

CNST V 2.0

问题

- 接口相关问题

#	作用	详情	是否必须	问题	方案
1	创建device	【sh (sh bridge service) 】 tinker在成功通过测试某一部件时 创建一个游离（未关联robot）的部件（device）	否	cnst中是否需要提供该接口 1. cnst中是否需要存在游离的部件(cnst本身是否要对出生前部件本身进行任何形式的管理) 2. 存在游离部件是否会导致系统使用的复杂度 3. 如果提供单独的接口 是否需要考虑 测试未通过相关情况	如果cnst需要关心出生前： 1.原cnst v1.0的方案不变 所有的成品机器人为一个asset 用户如果关心自己的robot 那么就去看asset。
2	关联测试报告	【sh (sh bridge service) 】 tinker在成功通过测试某一部件时 将相关测试报告与游离部件相关联	否	cnst中是否需要提供该接口 同上	同上
3	创建机器人灵魂	【sh(sh bridge service)】 tinker 在成功通过整机测试时 将所有相关联的（与tinker通信）的部件信息收集 创建机器人灵魂	是		
4	创建机器人灵魂 并关联相关部件时 同时创建asset	【sh(sh bridge service)】 tinker 在成功通过整机测试时 创建了机器人灵魂并与部件相关联后 创建asset并与之关联	否	cnst 中是否提供对外的关联asset接口 1. 是否需要自动关联asset 2. 提供了自动关联asset的能力是否需要提供手动关联asset的能力	
5	关联失败的测试报告	机器人因为某一部件出现问题 返厂维修（此时并不能确定 是什么的问题） 【sh(sh bridge service)】 tinker 在测试返厂的机器人时 对有整机进行测试 产出一份失败的测试报告	是	机器人的一部分生命周期是否需要在cnst中维护 其零部件的生命周期是否需要记录在cnst中	

- cnst 是否保留手动创建 device asset 的能力；

保留的优点	保留的缺点	存在的矛盾点：	解决方法
保留cnst 的独立性（即使没有sea horse依然可以通过手动创建设备来进行正常运转）	<ul style="list-style-type: none">由于操作不慎导致虚拟device 的数据污染 正常device的数据统计（cnst统计机器人卖出/返修比等）误将虚拟device分配给用户 导致无法得到正确的数据返回；	TB中只有TENANT可以 创建asset以避免 上述提到的缺点 目前在CNST中我们要对CUSTOMER的权限进行扩大（white公司 是否需要创建一些 未经测试的ROBOT 或者一些自己的设备来进行租售的业务）	是否可以通过某些权限的设计来控制某些 Account可以创建新的device与asset 如果这么做需要具体权限的设计

- seahorse 的webportal 需要实现哪些功能？