



Technische
Universität
Braunschweig

Anwendungsorientierte
Programmierung für
Ingenieure;

Bewertungsrichtlinien

Projektarbeit im Rahmen von Anwendungsorientierte
Programmierung für Ingenieure

9. April 2019

Die Bewertung eines Projekts gliedert sich in die vier Teilbereiche Planung, Sprintdokumentation, Implementierung und Anleitung. Für das Bestehen des Projekts ist eine Bewertung von mindestens 50 % erforderlich. Die Gesamtbewertung gliedert sich wie folgt:

Planung

■ Backlog mit User-Stories

Eine User-Story beinhaltet eine Funktion der zu entwickelnden Software. Sie ist definiert über die Aktion einer Rolle und dem Nutzen dieser Aktion. Eine User-Story soll kurz und prägnant formuliert werden. Eine User-Story gilt als verhandelbar und soll in dem Entwicklungsprozess als Diskussionsgrundlage dienen. Die Formulierung einer User-Story kann nach dem folgenden Schema erfolgen:

Als *<Rolle>* möchte ich eine *<Aktion>* ausführen, weil sie diesen *<Nutzen>* bringt.

Ein ausformuliertes Beispiel kann wie folgt aussehen:

Ich als Autofahrer möchte beim Parkvorgang den Abstand zu den angrenzenden Fahrzeugen messen, um eine Kollision mit anderen Fahrzeugen zu vermeiden.

Damit im Entwicklungsprozess überprüft werden kann, ob eine User-Story erfolgreich abgeschlossen wurde, besitzt jede User-Story Zielkriterien. Die Zielkriterien sollen voneinander unabhängig und messbar sein. Für das angegebene Beispiel können die Zielkriterien wie folgt aussehen:

- Parkvorgang wird automatisch erkannt
- Der Abstand zu den Fahrzeugen vor und hinter dem eigenen Fahrzeug wird gemessen
- Der Abstand zu den Fahrzeugen wird als Piepton wiedergegeben, der sich in seiner Frequenz reziprok proportional zu dem Abstand verhält
- Bei der Unterschreitung eines fest definierten Abstandes ertönt ein durchgängiger Ton

Jedes Gruppenmitglied soll für die Projektmappe **eine User-Story** mit jeweils **vier Zielkriterien** erstellen. Die User-Stories sollen voneinander abgrenzbar sein. Legt beispielsweise ein Studierender in seiner User-Story die Abstandsmessung nach vorne aus und ein anderes Gruppenmitglied legt die Abstandsmessung nach hinten aus, werden diese als nicht voneinander abgegrenzt gewertet. Die Punkte für eine User-Story werden auf beide Studierende verteilt.

- | | |
|--|------------|
| ■ Im Kontext sinnvolle Rolle vorhanden. | Individual |
| ■ User-Story enthält eine nachvollziehbare Aktion | Individual |
| ■ User-Story enthält einen nachvollziehbaren Nutzen für die definierte Rolle | Individual |
| ■ Zur User-Story passende Zielkriterien vorhanden | Individual |
| ■ Zielkriterien messbar | Individual |

■ Projektschema

Das Projektschema soll die Struktur der Software und - falls vorhanden - der Hardware sowie deren Interaktion darstellen. In dem Schema sollen die Funktionalitäten der Software möglichst modular abgebildet werden. Gleichzeitig sollen die Zielkriterien aller User-Stories in dem Schema auffindbar sein (zum Beispiel über Systemgrenzen, Beschriftung, etc.).

- Vollständigkeit (Aktualität und Übereinstimmung mit der finalen Version des Projekts)
- Konsistenz und adäquate Darstellung
- Verknüpfung mit Zielkriterien der User-Stories

Gruppe

Gruppe

Sprintdokumentation

Ein Sprint ist ein begrenzter Entwicklungszeitraum, in dem ein Inkrement der Software entwickelt werden soll. Das Ergebnis eines Sprints soll eine lauffähige Software sein, die Teile der User-Stories oder einer User-Story enthält. In der Projektmappe sollen **3 Sprints** dokumentiert werden.

■ Sprintplanung

In der Sprintplanung sollen aus den User-Stories Aufgaben generiert werden. Nach der Generierung der Aufgaben sollen von den Gruppenmitgliedern die Abhängigkeiten dieser Aufgaben identifiziert werden. Parallele Vorgänge sollen erkannt werden, damit diese gleichzeitig von unterschiedlichen Gruppenmitgliedern bearbeitet werden können.

Zielkriterien werden hier den Gruppenmitgliedern zugeordnet. Die Bepunktung erfolgt bei Abschluss des Zielkriteriums zugunsten der hier definierten Gruppenmitglieder.

- Es wurde für jedes Gruppenmitglied mindestens eine Aufgabe generiert.
- Abhängigkeiten jeder Aufgabe sind dokumentiert.
- Teilnahme an Sprintplanungs-Meeting

■ Sprintabschluss

Im Sprintabschluss sollen die Aufgaben für den abzuschließenden Sprint geprüft werden. Falls die Aufgaben nicht abgeschlossen wurden, sind die Gründe zu nennen. Des Weiteren soll ein kurzer Überblick über den aktuellen Stand des Projekts erstellt werden. Wurde im Rahmen des Sprints ein Zielkriterium erfüllt, ist anzugeben mit welcher Revisionsnummer dieses Zielkriterium abgeschlossen wurde. Die Punktevergabe erfolgt gleichmäßig unter allen Gruppenmitgliedern, denen das Zielkriterium in der Planung zugewiesen wurde. Soll ein Zielkriterium davon abweichend bewertet werden, so muss in dem Sprintabschluss der tatsächlich geleistete Beitrag der betreffenden Gruppenmitglieder angegeben werden (Anzahl von Aufgaben, Commits, etc.). Am Ende des Sprintabschlusses soll eine Bewertung der Sprintplanung erfolgen (waren die Aufgaben zu umfangreich/zu klein?; sollte die Methodik geändert werden?; konnte die Arbeitslast auf alle Mitglieder gleichmäßig verteilt werden?).

- Übersicht über Stand der User-Stories und Zielkriterien vorhanden (nicht begonnen, in Bearbeitung, abgeschlossen).

Gruppe

- Status der Aufgaben Gruppe
- Verständlichkeit und Nachvollziehbarkeit von der Bewertung der Sprintplanung Gruppe
- Dokumentation der abgeschlossenen Zielkriterien mit Revisionsnummer Individual
- Teilnahme an Sprintabschluss-Meeting Individual

■ Sprintmeeting

Sprintmeetings sollen wöchentlich stattfinden. In diesen Meetings wird festgehalten, welches Gruppenmitglied welche Aufgabe bearbeitet hat, welche Aufgaben in der kommenden Woche bearbeitet werden sollen und welche Probleme aufgetreten sind. Dieses Meeting sollte nicht länger als 15 Minuten dauern. Die Punkte dieses Meetings sollen in einem Protokoll zusammengefasst werden.

- Bearbeitete Aufgaben wurden niedergeschrieben Individual
- Geplante Aufgaben für die Folgewoche wurden niedergeschrieben Individual
- Teilnahme am Meeting Individual

Implementierung

Die API-Projektmappe enthält zum Zeitpunkt der Abgabe (12. August 2019 10:00 Uhr) die zu bewertende Version der Software und deren Dokumentation. Für die Bewertung wird ausschließlich der Stand des master-Zweigs zum Zeitpunkt der Abgabe verwendet. Im Speziellen sind die folgenden Repositories Teil der Abgabe: Das Software-Repository und das Wiki-Repository.

■ Quellcode

Das Software-Repository soll die fertige Software als Quelltext enthalten. Alle User-Stories sollen in dieser Software umgesetzt werden (eine ausführbare Datei pro Projekt). Eine Ausnahme bilden Projekte, in denen die Software zur Kommunikation zwischen verschiedenen Geräten verwendet wird. In diesem Fall dürfen mehrere ausführbare Dateien Bestandteil der Projektmappe sein. Die Software soll kompilierbar sein. Der Quellcode soll von anderen Dateien (Medien, Dokumente und weitere Ressourcen) abgrenzbar sein und in einem dedizierten Ordner (REPO-URL/src) abgespeichert werden.

- Oben genannte Projektstruktur eingehalten Gruppe
- Kompilierbarkeit des Projekts und aller etwaigen Unterprojekte Gruppe

■ Kommentare im Quellcode

Der Quelltext der Software soll sinnvoll mittels Code-Kommentaren dokumentiert werden (nicht jede Zeile). Als sinnvoll dokumentiert gilt der Quelltext, wenn die folgenden Punkte dokumentiert sind:

- Code-Blöcke, die eine eigene Logik enthalten (Schleifen, Kontrollfunktionen, etc.)
- Jede selbst geschriebene Funktion mit ihrer Aufgabe, den Ein- und Ausgabeparametern und entsprechend ihres Umfangs der Vorgehensweise
- Klassen mit ihrer Aufgabe und Besonderheiten bei der Initialisierung und Anbindung
- Jedes Gruppenmitglied soll eine eigene Funktion, die Ein- und Ausgabeparameter enthält, wie oben beschrieben, dokumentieren.

Gruppe

Individual

■ Commits

Jedes Gruppenmitglied soll in dem Software-Repository mindestens 10 Commits erstellt haben. Mit diesen Commits wird sichergestellt, dass jedes Gruppenmitglied ein Versionsverwaltungssystem verwendet hat. Die Commits sollen mit Nachrichten versehen werden, die die Änderung an der Software beschreiben (Einzeilige Zusammenfassung der Änderung und ggf. Angabe von: User-Story, Bug und/oder Feature). Commits sollen möglichst kleinteilig erfolgen und nicht zu viele Änderungen auf einmal enthalten (gut: Bug 121 in Funktion X behoben; schlecht: Implementierung von User-Story A; schlecht: Fortschritt am Projekt; schlecht: Datei xy hinzugefügt). Jedes Gruppenmitglied soll die 10 zu bewertenden Commits auf der Release-Wikiseite unter seinen Namen angeben.

- Nachvollziehbarkeit der Änderung über Commit-Nachricht
- Akzeptable Größe der Commits (max. 250 Zeilen je Commit)

Individual

Individual

■ Bug-Report

Im Laufe der Softwareentwicklung treten unweigerlich Fehler auf. Diese können sich bereits zur Kompilierung oder auch während der Programmlaufzeit manifestieren. Ein Bug-Report adressiert solche Fehler und soll den Programmierer in die Lage versetzen, diese Fehler zu reproduzieren und so bei der Fehlerbehebung helfen. In der Projektmappe soll jedes Gruppenmitglied im Entwicklungszeitraum mindestens einen Bug-Report erstellen, das einen Fehler zur Programmlaufzeit adressiert. Fehler sind Abweichungen von dem vorgesehenen Programmverhalten oder ein Programmabsturz. Bestandteil eines Bug-Reports sind

- die Revisionsnummer des Commits, das den unveränderten Quellcode der getesteten Software enthält.
- die Fehlerbeschreibung (Fehlermeldung beim Absturz, Beschreibung des abnormalen Programmverhaltens)
- die Schritte zum Reproduzieren des Fehlers angefangen bei dem Programmstart
- Behebung des Fehlers, dokumentiert mittels Revisionsnummer. Bepunktung erfolgt zugunsten des Gruppenmitglieds, welches den Commit erstellt hat.

Individual

Individual

Individual

Individual

Der Bug-Report muss als issue auf Github erstellt werden und auf der vorgesehenen Seite im Wiki verlinkt werden.

Anleitung

Mit der Anleitung soll die Inbetriebnahme der Software und - falls vorhanden - Hardware für einen Projektunbeteiligten ermöglicht werden. Jede User-Story soll in der Anleitung beschrieben werden. Es ist darauf zu achten, dass Schnittstellen (zum Beispiel der Anschluss der Hardware) und Systemvoraussetzungen in der Anleitung enthalten sind. Funktionalitäten und Konfigurationsmöglichkeiten sollten in der Anleitung gleichermaßen behandelt werden. Grafische Benutzerschnittstellen sollen anhand von Screenshots erläutert werden. Die Anleitung kann entweder als Wikiseite oder als pdf-Dokument erstellt werden. Das pdf-Dokument muss auf der entsprechenden Wiki-Seite verlinkt werden.

- Vollständigkeit und Aktualität
- Bezug zu User-Stories vorhanden
- Verständlichkeit
- Konsistenz der Darstellung

Individual

Individual

Gruppe

