# Decide whether your SE Process for implementing the MHC-PMS should be rather plan-driven, agile, or a combination of both, by collecting pros and cons, considering

* Pro Agile: Die meisten Teammitglieder sind bereits einen agilen Entwicklungsprozess gewohnt
* Pro Agile: Das Kundenfeedback kann schneller in die Entwicklung aufgenommen
* Pro Agile: Da wir alle keine Mediziner sind, ist damit zu rechnen, dass laufend Fragen zum System und zur Art und Weise der Verwendung auftauchen werden. Deshalb ist es von Vorteil, wenn wir flexibel auf Anforderungsänderungen reagieren können.
* Pro plan driven: Bei einem plan driven Prozess können die Anforderungen bis ins Detail definiert und darauf beharrt warden, was zu einer einfacheren Systemabnahme führt.
* Pro plan driven: Ein Kunde ist möglicherweise zu fixiert auf seine Bedürfnisse, was eine agile Entwicklung verhindert, wenn das System auch für andere Kunden eingesetzt werden können soll.
* Pro plan driven: Es sind kaum Redesigns nötig, da der Kunde keine Änderungswünsche anbringen kann.
* Pro combination: Es ist ein gesundes Mass an Redesings voraussehbar, da auf Kundenwünsche zwar eingegangen wird, aber nicht auf jeden Schnickschnack.
* Pro combination: Der Kunde muss weniger aktiv im Projekt tätig sein, weil nicht jedes kleinste Detail mit dem Kunden angeschaut wird.
* Pro combination: Der Kunde aus der medizin- und nicht aus einer technischen Branche kommt, ist es besser, wenn er nur die wesentlichen Systemfunktionalitäten mitentscheidet.

# Discuss which SE process models would be fitting for the project at hand – collect pros and cons and decide on the best one from your perspective.

|  |  |  |
| --- | --- | --- |
| SE process model | Pros | Cons |
| Agile | * Ermöglicht Kundeneinflüsse * Iterative Entwicklung | * Allgemeines Agile-Modell zu unspezifisch |
| Enterprise Unified Process | * Ermöglicht iteratives entwickeln * Ermöglicht regelmässiges Ausliefern stabiler Versionen dank regelmässiger Qualitätssicherung | * Zuviel Overhead für die Projektteamgrösse |
| Executable UML | * Dokumentation und Software kombiniert | * Sehr hohe Abstraktion * Team hat keine Erfahrung damit |
| Incremental model | * Durch die Testphase nach jeder Iteration werden Fehler schnell gefunden | * Wenn zusätzliche Funktionen hinzugefügt werden, können Fehler in bereits abgeschlossen Iterationen wieder auftreten |
| Iterative model | * Iteratives Entwickeln * Ermöglicht Einbezug des Kunden | * Keine klaren Schritte/Entwicklungsresultate Möglicherweise viele Redesigns notwendig |
| Rational Unified Process | * Keine | * Vorfahre vom Enterprise Unified Model |
| Scrum | * Flexibel dank Besprechung vor jeder Entwicklungsphase * Qualitätssicherung dank Reviews * Iterative Entwicklung | * Weniger geeignet, wenn Teammitglieder sich nicht regelmässig austauschen |
| Prototype Model | * Ermöglicht besser Integration des Kunden | * Ist anfällig auf ungenügende Analysen * Fehlprototyp könnte Kunden verwirren oder unter Entwicklern missverstanden werden Benötigt viel Entwicklungszeit |
| Spiral Model | * Viele Risikoanalysen * Programmierung beginnt relative früh | * Risikoanalyse benötigt viel Erfahrung * Eher für grosser als kleinere Projekte geeignet |
| V-Model | * Frühere Testphase gegenüber Wasserfallmodel 🡪 Fehler warden schneller gefunden * Jede Phase hat klare Endproduktive | * Genauso starr wie Waterfall-Modell * Keine frühen Prototypen, dadurch keine Möglichkeit, Designfehler früh zu finden |
| Waterfall Model | * Einfache Entwicklung dank unveränderbaren Vorgaben * Einfach zu verstehen, kein guter Teamzusammenhalt notwendig | * Keine Möglichkeit, Änderungswünsche einzubringen * Keine Möglichkeit, grosse Designfehler effektiv zu korrigieren * Grosser administrativer Overhead |
| Extreme Programming | * Iterative Entwicklung * Einfache Methoden passen zum kleinen Projekt * Motivierend für Entwickler | * Benötigt Erfahrung, dass es sonst eine code-and-fix Übung wird |

Siehe <http://en.wikipedia.org/wiki/Software_engineering#External_links> die blaue Software Engineering Box

Entscheid: Scrum

Begründung:

Wir sind ein Team, dass unterschiedlich Erfahren ist in der Software Entwicklung. Darum wählten wir ein iteratives Entwicklungsmodell, um nachträgliche Fehlentscheide oder auch Änderungswünsche des Kunden einfach und effizient in die Entwicklung einfliessen lassen zu können. Von der Art des Projektes her, ist es ebenfalls von Vorteil, wenn das Entwicklungsmodell klare Rollen und definierte Schritte hat, weswegen die Modelle Incremental und Iterative wegfallen. Auf Grund der Art und Weise der zeitlichen Rahmenbedingen und der geforderten Resultate fällt Extreme Programming weg. Da das Projekt eher klein ist, fällt auch das Spiral Modell weg und nur Scrum bleibt noch übrig.

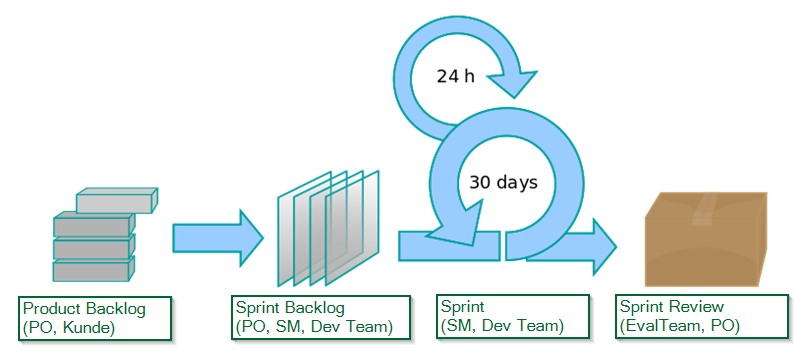
# Based on your decison, design - a process model with concrete activities - for each activity, the goals, tasks, and potential output(s) - the relation among outputs of different activities - the involvement of stakeholders (= customer, users, management) and development team (= you) and required interaction

## Rollenverteilung

|  |  |  |
| --- | --- | --- |
| Product Owner | PO | Die Dozenten, Hauptvertreter des Kunden |
| Development Team | DevTeam | Swen, Pascal v. O., Roger, Pascal B. |
| ScrumMaster | SM | Swen |
| Test and Evaluation Team | EvalTeam | ScrumMaster mit DevTeam |

## Aktivitäten (vereinfachtes Scrum)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Goals | Tasks | Potential Output |
| Product Backlog | Geordnete Liste der Anforderungen erstellt Sprint Endzeit ist bekannt Bisher erreichtes ist besprochen | Planning meeting: Product Owner legt die zu erledigenden Aufgaben fest | Folgende Fragen sind beantwortet: |
| Sprint Backlog | Konkrete Anforderungen ausgesucht, die im nächsten Sprint umgesetzt warden | Sprint planning meeting: Development-Team definiert mit Product Owner, was sie vom Product Backlog zeitlich alles umsetzten können | Die ToDo-Liste für den nächsten Sprint |
| Sprint | Sprint Backlog umgesetzt | Scrums: Regelmässige Treffen und Besprechend des Entwicklungsstandes | Beantwortete Fragen aus Scrum Meeting: Was habe ich seit letztem Meeting gemacht?  Was ist weiter geplant geplant?  Gibt es Dinge, die mich aufhalten? |
| Sprint Review | Der neue Softwareincrement ist ausgeliefert und vorgeführt | Sprint review meeting: Development Team liefert Software aus und führt die Neuerungen dem Product Owner vor | Eine bereinigte Liste, was im letzten Sprint erledigt werden konnte und was liegen geblieben ist |



Angepasst aus Quelle: <http://en.wikipedia.org/wiki/File:Scrum_process.svg>