job Tickets, zieht sie durch den Prozess und schließt sie durch die Abnahme.