# Vergleichskriterien der Agilen Prozesse untereinander

Anhand der nachfolgenden Vergleichskriterien werden die drei agilen Prozesse, Crystal, Kanban und Verglichen. Die Auswertung der einzelnen Prozesse erfolgt auf Grundlage der Recherchen der einzelnen Prozesse. Die Vergleichskriterien wurden anhand einzelner Besonderheiten der einzelnen Prozesse aufgestellt. Die Vergleichskriterien dienen auch nach Abschluss des Projekts zum Vergleich der agilen Prozesse anhand unserer gewonnenen Erfahrungen. Die Gegenüberstellung anhand der Matrix ist besonders hilfreich um Unterschiede oder auch Gemeinsamkeiten hervorzuheben.

Begriffserklärung

**Iterationen:** Iterationen sind Abschnitte die immer wieder nach einem gleichen Muster durchlaugen werden. In den einzelnen Iterationen werden Teile der Software innerhalb eines festgelegten Zeitraums entwickelt

**Teamgröße:** die Teamgröße beschreibt die Optimale beziehungsweise mindestens benötigte Teamgröße in einem Projekt mit der jeweiligen Entwicklungsmethode

**Rollen:** Unter Rollen versteht man die Zuweisung unterschiedlicher Aufgabenbereiche, welche in einem Projekt vorhanden sein sollen.

**Techniken:** die Techniken beschreiben die Vorgehensweise während der Entwicklung, welche Techniken zum Best Paractise gehören und welche möglich sind.

**Lieferung:** Die Lieferung beschreibt die Termine an denen Jeweils ein fertiger Softwareteil dem Kunden übergeben werden kann. Dies kann entweder nach bestimmten Iterationen oder immer zu festen Zeiten erfolgen.

**Änderung der Anforderungen:** In den Verschiedenen agilen Prozessen ist es an verschiedenen Stellen möglich die Anforderungen des Kunden zu ändern, dies kann durch unvorhergesehene Einflüsse neue Erkenntnisse oder durch wirtschaftliche Vorkommnisse unter anderem nötig sein

**Änderung der Arbeitsweise:** Eine Änderung der Arbeitsweise ist je nach Vorgehens-modell an verschiedenen Stellen möglich. Solche Änderungen sind nötig wenn festgestellt wird, dass eine bestimmte Arbeitsweise für dieses Projekt nachteilig ist

**Meetings/Kommunikation:** Die Kommunikation ist in den einzelnen Agilen Prozessen sehr unterschiedlich geregelt, ebenso die Zeiten wann eine Kommunikation untereinander stattfindet und auf welche weise.

**Dokumentation:** Unter Dokumentation wird unter anderem die Dokumentation des Fortschritts eines Projektes, sowie vorherige Planungsdokumente verstanden und in welcher weiße diese erfolgt.

**Comitment/ definition of Done:** Ist die Festlegung wann eine User Story als erledigt makiert werden kann.

**Größe der Tasks:** Die Größe der Tasks richtet sich danach in welcher Zeitspanne ein einzelner Task abgearbeitet werden kann.

**Aufwandsschätzungen:** Für die Aufwandsschätzungen gibt es verschiedene Möglichkeite. Zu Teil können diese einen Realen Ursprung wie die Zeit haben oder Punkte.

**Priorisieren:** Bei der Priorisierung geht es darum wer die einzelnen User Stories in einem Projekt priorisiert.

**Empirie:** Empirische Erhebung von Informationen zur Erstellung von Statistiken über den Entwicklungsprozess (z.B. Velocity)

Die Gegenüberstellung anhand der Matrix ist besonders hilfreich um Unterschiede oder auch Gemeinsamkeiten hervorzuheben. Die einzelnen Prozesse werden in ihrer Ursprünglichen, nicht modifizierten Art gegenüber gestellt. Denn gerade in Der Praxis werden die Prozesse nicht immer in Ihrer Ursprünglichen Art genutzt, sondern den Bedürfnissen des Teams angepasst.

|  |  |  |  |
| --- | --- | --- | --- |
| Kriterien | Crystal | Kanban | Scrum |
| Iterationen | Iterationszyklen innerhalb einer Lieferung | keine Pflicht - variable Länge oder gleichmäßige Iterationen | gleichmäßige Länge  in der Regel 1-4 Wochen |
| Teamgröße | 2-6 | keine Begrenzung | 7-10 |
| Rollen | - Chefdesigner  - Endanwender  - Auftraggeber  -Programmierer/  Designer | keine Vorgaben | - Scrum Master  - Product Owner  - Scrum Team |
| Techniken | Verschiedenste agile Techniken:  - Pairprogramming  - Side by side   Programming,  - Side by side  Testing | - WIP-Limit  - Kanban-Board zur  Visualisierung  - Service-Level-Agreement-Klassen (SLAs) | Verschiedenste agile Techniken:  - Pairprogramming  - Side by side   Programming |
| Lieferung | mind. 2 Lieferungszyklen  nach einer Iteration | Release - am Ende einer Iteration oder beliebig festgesetzt | nach jedem Sprint → gezeigt im Review |
| Änderungen der Anforderungen | vor jeder Iteration möglich | immer möglich | Vor jedem Sprint möglich |
| Änderung der Arbeitsweise | nach jeder Lieferung (Reflexion der Arbeitsweise) | erwünscht (kontinuierliche Verbesserung)  immer möglich | nach jeder Retrospektive |

|  |  |  |  |
| --- | --- | --- | --- |
| Meetings/ Kommunikation | osmotische Kommunikation, Reflexionsmeeting nach Iteration und Lieferung, weitere Meetings empfohlen | Empfehlungen aber keine Pflicht:  - Daily Standup  - After Meeting  - Queue  Replenishment  Meeting  - Release  Planning  Meeting  - Triage  - Issue Log Review  - Retrospektive | - Sprint  Planning  Meeting  - Daily Scrum  - Review  - Retrospektive |
| Dokumentation | umso besser die Kommunikation desto weniger Doku, aber generell empfohlen  Beipiele:  - Projektplan,  - Versionsplan,  - Risikoliste,  -Architekturbeschreibung | keine Pflicht –  Kanban Board  + elektronischem Trackingsystem | - Scrumboard  - elektronisches  Trackingsystem |
| Definition of Done/ Commitment | nicht festgesetzt | nicht Pflicht | Product Owner und Team gemeinsam |

|  |  |  |  |
| --- | --- | --- | --- |
| Größe der Tasks | nicht vorgeschrieben, möglichst klein | möglichst gleich groß | möglichst kleine Tasks  max Tag /Task |
| Aufwands-schätzungen | Blitzplanung,  vor jeder Iteration anpassbar | nicht Pflicht | Sprint Planning Meeting Abschätzung: Story Points |
| Priorisierung | in der Blitzplanung durch Auftraggeber, Anpassungen während des Projektes | Queue Replenishment Meeting: Nicht-Entwickler  bei der Umsetzung: Entwickler (FIFO) Hinwegsetzen: SLAs | vom Product Team Anpassungen bei Abhängigkeiten vornehmen |
| Empirie | nicht gegeben | Cumulative Flow, Lead und Cycle Time (Durchsatz) | Velocity (Story Points) |

In der direkten Gegenüberstellung fallen zuerst die Gemeinsamkeiten bei Iterationen auf, denn jeder Prozess hat, Beziehungsweise kann Iterationen haben. Lediglich in der Art der Länge unterscheiden sie sich. So haben Crystal und Scrum die Vorgabe, dass es mehrere Iterationszyklen innerhalb der Entwicklung geben muss. Bei Scrum sind die Bestimmungen noch enger gefasst, denn hierbei ist die optimale Länger einer Iteration vorgegeben. Im Gegensatz dazu ist Kanban hierbei sehr offen, denn hier ist nicht Vorgeschrieben ob es eine oder mehrere Iterationen gibt.

Ebenso bei der Teamgröße erweist Kanban die höchste Flexibilität, denn hier gibt es keine Begrenzung der Teamgröße. Anders sieht es hier bei Crystal und Scrum aus. Chrystal ist eher für kleinere Entwicklungsgruppen konzipiert, Wohingegen Scrum für ein etwas größeres Team ausgelegt ist.

Ebenso offen wie bei der Teamgröße ist Kanban auch bei der Rollenverteilung. Hier können sowohl die Rollen von Scrum wie auch von Crystal übernommen werden, oder ganz klassisch mit einen Projektleiter. Die Rollen bei Crystal und Scrum hingegen sind fest definiert.

Die Techniken sind auch wie die Rollen wieder sehr unterschiedlich, da bei Scrum und Crystal alle möglichen agilen Techniken wie Pairprogramming, Side by side Programming und Side by side Testing zum Einsatz kommen können. In Kanban hingegen stehen Techniken wie WIP oder Service-Level-Agreement-Klassen im Vordergrund. Bei WIP (Work in Progress) wird festgelegt, welche Anzahl von Tasks sich in einem Entwicklungsschritt befinden dürfen. Das Festlegen und einsetzten von Service-Level-Agreement-Klassen (SLAs) beschreibt den einsatz von verschiedenen Ticketarten. Darunter fallen unter anderem Bugs und Change Requests.

In allen Prozessen erfolgt die Lieferung nach einer Iteration. Jedoch können bei Kanban die Lieferungen auch außerhalb von Iterationen stattfinden, da es ja wie oben bereits erwähnt wurde nicht zwingend notwendig ist mehrere Iterationen zu durchlaufen.

Sowohl bei der Änderung der Anforderungen so wie der Arbeitsweise sind wieder Scrum und Crystal gleich. Die Änderungen der Anforderungen erfolgen bei beiden direkt von der Iteration. Bei Crystal hingegen sind Änderungen jeder Zeit möglich. Auch dies hat wieder den Hintergrund das es nicht zwingend Iterationen gibt.

Für die Änderung der Arbeitsweise gilt das gleiche, hier sind ebenfalls Crystal und Scrum gleich. Bei beiden kann die Arbeitsweisen nach einer Iteration geändert werden.

Um eine ungeschickte und eventuell fehleranfällige Arbeitsweise in der nächsten Iteration zu verbessern.   
Bei Kanban hingegen ist eine kontinuierliche Verbesserung der Arbeitsweise während der Iterationen gerade zu erwünscht. Denn jede Verbesserung der Arbeitsweise zieht lauf Kanban eine Verbesserung des Produktes nach sich.

Die Kommunikation und die Meetings sind bei allen Prozessen sehr unterschiedlich. So ist bei Crystal die osmotische Kommunikation, bei der alle Teammitglieder im gleichen Raum sitzen sollen, das best Parctice. Zudem kommen Reflexionsmeetings nach den Iterationen. Weitere Meetings sind keine Pflicht können aber zum Beispiel durch Dailys ergänzt werden. Bei Kanban lautet das Motto wie vorher schon „alles kann aber nichts muss“. So zählen zu den Empfohlenen Meetings zum Beispiel ein Daily Standup, ein After Meeting, das Queue Replenishment Meeting, ein Release Planning Meeting, ein Triage und ein Issue Log Review. Beim Triage wird nach alten Tasks im Backlog gesucht und diese gelöscht. Bei Scrum hingegen sind die Meetings fest vorgeschrieben, dazu zählen Sprint Planning Meeting, Daily Scrum, Review und Retrospektive

Zur Dokumentation haben alle drei Prozesse ein Board an dem die User Stroies und die dazugehörigen Tasks für das ganze Team sichtbar sind. Zudem werden Technische Hilfsmittel wie ein elektronisches Trackingsystem immer häufiger verwendet. Bei Crystal kommen zu dem Board und dem Trackingsystem noch die Dokumentation des Projektplans, des Versionsplans, der Risikoliste sowie eine Architekturbeschreibung hinzu.

Die Definition of Done oder auch Commitment besagt ab wann eine User Story vom Kunden als erledigt abgenommen wird. Dabei ist dies nur bei Scrum genau festgehalten und wird vom Product Owner festgelegt. Bei den anderen zwei Prozessen ist eine feste Definition of Done nicht festgeschrieben

Die größer der Tasks ist bei allen Prozessen sehr ähnlich. Dabei sollte darauf geachtet werden das die einzelnen Tasks eine geringe Größe haben. Jedoch sollte als Spezialisierung bei Kanban darauf geachtet werden das die einzelnen Tasks ungefähr die gleiche Größe haben. Bei Scrum hingegen gilt es als Sinnvoll darauf zu achten, das ein einzelner Task in der Größe nur so groß ist das er innerhalb eines Tages abarbeitbar ist.

Bei der Aufwandschätzung unterscheiden sich die drei Prozesse sehr. Crystal führt zu Anfang des Projektes eine Blitzplanung durch in dem die jeweiligen Zeiten für die einzelnen Tasks geschätzt wird. Vor jeder Iteration können die Zeiten bezüglich gewonnener Erfahrungswerte angepasst werden. Bei Kanban ist eine Zeitabschätzung nicht vorgeschrieben. Natürlich kann auch hier eine Schätzung zu jeder Zeit erfolgen. Bei Scrum wird im Sprint Planning Meeting die Zeiten für die einzelnen Tasks die im Sprint abgearbeitet werden sollen vom gesamten Team geschätzt.

Bei der Priorisierung der Tasks sind sich Crystal und Scrum wieder sehr ähnlich hier werden die Tasks vom Auftraggeber beziehungsweise vom Product Owner durchgeführt. Jedoch können die Entwickler während der Entwicklung Tasks oder ganze User Stories umpriorisieren, falls dazu die Notwendigkeit besteht. Dies kann der Fall sein wenn User Stories oder Tasks von andren Abhänig sind. Kanben (fehlt noch)

Die Erhebung von Statistiken so genannte Empiere gibt es nur in Kanban und in Scrum. Bei Kanban gibt es unter anderem den cycle time, dabei wird die Zeit der Arbeit am Ticket beginnen bis zu dessen Ende ins Verhältnis gesetzt. Bei Scrum gibt es das Velocity, dies besagt wie viele Storiepoints innerhalb eines Sprints abarbeitbar sind.