## 进度记录

1. 4月19号，刚刚从qt学习中出来，开始着手方案，明天找产品对接方案详细需求。
2. 4月20号（明天），对接方案需求，了解需求，分析需求。
   1. 现在是20号上午9点，已经和产品经理（邱辉军）沟通，并确认了大致的实现样貌，目前需要有版本服务器的支持，经理在协调是否能够使用华为云作为版本服务器，等待回复。
   2. 首先做一套客户端UI出来。（20号任务）
3. 4月21号，今天需要完成需求文档模板的搭建，完成基本需求内容填写，为后期正式需求下来，打一个基础。
4. 然后下一步是Qt的网络编程。因为需要网络进行最新版本的下载，替换本地旧的版本，从而达到升级的目的。
5. 4月22号，需要完成git仓库管理版本库，提供可供测试所用的版本库，然后从git版本库中实现软件下载自更新需求。
6. 作为假设：如果是直接拉取setup.exe下来直接执行安装，安装完成后执行旧版本的unset.exe，这样旧版本就会被清理，新版本创建桌面快捷方式，提示客户重新执行程序，这样就是一个完整的升级过程，安装路径会从（例如从1.0升级到2.0版本）1.0变成了2.0的路径，只是安装路径的改变完成了升级。
7. 过程：
   1. 第一步：更新方式，程序启动自动检查更新，检查方式，读取服务器版本文件（版本文件记录本版更新详细信息），与本地版本文件进行版本对比，如果服务器版本比本地版本新则执行更新，并替换最新版本文件。
   2. 第二步：如何更新，下载服务器上的最新版本安装包，并执行安装，安装目录选择本地旧版本安装目录，执行默认的自动安装，安装脚本执行完成后，提示客户重新启动客户端完成安装，并执行旧版本卸载脚本，将旧版本卸载掉。
   3. 这样就完成了简单的版本更新过程。
8. 4月23，基本完成client的UI。
9. 实现方案基本定型，更新程序为独立可执行程序，每次执行主程序调用更新程序，进行更新检查（XML文件提供<version>标签提供版本），没有更新则退出更新程序，有更新则弹出更新窗口，客户端选择是否更新，更新过程，下载最新可更新文件替换旧版本，更新完成重启客户端。
10. 4月26，完成了网络下载，XML文件对比，最新版本文件的筛选并下载；待完成主程序和更新程序的交互。
11. 4月27，和鹏哥讨论结果

**关于UI界面：**

1. 更新界面：更新日志显示版本基本信息，不需要显示更新文件是什么（客户不关系更新的文件内容，主要体现软件功能变更等信息）。并且XML文件的格式不适于后台开发人员的编辑，需要以一种更加简单的方式提供编辑（如txt文件，可以参考git上的版本文件）;
2. 正在更新界面：保留outputEdit（内容显示）部件，用于显示正在更新的文件，下载进度，如果由于网络原因导致的失败，进行显示，并显示解决办法。进度提示使用进度条的方式进行。
3. 更新完成界面：不要弹出多余的对话框，更新完成后提示客户端是否重启（重启的话执行更新后的版本，不重启则保留旧版本使用，下次启动的时候新版本启动）。

**UI界面处理方案：**

1. 更新界面：outputEdit中间显示部件，显示版本信息，版本信息以txt的方式存储在后台，txt的方式方便于开发人员的编辑，显示的方式是，直接读取整个txt文件内容进行逐行的显示在outputEdit中。同时增加一个取消更新按钮。
2. 正在更新界面：保留outputEdit部件，用于显示正在更新的文件信息，包括下载方式、当前下载到哪个文件、并且改文件下载到了哪个路径、网络错误导致的下载中断错误的提示和处理方式、下载完成后显示下载成功。在下载文件的时候，当前下载的文件前面显示一个图标（公司logo）。在outputEdit部件下方显示一个下载进度条。
3. 更新完成界面：保持正在更新界面的下载完成状态，进度条隐藏，显示两个按钮，确定和取消，并提示是否立即重启。

**关于下载：**

1. 使用FTP服务进行下载，处理下载中断，或者出错的情况。

**下载解决方案：**

1. 现在是使用的http服务进行下载，改为FTP服务，获取更优的下载体验。
2. 下载过程中的错误处理。
3. 4月27晚，UI界面基本按照上面的方案进行了改进，界面改进完成，下面是网络和逻辑。目前版本控制文件和版本文件读取解析都是正常的，更加xml版本控制文件筛选出来的需要下载的文件，下载过程也是正常的，但是在复制替换的过程中有一些问题，复制过来的文件都是空的。
4. 4月28号下午，下载拷贝过程正常，整体功能运行也正常，然后把协议改为FTP协议。