1. **中蛋实施的整体做法**

对于中蛋实施任务的功能或需求里，可以分为2种类型的需求功能：

1. 比较通用性和代表性的一些业务，是需要做到产品里，成为产品一部分的
2. 是新蛋特有的，不具有通用性和代表性的业务，不会成为产品的一部分的，比如DIY，阿斯利康等相关的业务

针对上面两种类型的功能，我们在后续的实施方式也不同，分别为：

1. 针对a)点的功能，我们会将代码直接做到现在的产品代码里就好了，因为其本身就是产品一部分的，以后需要拿出来为外包项目所用的
2. 针对b)点的功能，我们实施的做法将会是：仍然直接做到现有的产品代码里（方便以后中蛋直接拉个版本出来维护），但是需要和产品代码稍作隔离，以便产品后期的抽离；（通过在工程下建一个单独的目录来存放为中蛋定制的代码）
3. **中蛋实施的具体方案**

我们具体来分析Portal端和Service端的每一个工程：

1. **在Service端的处理方式：**

按照每个工程，会有不同的处理方式，具体如下：

1. **BizEntity工程**

在BizEntity下添加定制化目录，比如添加一个NeweggCN目录，在该目录里再按照Domain划分建立相应子目录，将每个Domain需要的BizEntity放到对应的NeweggCN/[DomainName] 目录里（注意命名空间请保持和原有Domain的命名空间一致）

在定制化的时候:

(a)如果是需要新增BizEntity的，那么直接新建新的BizEntity的class即可

(b)如果是要在原有的BizEntity里新增属性的，那么请使用部分类(partital class)的方式，将定制化的属性部分拆分出来，放到定制化的目录里（这里不使用继承的方式，因为json序列化会出问题）

1. **BizEntity.Enum工程**  
   直接在原有枚举上修改即可（但需把修改点记录下来，后面做产品抽离的时候，需要回滚掉，就现在梳理的功能来看这种情况不多）
2. **QueryFilter工程**

做法和BizEntity一样，参照上面 (1) BizEntity的说明

1. **IBizInteract工程**

在IBizInteract下添加定制化目录，比如NeweggCN，将每个Domain需要新增加的交互接口放到该目录下；新增的接口都需要派生自该Domain原有的接口，然后再添加所需要的新的方法定义

1. **Utility、Utility.DataAccess、Utility.WCF工程**

定制化的过程中基本不需要修改这3个技术框架层次的工程，如果确实遇到一些有需要的且比较通用的工具或特性需求，可以再在这3个工程中添加完善；

1. **每个Domain的Restful**
2. **每个Domain的AppService**
3. **每个Domain的BizProcessor**
4. **每个Domain的IDataAccess**
5. **每个Domain的SqlDataAccess**

这5个工程，都是在工程里建立相应的定制化目录，比如NeweggCN，将定制化添加的代码都放到该目录下；

对于新增功能，基本上就是增加相应的全新的类；

如果是对产品已有功能的调整（比如部分逻辑的调整定制化），那么建议是优先使用 新增类继承原有类后override相关逻辑的处理方式（主要增对AppService、BizProcessor和SqlDataAccess）；

对定制化所新定义的IBizInteract的实现类，也是先派生自新定义的IBizInteract接口，同时再继承原来的IBizInteract接口的实现类，这样就只用为新增的接口方法做实现即可；

（只是需要注意MEF的导出，这个类需要导出新旧2个IBizInteract接口，我们需要做个简单的配置，用新作的实现类来替换以前的实现类，而在需要使用新接口的方法的地方，就以新接口为ObjectFactory的泛型参数来创建对象）

1. **在Portal端的处理方式**：
2. **BizEntity工程**

直接引用Service端的BizEntity代码，所以跟随Service端的BizEntity的变化而变化

1. **BizEntity.Enum工程**

直接引用Service端的BizEntity.Enum代码，所以跟随Service端的BizEntity.Enum的变化而变化

1. **Basic工程**

定制化的过程中基本不需要修改这个工程，如果确实遇到一些有需要的且比较通用的工具或特性需求，可以在这个工程中添加完善；

1. **每个Domain的UI工程**
2. 对于新增的功能，在工程里建立相应的定制化目录，比如NeweggCN，将定制化新添加的代码都放到该目录下
3. 对已有的界面或者界面逻辑需要做调整的，直接在产品代码中调整即可（但需把修改点记录下来，后面做产品抽离的时候，需要回滚掉，就现在梳理的功能来看这种情况不多）