

Software Engineering - RWU

Prof. Dr. R. Schimkat

Wintersemester 2020/21

Beschreibung der Aufgabe für den benoteten Leistungsnachweis in Software Engineering im WS 2020/21.

Inhaltsverzeichnis

1	<i>Aufgaben zum benoteten Leistungsnachweis</i>	1
2	<i>Überblick</i>	1
3	<i>Themenliste</i>	2
4	<i>Lernvideos</i>	3
4.1	<i>Bewertungskriterien</i>	3
5	<i>Dokumentation</i>	4
5.1	<i>Bewertungskriterien</i>	4
6	<i>Lernfragen</i>	4
6.1	<i>Beispiel</i>	5
6.2	<i>Bewertungskriterien</i>	6
6.3	<i>Hinweise</i>	6

1 Aufgaben zum benoteten Leistungsnachweis

2 Überblick

Im Rahmen des benoteten Leistungsnachweises im Modul *Software Engineering* erstellen Sie für eine ausgewählte Fragestellung im Themenbereich des Moduls ein Lernvideo, ein Textdokument und Lernfragen. Die Aufgabe bearbeiten Sie in einem Team von 4 Personen.

- **Lernvideo** - Sie erstellen ein 6-10 minütiges Lernvideo (mp4), in dem Sie die Fragestellung näher erläutern, die Relevanz im ausgewählten Themenbereich illustrieren, gegebenenfalls Lösungen dafür darstellen und letztlich kritisch würdigen.
- **Dokument in Markdown** - Sie beschreiben in einem Textdokument (Markdown), wie Sie im Team vorgegangen sind, welche Verantwortlichkeiten jedes Teammitglied hatte und machen Angaben zur Literatur und Quellen, die Sie für die Bearbeitung der Aufgabe herangezogen haben.

- **Lernfragen** - Sie erstellen 4 Lernfragen a la Multiple Choice mit unterschiedlichen Schwierigkeitsgraden passend zum ausgewählten Themenbereich.
- **Teamgröße** - Sie finden sich in einem Team von 4 Personen zusammen und arbeiten kontinuierlich an den ausgewählten Fragestellungen.

3 Themenliste

Aus der folgenden Liste wählen Sie zu Beginn des Semesters ein Thema aus und stimmen sich mit dem Dozierenden ab.

Bereich - Vorgehensmodelle

1. Wasserfallmodell - Phasen - Was sind die grundlegenden Phasen in der Softwareentwicklung; insbesondere im Wasserfallmodell?
2. Agile - Agiles Manifest - Geben Sie einen Überblick über das agile Manifest!
3. Scrum - Rollen. Scrum Master. Wie sieht ein typischer Arbeitstag eines Scrum Masters aus?
4. Scrum - Rollen. Scrum Master. "In einem Unternehmen der Region ist der dortige Abteilungsleiter (im Bereich Entwicklung) auch Scrum Master für seine beiden Entwicklungsteams." Nehmen Sie kritisch Stellung dazu!
5. Scrum - Rollen. Scrum Team. Welche Entwicklungspraktiken sind für ein Scrum Team besonders relevant? Stellen Sie 2 näher vor.
6. Scrum - Rollen. Product Owner. Welche Eigenschaften sollte ein Product Owner typischerweise besitzen?
7. Agile - Benutzergeschichten. Was sind Benutzergeschichten und warum erzählt man Geschichten in der Softwareentwicklung überhaupt?
8. Agile - Story Mapping. Warum kann das Story Mapping für die Verwaltung von Anforderungen bzw. Spezifikationen wichtig sein?
9. Scrum - Meetings. Sprint Planning 1 und 2. Beschreiben Sie detailliert einen möglichen Ablauf dieser beiden Meetings zu Beginn eines neuen Sprints!
10. Scrum - Meetings. Sprint Retrospective. "Kaffepause für die Entwickler - weit gefehlt! Hier wird gelernt!" - Nehmen Sie Stellung dazu!
11. Scrum - Meetings. Sprint Review und Show. "Wo liegen die Unterschiede zwischen einem Review und einer Show - wir reden doch von demselben Meeting bei Scrum?" - Nehmen Sie Stellung dazu!
12. Scrum - Artefakte. Burndown Chart. Erläutern Sie das Konzept und Werkzeug des Burndown Charts an verschiedenen plakativen und beispielhaften Verläufen!
13. Scrum - Artefakte. Product Backlog. "Im Product Backlog spiegelt sich das komplette Softwareprodukt wider - das jetzige und das künftige

Produkt" - Erläutern Sie, was damit gemeint sein könnte!

14. Scrum - Artefakte. Sprint Backlog und Tasks. Erläutern Sie was man unter einem Sprint Backlog und Entwickler Aufgaben (tasks) in Scrum genau versteht!

Bereich - Werkzeuge

15. Versionsverwaltung mit git - Wie unterstützt das Werkzeug *git* eine verteilte Softwareentwicklung?
16. Buildmanagement mit maven - Wie können wir große - sehr große - Softwaresysteme bauen und pflegen?
17. Kontinuierliche Integration mit jenkins - "Den aktuellen (wirklichen) Gesundheitszustand unseres Softwareprodukts zu kennen wirkt beruhigend." - Nehmen Sie Stellung dazu.

Bereich Testing

18. Unit Testing - "Qualitäten zu sichern ist zwingend notwendig - aber wie soll das in der Entwicklung funktionieren?" (am Beispiel von Unit Testing).
19. Mock Testing - "Warum setzt man in der Softwareentwicklung auf Attrappen?"
20. Test-getriebene Entwicklung (TDD) - "Test ist nicht gleich Test - es gibt feine Unterschiede zwischen Test-First und Test-After!" - Erläutern Sie die Aussage!

4 Lernvideos

Sie erstellen im Team ein Lernvideo zwischen 6-10 Minuten Dauer. Für die Abgabe des Lernvideos benutzen Sie das mp4-Format.

Die gegebene Problemstellung - insbesondere die Relevanz und die Einordnung des Thema, sowie verschiedene Aspekte im Zusammenhang mit der Problemstellungen als auch eine kritische Beurteilung sollten sich im Lernvideo "wiederfinden" lassen. Sie haben aber grundsätzlich große Freiheiten in der künstlerischen und inhaltlichen Gestaltung der Lernvideos.

4.1 Bewertungskriterien

Im folgenden sind die Bewertungskriterien für die Lernvideos aufgeführt:

1. Die Problemstellung ist verständlich und ansprechend thematisiert.
2. Die Einordnung in den Kontext *Software Engineering* ist nachvollziehbar erfolgt.

3. Eine kritische Beurteilung hat nachvollziehbar und schlüssig stattgefunden.
4. Das Lernvideo ist für die Zielgruppe Bachelor adäquat und ansprechend aufbereitet.

5 Dokumentation

Sie dokumentieren und erläutern ihre Vorgehensweise sowie ihre Verantwortlichkeiten im Team in einem Textdokument, das Sie in Markdown (<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>, <https://daringfireball.net/projects/markdown/syntax>) verfasst haben.

Besondere Aspekte in der Bearbeitung der Aufgabe oder Themenstellung halten Sie im Textdokument ebenfalls textuell fest.

Im Dokument führen Sie auf, welche Quellen (Literatur, ...) Sie für die Erstellung des Lernvideos und der Lernfragen benutzt haben. Der Umfang der Dokumentation umfasst wenige Seiten (zwischen 3-4 Seiten).

5.1 Bewertungskriterien

Die Bewertungskriterien orientieren sich an den oben aufgeführten Anforderungen.

6 Lernfragen

Sie erstellen 4 Lernfragen zu ihrer ausgewählten Themenstellung aus dem Bereich *Software Engineering*. Jeder der Lernfragen ist eine Multiple-Choice Frage mit 4 Antwortmöglichkeiten. Genau eine der Antworten ist korrekt. Für jede falsche Antwort wird ein kurzes Feedback (als Antwort) mit Begründung bereitgestellt.

Die 4 Lernfragen inklusive der Antworten hinterlegen Sie als JSON-Datei (<https://www.json.org/json-de.html>).

Jede Frage wird von ihnen hinsichtlich des Schwierigkeitsgrads eingestuft und klassifiziert: leicht, mittel, schwer. Für jede Klassifikation erstellen Sie mindestens 1 Frage; also beispielsweise 2 leichte, 1 mittlere und 1 schwere Frage.

In einem separaten Textdokument (kein Word!) erläutern Sie *kurz* ihre Einschätzung und Klassifizierung: Warum haben Sie die jeweilige Frage in welche Kategorie gelegt.

6.1 Beispiel

Im folgenden ist ein Beispiel aus dem Bereich Vorgehensmodelle abgebildet (mit lediglich 3 Antwortmöglichkeiten).

- Frage: *Der Scrum Master spielt eine wichtige Rolle im agilen Vorgehensmodell Scrum. Welche der folgenden Aussagen trifft zu?*
- Bereich Vorgehensmodelle
- Schwierigkeitsgrad mittel
- Antwort 1: *Der Scrum Master ist verantwortlich für das Projekt.*
 - falsche Antwort.
 - Begründung und Feedback: Er fungiert als Vermittler zwischen Product Owner und Team ohne, dass er dabei eine fachliche oder technische Verantwortung übernehmen müsste.
- Antwort 2: *Der Scrum Master sollte ein guter Programmierer sein, damit er seinem Team bei Problemen helfen kann.*
 - falsche Antwort
 - Begründung und Feedback: Der Scrum Master programmiert nicht; er ist damit auch nicht Teil des (Entwicklungs)teams.
- Antwort 3: *Der Scrum Master vermittelt zwischen dem Scrum Team und Product Owner. Insbesondere schützt er das Team vor äusseren Störgößen.*
 - richtige Antwort.

Die Frage, Antworten und Begründungen als JSON-Datei haben folgenden Aufbau:

```
{
  "fragen": [
    {
      "text": "Der Scrum Master spielt eine wichtige Rolle im
      agilen Vorgehensmodell Scrum. Welche der folgenden Aussagen
      trifft zu?",
      "bereich": "vorgehensmodelle",
      "level": "mittel",
      "antworten": [
        {
          "text": "Der Scrum Master ist verantwortlich für das Projekt.",
          "korrekt": false,
          "warum": "Er fungiert als Vermittler zwischen Product Owner
          und Team – ohne, dass er dabei eine fachliche
          oder technische Verantwortung übernehmen müsste."
        },
        {
          "text": "Der Scrum Master sollte ein guter Programmierer sein,
          damit er seinem Team bei Problemen helfen kann.",
```

```

    "korrekt": false,
    "warum": "Der Scrum Master programmiert nicht; er ist damit
auch nicht Teil des (Entwicklungs)teams."
  },
  {
    "text": "Der Scrum Master vermittelt zwischen dem Scrum Team
und Product Owner. Insbesondere schützt
er das Team vor äusseren Störgößen.",
    "korrekt": true,
    "warum": "super!"
  }
]
}

```

6.2 Bewertungskriterien

Im folgenden sind die Bewertungskriterien für die Erstellung der Fragen, Antworten und Begründungen aufgeführt:

1. Jede Frage ist relevant für den ausgewählten Themenbereich.
2. Jede Frage ist verständlich formuliert und benutzt die Fachterminologie des Themengebiets.
3. Die bereitgestellten Antworten sind verständlich formuliert und fordern den Leser bzw. Lernenden zum Nachdenken und Reflektieren auf. Zu jeder Frage gibt es 4 Antwortmöglichkeiten.
4. Die Begründungen zu den nicht korrekten Antworten sind verständlich formuliert und verdeutlichen den Zusammenhang mit der gestellten Frage.
5. Die Einstufung der Frage ist angemessen und nachvollziehbar.
6. Die JSON-Datei ist syntaktisch korrekt und validiert.

6.3 Hinweise

- Sämtliche Fragen aller Studierenden werden in Moodle hochgeladen und stehen allen Teilnehmern am Ende der Veranstaltung insbesondere zur Klausurvorbereitung zur Verfügung.
- Für Erstellung der JSON-Datei können Sie auf verschiedene Werkzeuge zurückgreifen (u.a. in Eclipse, <https://jsoneditoronline.org>, ...)
- Validieren Sie die JSON-Datei zwingend vor Abgabe!