

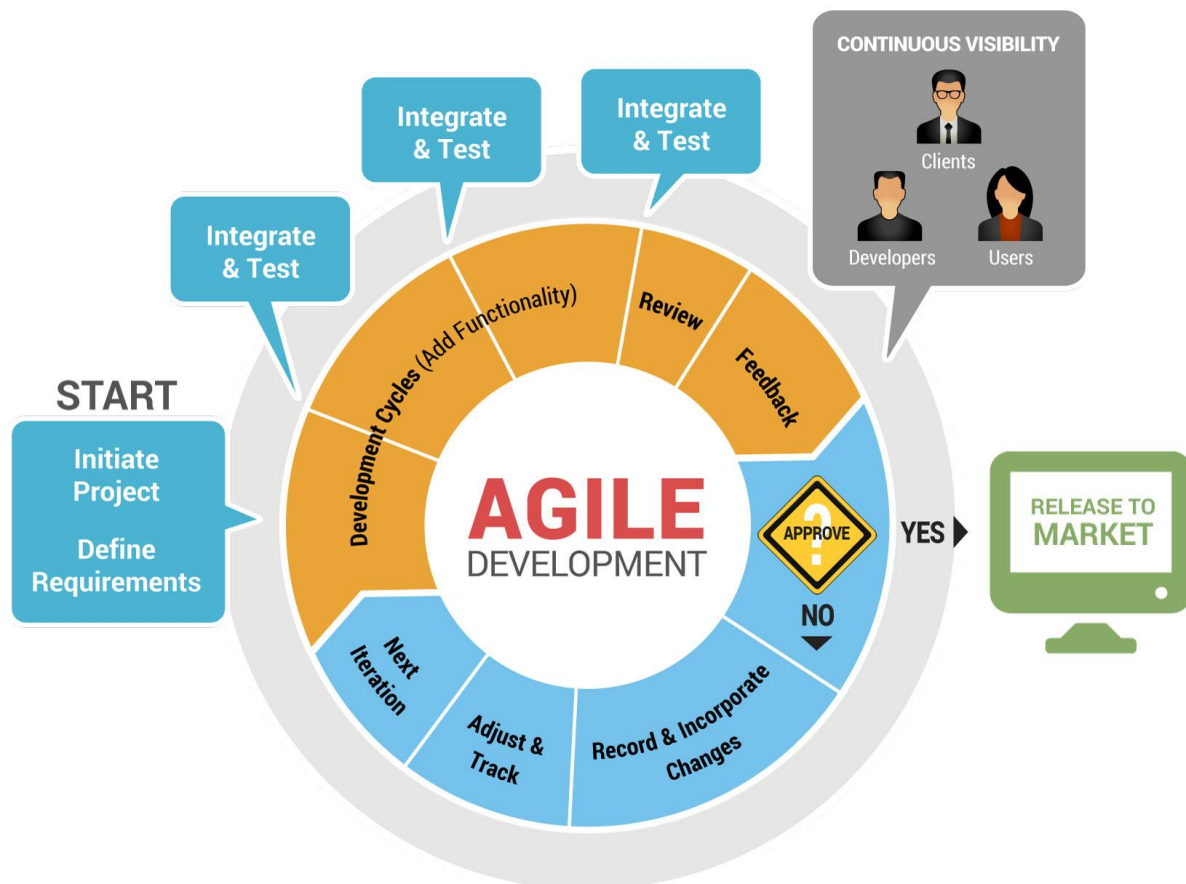
Software Engineering Prozess

Plangetrieben vs. Agile Entwicklung

| Faktoren | Plangetrieben | Agil |
|--|---------------|------|
| Es handelt sich um einen staatlichen Auftrag (Es wird viel Dokumentation erwartet). | X | |
| Der exakte Umfang ist nicht definiert. | | X |
| Die Zeit ist knapp bemessen. | | X |
| Das med. Personal hat wenig Erfahrung mit SE und kann daher das GUI nicht zu Projektbeginn abschliessend definieren. | | X |
| Das Entwicklungsteam hat wenig Erfahrung in der fachlichen Domäne. | | X |
| Zeitaufwand kann nicht genau geschätzt werden. | | X |
| Der Zugang zu den Endbenutzer ist erschwert. Diese haben wenig Zeit um am Entwicklungsprozess teilzunehmen. | X | |

Aufgrund der Ergebnisse aus der Analyse hat das Projektteam sich dazu entschieden die agile Softwareentwicklung zu verwenden. Die vielen Vorteile der agilen Entwicklung überwiegen klar die Nachteile einer plangetriebenen Entwicklung.

Prozessmodell



Aktivitäten

Projektstart

- Benutzeransprüche bestimmen
- Gesetzliche Anforderungen festlegen
- Verfügbare Ressourcen bestimmen
- Zusammenarbeit im Entwicklerteam festlegen
- Ansprechpartner beim Kunden definieren

Sprint 1

- Benutzeroberfläche realisieren
- Akzeptanz der Benutzeroberfläche prüfen

Sprint 2

- Patientenverwaltung realisieren

Sprint 3

- Terminverwaltung realisieren

Sprint 4

- Statistikmodul realisieren

Sprint 4

- Optionale Anforderungen umsetzen

Beziehungen zwischen Outputs

Benutzeroberfläche

Die Benutzeroberfläche ist abhängig von folgenden Komponenten:

- Middleware (Patientenverwaltung, Terminverwaltung)
- Datenbank

Middleware

Die Middleware ist abhängig von der Datenbank.

Datenbank

Die Datenbank hat keine technische Abhängigkeit von anderen Komponenten.

Interaktionen mit den Anspruchsgruppen

Auftraggeber:

- Einführung in die agile Softwareentwicklung
- Kostenrahmen des Projekts klären (wie wird das Projekt verrechnet)
- Informationssicherheit und Datenschutz Anforderungen bestimmen.
- Weitere Bedingungen bestimmen (muss das Projekt Open Source sein usw.)
- Regelmässige Information der Verantwortlichen auf Seite Auftraggeber

Benutzer:

- Informieren der Benutzer über das Projekt
- Auswahl der Testgruppe
- Release Zyklus bestimmen
- Anforderungen bestimmen (GUI, nächstes Ziel usw.)
- Zwischenversionen ausliefern
- Feedback einholen
- Super User ausbilden

Management:

- Aufwands- /Ertragsrechnung
- Regelmässige Reports über den Projektverlauf
- Terminplanung
- Ressourcenplanung