Softwarepraktikum 2022 Hinweise zur Meilensteinabnahme 1

Abgabefrist: 22.5.2022

Allgemeine Hinweise:

- Die Meilensteinabnahme erfolgt unter aktiver Beteiligung aller Gruppenmitglieder. Im Fall der Abwesenheit eines Gruppenmitglieds während der Meilensteinabnahme muss ein ärztliches Attest vorgelegt werden.
 Selbiges gilt auch für die mit dem jeweiligen Tutor/der jeweiligen Tutorin vereinbarten Besprechungstermine in den Tutorien.
- Die Abgabe aller Artefakte erfolgt über das teameigene Git-Repository.
- Für Fragen bzw. Anregungen außerhalb der betreuten Rechnerzeiten nutzen Sie bitte auch das Forum in Ilias.

1 GitHub

Alle Artefakte, die bei der Bearbeitung des Praktikums entstehen, sollen in einem Git-Repository gespeichert und damit unter Versionskontrolle gestellt werden. Hierzu empfehlen wir https://github.com/ an. Legen Sie dort für Ihr Team ein privates Repository an. Benennen Sie dieses nach dem Schema:

SP22_Gruppe[A-E]

Fügen Sie alle Teammitglieder, den/die Tutor/in Ihrer Gruppe und die Nutzer *lucieflek* und *caisa-admin* als *Collaborator* hinzu. Die Nutzernamen der Tutoren finden Sie in der Gruppenbeschreibung in Ilias.

Nutzen Sie Git aktiv! Verwenden Sie insbesondere Branches um parallel zu arbeiten und Ihre Ergebnisse später zusammenzuführen. Führen Sie insbesondere **vor jeder Meilensteinabgabe** Ihre Ergebnisse zusammen!

Eine Einführung in die wichtigsten git-Befehle, verfügbar online: https://git-scm.com/docs/gittutorial, https://www.atlassian.com/git/tutorials...

Organisation

Bitte nutzen Sie die folgende Verzeichnisstruktur in Ihrem Repository für Ihr Projekt:

src/ Programmcode für Ihr Projekt
ms1/ Dokumente für Meilenstein 1
ms2/ Dokumente für Meilenstein 2
ms3/ Dokumente für Meilenstein 3
fin/ Dokumente für die finale Version
doc/ Allgemeine Dokumente

Formatvorgaben und Dateiformate

Bilder und Graphiken:

Geben Sie Bilder und Graphiken in einem geeigneten Format ab. Geeignete Formate sind jpeg, png und pdf. Für am Computer erstellte Graphiken empfehlen wir png, für Vektorgraphiken pdf. Alle Bilder sollen sowohl am Bildschirm als auch auf Papier lesbar sein. Achten Sie auf einen guten Kontrast.

Tipp: Wenn Sie Ihre Graphiken auf Papier entwerfen, achten Sie auf die Qualität der Digitalisierung: Ein abphotographiertes Blatt kann meist nur mit aufwendiger Nachbearbeitung die notwendige Qualität erreichen. Nutzen Sie die Scanner, die Sie in einigen Computerräumen im Mehrzweckgebäude finden können, beispielsweise in den PC-Pools auf D4 und D5.

Text:

Geben Sie Text im pdf-Format ab. Erstellen Sie für die Abnahme für jeden Meilenstein 1 pdf-Datei mit dem Namen ms<#>-<gruppenname>.pdf. Diese Datei sollte alle Textdokumente und Bilder enthalten, die für den jeweiligen Meilenstein relevant sind. Die Bilder sollen so enthalten sein, daß sie sowohl am Bildschirm als auch auf einem Ausdruck noch gut erkennbar sind.

Tipp: Wenn Sie Ihre Texte mit LATEX erstellen, empfehlen wir Ihnen, jeden Satz in eine neue Zeile zu schreiben. Dadurch können Sie mit git Ihren Text— genau wie den Programmcode—gemeinsam unabhängig voneinander bearbeiten und nachher automatisch zusammenführen.

Abgabemodalitäten:

Reichen Sie Ihr Projekt über das github-Repository ein. Bitte versehen Sie Ihre Einreichung mit einem tag: ms1 für Meilenstein 1, ms2 für Meilenstein 2, ms3 für Meilenstein 3 und final für die Endabnahme. Beachten Sie den Einreichungszeitpunkt (Mitternacht der Einreichungstag; nicht der Tag der Meilensteinabnahme!)

Fragen Sie bei Unklarheiten bitte frühzeitig nach. Alle nachträglichen Anderungen können gegebenenfalls nicht berücksichtigt werden.

Andere Abgabemodalitäten werden nur in Ausnahmefällen und nur nach vorheriger Rücksprache akzeptiert.

Tipp Prüfen Sie vor der Einreichung, ob:

- das Projekt auf dem aktuellen Stand ist
- alle Bilder und Graphiken den Kriterien entsprechen
- Sie alle Dokumente in einer pdf-Datei zusammengefaßt haben

Inhalt des Dokuments für den Meilenstein 1

1. Kurzbeschreibung

- Inhaltliche Beschreibung des Projekts
- Rahmenbedingungen
 - Eventuell Einbettung in großeres Projekt
 - Beschreibung des Auftraggebers
 - Eventuell besondere Vorgaben Zielplattform, zeitliche Vorgaben, etc.

2. Entwicklungsprozess

- Kurzbeschreibung des Prozessmodells inklusive der Artefakte, die in dem gewa hlten Prozessmodell anfallen (Dokumente, Code, Konfigurationen, etc.)
- Aufgabenverteilung über die Teammitglieder.
 - Falls Sie das Rotationsprinzip anwenden, beschreiben Sie auch, nach welchem Mechanismus Sie Rollen rotieren.
 - Rollen wie Projektleiter, Product Owner, Architekt, Entwickler, etc.
 - Verantwortlichkeit für Dokumente, Technologien, Werkzeuge, Protokolle, etc. •
 - Liste der Praktiken, die Sie anwenden wollen, und einer Erklärung, wie diese angewendet werden sollen. Zum Beispiel Code-Konventionen, Pair-Programming, Test-First, Planning Game, Collective code ownership, Develop iteratively, Manage requirements, Model visually, etc.

3. Anforderungen

- Definition des Zielsystems (Hardware, Software, Versionen)
- Funktionale Anforderungen
- Kurzbeschreibungen der Anforderungen (Niveau "Geschäftsanforderungen")
 - Überblick mittels UML Use Case Diagramm
 - Detaillierte Anforderungen (Niveau "Benutzeranforderungen" bzw.
- "Systemanforderungen")
 - Für Meilenstein 1 reicht es hier die vorgegebenen Kernanforderungen zu präsentieren. Sollten Ihnen bei der Analyse der

Anforderungsspezifikation Mehrdeutigkeiten, Lücken oder Fehler auffallen, sollten Sie diese aber verbessern und Ihre überarbeiteten detaillierten Anforderungsbeschreibungen präsentieren. Für die Detaillierte Anforderungsspezifikation sollten Sie einem Prinzip folgen, das zu Ihrem Vorgehen passt. Insbesondere sollten Sie Anforderungen nicht einfach als beliebigen Text aufschreiben, sondern eine möglichst präzise Ausdrucksweise verwenden. Ein einfacher Ansatz hierfür ist die Verwendung von "kontrollierter Sprache", also das Schreiben in Sa tzen, die immer eine vorgegebene Struktur haben und bestimmte Signalwörter verwenden. Die vorgegebenen Anforderungsbeschreibungen für die Kernanforderungen sind ein Beispiel hierfür. Die kontrollierte Sprache lehnt sich dabei an das Prin-

zip "Behavior-Driven Development" (siehe z.B.

https://de.wikipedia.org/wiki/Behavior Driven Development) an.

- Unterteilung der Anforderungen nach Wichtigkeit und Komplexität
 - Nicht-Funktionale Anforderungen
- Externe Qualität des Systems, z.B. Performanz, Skalierbarkeit, Zuverlässigkeit, etc.
- Interne Qualität der Codes, z.B. Ziele für Code-Metriken, Kommentierung, etc.
 - Glossar

4. Planung

- Verfügbarkeitsplanung
- Zuordnung der Anforderungen und Dokumente zu Meilensteinen. Außerdem sollen die Anforderungen priorisiert werden, so dass sie sinnvoll in der Reihenfolge der Prioritäten abgearbeitet werden können.
- Insbesondere für die kurzfristig umzusetzenden Anforderungen bzw. zu erstellenden Dokumente sollte eine detaillierte Zeitplanung aufgestellt werden und als Gantt- oder Netzwerkdiagramm präsentiert werden.

5. Entwurf

Grober Entwurf / Architektur

Teilen Sie das System in Komponenten auf und dokumentieren Sie die Architektur mit Hilfe eines UML Komponentendiagramms. Stellen Sie hier auch externe Komponenten dar, mit denen Ihr System kommunizieren muss. Verwenden Sie hierzu nur Informationen, die aus der Anforderungsbeschreibung ersichtlich sind. Sollten Sie feststellen, dass wesentliche Informationen fehlen, treffen Sie wenn no itig Annahmen und dokumentieren Sie diese. Dokumentieren Sie, welche architekturellen Entwurfsmuster Sie verwenden.

Detaillierter Entwurf der Struktur

Modellieren Sie Ihr System auf dem Niveau eines "Analysemodells" vollst ändig als UML Klassendiagramm. Teile des Systems, die durch noch nicht ausgearbeitete Anforderungen spezifiziert werden, werden dabei zwangsläufig weniger ausführlich modelliert. Legen Sie insbesondere Wert auf die Modellierung der benötigten Daten. Dokumentieren Sie die Verwendung von Entwurfsmustern.

Detailierter Entwurf des Verhaltens

Modellieren Sie für ausgewählte Aufgaben das Verhalten des zu entwickelnden Systems mittels geeigneter UML Diagrammtypen. Zum Beispiel können Sie Objektdiagramme, Interaktionsdiagramme oder Aktivitätsdiagramme verwenden.