

Kompetenzerreichung

Handlungsziel 1 - Scrum

Beschreibung

Dieses Dokument dient zur Anleitung und Führung zu der Erarbeitung der Kompetenzgruppe 1.1 – 1.4, sowie als Einführung ins Agile Manifesto.

Handlungsziel 1	Vorgegebene Funktionalität im Rahmen eines Softwareprojekts mit einer agilen Methode umsetzen.
Kompetenzen	1.1 – 1.4

Inhaltsverzeichnis

1 KOMPETENZERREICHUNG	2
1.1 HANDLUNGSZIEL 1	2
1.2 KOMPETENZEN IM HANDLUNGSZIEL	2
1.3 AUFTRÄGE ZU DEN KOMPETENZEN 1.1 BIS 1.4	2
2 AGILES MANIFESTO	3
2.1 4 LEITSÄTZE DES AGILEN MANIFESTOS	3
2.1.1 1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge	3
2.1.2 2. Funktionierende Software ist wichtiger als umfassende Dokumentation	4
2.1.3 3. Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlung	4
2.1.4 4. Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans	5
2.2 12 PRINZIPIEN HINTER DEM AGILEN MANIFEST	6
3 AUFGABEN	8
3.1 FRAGEN FORMULIEREN	8

1 Kompetenzereichung

1.1 Handlungsziel 1

Vorgegebene Funktionalität im Rahmen eines Softwareprojekts mit einer agilen Methode umsetzen.

1.2 Kompetenzen im Handlungsziel

- 1.1 - Ich kann den Inhalt der vier Leitsätze und der zwölf Prinzipien des agilen Manifests sinngemäss wiedergeben.
- 1.2 - Ich kann aufzeigen, wie und wo das agile Manifest mit seinen vier Leitsätze und der zwölf Prinzipien im praktischen Einsatz von Scrum zum Ausdruck kommt.
- 1.3 - Ich kann erläutern wie Scrum, dessen Artefakte (Product Backlog, Sprint Backlog, User-Story, Task), Ereignisse (Pre-Planing (Grooming), Sprint Planning, Sprint, Daily Scrum, Review, Retrospektive) und Rollen (Product Owner, Scrum Master, Team Member) als agile Methode funktionieren.
- 1.4 - Ich kann die Aufgaben der Rollen von Scrum und von anderen Rollen in Softwareprojekten (Produktmanager, Projektleiter, Architekt, Softwareentwickler, Softwaretester, Systemtechniker) differenzieren.
- 1.5 - Ich kenne den Sinn und Aufbau von User-Stories und kann diese korrekt formulieren (Ziel, Pre-Condition, Aufgabe, Akzeptanz Kriterium).
- 1.6 - Ich kenne den Unterschied zwischen Tasks und User-Story und kann Tasks für User-Stories aussagekräftig formulieren.
- 1.7 - Ich kann mit mindestens zwei Methoden den Aufwand für eine Story im Team abschätzen. (Storypoint, Planning Poker, Verfeiner, Expertenschätzung, Analogieschlüsse, Zwei/Dreipunkte)
- 1.8 - Ich kann Tasks aus dem Sprint-Backlog aktiv bearbeiten. (Umsetzen, Definition of Done entwerfen, erweitern und anwenden, Status im Board nachführen).

1.3 Aufträge zu den Kompetenzen 1.1 bis 1.4

01_1_EinführungInScrum.pdf

Lesen Sie diese kurze Zusammenfassung. Sie beschreibt die Kernelemente vom Scrum-Entwicklungsprozess.

01_2_AufgabeScrumFilm.pdf

Dieses Dokument beinhaltet drei Aufgaben. Schauen Sie sich allenfalls den Film nochmals genauer an und machen Sie sich Notizen. Lesen Sie die erste Aufgabe und erstellen Sie sich eine Zusammenstellung der Begriffe, wie gefordert.

01_3_ScrumMemoryBilder

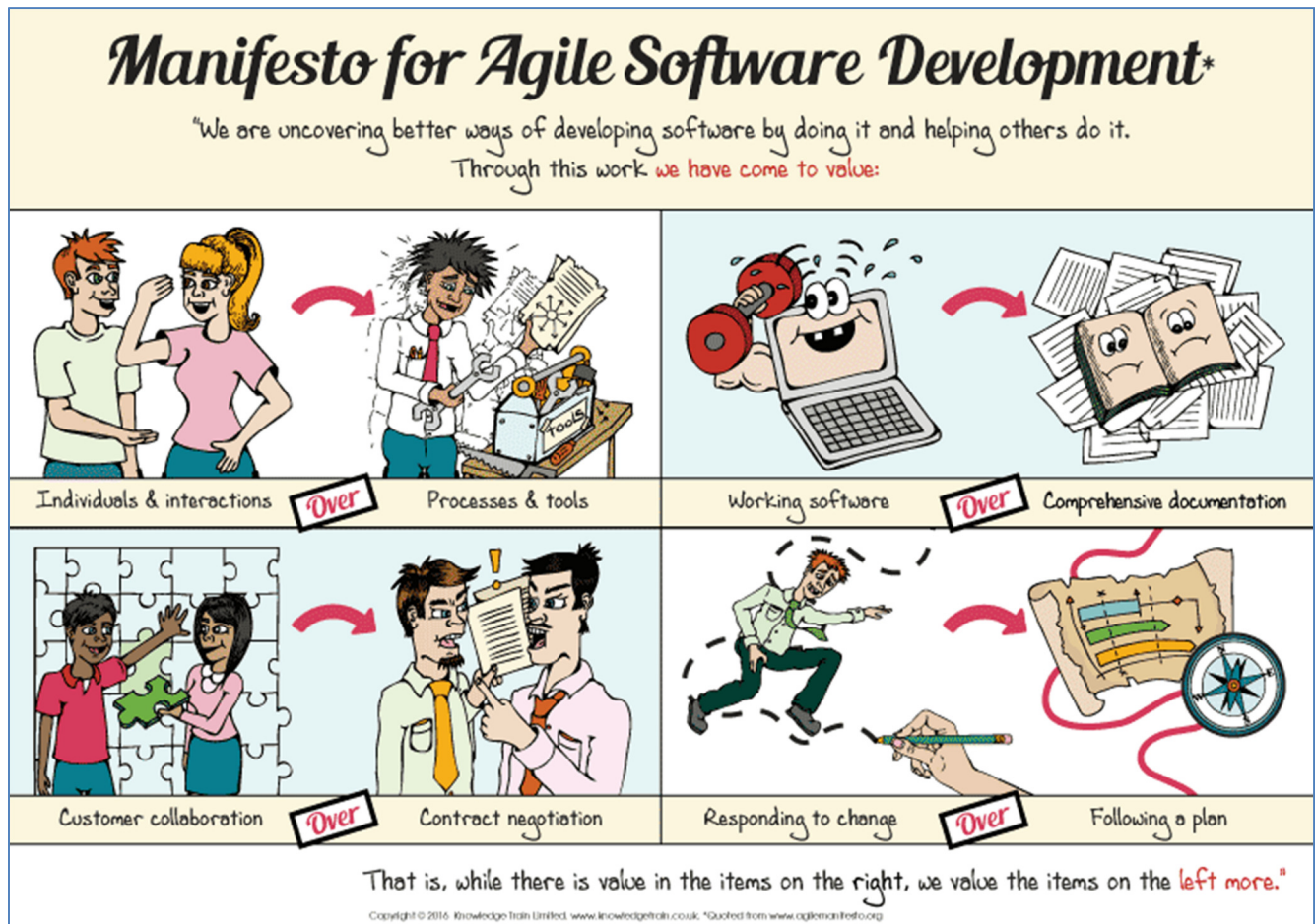
Ordnen Sie die Texte den Bildern zu. Versuchen Sie die Beschreibungen zu verstehen und herauszufinden um welchen Begriff, Rolle, etc. es sich handelt

01_3_ScrumMemoryBilder Liste leer.docx

Agiles Manifesto

Lesen Sie das Agile Manifest durch und ergänzen Sie Ihre Zusammenstellung.

2 Agiles Manifesto



2.1 4 Leitsätze des Agilen Manifestos

Die vier Leitsätze teilen sich jeweils in zwei, bzw. genau genommen 3 Teile auf, einen vorderen, einen mittleren und einen hinteren Teil. Die äusseren Teile enthalten dabei die eigentlichen Aussagen, der mittlere Teil stellt eine Bewertung dar, bzw. verbindet sie sprachlich.

Quelle: <http://doktor-scrum.de/agiles-manifest-4leitsaetze>

1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge
2. Funktionierende Software ist wichtiger als umfassende Dokumentation
3. Zusammenarbeit mit dem Auftraggeber ist wichtiger als Vertragsverhandlung
4. Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans

2.1.1 1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge

Was ist damit gemeint?

Damit wird ausgedrückt, dass die an einem Vorgang oder Projekt beteiligten Personen, ihr Umgang und Austausch, ihre Kommunikation und ihre Zusammenarbeit einen grösseren Einfluss auf die erfolgreiche Durchführung, bzw. den erfolgreichen Abschluss des Projektes haben und damit wichtiger sind, als die Wahl der eingesetzten Prozesse und Werkzeuge, zumal die Prozesse letztlich auch nur Werkzeuge sind.

Dieser Leitsatz stellt die am Projekt beteiligten Personen in den Vordergrund und grenzt die agile Vorgehensweise gegenüber den klassischen Vorgehensweisen — die mit ihren eng gefassten und kleinteilig definierten Handlungsspielräumen den Prozess als solchen in den Vordergrund stellen — ab.

Die agilen Vorgehensweisen zielen darauf, den beteiligten Personen den notwendigen Freiraum zu schaffen, damit sie ihre kreativen und handwerklichen Potentiale und Kompetenzen zum Vorteil des jeweiligen Vorganges voll ausschöpfen können, wohingegen die klassischen Vorgehensweisen ein enges Korsett abzuarbeitender Handlungsanweisungen vorgeben. Letzteres funktioniert jedoch nur dann gut, wenn im Plan bereits alle eventuell auftretenden Zustände bedacht und dokumentiert worden sind, was, realistische betrachtet, insbesondere bei einem Softwareentwicklungsprojekt, so gut wie unmöglich ist.

Zudem sind es letztlich immer die Individuen und ihr gemeinsames, permanent aufeinander fein abgestimmtes Handeln, die den Vorgang oder das Projekt vorantreiben und nicht die Präsenz von Prozessen oder Werkzeugen; es sind immer die Individuen, die die Werkzeuge einsetzen und nicht umgekehrt.

Anders gesagt, sind die beteiligten Individuen und ihre Fähigkeiten für die Durchführung eines Vorganges oder Projektes essenziell, die zur Verfügung stehenden Werkzeuge und Prozesse sind es nicht in jedem Fall.

Was ist damit nicht gemeint?

Damit wird nicht gesagt, dass Werkzeuge unwichtig wären, im Gegenteil; jedoch ist ihre Bedeutung geringer als die der Individuen, die sie einsetzen, zumal die Werkzeuge aus sich heraus ohne die Individuen, die sie einsetzen oder bedienen, nichts leisten können; die Individuen können das ohne die Werkzeuge schon, wenn auch wahrscheinlich viel langsamer.

2.1.2 2. Funktionierende Software ist wichtiger als umfassende Dokumentation

Was ist damit gemeint?

Das Ziel von Softwareentwicklung ist in erster Linie, lauffähige, einsetzbare Softwareprodukte zu entwickeln und erst in zweiter Linie, sie zu beschreiben oder zu dokumentieren. Eine Dokumentation ist ohne Frage sehr wichtig, aber wenn sich der Auftraggeber zwischen dem Produkt und der Dokumentation entscheiden müsste, würde er sich ohne Zweifel für die für ihn hergestellte Software entscheiden.

Soweit sollte man es sicherlich nicht kommen lassen, aber wenn eine Entscheidung zwischen der Fertigstellung der Software und der Fertigstellung der Dokumentation zu treffen ist — so drückt es dieser Leitsatz aus — ist es für den Auftraggeber von weitaus höherem Wert, das bestellte Softwareprodukt in einem für ihn einsetzbaren Zustand zu erhalten und einsetzen zu können, als die Auslieferung des Produktes von der lückenlosen Fertigstellung der Dokumentation abhängig sein zu lassen.

Was ist damit nicht gemeint?

Dass soll nicht bedeuten, dass die Dokumentation überflüssig wäre; ihre Bedeutung ist ohne Frage sehr hoch, aber im Zweifel geringer einzuschätzen, als die Bedeutung der lauffähigen, einsetzbaren Software, die in ihr beschrieben wird.

2.1.3 3. Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlung

Was ist damit gemeint?

Software zu entwickeln ist zumeist ein sehr teures Unterfangen und man tut gut daran, ein solches Projekt mit einem für alle Seiten akzeptablen Vertrag zu regeln. Diese Vertragsverhandlungen können sehr schwierig und zeitaufwändig sein, insbesondere dann, wenn es sich für die beteiligten Parteien um grosse Projekte handelt.

Der Leitsatz rückt die Zusammenarbeit zwischen dem ausführenden Unternehmen, bzw. den Entwicklern und dem Auftraggeber, in den Mittelpunkt des Projektes. Statt sich schon im Vorwege in Vertragsverhandlungen zu

verlieren, sollte man statt dessen frühzeitig prüfen, ob nicht bereits mit der Arbeit begonnen werden kann, auch wenn der Vertrag noch nicht bis ins letzte ausformuliert ist oder sich noch in der Bearbeitung befindet.

Selbstverständlich können sich auch im Verlauf des Projektes Änderungen ergeben, die dazu führen, dass Teile des Vertrages neu verhandelt und angepasst werden müssen. Hier mahnt der Leitsatz an, dass sich daraus keine Unterbrechungen des Projektverlaufes ergeben sollten, sondern dass das Projekt parallel zu den Verhandlungen unterbrechungsfrei fortgeführt wird.

Was ist damit nicht gemeint?

Damit ist nicht gemeint, dass man keine Verträge machen sollte, oder dass Verträge unwichtig wären, sondern nur, dass einer konstruktiven Zusammenarbeit mit dem Auftraggeber mehr Raum und Zeit als den Vertragsverhandlungen eingeräumt werden sollte. Es bedeutet weiterhin, dass der Fokus des Projektes im Anschluss an die Vertragsverhandlungen nicht allein darauf gesetzt werden soll, ausschliesslich den geschlossenen Vertrag zu erfüllen, sondern dass die Maximierung des Nutzens, den der Auftraggeber aus dem für ihn erstellten Produkt ziehen kann, wichtiger sein sollte als das, was ursprünglich im Vertrag festgehalten wurde.

Verträge kann man relativ leicht ändern, verlorenegegangene Zeit aber nicht zurückholen.

2.1.4 4. Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans

Was ist damit gemeint?

Mangelnde Flexibilität ist einer der wesentlichen Gründe für das Scheitern vieler (Softwareentwicklungs-) Projekte. Ansichten, Erfahrungen, Meinungen, Bedürfnisse und Anforderungen können sich gerade in der Softwareentwicklung sehr schnell ändern; was gestern noch sehr wichtig und „alternativlos“ war, kann heute schon wieder aus guten Gründen obsolet sein.

Ein Plan stellt eine Richtschnur dar, an der man sich entlang arbeitet, um am Ende das im Plan festgelegte Ziel zu erreichen. Ändern sich im Laufe der Umsetzung jedoch die Anforderungen und/oder die Voraussetzungen der ursprünglichen Planung, ist es nicht unwahrscheinlich, dass diese Änderungen sich auch in einer Aktualisierung des Zieles niederschlagen.

Verfolgt man statt dessen jedoch den (zu diesem Zeitpunkt wahrscheinlich bereits veralteten) Plan weiter, ohne die Änderungen zu berücksichtigen, erhält der Auftraggeber bei der Ankunft am Ziel mit sehr hoher Wahrscheinlichkeit ein analog zum Plan zu diesem Zeitpunkt wahrscheinlich bereits veraltetes Projektergebnis, dass dann im Nachhinein oftmals nur noch mit grossem Aufwand (Zeit UND Geld) auf den neuesten Stand gebracht werden kann.

Dieser Leitsatz fordert daher die beteiligten Personen auf, offen für Veränderungen des Plans zu sein, um auf eventuell veränderte Rahmenbedingungen schnell und flexibel reagieren zu können und sie nicht erst nach vollständiger Abarbeitung des (in diesem Fall bereits veralteten) Planes zu berücksichtigen.

Was ist damit nicht gemeint?

Damit ist — wie den agilen Prozessen gerne immer wieder unterstellt wird — auf gar keinen Fall gemeint, dass man keine Pläne machen sollte (Zitat: „in agilen Prozessen wird nicht geplant“), vielleicht z.B., weil sie in dem Moment, wo sie aufgestellt werden, potentiell schon wieder veraltet sein könnten. Pläne und Planung sind (auch und gerade in den agilen Prozessen) von essentieller Bedeutung, weil sie dem Team und dem Auftraggeber ein gemeinsames Bild davon geben können, in welche Richtung zu gehen ist.

Gemeint ist lediglich, dass es wichtiger ist, flexibel auf Veränderungen zu reagieren, statt sie zu ignorieren oder nicht anzunehmen und auf das Einhalten eines zuvor erstellten und potentiell veralteten Planes zu bestehen.

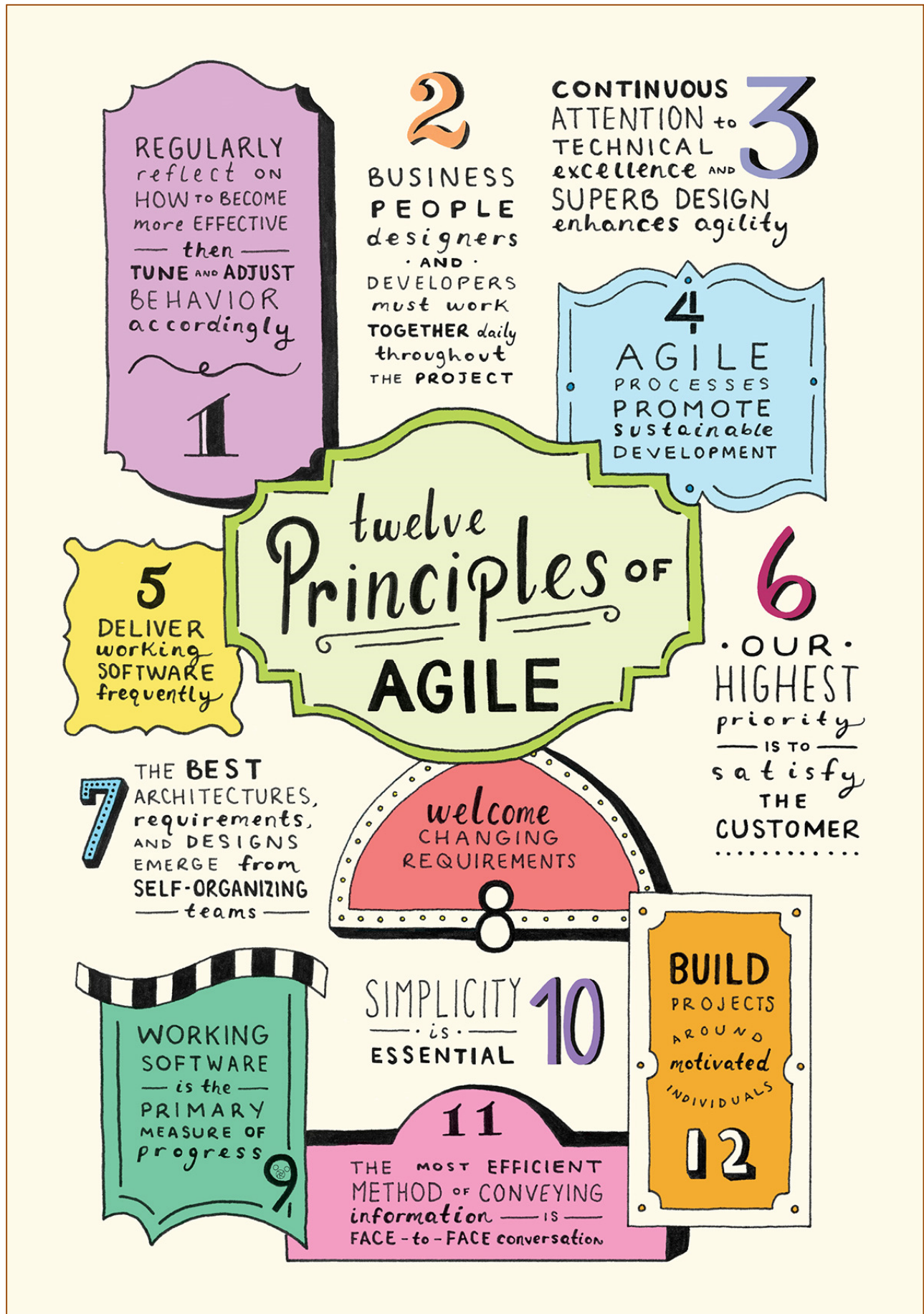
Gemeint ist damit natürlich auch nicht, dass das Prinzip der Veränderung des Planes zum Kernprinzip agiler Softwareentwicklung als Selbstzweck wird, also quasi am Vortag der Endabgabe die Anforderungen ins Gegenteil zu verkehren, um dem Prinzip der Agilität maximal zu entsprechen, gleichzeitig aber auf die Einhaltung des Abgabetermins zu pochen ...

2.2 12 Prinzipien hinter dem Agilen Manifest

Quelle: <http://www.think-pi.de/agile%20manifesto>

Die „Agile Alliance“, siebzehn unabhängige ‚leichtgewichtige‘ Software-Methodologen, trafen sich im Februar 2001 um miteinander zu reden und eine gemeinsame Basis zu finden. Sie einigte sich auf ein Manifest für die Agile Softwareentwicklung. Des Weiteren definierten sie eine Reihe von zwölf Prinzipien hinter dem Manifest, die hier in voller Länge wiedergegeben sind:

1. Den Kunden durch die rechtzeitige und kontinuierliche Lieferung wertvoller Software zufrieden zu stellen ist unsere höchste Zielsetzung.
2. Wir begrüßen Änderungen der Anforderungen, auch wenn diese spät in der Entwicklung kommen. Agile Prozesse nutzen den Wandel für den Wettbewerbsvorteil des Kunden.
3. Wir liefern funktionierende Software häufig, alle paar Wochen oder Monate, wobei wir die kürzeren Zeiträume bevorzugen.
4. Geschäftsleute und Entwickler müssen täglich im Projekt zusammen arbeiten.
5. Setze motivierte Individuen in den Projekten ein. Gib ihnen die Umgebung und Unterstützung, die sie brauchen, und vertraue ihnen den Job bestens zu erledigen.
6. Die Konversation von Angesicht zu Angesicht ist die effizienteste und effektivste Art der Informationsweitergabe an und in einem Entwicklungsteam.
7. Das primäre Mass für Fortschritt ist die funktionierende Software!
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Sponsoren, Entwickler und Benutzer sollten ein gleichbleibendes Tempo ohne Unterbrechung einhalten.
9. Technische Exzellenz, gutes Design und deren fortwährende Beachtung verbessern die Agilität.
10. Einfachheit ist essentiell.
11. Die besten Architekturen, Anforderungen und Designs entstehen in Teams, die sich selbst organisieren.
12. Das Team reflektiert in regelmässigen Intervallen darüber, wie es effektiver werden kann und passt sein Verhalten entsprechend an.



3 Aufgaben

3.1 Fragen formulieren

Formulieren Sie in Ihrer Gruppe fünf Fragen für die Ihnen zugewiesene Handlungskompetenz. Schreiben Sie die Fragen mit den Lösungen im Word bzw. in einer PDF Datei und legen Sie diese in den entsprechenden Ordner auf Olat.

(Die Lehrperson wird für Sie einen Online-Fragebogen zusammenstellen)

3.2 Nutzen Sie die folgenden Fragetypen:

Multiple Choice (eine aus n)

Welche Antwort trifft zu?

- ☐ Antwort 1
- ☐ Antwort 2
- ☒ Antwort 3
- ☐ Antwort 4

Mehrfachauswahl

Welche Antworten treffen zu?

- ☐ Antwort 1
- ☒ Antwort 2
- ☒ Antwort 3
- ☐ Antwort 4

Zuordnungsfragen

Ordnen Sie die Pflanzen ihrer jeweiligen Pflanzenfamilie zu!

Frühjahrsblüher

Kaktus

Rosengewächs

Apfelbaum

Nachtschattengewächs

Osterglocke

Tomate

Richtig oder falsch Fragen

Für jede Aussage muss entschieden werden: [richtig] oder [falsch]

richtig

falsch

- | | | |
|-----------------------|-----------------------|-----------|
| <input type="radio"/> | <input type="radio"/> | Aussage 1 |
| <input type="radio"/> | <input type="radio"/> | Aussage 2 |
| <input type="radio"/> | <input type="radio"/> | Aussage 3 |
| <input type="radio"/> | <input type="radio"/> | Aussage 4 |