**10. Backlog**

**Einleitung:**

Zwei Vorgehensweisen beim Abrieten mit dem Backlog sind „**Prune the Product Tree**“ und „**Remember the Future**“.

//10.1 Arbeiten

Beim Arbeiten am Backlog geht es darum ein grobes Bild des gewünschten Ergebnisses zu erschaffen und nicht den Weg zu diesem Ergebnis zu Planen. Was bedeutet das noch keine diskreten Lösungsansetzte/Probleme behandelt werden sollen. Das Arbeiten am Backlog fördert den Austausch der Einzelnen Stakeholder und ermöglicht gleichzeitig allen einen übersichtlichen und Strukturierten Überblick an die Anforderungen/Ziele. //

**10.2 Prune the Product Tree**

Prune the Product Tree bedeutet übersetzt „trimme den Produktbaum“, was bereits eine gute Beschreibung dieser Vorgehensweise ist. Hierbei wird das Product mit seinen Funktionsbereichen /Themen auf einen Baum gebracht wobei die Dickeren Äste/der Stamm höher Priorisierte Themen haben wohingegen dünnere Äste niedriger Priorisierte Themen. Im Anschluss trimmt man dann den Baum um die nicht essenziellen Themen, wodurch man den Baum/das Produkt in die gewünschte Form bringt.

Es gibt zwei Möglichkeiten diesen Baum anzuwenden:

* Um ein neues Produkt zu gestallten.
* Um ein Bereits existierendes Produkt weiterzuentwickeln.

**Vorgehen beim Erstellen eines Produkts**

Schritt 1: Brainstorming hierbei werden erstmalig alle Anforderungen der Stakeholder gesammelt und gezeigt.

Schritt 2: Clustern hierbei werden die Anforderungen in Gruppen gesammelt und diesen dann Überthemen gegeben. Diese Überthemen werden dann auf den Stamm oder die Dicken Äste des Baumes gehangen.

Schritt 3: grobe Priorisierung, hierbei werden die wichtigen Anforderungen an dickere Äste gehängt und die weniger wichtigen weiter außen.

Im Anschluss wird kontrolliert, ob die entstandene Baumstruktur die Interessen der Stakeholder passend repräsentiert. Gibt es Themengebiete mit Übergewicht? Gibt es Themengebiete mit zu wenig Fokus?

**Vorgehen bei der Weiterentwicklung**

Hier werden bereits vorhandene Features an dickere Äste platziert und neue Wünsche/Features weiter Außen an dünnere. Es besteht auch die Möglichkeit Features zu entfernen.

**10.3 Remember the Future**

Remember the Future bedeutet übersetzt „erinnere dich an die Zukunft“, was auch hier die Vorgehensweise gut beschreibt. Es wird eine zeitleiste mit drei Zeitpunkten vorbereitet.

1: der Momentane Zeitpunkt.

2: Der Zeitpunkt eines Ereignisses.

3: Ein Zeitpunkt nach diesem Ereignis.

Nun stellen sich die Stakeholder vor das Sie bei Zeitpunkt Drei Revue passieren lassen und geben Gründe für den Erfolg bei Zeitpunkt Zwei an. Im Anschluss werden diese Gründe dann wieder geordnet und priorisiert.

**10.4 Der initiale Stand**

Beim Arbeiten mit den Backlog durch „Prune the Product Tree“ oder „Remember the Future“ wurde eine erste intuitive Reihenfolge erstellt diese wird noch durch das Systematische **Priorisieren**, welches die genaue Reihenfolge der Abarbeitung bestimmt und das **Schneiden** angepasst. Weiterhin muss noch Risikomanagement betrieben werden.

**11 Priorisierung**

Im Voraus wurde das Backlog mit Anforderungen/To-dos gefüllt und erstmalig intuitiv geordnet. Nun müssen diese weiter sortiert und priorisiert werden, um eine Reihenfolge der Abarbeitung zu bestimmen. Diese kann sich mit der Zeit noch ändern, wenn neue Erkenntnisse oder Erfahrungen gemacht worden sind, aufgrund von Feedback oder wenn neue To-dos hinzukommen.

Kriterien zur Bestimmung der Reihenfolge sind

* **Nutzen oder Geschäftswert**

Wertsteigernder oder kostensparender Inhalt

**Bedeutung aus Kundensicht**

Wie wichtig ist der Inhalt aus Kundensicht.

* **Risiko und Wert**

Positive Risiken, in Form von Gewinnchancen fördernde oder Negative Risiken unterdrückende Maßnahmen.

Zum besseren Sortieren kann man diese in eine Matrix mit den Achsen Risiko und Geschäftswert einsortieren.

* **Cost of Delay**

Die Kosten die entstehen durch verzögerten Abschluss eines To-Dos/ verpasste Gewinnchancen.

* **Komplexität, Aufwand, Kosten**

Bei der Umsetzung eines To-dos gibt es Entwicklungskosten, Aufwand/Komplexität und Wartungskosten, welche dann mit den Gewinnpotenzial des Ergebnisses abgewogen werden müssen. To-dos mit höheren Gewinnchancen werden priorisiert.

Wobei eher die Komplexität beachtet wird und weniger der Aufwand, da dieser in Absoluten Werten, welche subjektiv eingeschätzt werden, angegeben wird. Hingegen wird die Komplexität Relativ bewertet welches den vergleich vereinfacht.

* **Gesetzliche oder andere reguläre Vorgaben**
* **Abhängigkeit**

Im Normalfall versucht man die Einzelnen To-dos bei der Erstellung so zu gestalten, dass sie voneinander unabhängig sind, um besser nach Wert sortieren zu können, allerdings ist dies nicht immer möglich.

* **Erwerb von Wissen und Erfahrung**

Reduktion über Unsicherheit über das Wie oder Was

Zum Bestimmen der Reihenfolge, nach diesen Kriterien, kann man verschiedene Tools benutzen, wobei jeder dieser ihre eigenen Stärken haben und Fokus auf verschiedene Kriterien werfen.

**11.3 Priorisieren nach Einschätzen der Komplexität**

Zum priorisieren nach Komplexität kann man die Tools „**Relatives schätzen nach T-Shirt Größen**“ oder „**Planungspoker**“ nutzen.

Beim Relativen Schätzen nach T-Shirt Größen werden den Unterschiedlichen To-dos die Größen S, M, L, XL zugeordnet, von unkompliziert zu kompliziert. Hierfür brauch das Team eine Referenz. Am ende erhält man eine Grobe Struktur, die die Auf Wendigkeit einteilt.

**11.4 Priorisieren nach Einschätzung des Wertes**

Für die Priorisierung nach dem Wert, gibt es auch wieder verschiedene Vorgehensweisen.

* **Einfaches Priorisieren**

Hierbei werden Prioritätsstufen verteilt zum Beispiel 1, 2, 3 oder A, B, C.

Worauf sich die Stufen beziehen, ist hierbei nicht festgelegt. Weiterhin gibt es keine klare Einteilung ab wann welche stufe beginnt.

* **Abstufung nach MuSCoW**

Hier werden die To-dos in die folgenden Stufen eingeteilt, welche eine Strengere form bieten als das einfache Priorisieren:

**Mu (Must have):** Features die essenziell wichtig sind für die Inbetriebnahme des Produkts.

**S (Should have):** Features die vorhanden sein sollten, allerdings nicht müssen

**Co (Could have)** : Features die nicht vorhanden sein müssen, allerdings dennoch Vorteile bieten und Implementiert werden sollten wenn nach „Must have“ und „Should have“ Features noch Ressourcen übrig sind.

**W (Would like to have, but not this Time)**

Features die momentan nicht in Betracht gezogen werden und nur notiert werden.

**11.4.3 Priorisierung nach Wert oder Attraktivität für Kunden**

Bei diesen verfahren werden die To-dos nach der Attraktivität für den Kunden sortiert. Dies geschieht dadurch, dass eine Gruppe von Schätzern Punkte an die To-dos verteilt, ignoriert werden essenzielle To-dos wie Dokumentation, die aus Kundensicht irrelevant sind. Anstatt dessen wird nur auf funktionale To-dos gesetzt.

Die konkreten verfahren sehen wie folgend aus:

**Monopoly Money** in diesen Fall wird Monopoly Geld auf die To-dos gesetzt. Die Menge an Geld sollte ungefähr das Projekt Budget betragen vorzugsweise etwas weniger. Der Betrag den die einzelnen To-dos erhalten stellt das Interesse der Gruppe da.

**100-Punkte-Methode** bei dieser Methode hat jeder Stakeholder die Möglichkeit 100 Punkte auf die Features zu verteilen, wobei jeden frei überlassen ist wie er dieses tut (Alle auf eins oder weit gestreut).

**Buy a Feature – Kunden „kaufen“ Features zu vorher definierten Preisen**

Hierbei erhalten die Features festgelegte Preise, welche durch Aufwand, Risiko oder auch aus Strategischen Gründen wie Kollision von Eigenschaften berechnet werden. Diese Preise können auch die Kaufkapazität eines einzelnen Kunden überschreiten und somit eine Bündelung der Gelder benötigen. Kunden sollten so viel Geld haben, das sie allein zwei Drittel der Features erwerben können. Sollten keine Echten Kunden zur Verfügung stehen können auch Teammitglieder wählen.

**11.4.4 Planning Poker mit Business value Points**

Dieses Verfahren wurde entwickelt da Menschen sich bei Abstimmungen in einer Gruppe unterbewusst von anderem Mitziehen lassen und somit ihre Meinung verfälscht wird. Es soll versucht werden eine Meinung aller zu erhalten, um dann auf Basis dieser weiter diskutieren zu können, wodurch man am ende ein von allen Akzeptiertes Ergebnis erhält, welches deren unbeeinflusste Meinung wieder spiegelt.

Für dieses Verfahren benötigt man Karten mit Punktewerten (z.B. in Fibonacci Reihenfolge) bis 100. Diese erhält jeder Stakeholder.

Wenn jeder Stakeholder Karten hat, werden To-dos vorgelesen und erklärt im Anschluss kann jeder eine Karte mit Punkten vergeben, ohne zu sehen was die andere tun. Dann wird über die Ergebnisse diskutiert und nochmal gevotet, bis ein Übereinstimmendes Ergebnis entsteht.

**11.5 Kombinierte Betrachtung von Komplexität und Wert-Wall Estimation**

Dieses Verfahren fügt verschiedene Verfahren zur Sortierung nach Komplexität/Größe (z.B. T-Shirt verfahren) und Priorität/Wichtigkeit für Kunden zusammen (z.B. **Einfache Priorisierung**). Dies geschieht durch das Erstellen eines Koordinaten System wobei Größe und Komplexität die Achsen darstellen. Wenn man die To-dos eingeteilt hat, hat man drei Kategorien

1: niedrige Komplexität, hohe Priorität. (diese Aufgaben werden zuerst erledigt)

2: hohe Komplexität, hohe Priorität. (Werden gegebenenfalls in kleinere Aufgaben untereilt)

3: niedrige Priorität (werden erstmal nicht umgesetzt)

**Zusammenfassung:**

Es gibt vier Faktoren nach denen Priorisiert wird

**Geschäftswert** (Business Value Points oder Verzögerungskosten),

**Kundennutzung** (Bestimmt durch Monopoly Money, 100 Punkte oder Buy a Feature),

**Komplexität/Aufwand** (T-Shirt Schätzen),

**Risiken**.

Bei kleineren Projekten reicht es aus die verschiedenen Einflussfaktoren im Gedächtnis zu halten und muss diese verfahren nicht unbedingt anwenden.

12 Themes, Epics, Features, User Stories, Tasks To-dos in unterschiedlicher Granularität

Auf einen Backlog gibt es To-dos die bereits einen Festen Arbeitsschritt beschreiben und To-dos die nur grob eine Idee beschreiben. Um diese zu unterscheiden, werden sie Hierarchisch in die Typen Theme, Epic, Feature, User Story oder Task eingeteilt.

**Theme,** beschreibt die grobsten To-dos welche lediglich ein Thema oder eine Idee sind. Und bildet Somit die Basis der Hierarchie.

**Epic/Feature** können Synonym verwendet werden, muss aber nicht, es gibt in den Fall keine feste Regel welches höher gestellt ist in der Hierarchie. Features sind Teil eines Themas und entstehen, wenn man sich näher mit seiner Planung beschäftigt.

**User-Story** ist die Untermenge von Features. Gleichzeitig beschreibt dies aber auch die Art wie Anforderungen/To-Dos formuliert werden sollen. User-Storys Entstehen, wenn man ein Feature so weit in kleinere To-dos aufsplittet, so dass diese in begrenzter Zeit bearbeitet werden können.

**Task** ist die konkrete Planung einer User Story und werden erst kurz vor der Umsetzung dieser Formuliert.

**Wofür braucht man diese Kategorien?**

Da beim agilen Arbeiten versucht wird ein Fokus zu behalten werden nur die hoch Priorisierten Themen in Features und User-Storys runtergebrochen. Wodurch die weniger Priorisierten Themen zwar notiert sind, allerdings noch nicht geplant werden. Um womöglich unnötige Planung und Arbeit, bei Themen/Ideen Verwurf, zu vermeiden. Die Hierarche hilft das Akronym (Deep) beim Arbeiten mit den Backlog besser einzuhalten, da es eine Einteilung der To-dos gibt, die deren Zeitlichen Aufwand beschreibt.

**Deep**

**E = emergent** es kann jederzeit zu Umstrukturierung der Prioritäten kommen was teilweise zu unnötiger Planung führen kann, wenn Themen im Voraus bereits zu Tiefgründig behandelt wurden.

**D = detailed appropriately** es werden keine Unnötigen Themen aufgearbeitet, solange sie nicht oben stehen in der Prioritätenliste, denn auch dies würde wieder zu Unnötiger Arbeit führen.

**P = priorisiert** Die Wichtigsten Themen stehen oben in runtergebrochener Form von User Storys.

**12.2 User Stories Formulierung und Bezeichnung**

User Stories sind nicht fest definiert, da sie einerseits eine Schablone, welche die Formulierung von Anforderungen als solche vorgibt sind und andererseits stellen sie eine Stufe in der Aufwands-Hierarchie da.

**Als<<Rolle>>** wer benötigt die Funktionalität (Personas oder Stakeholder)

**will ich<<Funktionalität>>**

**damit<<Business Begründung>>** Nutzen der Funktionalität, gibt Hinweis auf Priorität und gibt Kontext.

Der Vorteil dieser Form ist, dass sie eine Beziehung zum Nutzer und Kontext herstellt. Allerdings ist diese sehr knapp, weshalb es die ergänzende Regel 3C gibt. Die 3 C‘s stehen für

**Card**: Karte mit User Story.

**Conversation**: Zusätzliche Unterhaltung und Erklärung. Hierbei erklärt der Product Owner den Entwicklerteam die User Storys, um Missverständnisse und Fehlinterpretationen zu vermeiden. Welche bei herkömmlichen Vorgehensweisen bei denen lediglich ein Aufgabenblatt, welches von einem Analysten gestellt wurde, übergeben wird oft passieren.

**Confirmation**: Hier werden weitere Kriterien, die das fertige Produkt erfüllen muss, um als abgeschlossen zu gelten aufgelistet.

**User Stories auf Featureebene**

Da User Stories auch ein Format darstellen kann man diese auch nutzen, um Anforderungen auf Feature Ebene zu stellen. In diesen Fall kann es sein das man die 3C’s nicht einhalten kann, da eine Konversation über das Produkt noch nicht möglich ist oder keine Akzeptanzkriterien gestellt werden können, da dieses noch nicht ausführlich genug geplant wurde.

**12.3 Große User Stories in kleinere unterteilen**

User Storys enthalten Eigenschaften, die sie erfüllen müssen, wenn man nun User Stories aufteilt/schneidet oder Features in User Stories aufteilt müssen die diese Eigenschaften wieder erfüllen. Beschrieben werden diese Eigenschaften durch das Akronym INVEST

**I**ndepenent: die Storys sollen voneinander unabhängig sein.

**N**egotiable: Nicht alles ist niedergeschrieben es gibt verhandlungsraum für Product Owner und Kunden.

**V**aluable: User Stories sollen etwas zu dem Produkt beitragen.

**E**stimatable: Der Aufwand für die Umsetzung muss einschätzbar sein.

**S**mall: Klein genug um, mehrere Umzusetzen oder den Fortschritt von Abreitschritt zu Arbeitsschritt verfolgen zu können.

**T**estable: Es muss die Möglichkeit geben eine User Story auf erfolgreiche Umsetzung zu testen.

Um User Stories nun so zu schneiden, dass sie diese Eigenschaften Erfüllen gibt es verschiedene Ansätze.

* **Splitten nach Workflow,** hier wird nach den schritten im Workflow geschnitten, Gefahr besteht hier darin das ein Abhängigkeit zwischen den User Stories entstehen kann.
* **Splitten nach Geschäftsregeln,** wenn es verschiedene Regeln für die Umsetzung gibt, kann nach diesen getrennt werden
* **Splitten nach Aufwand** hier werden die User Stories in verschieden Aufwendige Aufgaben aufgeteilt.
* **Variationen in den Daten**, für die Entwicklung des Produkts in verschiedenen Sprachen werden verschiedene User Stories erstellt.
* **Nach Operationen wie Lesen, Neuerfassen,Ändern,Löschen**