## Chosen Changes

Scripting –

Looped Scripting –

Test -j –

Quis custodiet? –

### Motivation to not choose the other changes

All thumbs and Emulate Schemulate are not chosen since our analysis is that they would have a impact too significant to refactor within the timeframe given by the deadline.

Emulate Schemulate would require us to add Crash detection and checkpoints to the emulation component. This would break the current standard we have for seperating different functionality in different components.

With All thumbs we felt that we lack the knowledge of mobilephones and touchscreen for finding valid solutions to this change. With enough time to researcht the subject it could be solved but this does not fit the staff assumptions made by our team.

## Analyzis

There was no need to change to current factors, instead we added four new factors, one for each change. These are marked in the factor table with red. All the new factors are product factors.

We added two new issue-cards that will also be marked with red.

There will be some changes to the architecture to accommodate the new functionallity. Thankfully we already designed the system to handle generating new input from output received for different parts of the MIB so most changes will be localized to the testing wrapper and the new scripting component.

A scripting component will be added along with the corresponding modules for the module and execution view. It will be connected to the input/output component to enable it to feed new input into the system as well as define new output validation rules.

Running parallel tests on the same intance will require the testing wrapper to keep track of several testing techniques at the same time. With our current model this should not be a problem as long as they both have unique identifiers so that the correct input/output channels can be maintained.

All other components of the system should be unaffected on a architectual level.

## Evaluation scenarios

* Customer asks us to implement a specific script-language for the input-scripting
* A tester starts the MIB in automatic mode and later wants a hands on aproach, thus wanting to switch into a monitoring mode.
* Output validator receives output to be validated from two parallel tests on the same tested system, how does it tell them apart?
* The customer wants 10 different scripting languages implemented
* One testing technique crashes the tested system and the system is restarted from a checkpoint. What happens to the testing technique that did not crash the system?
* We are testing a system with a extreme throughput of data and the tester wants to run in monitored mode.