# Qt多项目工程安排

## 原则

结构不同的两个分支，而是两个仓库，两个项目部门，两个管理部门。这些结构方式不同的仓库在一起构成架构，架构总和就是一个系统。

## 前边的问题

在单个工程部署中，对生成文件的位置的考量引发了这些。

1. 对于版本注明重点功能和日期。
2. 即便有已知bug也不进行替改。直到下一个时间点下一个版本发布。紧急修正不作为版本发布，作为fix存在，将fix及时合并到develop当中。
   1. Fix分支，作为master的延续发布分支，所有和fix有关的问题都在fix上解决。
   2. Fix分支每隔一段时间，都要合并到develop分支，而不是master分支。
3. 现实不是这个样子，现实是develop中的更改不被版本关注者重视，他们采取的方案是发现了新问题解决新问题。“改出了新问题”似乎是他们的笑点和批评的位置。所以这块的目标添加：不能出现改出新问题，但是可以出现我改的还没改完。
   1. 禁止连带问题；
   2. 允许还有改动；
   3. 要求产品合格。
      1. 合格率，bug数目。
      2. 提高成品率和降低问题率，方向是关注到更多的问题，细节，解决更多的问题。
      3. 一个产品的开始全是问题？不，已经会了一些，还有一些，这些都要实现。
4. 到了发版本的时间合并develop到master。
5. 工作环境，包括我，无法承担这种开发能力要求，依然按照svn的开发方式，有种新结构的想法就要新创建工程。
6. 发展方向是，以上这个样子，但是android开始以后，就能从svn转变过来吗？呵呵，git不符合人的新结构思维方式，也不符合现实的发展方式，怎么可能完全凭想象和深陷其中的对比？
7. 从现实情况，我创作新结构的时候，就会更换工程，所以git的分支肯定不能达到svn的分支的效果，但是git的效率很高。Git的分支仅仅是特征，特性。我认为git还是很好的，新结构开始新工程，还在一个svnrepo里干什么呢？Svn的分支不应当还有什么关系，但是git的分支却容易处理关系。
8. Svn在分支的问题上，不容易处理，而git容易处理。而且git抛弃了不同结构的代码还在一起开发的弊端。结构，方式都不一样了，为什么还在一起？所以git比svn先进，思想先进，表现行为先进。

## 考量

结构不同为什么还要提交在一起管理呢？结构/方式不同，不往相同的代码库提交，也不是

不知道svn服务器的实现是否将不同分支分为了不同镜像。但是在此处，结构不同，不在一起管理。

这就是现在流行的树形管理结构。

Git的分支，指特性分支，达不到结构分支。

SVN的分支，结构分支，还能在一起管理吗？特性代码分支为什么是两套工程？上层封死，还能和其他的工程有效沟通吗？历史丢失不管吗？

Git下一个更新，需要能够容易的将两个仓库进行合并。容易的将两个仓库进行分离。

## 多工程部署

结构不同，构成相同功能层次的，并列放置。

构成不同功能层次的，根据层次分开放置。

总体达成一个操作系统的结构。

没有接触到，所以持续更新。

1. 资源位置
   1. 从网络和从其他等处获取到的资源存在此处。
   2. 放在Refrence文件夹里。