# Labskaus

Matthias Baumann

28. Dezember 2017

# Inhaltsverzeichnis

# 0.1 Fehlerkonzept

Fehler passieren! Aber was dagegen tun. Hier wurde sich dagegen entschlossen einfach nur die Augen zu verschließen und zu hoffen das schon alles gut gehen wird.

## 0.1.1 Definitionen

**Fehler:** Allgemeiner Ausdruck der für die verschiedensten Arten von Fehlern verwendet wird.

**Programmierfehler:** Ein Fehler welcher entweder im Konzept entstanden ist oder durch die Ausprogrammierung mit in das Programm eingeflossen ist.

**Warnung:** Warnungen sind keine Fehler des Programms Labskaus. Warnungen werden genutzt um darzustellen das ein Ereignis aufgetreten ist welches zum Abbruch einer Funktionalität führt und entweder durch Verbindungsabbrüche (Timeouts), Fehlbedienung oder durch Einlesen nicht kompatibler Dateien verursacht wurde.

**Events:** Events sind Meldungen die zusätzlich in einem Log aufgenommen werden sollten. Sie können helfen zu verstehen welches Ereigniss vor einem Fehler aufgetreten ist. Sie erlaubt die grobe Rekonstruktion die eventuell zu einem Fehler geführt hat.

**Debug:** Debug-Meldungen dienen nur dem Entwickler zum Abfragen von Zwischenergbnissen und sollte in einem Release nach Möglichkeit nicht Implementiert sein.

## 0.1.2 Anforderungen an das Fehlerkonzept

**Req1**: Die Ausgaben von cout und cerr sollten in eine Datei geschrieben werden damit diese im Fall eines Programmabsturzes genutzt werden können um die Fehler zu analysieren.

**Req2**: Das Programm sollte auch im Terminal gestartet werden können. Die Ausgaben die laut Req1 in die Dateien geschrieben werden, sollten auch auf dem Terminalfenster ausgegeben werden. Dieses sollte umfassen: Programmierfehler, Warnungen, Events und Debugmeldungen. Es kann natürlich sein, dass nun das Terminalfenster kontinuierlich zugemüllt wird. Dieses wird hier aber bewusst in Kauf genommen, da in dem Fall ein Fehlers jede Information wichtig sein kann.

**Req3**: Alle Fehler sollten nach dem gleichen Muster in den Code gebaut werden so diese später auch noch programmatisch geändert werden könnten.

**Req4**: Ereignisse die auf einen Programmierfehler hindeuten sollten über cerr (also ungepuffert) weggeschrieben werden.

**Req5**: Die Ereignisse sollten direkt beim Auftreten in eine Datei geschrieben werden so das sie im Fall eines Programmabsturzes verfügbar sind.

**Req6**: Das Fehlerkonzept sollte gut kapselbar sein, so das es eventuell auch in weiteren Programmen wie eingesetzt werden kann.