

#### 4 Verwandte Arbeiten

Dieses Kapitel umfasst die offiziellen Arbeiten und Dokumente, die in diese Ausarbeitung einfließen. TODO

##### 4.1: MechatronicUML: [Von Proposal]

##### 4.2 Auswertung bestehender Ansätze für die Integration von Arduinosoftware und Automatisierung in Eclipse [Eigenes Dokument]

##### 4.3 Taxonomie:

In diesem Unter-Kapitel werden die Taxonomien für die verschiedenen umgesetzten Implementierungen eingeführt.

Laut Usman et al. [Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method] ist Taxonomie hauptsächlich als ein Klassifizierungssystem definiert.

[[[ Teilweise zusammengefasster Teil über Absicht und Terminologie:]]]

In diesem sollen beim Aufstellen einer Taxonomie zu einer Umsetzung jeweils die in dem Forschungsfeld oder in der Branche [[[(evtl. bessere(r) Begriff(e) )]]] allgemein akzeptierten Aspekte, Attribute, Faktoren oder Funktionen unter Verwendung der jeweiligen üblichen Terminologie berücksichtigt werden.

##### 4.4: Codequalität:

Für die Code-Qualität wurde der ISO-2510-Standard [<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>] (23.2.24) gewählt. Dieser beschreibt verschiedene Kriterien, von denen diejenigen, die für diese Ausarbeitung bzw. für den im Rahmen des von ihr erstellten Quellcodes relevant sind, im folgenden zusammengefasst beschrieben werden.

Hinweis: Inzwischen wurde bemerkt, dass die Webseite und deren Punkte geändert wurden. Über <https://web.archive.org/> können alte Versionen eingesehen werden, soweit diese und deren Unter-Seiten archiviert wurden. Für die Ausarbeitung wurde mit einer lokalen Kopie gearbeitet, d.h. es wurde alles am 23.2.24 als Notiz in ein Textdokument kopiert. Diese alte Version wurde beibehalten, weil sie bei einer Besprechung über die Kriterien vorgelegt und für gut befunden wurde.



Funktionale Eignung:

Wie weit Funktionen, deren „angegebenen und angedeuteten Anforderungen unter den spezifizierten Bedingungen“ [ISO25010] erfüllt sind, bereitgestellt werden.

Unter-Charakteristiken:

- Funktionale Vollständigkeit
- Funktionale Korrektheit
- Funktionale Angemessenheit

Performanceeffizienz:

Weder bei dem Anwendungsszenario noch bei den Aufgaben ist die Performance relevant oder als Kriterium genannt. Deshalb wird zu diesem Aspekt keine Bewertung durchgeführt. Das Evaluieren des Zeitverhaltens der Pipeline wird von dem Attribut „Geschwindigkeit“ aus

[<https://www.techtarget.com/searchsoftwarequality/CI-CD-pipelines-explained-Everything-you-need-to-know>] übernommen.

Kompatibilität:

Innerhalb derselben Hardware oder Softwareumgebung können „Informationen mit anderen Produkten, Systemen oder Komponenten [ausgetauscht] und/oder die vorausgesetzten Funktionen [durchgeführt]“ [ISO25010] werden.

Unter-Charakteristiken:

- Koexistenz
- Interoperabilität

Verwendbarkeit:

„Das Erreichen von spezifizierten Zielen in einem spezifizierten Kontext mit Effektivität, Effizienz und Zufriedenstellung“ [ISO20100]

Unter-Charakteristiken:

- Eignungserkennbarkeit
- Lernbarkeit
- Bedienbarkeit
- Nutzerfehlerschutz
- Benutzerinterface-Ästhetik
- Zugänglichkeit

#### Zuverlässigkeit:

Für die spezifizierten Bedingungen und spezifizierte Zeit können die angegebenen Funktionen ausgeführt werden.

#### Unter-Charakteristiken:

- Ausgereiftheit
- Verfügbarkeit
- Fehlertoleranz
- Wiederherstellbarkeit

#### Sicherheit:

Weder bei dem Anwendungsszenario noch bei den Aufgaben ist die Sicherheit relevant oder als Kriterium genannt. Deshalb wird zu diesem Aspekt keine Bewertung durchgeführt.

#### Wartbarkeit:

Die Effektivität und Effizienz, mit denen [das] Produkt oder System geändert werden kann.

#### Unter-Charakteristiken:

- Modularität
- Wiederverwendbarkeit
- Analysierbarkeit
- Testbarkeit

#### Portabilität:

Effektivität und Effizienz, mit der das Produkt „von einer Betriebs- oder Verwendungs-Umgebungen zu einer anderen übertragen werden kann“ [ISO250101].

#### Unter-Charakteristiken:

- Adaptierbarkeit
- Installierbarkeit
- Ersetzbarkeit

#### 4.5: CI/CD-Pipeline:

Es wird die Qualität der im Rahmen dieser Ausarbeitung geschriebenen Eclipse-Plug-Ins bewertet werden.

Im Rahmen dieser Ausarbeitung wurde eine Implementierung einer Pipeline für die Eclipse-IDE umgesetzt, damit die Entwicklungspraxis CI/CD ermöglicht wird. Es wird geprüft werden, wie gut sie die Eigenschaften und Funktionen von CI/CD-Pipelines nach Stephen J. Bigelow [Bigelow , <https://www.techtarget.com/searchsoftwarequality/CI-CD-pipelines-explained-Everything-you-need-to-know>, Abschnitt „Attributes of a good CD/CD pipeline“] erfüllt.

Hier folgt eine Zusammenfassung aus dem Englischen von [Bigelow]:

**Geschwindigkeit:**

Schnelle Durchführung von vielen Schritten bzw. einer großen Pipeline.

**Konsistenz:**

Der automatische Ablauf soll immer dieselben Ergebnisse liefern. Sowohl Variationen als auch manuelle Schritte „verlangsamen die Pipeline und laden Fehler und Ineffizienz ein.“ [ Bigelow]

**Enge Versionskontrolle:**

Eine integrierte zuverlässige Versionskontrolle, um mögliche erforderliche Rückgängigmachungen jederzeit durchführen zu können.

**Automatisierung:**

Extensive Automatisierung bringt den Code durch die Pipeline mit so wenig manueller Interaktion wie möglich.

**Integrierte Rückmeldungsschleifen:**

Bei Problemen möglichst frühe Reaktionen und ein möglichst früher Abbruch, um diese (die Probleme) möglichst früh zu beheben.

**Durchgängige Sicherheit:**

Weder bei dem Anwendungsszenario und noch bei den Aufgaben ist die Sicherheit relevant oder als Kriterium genannt. Deshalb wird zu diesem Aspekt keine Bewertung durchgeführt.