Atitit 软件开发体系法规大全v2.docx

Atitit 软件开发体系大全

目录

[1. 基本法（类似宪法） 2](#_Toc27611)

[2. 行政法 框架使用法 工具使用法 4](#_Toc27275)

[2.1. Vue使用用法 4](#_Toc18902)

[2.2. Pdo使用法 4](#_Toc12560)

[2.3. Git使用法 4](#_Toc26240)

[2.3.1. Atitit git删除大文件法 4](#_Toc21898)

[2.4. Jqury使用法 4](#_Toc10316)

[2.5. Layui使用法 4](#_Toc9879)

[2.6. Phop法 4](#_Toc1396)

[3. 流程法民商法 4](#_Toc25427)

[3.1. 数据类型规范法 4](#_Toc837)

[3.2. Ui提升法 4](#_Toc15344)

[3.2.1. Atitit 数据与模板绑定法 4](#_Toc14555)

[3.2.2. 数据表格绑定法 5](#_Toc15367)

[3.3. 功能开发法 5](#_Toc14510)

[3.4. 可读性提升法 5](#_Toc5223)

[3.5. 性能提升法 5](#_Toc8097)

[3.6. 对外接口提供法 5](#_Toc10353)

[3.7. 可用性提升法操作性 5](#_Toc15732)

[3.8. 可扩展性 维护性提升法 5](#_Toc9028)

[3.9. 调试法 5](#_Toc10686)

[3.10. 重构法 5](#_Toc15165)

[4. Sin法（安全法） 5](#_Toc29161)

[4.1. Ui安全法 5](#_Toc16031)

[4.2. Atitit 事务使用法 5](#_Toc12217)

[4.3. Sql安全法 5](#_Toc6756)

[4.4. Atitit 数据安全法 5](#_Toc5482)

[4.5. 后台admin安全法 5](#_Toc21622)

[4.6. 反模式罪名一览表 6](#_Toc9067)

[5. 可读性提升法 6](#_Toc31979)

[6. Ui法 6](#_Toc7487)

[7. 架构法 6](#_Toc30205)

[8. 数据库法 6](#_Toc15878)

[9. Ref 6](#_Toc12493)

# 基本法（类似宪法）

Atitit.软件开发的几大规则，法则，与原则。。。

1. 设计模式六大原则 2

1.1. 设计模式六大原则（1）：单一职责原则 2

1.2. 设计模式六大原则（2）：里氏替换原则 2

1.3. 设计模式六大原则（3）：依赖倒置原则 2

1.4. 设计模式六大原则（4）：接口隔离原则 2

1.5. 设计模式六大原则（5）：迪米特法则 2

1.6. 设计模式六大原则（6）：开闭原则 2

2. 面向对象的S.O.L.I.D原则 2

2.1. Single Responsibility Principle (SRP)–职责单一原则 3

2.2. Open/Closed Principle (OCP)–开闭原则 3

2.3. Liskov substitution principle (LSP)–里氏代换原则 3

2.4. Interface Segregation Principle (ISP)–接口隔离原则 3

2.5. Dependency Inversion Principle(DIP)–依赖倒置原则 3

3. Other原则 3

3.1. Don’t Repeat Yourself(DRY) 4

3.2. Keep It Simple, Stupid (KISS) 4

3.3. Program to an interface, not an implementation 4

3.4. Command-Query Separation (CQS)–命令-查询分离原则 4

3.5. You Ain’t Gonna Need It(YAGNI) 4

3.6. Common Closure Principle（CCP）–共同封闭原则 4

3.7. Common Reuse Principle (CRP)–共同重用原则 4

3.8. Hollywood Principle–好莱坞原则 4

3.9. High Cohesion & Low/Loose coupling & –高内聚，低耦合 5

3.10. Convention over Configuration（CoC）–惯例优于配置原则 5

3.11. Separation of Concerns (SoC)–关注点分离 5

3.12. Design by Contract (DbC)–契约式设计 5

3.13. Acyclic Dependencies Principle (ADP)–无环依赖原则 5

# 行政法 框架使用法 工具使用法

## Vue使用用法

## Pdo使用法

## Git使用法

### Atitit git删除大文件法

## Jqury使用法

## Layui使用法

## Phop法

# 流程法民商法

## 数据类型规范法

## Ui提升法

### Atitit 数据与模板绑定法

### 数据表格绑定法

## 功能开发法

## 可读性提升法

## 性能提升法

## 对外接口提供法

## 可用性提升法操作性

## 可扩展性 维护性提升法

## 调试法

## 重构法

# Sin法（安全法）

## Ui安全法

## Atitit 事务使用法

## Sql安全法

## Atitit 数据安全法

## 后台admin安全法

## 反模式罪名一览表

**[原创 Atitit 反模式 黑名单 异常处理 反模式（antipatterns） 目录 1.1. 记录并抛出（log and throw） 1 1.2. 抛出异常基类（Throwing Excepti](https://blog.csdn.net/attilax/article/details/103550541" \t "https://blog.csdn.net/_blank)**

# 可读性提升法

# **U**i法

# 架构法

# 数据库**法**

# Ref

Atitit 软件开发基本法

非功能性需求\_百度百科.html