

AUFGABE 1:

A.)Wasserfallmodell

- 1) Beim Wasserfallmodell führt man eine Softwarevalidierung durch, bei der durch mehrere Tests (Komponententest, Integrationstest, Systemtest, Abnahmetest) sichergestellt wird, dass die Spezifikationen der des Kunden entspricht. Hierbei kann es jedoch auch zu einer Rückkopplung von einer Phase zur vorheriger kommen.
- 2) Jede Phase wird in der gesamten Breite durchgeführt und danach gibt es einen sequentiellen Durchlauf der einzelnen Schritte.
- 3) Eine frühzeitige Rückmeldung vom Kunden ist wichtig. Nach jedem fertigen Schritt kann der Kunde durch Absprache mit dem Team Einfluss ausüben. Ein Beurteilen des Entwurfes könnte dem Kunden eher schwer fallen.
- 4) Wird das Wasserfallmodell streng sequentiell durchgeführt, kann durch Meilensteine der Fortschritt gut anhand des Erreichen eines Meilensteins gemessen werden.
- 5) Das Wasserfallmodell ist nicht für Großprojekte geeignet.
- 6) Für jede Phase wird eine Dokumentation erzeugt, die eventuell später noch abgeändert werden müssen.
- 7) Da es nicht für Großprojekte ausgelegt ist, ist es nicht für sehr komplexe Projekte geeignet.
- 8) Die iterative Vorgehensweise wird beim Wasserfallmodell nicht unterstützt. Wird jedoch bei der Softwarevalidierung benutzt.
- 9) Eine inkrementelle Entwicklung ist beim Wasserfallmodell vorgesehen.
- 10) Bei der Softwarearchitektur handelt es sich um einen Teilprozess, der in der zweiten Phase definiert und ausgearbeitet wird.

B.)Rational Unified Process

- 1) Die Qualitätssicherung spielt beim RUP-Modell eine hohe Rolle, denn hier werden Qualitätsmerkmale definiert, die vom Softwaresystem erreicht werden sollen. Hierfür gibt es eine gesonderte Phase. Für die Teststrategien gibt es einen eigenen Punkt, der einen Arbeitsablauf darstellt. Hierbei werden die Ergebnisse der Implementierung überprüft.
- 2) Das Projektteam stimmt sich durch kleine Releases ab.
- 3) Der Kunde hat keine Auswirkung auf die Entwicklungsphase. Die Auslieferung eines Produkts erfolgt am Ende eines Zyklus.
- 4) Nach jedem Release wird eine Iteration durchgeführt, die ein mess- und bewertbares Ergebnis liefern.
- 5) Kann für alle Projektgrößen verwendet werden, aber bietet sich erst für Projekte mit einer gewissen Größe an (z. B. für Projekte mit sehr vielen Beteiligten).
- 6)
- 7) Da das Modell für große Projekte vorgesehen ist, ist davon auszugehen, dass es auch für komplexere Projekte geeignet ist.
- 8) Eine iterative Vorgehensweise wird beim RUP-Modell unterstützt. Die Phasen sind durch die einzelnen Releases abgegrenzt.
- 9) Beim RUP-Modell ist eine inkrementelle Entwicklung vorgesehen.
- 10) Bei der Entwurf-Phase wird ein Architekturrahmen festgelegt, der jedoch durch eine Architektur-Iteration überarbeitet werden kann.

C.)Extreme Programming

- 1) Beim Extreme Programming gehören Testfunktionalität und -automatisierung zu den Grundprinzipien. Über die Tests wird eine Qualitätssicherung und eine Einbeziehung des Kunden gewährleistet.
- 2) Durch eine direkte Kommunikation zwischen allen Projektbeteiligten, mit dem Auftraggeber und Kunden wird sich abgestimmt.
- 3) Der Kunde kann permanent Einfluss ausüben, da dieser immer in einer direkten Kommunikation zum Team steht.
- 4) Durch kleine Releases und regelmäßige Tests wird eine Fortschrittskontrolle im Team gewährleistet.
- 5) Das Modell ist eher für große Projekte geeignet, da sich das gesamte Projekt in kleine Releases unterteilen lässt. Dies wäre bei kleineren Projekten nicht von Nöten.
- 6) Jeder muss sein Teil dokumentieren.
- 7) Das Modell ist eher komplex, da jeder mit jedem in einer direkten Kommunikation steht.
- 8) Es wird eine iterative Vorgehensweise durch die kleinen Releases unterstützt.
- 9) Eine inkrementelle Vorgehensweise ist vorgesehen, da es zu den Grundprinzipien gehört.
- 10) Die Software-Architektur entsteht während der Entwicklung und spielt dadurch keine hervorgehobene Rolle.

D.)Scrum

- 1) Durch Sprints und regelmäßige Meetings werden Probleme und Fortschritte behandelt/besprochen. Die Schnittstelle an sich wird durch die Absprache mit dem Kunden verbessert.
- 2) Bei Scrum werden täglich kurze Treffen vereinbart, bei denen man sich austauscht.
- 3) Der Kunde hat die Rolle des Stakeholders inne und steht mit dem Product Owner oder Scrum Master im engen Kontakt.
- 4) Sprint-Ziele dienen als Fortschrittskontrolle, die durch einen Produkt-Backlog vorgegeben werden. Zusätzlich zu den großen vier wöchentlichen Sprint-Meetings gibt es kleinere, tägliche Meetings.
- 5) Scrum eignet sich für alle Projektgrößen, da es individuell anpassbar ist.
- 6) Die Dokumentation wird von dafür zuständigen Mitarbeitern verfasst oder vom Scrum Master übernommen.
- 7) Scrum ermöglicht durch das breite Auffächern eines Projekts in Produkt-Backlogs und Sprints eine hohe Komplexität. Diese hat trotzdem eine klare Struktur.
- 8) Durch Sprints und täglich kleine Treffen ist ein iteratives Vorgehen möglich.
- 9) Das System wird hier auf einzelne Sprints angewendet.
- 10) Im Scrum spielt die Architektur auch eine große Rolle, obwohl sie vom Entwicklerteam regelmäßig überprüft, angepasst und weiterentwickelt wird.

AUFGABE 2:

| Nr. | Stärke | Nützlichkeit | mögliche Schwachpunkte |
|-----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | schnelle Auslieferung | Feste Ziele | Hoher Druck durch ständige Auslieferung |
| 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | Perfekte Software | Fehler die man selbst übersehen/ nicht wahrgenommen hätte | Es können größere Verzögerungen zustande kommen und der Zeitplan kann nicht eingehalten werden |
| 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. | Schnelleres Vorankommen | Durch die Auslieferung von neuen Versionen o.ä. bleibt die Software modern und im Gedächtnis des Kunden schon durch das öftere sehen | Schlechtere Qualität durch schnelleres Abarbeiten |
| 4. Business people and developers must work together daily throughout the project. | Abstimmung zwischen den Parteien ist besser | Entwickler können gezielter am Projekt arbeiten | Abstimmungen nehmen zu viel Zeit in Kauf |
| 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | Mehr Zeit für andere Aufgaben | Die Mitarbeiter sind motivierter und dadurch steigt die Qualität ihrer Arbeit | Unmotivierte Menschen können das Projekt in die Länge ziehen. |
| 6. The most efficient and effective method of conveying information to and within a development | Es können keine Missverständnisse durch fehlerhafte Mails auftreten | Durch Gespräche werden Probleme auf den Punkt gebracht ohne Mails etc. mehrmals hin- und herschicken | Es kann dazu kommen, dass man beim Gespräch vom Wesentlichen abkommt. |

| | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| team is face-to-face conversation. | | | |
| 7. Working software is the primary measure of progress. | Der Erfolg ist dadurch messbar | Ziele und Erfolge lassen sich genau visualisieren | Bei abstrakter/komplexerer Software wird das System schwieriger. |
| 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. | Das Team und der Kunde sind auf einer Augenhöhe. Mitarbeiter müssen nicht Überstunden machen, weil die Entwicklung kontinuierlich voranschreitet. | Alle Beteiligten sind immer auf dem neusten Stand. Des Weiteren kommt es nicht zu unnötigen Verzögerungen | Das System ist schwer einzuhalten und alle stände mit allen Beteiligten abzustimmen. Es können Probleme auftreten, falls Mitarbeiter ausfallen oder Termine verschoben werden müssen. |
| 9. Continuous attention to technical excellence and good design enhances agility. | Ein schönes Design kommt beim Kunden immer besser an | Der Mitarbeiter ist motivierter an einem guten Produkt zu arbeiten, das auch Spaß macht. | Das Wissen hierfür anzueignen kann Zeit kosten. |
| 10. Simplicity--the art of maximizing the amount of work not done--is essential. | Je einfacher eine Aufgabe ist, desto mehr Menschen können am Projekt arbeiten und desto schneller wird es fertig | Die unterschiedlichen Aufgaben können von allen bearbeitet werden und jeder weiß sofort, was zu tun ist. | Die Motivation kann sinken, da die Aufgaben einen nicht mehr fordern. |
| 11. The best architectures, requirements, and designs emerge from self-organizing teams. | Konzentration auf die Software | Die Gruppe kann ihren Fokus selbst bestimmen. | Schlechte Kommunikation kann zu Missverständnissen und ungewollten Fehlern in der Software führen. |
| 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. | Durch Selbstreflexion und Möglichkeiten das Projekt umzugestalten, falls Das Projekt in eine falsche Richtung läuft | Die Teams kommunizieren mehr und der Fortschritt/ die Fehler können schneller entdeckt werden | Nur möglich, wenn genug Diskussionsthemen vorhanden sind. Kann auch zu Spannungen führen, falls sich Teammitglieder nicht vertragen. |