Qualitätssicherungsmaßnahmen

**Gebrauchsanleitung:**

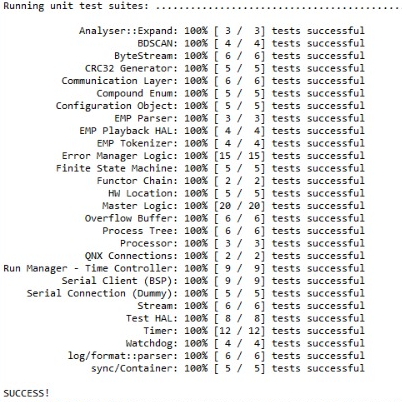
Für den sicheren und bestimmungsgemäßen Umgang mit der Maschine haben wir eine Gebrauchsanweisung verfasst. Diese ist unter *“doku/Betriebsanleitung“* zu finden.

**Programmierstil:**

Wir erstellen Quellcode nach bestimmten, vorab festgelegten Regeln. Dies ist insbesondere für die Verständlichkeit und Wartbarkeit unserer Software wichtig. Der festgelegte Programmierstil ist unter *“doku/CodingStyle“* zu finden.

**Unit Test:**

Um den konkreten Code zu testen, benutzen wir Unit-Tests. Dafür wurde ein Framework erstellt *“ESEP/src/test/unit“.* Die Unit-Tests sind unter *“ESEP/src/test/ut“* zu finden.



**Code Qualität:**

Wenn zuviel Code in einer Quelltextdatei oder einer Funktion ist, wird es schnell unübersichtlich. Deshalb lagern wir in so einem Fall den Code auf mehrere Dateien bzw. Hilfsfunktionen aus. Dies dient in erster Linie dazu die Leserlichkeit und die Wartbarkeit zu verbessern. Dies ist z.B. in unserem RunManager-Modul zu sehen. Hier wurde der Code für die Initialisierung in eine Hilfsfunktion ausgelagert *“ESEP/src/base/run/init\_Logic.cpp“.*

**Funktionale Tests:**

Wir überprüfen unser System in Bezug auf funktionale Anforderungsmerkmale, also ob das Verhalten, der Software bzw. des Systems den Anforderungen(Requirements) im Pflichtenheft entspricht. Ein Beispiel dafür ist der funktionale Test des Höhensensors. Dieser und andere Funktionstests sind in *“Esep/src/test/ft“* zu finden.

**Akzeptanztest:**

Wir überprüfen ob unsere Software die funktionalen Erwartungen und Anforderungen im Gebrauch erfüllt. Dies ist z.B. an dem Run-Manager-Test unter *“Esep/src/test/ft/runmanager“* zu sehen.

**Dokumentation:**

Die mit dem Kunden in den Meetings festgelegten Beschlüsse sind unter *“doku/Spezifikationsbeschluesse mit dem Kunden“* zu finden. Die Beschlüsse, die im Team getroffen worden sind, sind unter *“doku/beschluesse“* zu finden.

**Modellierung & Design:**

Die Modellierung unserer Software ist unter *“doku/diagrams“* zu finden. Dieser Ordner spiegelt unseren Entwurfsprozess zur Planung unserer Software-Lösung wieder. Es dient vor allem dazu, die Komplexität des Projektes für uns als Programmierer handhabbar zu machen und das Risiko von Fehlentwicklung zu verringern.