|  |
| --- |
|  |
| 问题汇总 |
|  |
|  |
| ©MetaShare Inc. |
| **8/3/2017** |

|  |
| --- |
| 本文eService开发过程中的问题 |

目录

[1. 引言 1](#_Toc500859620)

[2. 第五期总结 1](#_Toc500859621)

[2.1 需求类和管理类问题 1](#_Toc500859622)

[2.1.1 任务划分不够细 1](#_Toc500859623)

[2.1.2 需求传递不够全面和及时 1](#_Toc500859624)

[2.2 好的地方 1](#_Toc500859625)

[2.3 不好的地方 1](#_Toc500859626)

[2.4 需要改进的地方 1](#_Toc500859627)

# 引言

Eservice项目从2017-05-11开始对整个架构需要调整，本文包括每天早会后对早会上讨论事情的汇总，包括早会讨论了什么事情，每天做什么改进，接下来要安排什么会议讨论什么事情等等，以供大家查看。

# 第五期总结

## 需求类和管理类问题

### 任务划分不够细

对testdata/service/dao/database的任务没有区分开，本次实现是需要将页面用Mockservice先展示出来，所以测试数据应该是先写的，应该放在每个任务下面去实现。

### 需求传递不够全面和及时

有些时候需求更新并整理后未与整个团队沟通达成共识，导致有些人对新需求理解不够。

## 好的地方

1. 整个团队成员分工比以前明确，每个人都清楚自己的职责，协作也更顺畅。
2. 每天坚持写Sprint memo和工作汇总，这样能及时的把工作进展告知Adam，方便双方了解彼此的进展情况。
3. 在与Adam协作过程中，发现了团队很多问题，并及时指出来，这样对团队的提高很有帮助。
4. Adam作为架构师能在架构上把控的非常好，这样不会让我们走弯路。

## 不好的地方

1. 有的任务划分不够合理，比如将service\dao\testdata\database
2. 对于不清楚的任务没有及时确认，比如登录，在没有确认的情况下先做了登录页面，而这个页面是不需要的。
3. 本次迭代关于计算的需求理解花费时间较多，离交付还有最后一周的情况下才把需求整个理顺，导致我们的时间很紧张，造成按时交付有风险。
4. 怕做错事情挨说所以做事情时异常小心翼翼，导致有的事情可以开始做了，但有一点点不太确认，还要等着Adam第二天确认后再去做，这样会导致时间延迟问题。

## 需要改进的地方

1. 开发过程中代码规范问题较多，比如命名、逻辑处理

改进建议：在项目开始之前，组织会议将命名规范以及其他常用的代码书写方式让每个人都清楚并达成统一

1. 测试代码的问题较多

改进建议：组织项目成员review本次项目中测试代码，并完善实现规范，让每个成员理解达成一致。

1. 对Syncfusion控件的应用很不熟悉，导致前端开发费时较多

改进建议：在公司自己的项目（Scrum Management）中应用Syncfusion常用控件，并掌握各个控件的使用方法。

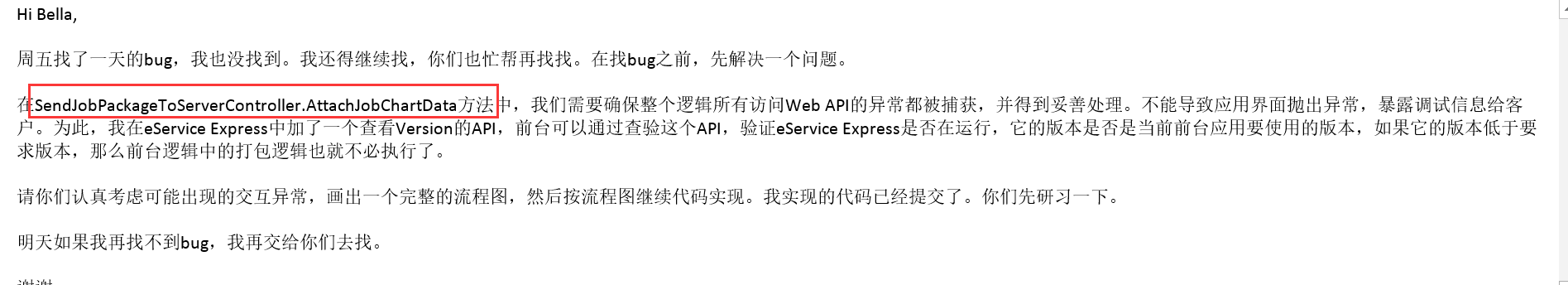
1. 项目经理Bella对代码质量监控不够

改进建议：项目经理应加强对代码的理解，抽时间review代码，及时发现问题并提醒改正。对于架构的设计我们目前能力较差，这方面能力急需提高

改进建议：适当安排一些架构设计的工作给年长的开发员，往架构师方向培养

# 20181211团队检讨

## 上下文

1. 20181210Adam发来邮件需要我们改个问题，已经指出了具体修改的方法，但Colin和Bright还是修改错误，改了别的方法。
2. eService的代码Adam已经提交，WebAPI到接口是已经定义了，但是我们在实现的时候又修改了这个定义。
3. Adam交代的任务有明确到代码位置，而且所有到描述都是前台逻辑，但是我们画的图却是Express中的打包逻辑。

## 我们的问题

1. 对于Adam发的邮件，我们没有认真阅读，导致已经非常明确的修改位置也改错了。
2. 我们对上下文的理解不到位，或者对已经有的接口没有认真思考。

## 改进方法

1. 对于那些因为自己读邮件不仔细造成的错误以后坚决避免，我们已经制定了惩罚措施。
2. 上下文的理解一直都是我们到软肋，我们必须花心思在理解问题的上下文上。所以要求每位项目成员严格遵守我们到开发规范，按照开发过程进行开发，所有的TestCase必须认真写，写完后必须其他团队成员Review，同样提交的代码也一样。如果这个任务客户指出有问题，责任人和Review的人一起受罚。

# 第十二期总结-20190606

这几天我们也好好总结和反思了一下问题，就如你所说，因为eService的规模太大，所有有些问题在其他项目上没有遇到过相应的处理方式也不太好。

我们主要针对你的工作方式和我们双方的配合上总结了以下几点：

1. 每次你那边的发布范围我们不明确：比如你一般会告诉我们下周三要发布online，我们可能只关注online新做的功能，而你可能还会发布eService，或者其他项目，希望你能把整个发布范围明确告知我们，我们进行全面测试；
2. 每个功能影响到的哪些其他项目的功能，我们并不是非常清楚：一般你让我们做哪些功能，可能我们只做了功能，而对这个功能可能会引起其他项目的功能改变，我们并不是非常了解；
3. 业务逻辑没有及时传达到我们，有些业务逻辑你觉得比较复杂所以你自己做了，但是因为我们并不清楚所以这些不清楚的地方就会为以后留下隐患；
4. 如果出现了错误，希望能够告诉我们是哪里出错了，我们需要注意些什么，比如这次的错误我们只能从代码中看到你的修改记录，而不清楚问题的根源在哪里，以后可能这块还会出错；
5. 每次创建的数据都用在哪里，尤其是用在其他项目的哪些地方，我们希望你能清晰的传达给我们；
6. 咱们项目的开发周期较长，比如Online咱们从17年就开始做了，有很多需求变动较大，而我们没有维护一个相对完整的文档，导致后面进来的成员对整个业务逻辑不是非常了解，这是我们的问题。
7. 对于设计上你对我们的指导已经非常细致了，我们觉得受益很大，而且设计也非常清晰，这块没什么问题。

解决方案：

1. 我们需要整理一个完整的需求文档；

                虽然Trac很好用，但是如果要找到两年前做的某个需求以及上下文还是挺不容易的。所以建议由我们这边针对每个项目整理一个完整的Word版需求文档，只保留最新的需求，历史记录可以在Comments中记录或者查看SVN记录得到。

1. 对于新需求希望你能传达更多的信息，比如上面2-5点所说；
2. 增加代码review的时间，每周我们可以抽出1-2个小时互相Review code，或者你有时间的话也每周抽出固定的时间来review，这个咱们再讨论。

最后我们相信我们会改正已有的问题，会做的更好！

顺祝端午节快乐！