# RB&WR技术OPL

# 车端快速迭代

1. 需要搭建车端工程师，打通全部软件链路。
   1. 所有车上的环境，detail很多，每个team有自己的流程和注意的点，缺一个dedicate启系统很熟悉的人，目前方式是所有组standby。有下述问题：
      * 从起传感器到最后control能拿到轨迹中间涉及模块多， 不是所有模块都非常稳定，硬件和其他模块缺少monitor方法，如camera出问题时，只能在viper dump log看，需要build up 各模块的健康度monitor
      * 每个模块有自己的trick 和注意事项，没有整合到一份完整的文档让联调的同学（or测试工程师）能够清楚知道问题。
      * 打包cuda相关的库单独放， 需要setup.bash 来 export环境变量， 但是setup.bash需要在bringup 一辆车的时候工程师自己写（很容易忘记）
      * 环境不统一，没有docker，各端debug会消耗很长时间；
      * 车辆无法被工程师远程接入，只能上车debug
      * 需要有onboard资源进程线程监控系统
   2. 对于如何收集incident数据和日志、coredump文件等拿回去debug，目前没有标准流程和做法。
   3. 需要尽快按照咱们之前沟通清楚的incident pipeline提供问题数据，才能加快迭代
   4. 车上录哪些topic的是很随机的，需要有人告知车端同学要录哪些topic，否则都会随机录。需要维护一份topic list。
   5. 需要有专人负责路测，收集完整数据（包括bag，各模块log）上传，反馈问题。现在都是各个模块自己上车。车较少需要排队。起车比较复杂，没有SOP指导，经常起不来，浪费大量时间排队等待。
2. PC & A sample图片压缩，完整车端数据闭环
   1. PC端图片需要继续压缩，才能保证输入和输出都能同时落盘。目前PC端经常无法完整落盘（例如只能录下来2、3个相机，其余topic严重掉帧）
   2. A sample端也需要优化，输入和输出都能同时落盘
3. 车辆标定参数
   1. 目前已经开始多台车进行路测，多台车的上发现的问题需要debug

# 数据闭环和消费

1. 架构interface适配的版本，ms5都没有能够release interface，会block 交付。
2. 需要建立两个分支来进行迭代开发——功能测试分支 and 稳定分支（主分支）
   1. 需要尽快维护一个稳定版本作为baseline，以及定义该分支为稳定分支
   2. 建立功能测试分支，用于功能合进主分支前台架以及车端验证，保证主分支功能稳定提升
   3. 需要有个方式，非WR的软件可以保持版本不变，只有WR 代码update的情况下编出包来做独立验证
   4. 有update的稳定版本和功能测试版本都需要每天打包和上车验证（在缺乏CICD的情况下）。
3. Incident追踪及测试
   1. 按照Jira incident的方式进行fix和track，WR按照一定的频率（一周两次？）持续迭代到集成分支，维护版本稳定性，目前集成分支缺乏CICD，频繁迭代会导致系统出问题概率增大，车端有效跑起来时间变短。
   2. 需要提供一台车给到WR 做功能测试和验证
4. Incident及branch 编译进专区（目前了解的信息是可能在430之后）
   1. yaml文件和interface 维护放在专区，更新可以通过diff来update各端
   2. 专区中才能使用更多的工具保证软件质量和效率（包括工具链，快速上车，CICD）

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **WeRide Request 4/4** | | **RB Reply on 4/6** |
| 车端快速迭代 | 1. 需要搭建车端工程师，打通全部软件链路 |  |  |
| a. 所有车上的环境，detail很多，每个team有自己的流程和注意的点，缺一个dedicate启系统很熟悉的人，目前方式是所有组standby。 | 有下述问题： |  |
|  | * 从起传感器到最后control能拿到轨迹中间涉及模块多， 不是所有模块都非常稳定，硬件和其他模块缺少monitor方法，如camera出问题时，只能在viper dump log看，需要build up 各模块的健康度monitor | RB匹配工程师需要知道如何起各个模块以及各个模块的输入输出是否正常，判断一场模块 |
|  | * 每个模块有自己的trick 和注意事项，没有整合到一份完整的文档让联调的同学（or测试工程师）能够清楚知道问题。 |  |
|  | * 打包cuda相关的库单独放， 需要setup.bash 来 export环境变量， 但是setup.bash需要在bringup 一辆车的时候工程师自己写（很容易忘记） | 单独和文远约会，讨论  1）单独打包cuda库，是否可以改成用博世底软释放的OS包  2）关于环境变量，能否提供完成的需要export的环境变量清单，集成统一放在脚本里 |
|  | * 环境不统一，没有docker，各端debug会消耗很长时间； | 博世内部需要讨论合适的部门来负责维护车端编译运行的环境 |
|  | * 车辆无法被工程师远程接入，只能上车debug | No.  这个无法满足， 不符合security要求 |
| b. 对于如何收集incident数据和日志、coredump文件等拿回去debug，目前没有标准流程和做法 |  | Yes ongoing  draft version by MT on Apr 14th |
| c. 需要尽快按照咱们之前沟通清楚的incident pipeline提供问题数据，才能加快迭代 |  | Yes ongoing  Data compliance kick off on Apr 7th |
| d. 车上录哪些topic的是很随机的，需要有人告知车端同学要录哪些topic，否则都会随机录。需要维护一份topic list。 |  | Yes  PO of each component shall provide the topic list of his/her own component  MT or VER shall collect the list and record the topics based on the list  enable MTA and topic list by component |
| e. 需要有专人负责路测，收集完整数据（包括bag，各模块log）上传，反馈问题。现在都是各个模块自己上车。车较少需要排队。起车比较复杂，没有SOP指导，经常起不来，浪费大量时间排队等待。 |  | 上位机guide已经ready，VER已经熟悉（6，7，8，9 ver）  ECU各模块是ASW的开发比较熟悉，有必要集中起来 |
| 2. PC & A sample图片压缩，完整车端数据闭环 |  |  |
| a.PC端图片需要继续压缩，才能保证输入和输出都能同时落盘。目前PC端经常无法完整落盘（例如只能录下来2、3个相机，其余topic严重掉帧） |  | Status, highly dependent on the ECU resource consumption  如果跑算法后ECU资源比较紧张，确实能录下的资源比较少 |
| b.A sample端也需要优化，输入和输出都能同时落盘 |  | Yes ongoing  Apr 14th |
| 3.incident机制，每30s切一个bag |  |  |
| 4.车辆标定参数 |  |  |
| a.目前已经开始多台车进行路测，多台车的上发现的问题需要debug |  | ? |
| 数据闭环和消费 | 1.需要建立两个分支来进行迭代开发——功能测试分支 and 稳定分支（主分支） |  |  |
| a.需要尽快维护一个稳定版本作为baseline，以及定义该分支为稳定分支 |  | Yes, ongoing after Apr 10th SW Arch for SOP is released |
| b.建立功能测试分支，用于功能合进主分支前台架以及车端验证，保证主分支功能稳定提升 |  | Yes, ongoing after Apr 10th SW Arch for SOP is released |
| c.需要有个方式，非WR的软件可以保持版本不变，只有WR 代码update的情况下编出包来做独立验证 |  | No  not reasonable |
| d.有update的稳定版本和功能测试版本都需要每天打包和上车验证（在缺乏CICD的情况下）。 |  | Yes after a & b are realized |
| 2.Incident追踪及测试 |  |  |
| a.按照Jira incident的方式进行fix和track，WR按照一定的频率（一周两次？）持续迭代到集成分支，维护版本稳定性，目前集成分支缺乏CICD，频繁迭代会导致系统出问题概率增大，车端有效跑起来时间变短。 |  | Yes  每周二，四定期看jira bug list |
| b.需要提供一台车给到WR 做功能测试和验证 |  | 截至430之前, PC车可以给 |
| 3.Incident及branch 编译进专区（目前了解的信息是可能在430之后） |  |  |
| a.yaml文件和interface 维护放在专区，这样可以方便user更新的时候可以通过diff来看美版的差异，便于更新效率 |  | Yes, on going |
| b.专区中才能使用更多的工具保证软件质量和效率（包括工具链，快速上车，CICD） |  | Yes, on going  Continuous integration+continuous SW package delivery+selected function test |