

Bildungszentrum Zürichsee

Bewertungskriterien Projektdokumentation

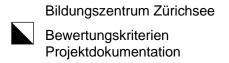
Bewertungskriterien

Die Bewertungskriterien werden entsprechend der Gütestufen mit 1 (Minimum) bis 3 (Maximum) bewertet und in Meilensteinen gruppiert und gewichtet. Fehlt ein Kriterium, wird dieses mit 0 gewertet. Es werden Punkte für zu spät eingereichte Artefakte abgezogen.

Zu jedem Meilenstein werden konkrete, messbare Bewertungskriterien gelistet.

Übersicht

| Meilensteine | Gewichtung |
|-----------------------------------|------------|
| Anwendungsfälle und Anforderungen | 20% |
| Klassen und deren Beziehungen | 20% |
| Architektur-Design | 20% |
| GUI-Design | 10% |
| Testfälle | 10% |
| Anwendung und API | 20% |
| Total | 100% |



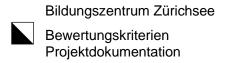
Anwendungsfälle und Anforderungen

| Der Projektauftrag wird sinngemäss in ein Anwendungsfalldiagramm übersetzt. | | |
|---|--|--|
| Das Diagramm genügt weder den formalen Anforderungen noch gibt es den Projektauftrag sinngemäss wieder. | Das Diagramm genügt entweder den formalen Anforderungen ODER gibt den Projektauftrag sinngemäss wieder. | Das Diagramm genügt den formalen Anforderungen und gibt den Projektauftrag sinngemäss wieder. |
| Jeder Anwendungsfall wird in einer Anwendungsfallspezifikation präzisiert. | | |
| Beschreibung wie auch ein | Es fehlen wichtige Merkmale der Beschreibung. Der beschriebene Ablauf ist sinnvoll. | Die Beschreibung ist vollständig und gibt einen sinnvollen Ablauf wieder. |
| Die Anforderungen sind aus den Anwendungsfällen sinnvoll abgeleitet. | | |
| unvollständig, unverständlich UND nicht aus den Anwendungsfällen abgeleitet. | teilweise unvollständig und unverständlich ODER sind | Die Anforderungen sind vollständig, verständlich und aus den Anwendungsfällen abgeleitet. |

Klassen und deren Beziehungen

| Die fachlichen Klassen liegen als CRC-Karten vor. | | |
|---|---|---|
| Es fehlen wichtige Klassen. ODER Bei den dargestellten Klassen fehlen wichtige Aspekte. | Das Diagramm weist Mängel in der Beschreibung der dargestellten Klassen auf. | Das Diagramm ist bezüglich der dargestellten Klassen vollständig. |
| Der Entwurf ist mit den benötigten technischen Klassen und Schnittstellen in einem Klassendiagramm ergänzt. | | |
| Die technischen Klassen sind erwähnt aber im Entwurf nicht dokumentiert. | Die technischen Klassen sind im Entwurf teilweise dokumentiert. | Die technischen Klassen sind vollständig im Entwurf dokumentiert. |
| Die Beziehungen der Klassen sind vollständig dokumentiert. | | |
| Die Beziehungen der Klassen sind unklar dokumentiert. | Die Beziehungen der Klassen sind teilweise dokumentiert. | Die Beziehungen der Klassen sind vollständig dokumentiert (inkl. Kardinalitäten) |
| Der Entwurf weist eine passende Paketstruktur auf. | | |
| Der Entwurf basiert auf einem einzigen Paket (nicht default Paket). | Der Entwurf weist eine (eher zufällige) Paketstruktur auf. | Der Entwurf weist eine begründete Paketstruktur auf. |

26. April 2020 Seite 2 von 4



Architektur-Design

| Die Analyse beinhaltet Aktivitätsdiagramme. | | |
|--|--|--|
| | Die Aktivitäten werden mehrheitlich durch Diagramme dokumentiert | Die Aktivitäten werden vollständig durch Diagramme dokumentiert. |
| Das Design beinhaltet Zustandsdiagramme. | | |
| Es werden nur vereinzelte Zustände dokumentiert. | Es werden einige wichtige Zustände dokumentiert. | Kritische Zustände sind vollständig dokumentiert. |
| Das Design beinhaltet Sequenzdiagramme. | | |
| | Es werden einige wichtige Abläufe dokumentiert. | Kritische Abläufe sind vollständig dokumentiert. |
| Das gewählte Design unterstützt die einfache Realisierung der Anwendung. | | |
| | Design weist schwächen auf. | Design ist klar und auf Problemstellung bezogen. |

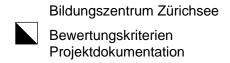
GUI-Design

| Die Mockups geben den Anwendungsaufbau und deren Verwendung wieder. | | | |
|---|---|--|--|
| Aspekte abgedeckt. | Es werden nur einzelne Es wird eine Mehrheit der Die grafische Oberfläche | | |

Testfälle

| 51 W. H. T. (1911) | | | |
|---|---|--|--|
| Die gewählten Testfalle geb | Die gewählten Testfälle geben die Funktionalität der Anwendungsfälle wieder. | | |
| Es werden nur einige Anwendungsfälle abgedeckt. | Es wird eine Mehrheit der Anwendungsfälle abgedeckt. | Es werden alle Anwendungsfälle abgedeckt. | |
| Die gewählten Testfälle geb | en die Anforderungen wied | er. | |
| Es werden nur einige Anforderungen abgedeckt. | Es wird eine Mehrheit der Anforderungen <i>abgedeckt.</i> | | |
| Die Testfälle sind durch JUi | Die Testfälle sind durch JUnit abgedeckt. | | |
| Einige wenige Testfälle können durch JUnit nachgewiesen werden. | Eine Mehrheit der geeigneten Testfälle können durch JUnit nachgewiesen werden. | Alle geeigneten Testfälle können mit JUnit-Tests nachgewiesen werden. | |
| Das Testprotokoll ist vollständig. | | | |
| Das Protokoll ist bezüglich Soll-Ist-Vergleichs unvollständig. | Das Protokoll enthält korrekte Soll-Ist- Vergleiche. Es fehlen Hinweise betreffend notwendige Massnahmen. | Das Testprotokoll enthält korrekte Soll-Ist-Vergleiche und wo notwendig zu ergreifenden Massnahmen. | |

26. April 2020 Seite 3 von 4



Anwendung und API

| Die Klassen sind entsprechend den UML-Diagrammen implementiert und paketiert. | | | |
|---|---|--|--|
| Im Code finden sich zusätzliche Elemente (Attribute, Methoden, Klassen, usw.), die im Entwurf fehlen. | Die Klassen sind korrekt und vollständig implementiert. Es fehlt die Paketierung. | Alle Vorgaben des Entwurfs sind im Code umgesetzt. | |
| Die Dokumentation jeder Kl API erstellt. | lasse ist vollständig (Autor, I | Beschreibungen) und die | |
| Die Beschreibung der Klasse und deren Methoden erklärt deren Anwendung nicht oder zu wenig klar. | Die Beschreibung ist nicht vollständig. Es fehlen Elemente wie Version, usw. | Zu jeder Klasse liegt eine vollständige und aussagekräftige API vor. | |
| Der Code ist verständlich u | nd wartbar. | | |
| Die verwendeten Programmstrukturen wirken zufällig, so dass Änderungen nur schwer zu implementieren sind. | Der Code ist mehrheitlich gut strukturiert. Für Änderungen bedarf es einer vertieften Analyse, um den Code umfassend zu verstehen. | Der Code ist gut strukturiert (funktionale Blöcke) und wo nötig mit erklärendem Kommentar unterlegt. Änderungen können einfach vorgenommen werden. | |
| Die für die Aufgabe typische | Die für die Aufgabe typischen Entwurfsmuster sind begründet und dokumentiert. | | |
| Entwurfsmuster werden im Code angewendet, aber nicht dokumentiert. | Die Entwurfsmuster sind im UML-Diagramm ersichtlich. | | |
| Die Applikation erfüllt die geforderten Funktionalitäten. | | | |
| Es werden nur wenige der Anforderungen erfüllt. | Die Anforderungen werden zu einem grossen Teil erfüllt. | Es werden alle Anforderungen des Auftrags erfüllt. | |

26. April 2020 Seite 4 von 4