|  |
| --- |
|  |
| Meeting Record |
|  |
|  |
| ©MetaShare Inc. |
| **5/11/2017** |

|  |
| --- |
| 本文包括EService项目在实施过程中的所有会议纪要 |

目录

[1 引言 1](#_Toc487622467)

[2 2017-05-11会议纪要 1](#_Toc487622468)

[2.1 5-11下午 1](#_Toc487622469)

# 引言

Eservice项目从2017-05-11开始对整个架构需要调整，本文包括整个项目开发过程中的会议纪要，以供大家查看。

# 2017-05-11会议纪要

会议时间：2017-05-11 11:00-12:00 14:00-17:00

参会人员：Adam，Luke，凌飞，赵宝红，鬲文娟，邵保卫、林鑫、李英，毕欣欣

会议主题：关于EService项目的规划和目标

会议内容：

1. Adam阐述了EService项目的目标：
2. 在可预期的未来（两三年之内），WinForm的eService将做为Admin Console存在，与Web应用并存，直到Web应用足够成熟。所以说，如果有新增功能，也许我们两边都做开发，WinForm只按现有标准架构，最小工作量的方式去做，不做可用性改进考量，而Web要考虑易用性。所以在重构的考虑上要复用。由于使用WinForm有限制，Datagrid的处理尤其突出，所以打算将EService迁移到web平台，采用Web逐步替代Winform的策略。
3. Adam对接下来的工作安排做了简单描述
4. 5月12日会将已经完成的原型发给我们，我们根据原型做正式的架构原型。
5. 下周需求讨论，正式开发预计从5月22日开始
6. 预计6月中旬完成第一期的开发，6月下旬现场测试。
7. Luke对资源进行了初始分配

第一期我们会投入4-5个开发人员进行架构设计和对新人的培训，涉及到的人员有：凌飞、邵保卫、赵宝红、鬲文娟、林鑫、张磊

1. Luke对设计文档进行了讲解

文档位置：<https://metashare.ca:9880/Projects/trunk/P20-EService/D-设计>

在讲解过程中大家提出了问题并进行了讨论。

1. 最后对接下来的工作进行安排
2. 参会的每个人都必须尽快熟悉并理解设计文档中的内容
3. 对.net core相关的技术需要尽快熟悉并做Presentation

## 5-11下午

注意创建branch以实现持续化开发

做Prototype-Windows service or Concole ap

完成一个不支持互联的service

完成一个支持web client的service

完成一个通过local gateway访问的

回归测试需要提高，自动化测试脚本

# 2017-10-12会议纪要

会议时间：2017-10-12 10:00-11:00

参会人员：Adam、林鑫、毕欣欣

会议主题：关于Eservice第六期需求讨论-Dispatch Process

会议内容：

1. Adam介绍了接下来要做的工作：

由于详细需求尚未整理出来，所以Adam先把后面要做的工作大致做了介绍，本次预计11月底之前要做出东西，Scope需要再确认。

* 客户通过电话跟Sanjel的人联系，告诉几天后Rig会到哪个地方，Sanjel需要提前做好准备，待时间确认后再次电话联系Sanjel，Sanjel会根据Program中的内容以及时间创建CallSheet。需要对人员、设备、工程车、特殊设备进行调度。需要知道它们的可用时间，类似于日程管理或调度系统。第一阶段先做单一的调度系统，后面再做团队调度系统。
* 另外需要对Rig进行追踪，记录Rig的位置、状态等。类似于课程表、排课系统，需要找相关资料进行模型的研究。

1. 接下来对工作进行了安排：

林鑫需要尽快熟悉eService的架构，并将online的环境跑通，目前数据量大时页面加载较慢，需要解决这个问题。

# 2017-12-22 会议纪要

会议时间：2017-12-22 10:00-11:00

参会人员：Adam、刘斌、毕欣欣

会议主题：关于Eservice第七期事务日志处理的需求

会议内容：

1. Adam介绍了关于Transaction logging functionality部分的需求：

第一种实现：

对于Logging部分，现在的问题是通过在Online上对Call Sheet等业务实体的操作，无法追踪到谁修改了哪些内容，应该可以记录谁在什么时间修改了什么内容。应该从服务层来做而不是逻辑层来做，用注册事件的方式，对某个实体进行监控，写到日志中。.NET FrameWork或者.NET Core中应该有Logging的功能，需要去了解下。同时参考OPS

第二种实现：记录用户登录和登出的时间和状态，在代码中建一个新的实体来记录用户的状态，每个用户只有一条记录，当用户登录或者登出时只修改该条记录。

目前我们先实现第二种。第一种可以考虑，后面再实现。

# 2018-03-09会议纪要

会议时间：2018-03-09 13:00-15:00

参会人员：Adam、刘斌、林鑫、赵园园、毕欣欣

会议主题：关于数据库迁移的问题讨论

会议内容：

1. Adam对于数据迁移部分提出了要求，数据迁移需要分批进行。
   1. 整理数据库表，数据分为Reference数据和Transaction数据，其中Transaction主要包括四个主表：CallSheet、JobPackage、Service Ticket、Service Report。
   2. 先查出主表的ID，然后再根据ID查出来跟它相关的数据，分批转移；
2. 数据库迁移需要制定执行计划，包括：
   1. Sql Server的准备工作
   2. Service停掉
   3. 备份Oracle
   4. 开始执行迁移工作
   5. 测试
   6. 恢复服务
   7. 应急方案
   8. ……

我们需要跟Adam一起完成。

1. 我们目前只关注了Oracle和SQL Server不同的数据类型，而没有关注实体中类型的处理，需要做的工作：
   1. 列出所有的实体包括的数据类型，它们对应的Oracle字段类型，精度，长度，对应到SqlServer数据库中使用哪种字段类型更合理。
   2. 逐个检查实体，并分析不同数据类型的处理。比如相同的字段类型可能出现在不同的实体中，它对应的SQL Server的数据类型会不一样。
2. 对于SQLServerDaoBase的重构：
   1. OracleDao和SQLDao要严格区分开
   2. DaoBase中的方法：LimitRecords( ) PaginatedQuery( )需要重构，移到OracleDaoBase和SQLDaoBase中，不需要重载。
   3. 对于DateTime的处理，需要分析一下如果SQL Server中对应DataTime2类型，是不是代码中就不需要再进行处理。
3. SQLDao写完后需要写测试，从Oracle Dao将数据读出来写到SQL Dao中。将结果与Stone整理的脚本执行的结果进行对比和分析。
4. 为了完成1中所说的分批转移，需要做的工作：
   1. 整理所有的表，并进行分类；
   2. 数据库转移工具修改：
      1. 先按照大体原则先转，生成数据库脚本；
      2. 其他根据步骤3中分析整理出来的规则需要单独处理，附在i生成的脚本之后
      3. 整个过程需要可以反复测试和执行。
5. 生产数据库会用SQL Server 2014 Standard版本，需要Oliver协助准备测试环境；
6. 由于Sql 2014支持Sequence对象，所以之前对于Sequence的处理需要调整；
7. eService Dao的处理：
   1. 将eService项目中的IDao/Dao/DaoTest/TestData复制到CommonLibrary/branch上，并修改为SQL Server的，在Branch中完成所有的切换，不需要再对eService项目进行处理。
8. 数据库转移的时间尽量控制在半小时之内，Stage环境需要Adam与公司讨论之后再定。

# 2018-03-20会议纪要

会议时间：2018-03-20 11:00-12:30

参会人员：Adam、刘斌、林鑫、赵园园、毕欣欣

会议主题：关于数据库迁移的问题讨论

会议内容：

就目前数据库迁移过程中的问题进行了讨论：

1. Datetime2类型：在对数据类型处理时，发现使用datetime2还是不能解决日期溢出问题。经过讨论目前的实现方案是：对于SQL Server数据库中的最小值1753-01-01和Oracle数据库中的最小值0001-01-01在代码中统一作为MinValue看待。需要Bright测试一下取数据和存数据是否正常运行，最好能支持datetime2类型，这样以提高数据转换速度。
2. 测试数据正确性：在测试过程中需要建一个测试项目，同时对SQL Dao和Oracle Dao进行测试，使用相同的数据对比获取到的对象值，以保证转换后的数据正确性。
3. 数据转换：按照四大类准备工作模型，记录实体的ID，测试每一大类数据的转换时间。可以按照实体中的时间属性获取到一定时间之前的数据，Security Data和Reference Data在小窗口重新转一次。各个实体对应的时间字段：

|  |  |
| --- | --- |
| TABLE\_NAME | COLUMN\_NAME |
| CALLSHEET\_HEADERS | GENERATED\_DATETIME |
| JOBS | TIME\_CREATED\_ON\_SERVER |
| SERVICE\_REPORT\_HEADERS | TIME\_CREATED\_ON\_SERVER |
| SERVICE\_TICKETS | TIME\_CREATED\_ON\_SERVER |

接下来继续实现架构原型和全面测试，测试完成后进行Security数据的处理，接着是Reference数据、业务数据。