Atitit。系统项目整合方案 v1

主要是整理现有项目，从中将其整合起来，提升效率，加快时间

[1.1. 跨平台化 1](#_Toc14320)

[1.2. 逻辑图形化，脚本化 1](#_Toc24737)

[1.3. 跨项目化，产品化 2](#_Toc31826)

[1.4. 配置切换管理 2](#_Toc28707)

[1.5. 平台化 2](#_Toc26487)

[1.6. 跨语言化 2](#_Toc21206)

[1.7. 跨api，统一api化 2](#_Toc32661)

[1.8. 界面统一h5+ajax化 2](#_Toc17396)

[1.9. 界面流程调整与路由 2](#_Toc4467)

[1.10. 界面模板理论与实践 2](#_Toc10706)

[1.11. 功能模板 3](#_Toc11285)

[1.12. 统一验证valide，数据约束 3](#_Toc6492)

[1.13. jsbridge 3](#_Toc12997)

[1.14. 数据查询linq化，stream化 3](#_Toc19071)

[1.15. Orm实践 3](#_Toc96)

[1.16. 跨语言Rmi远程方法调用理论与实践 3](#_Toc11453)

[1.17. 热部署实践 3](#_Toc11795)

[1.18. 模块化 3](#_Toc1407)

[1.19. 插件化，业务触发器 4](#_Toc24804)

[1.20. Page,中阶层事件化 4](#_Toc3557)

[1.21. html组件化 4](#_Toc23751)

[1.22. Js模块化导入 4](#_Toc29727)

[1.23. 国际化 4](#_Toc20748)

## 跨平台化

使用跨平台的技术。。跨越pc android ios等不同的设备与场景。。

## 逻辑图形化，脚本化

工作流，与脚本化

## 跨项目化，产品化

目前根据我们的实际情况，，跨项目化，在小范围内使用效果良好，大范围内对现有模块需要进行较大修改，可能不太适合。。所以先限制在小范围内使用即可。。新的模块可以直接跨项目化设计。。

## 配置切换管理

针对不同的项目配置，动态切换配置状态

## 平台化

平台化带来更好的效率与统一体验

## 跨语言化

尽可能的使用跨语言化的技术，方便切换后端语言

## 跨api，统一api化

在不同的源码api级别统一化，方便源码级别的移植与复用

## 界面统一h5+ajax化

尽可能的使用静态h5+ajax技术，构建界面。可以得到更加统一通用的界面。一套界面通用语各个设备与平台

不要使用服务端界面技术

## 界面流程调整与路由

Mvc服务端调整，urlrewriting

Js端调转 推荐

## 界面模板理论与实践

可以更具excel等结构化文档。。快速生成原型界面

## 功能模板

## 统一验证valide，数据约束

Js层面，中阶层，数据库层

## jsbridge

沟通前后端的桥梁

## 数据查询linq化，stream化

## Orm实践

## 跨语言Rmi远程方法调用理论与实践

## 热部署实践

## 模块化

纵向模块化优先。。然后横向分层化，便于模块复用

## 插件化，业务触发器

## Page,中阶层事件化

## html组件化

## Js模块化导入

## 国际化