### 1. Parameterimport und Visualisierung

**1.1** Doors beinhaltet vereinbarte Anforderungen mit definierten Parametern. Diese Parameter repräsentieren Werte, die je nach Variante, unterschiedlich sein können (zum Beispiel: die maximale Geschwindigkeit eines Autos). Diese Menge von Werten muss nach TESTONA importiert und gespeichert werden. Zu beachten ist, dass jeder Wert zu der richtigen Produktvariante zugeordnet wird. Dafür kann ein *Enum* oder ein *Zweidimensionales* Array (Variante *Tag/ID* und Parameterwert) benutzt werden. Diese Information muss lokal auf dem Rechner gespeichert werden (Metadaten ?), da sie zur Verfügung stehen muss, auch wenn keine Verbindung zur Datenbank gibt. Die Aufrufe zur Datenbank werden über eine Java-API erfolgen.

**1.2** Wenn der Tester die Ansicht zwischen Varianten ändert, muss der Wert hinter dem Parameter auf die jeweilige Variante aktualisiert werden (maximale Geschwindigkeit von Cabrio = x, Limo = y, Kombi = z).

Zu klären:

* Wie definiert man Parameter in DOORS? „Playground“ definieren für die Implementierung.
* Zuordnung vom DOORS importierten Parametern zu Baumelementen (Autobahngeschwindigkeit 🡪 max.Gesch. = x,y,z) wird vom Tester gemacht?
* Metadata für Parameterwerte, wie werden andere Infos gespeichert?

### 2. Testgültigkeit

Nachdem das Import und Anzeigen von Parameter in TESTONA funktioniert hat, soll TESTONA irrelevante Testfälle (Duplikate) innerhalb einer Varianten erkennen und löschen/den Benutzer warnen. Das heißt, ist die maximal Geschwindigkeit gleich die Landgeschwindigkeit (Cabrio) und in beiden Fällen wird eine Vollbremsung betrachtet, sind beide Testfälle gleich.

### 3. Optimierungskriterien

Es wird den Tester die Möglichkeit gegeben, den Testablauf zu optimieren. Im Fall dass verschiedene Varianten gleiche Parameter besitzen, so können sich Testfälle unnötigerweise wiederholen. So ist der Fall bei der Stadtgeschwindigkeit und einen leichten Bremsdruck. Die Parameter in den Varianten (Cabrio, Limo, Kombi) sind gleich. Zu beachten ist, dass im diesem Beispiel es sich um einen sehr einfachen Baum handelt. Mit dieser Taktik können gezielt Tests aus einer Variante ignoriert werden, da diese schon in einer anderem vorkommen. Dafür muss ein Algorithmus entwickelt werden.

|  |  |  |  |
| --- | --- | --- | --- |
| Auto/  Geschwindigkeit (km/h) | Cabrio | Limo | C:\Users\hansert\Documents\Masterarbeit\bremsen.PNGKombi |
| Stadt | 50 | 50 | 50 |
| Land | 100 | 100 | 100 |
| Maximale | 100 | 200 | 200 |