**Netsharp企业业务基础平台**

**体系结构**



作者： 秋时

日期： 2013-12

版本： v1

# 介绍

Netsharp是一个企业基础业务管理平台，介绍Netsharp分三个系列，分别是：

1. Netsharp快速入门
2. **Netsharp体系结构 (本文)**
3. Netsharp开发

本文是第二个系列。本文首先介绍了Netsharp的设计思路，然后说明业务基础平台的定义；本文的重点是介绍Netsharp中四个核心的概念:插件和资源、业务模型、工作台、六架马车；最后本文还介绍了Netsharp的功能清单以及路线图。

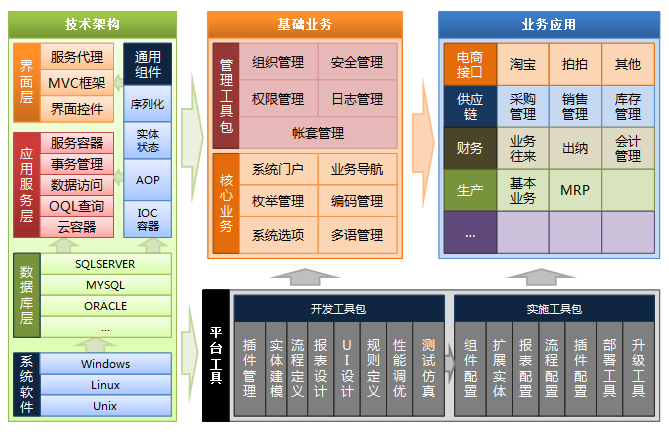
本文读者需要有5年以上管理软件行业研发经验。

# Netsharp设计思路

## 管理软件七阶段

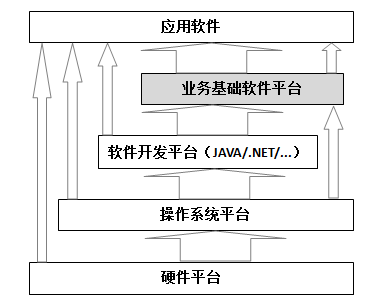
|  |  |  |  |
| --- | --- | --- | --- |
| **编码** | **阶段** | **特征** | **达到的条件** |
| 1 | 硬编码 | 直接使用IDE提供的工具进行代码的编写，很少考虑使用第三方的组件，或者自己封装正规的组件。一切代码只要能实现客户需求即可，不愿意或不懂得多做一些从技术上有益于产品的工作。 |  |
| 2 | 部分组件 | 1，开始需找第三方的组件，包括界面控件、持久层、配置组件  2，自己写一些常用的组件  3，可能使用一些第三方工具，如PD等进行快速的代码生成  4，有初步的分层思想 |  |
| 3 | 系统框架 | 自己写或者使用第三方的组件，特点是框架的全面性，在每个主要的技术点都有系统框架的支撑 | 1，团队技术负责人对技术热爱，希望写出高质量的代码  1，会对比分析各种部署方式下的开发模式包括，会分析前段的各种展现方式（Silverlight/WinForm/WPF/HTML4/HTM5）,会分析分布式的各种组件（webservice/json/remoting/wcf/socket）,分析各种持久层框架，等等    2，熟悉常见的业务展现形式，从技术角度进行整理，包括单据、列表、报表、选项、工作流、等等 |
| 4 | 技术平台  (业务基础软件平台) | 1，技术平台本身即为产品  2，在系统框架基础之上，提供友好稳定的工具维护框架下的元数据，并提供二次开发接口和工具。    3，提供基础业务功能，包括权限管理、组织机构管理、单据编码管理、枚举管理、选项管理、工作台界面等等 | 1，被动使用平台：业务本身复杂，不使用技术平台，导致研发成本太高  2，主动使用平台：技术负责人愿意在系统框架之上提供优良的开发工具让业务开发人员进行元数据的维护。     技术负责人愿意对实施人员，或者有二次开发能力的客户提供接口和平台开发工具  3，产品经理有明确的文档要求客户要自定义产品行为的 |
| 5 | 业务平台 | 1，在技术平台之上，业务按照领域对象为粒度，每个领域对象可以单独存在，可以和其他领域对象组合成新的的业务，可以扩展功能，可以二次开发覆盖原有的功能  2，在第一点的基础之上，可以基于流程调整领域对象的流程关系 | 1，领域的总体设计师期望做出结构优良的产品  2，产品经理对业务的分析抽象比较合理 |
| 6 | 软件生命周期管理 | 1，技术平台和业务平台是进行产品的研发，本阶段是为这两者提供项目管理工具，此工具可以是自己研发的，也可以是继承第三方产品到自己的产品中。  2，功能包括源代码管理、需求管理、BUG管理、升级和补丁管理等 | 1，研发负责人愿意使用科学的管理手段进行项目管理  2，研发负责人愿意在自己的管理软件平台上扩展出项目管理的功能 |
| 7 | 软件生态环境 | 1，前六个阶段是在产品厂商的角度来看的，此节点是站在整个社会的角度来看的。  2，生态环境允许第三方开发人员和厂商在现有的业务平台上扩展业务功能。也允许在技术平台上构建自己的特殊的业务。典型的例子有Eclipse和淘宝电商平台。  3，产品厂商和第三方互惠互利，合作共赢共同达成一种生态圈。 | 1，技术团队、业务团队、销售团队三者目标都有志向，目标远大  2，产品真的能为客户带来价值，物美价廉  3，平台能为第三方的开发人员、销售人员带来利益  4，团队有卓越的业务能力和推广能力 |

## Netsharp逻辑结构图



# Netsharp平台定义

## 业务基础平台



## 平台定义

业务基础平台 = 1.系统框架

+ 2.元数据

+ 3.平台工具

+ 4.基础业务

+ 5.二次开发

此五个概念请参考[管理软件七阶段](#_管理软件七阶段)。

## 平台即产品

### 最终产品和平台产品

1. 最终产品：

使用Netsharp做出来的产品，如ERP系统，用户是最终客户；

1. 平台产品：

即Netsharp平台，用户是实施和开发人员，他们使用Netsharp开发最终产品，或者开发项目。平台产品有自己独立的版本。

### 平台产品使用平台开发

平台的功能是通过平台自身的功能开发的，如：

* 1. 平台提供了插件机制，平台自身也是基于插件的，而且平台工具也复用了平台的通用插件功能
  2. 业务模型时实体元数据描述的，实体元数据自己也有实体元数据
  3. 界面显示的列表是列表元数据管理的，而列表管理界面的列表本身也有列表元数据

这样的做的带来的特点和好处是：

1. 平台易用：

平台开发人员是最早的平台用户，所以更能站在用户的角度不断改进工具的易用性。这和目前国内大部分的平台工具平台开发人员自己不使用用有明显的区别。

# Netsharp平台管理

## 插件

插件框架是越来越流行的一种开发方式。插件做的比较好的常见于IDE，如Eclipse（基于OSGI标准）、Netbean、SharpDevelop。国内西安有一家公司实现了.NET平台的OSGI标准，算是做得比较规范的。除此之外笔者没有见过国内其他的比较正规的插件体系，大部分声称系统支持的所谓插件都是伪插件，从技术角度来说不过是工厂模式加根据配置反射创建对象而已。笔者认为插件系统有三个必须要素：

* 1. 插件，一个插件有多个路径；插件从物理上可脱离产品独立存在
  2. 扩展点（Netsharp叫路径），扩展点可以横穿多个插件
  3. 插件项支持多种类型，如菜单、类型、图标等。插件项类型本身也有扩展点。

## 实体模型

## 资源

## 工作台

### 部件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **编码** | **名称** | **控制器** | **工具栏** | **说明** |
| 1 | EntityList | 列表.主列表 | Netsharp.Commerce.EntityListPart | Netsharp/Commerce/EntityList/Toolbar |  |
| 2 | Datagrid | 列表.只读 | Netsharp.Commerce.DatagridPart | Netsharp/Commerce/Datagrid/Toolbar |  |
| 3 | EntityDetail | 列表.单据列表 | Netsharp.Commerce.EntityDetailPart | Netsharp/Commerce/EntityDetail/Toolbar |  |
| 4 | VoucherDetail | 列表.单据明细列表 | Netsharp.Commerce.VoucherDetailPart | Netsharp/Commerce/Voucher/DetailMenu |  |
| 5 | Detail | 列表.细列表 | Netsharp.Commerce.DetailPart | Netsharp/Commerce/Voucher/DetailMenu |  |
| 6 | DatagridSolution | 列表方案 | Netsharp.Commerce.DatagridSoltionPart | Netsharp/Commerce/QuerySolution/MainMenu |  |
| 7 | ReportList | 列表.报表 | Netsharp.Commerce.DatagridReportPart | Netsharp/Commerce/Voucher/DatagridReportMenu |  |
| 8 | VoucherForm | 表单.单据 | Netsharp.Commerce.VoucherFormPart | Netsharp/Commerce/Form/Navigation/Toolbar |  |
| 9 | CustomerForm | 表单.可保存 | Netsharp.Commerce.FormPart | Netsharp/Commerce/Form/Edit/Toolbar |  |
| 10 | FormPart | 表单.属性 | Netsharp.Commerce.FormPart | Netsharp/Commerce/Form/Property/Toolbar |  |
| 11 | ReferencePart | 表单.引用表单 | Netsharp.Commerce.ReferencePart | Netsharp/Commerce/Voucher/ReferenceMenu |  |
| 12 | ReportStatistics | 报表.统计表 | Netsharp.Grus.SummaryReportPart | Netsharp/Commerce/Datagrid/Toolbar |  |
| 13 | ReportRdlc | 报表.交叉表 | Netsharp.Grus.RdlcReportPart | Netsharp/Grus/RdlcReport/Toolbar |  |
| 14 | ReportGrus | 报表.GRUS | Netsharp.Grus.GrusPart |  |  |
| 15 | ReportQueryFilter | 报表查询条件 | Netsharp.Commerce.RdlcQueryFilterPart | Netsharp/Commerce/QueryFilter/MainMenu |  |
| 16 | ReportSolution | 报表方案 | Netsharp.Grus.ReportSolutionPart | Netsharp/Grus/ReportSolution/Toolbar |  |
| 17 | TreePart | 分类树 | Netsharp.Commerce.TreePart | Netsharp/Commerce/CategoryTreePart/Toolbar |  |
| 18 | QuerySolution | 查询方案 | Netsharp.Commerce.QuerySolutionPart | Netsharp/Commerce/Solution/MainMenu |  |
| 19 | QueryFilter | 查询条件 | Netsharp.Commerce.QueryFilterPart | Netsharp/Commerce/QueryFilter/MainMenu |  |
| 20 | Customer | 自定义 |  |  |  |

## 六驾马车

Netsharp有六驾马车的概念，六驾马车是从技术角度对开发六种场景抽象，他们分别是：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **说明** | **场景** |
| 1 | Command | 命令 | 界面操作的一个触发，常用的有Button、MenuItem、Toolbar的事件 | 系统点击“退出”按钮，执行Netsharp.Platform.ApplicationExitCommand |
| 2 | Trigger | 字段触发器 | 某一个字段被修改之后，引起其他字段的计算 | 修改单价和数量时，自动计算金额 |
| 3 | Validation | 字段验证 | 数据合法性验证 | 保存时的邮件验证 |
| 4 | Condition | 字段条件 | 某个字段在某种场景下不可以编辑或者隐藏 | 审核后数量不能修改 |
| 5 | OperationCondition | 操作条件 | 操作按钮在某个场景下不可用或者隐藏 | 审核后不能删除 |
| 6 | Action | (无) | 一个实体的操作分成多个步骤执行，每个步骤叫一个Action。 |  |

这六驾马车都基于插件使用，可以拥有非常强大的功能。

# Netsharp组件清单

# Netsharp路标规划

框架合理稳定

1.平台工具产品化

平台产品化

1.工作流

2.多组织

3.项目管理

集团业务

海量数据

生态环境