



- 敏捷过程（2001/2—敏捷软件开发宣言
The Manifesto of the Agile Alliance）
- 敏捷过程的价值观
 - 个体和交互胜过过程和工具
 - 可以工作的软件胜过面面俱到的文档
 - 客户合作胜过合同谈判
 - 响应变化胜过遵循计划

敏捷过程

■ 敏捷过程的原则

- 我们最优先要做的是通过尽早的,持续的交付有价值的软件来使客户满意
- 即使到了开发的后期,也欢迎改变需求.敏捷过程利用变化来为客户创造竞争优势
- 经常性地交付可以工作的软件,交付的间隔可以从几周到几个月,交付的时间间隔越短越好
- 在整个项目开发期间,业务人员和开发人员必须天天都在一起工作
- 围绕被激励起来的个人来构建项目.给他们提供所需要的环境和支持,并且信任他们能够完成工作

敏捷过程

■ 敏捷过程的原则 (续)

- 在团队内部,最具有效果并且富有效率的传递信息的方法,就是面对面的交谈
- 工作的软件是首要的进度度量标准
- 敏捷过程提倡可持续的开发速度.责任人、开发者和用户应该能够保持一个长期的、恒定的开发速度
- 不断地关注优秀的技能和好的设计会增强敏捷能力
- 简单是根本的
- 最好的架构、需求和设计出自于自组织的团队
- 每隔一段时间,团队就会在如何才能更有效地工作方面进行反省,然后相应地对自己的行为进行调整

敏捷过程

- **SCRUM** : Schwaber, K., & Beddle, M. (2002). *Agile Software Development with Scrum*. NJ: Prentice Hall.
- **Crystal** : Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- **Feature Driven Development (FDD)** : Peter Coad, Eric Lefebvre, and Jeff De Luca (1999). *Java Modeling In Color with UML: Enterprise Components and Process*. Prentice Hall.
- **Adaptive Software Development (ADP)** : James A. Highsmith III (2000). *Adaptive Software Development*, Dorset House Publishing.
- **eXtreme Programming (XP)**

极限编程 (eXtreme Programming, XP)

- 极限编程是敏捷过程中最富盛名的一个，其中“极限”的含义是指把最好的开发实践运用到极致。目前极限编程已经成为一个典型的开发方法，广泛应用于需求模糊且经常改变的场合。
- 特点：
 - 对变化和不确定性反应更快速，更敏捷
 - 快速的同时保持可持续的开发速度

极限编程的有效实践

- 客户作为开发团队的成员
- 使用用户素材
- 短交付周期（每两周完成一次迭代）
- 验收测试
- 结对编程
- 测试驱动的开发
- 集体所有（程序代码属于整个开发小组，每个成员都有修改代码的权利，都对全部代码负责）

极限编程 (eXtreme Programming, XP)

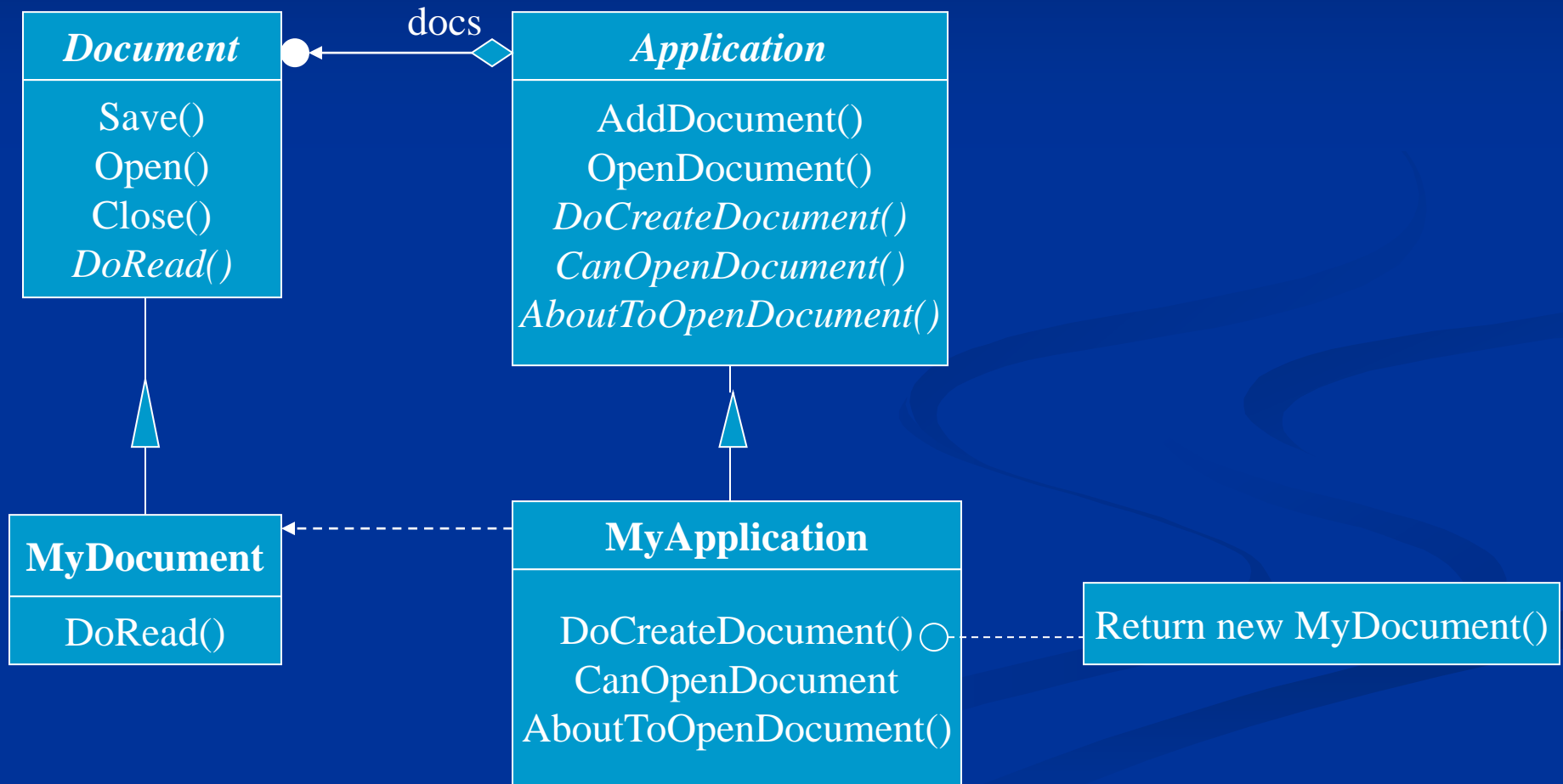
- 持续集成（一日内多次集成，不断回归测试）
- 可持续的开发速度（周工作时间不超过40小时，连续加班不超过两周）
- 开放的工作空间
- 及时调整计划
- 简单的设计
- 重构
- 使用隐喻（隐喻是把整个系统联系在一起的全局视图，描述系统如何运做，如何把新功能加入到系统中）

使用TEMPLATE METHOD重构的一个例子

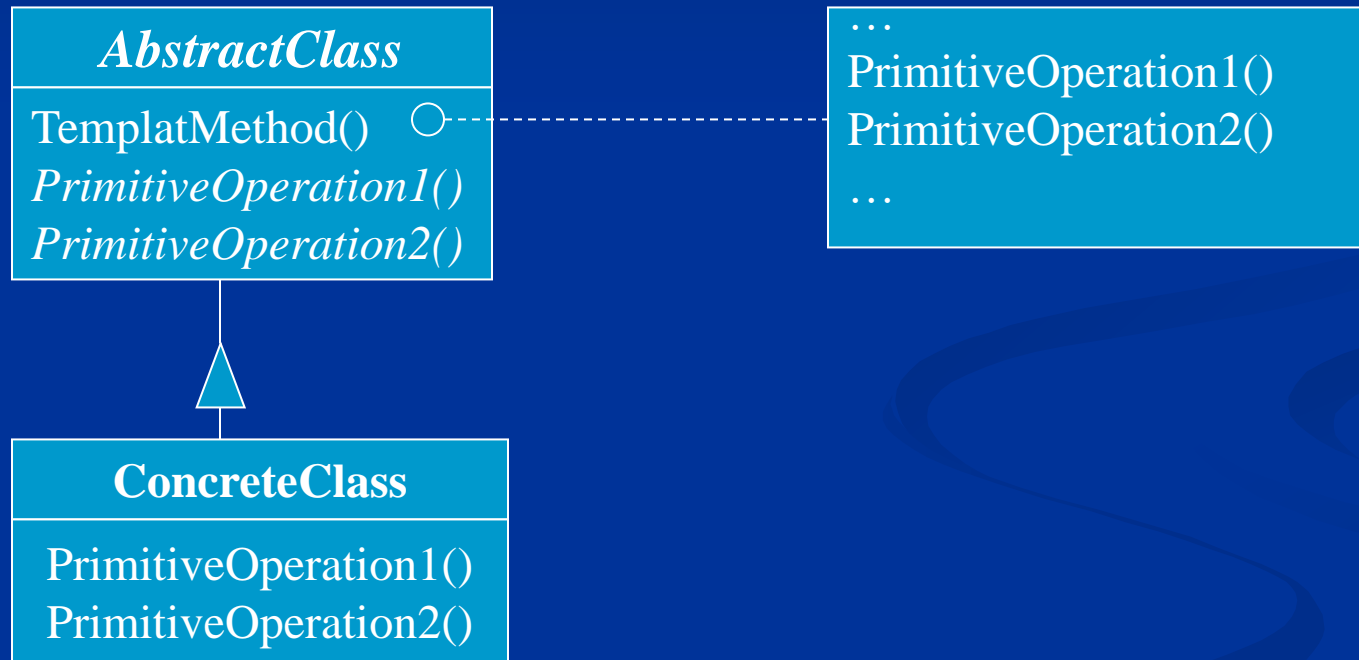
定义一个操作中的算法的骨架，而将一些步骤延迟到子类中。Template Method使得子类可以不改变一个算法的结构即可以重新定义该算法的某些特定步骤。

```
void Application::OpenDocument(const char*name) {  
    if (!CanOpenDocument(name)) {  
        return;  
    }  
    Document* doc=DoCreateDocument();  
    if (doc) {  
        _docs->AddDocument(doc);  
        AboutToOpenDocument(doc);  
        doc->Open();  
        doc->DoRead();  
    }  
}
```

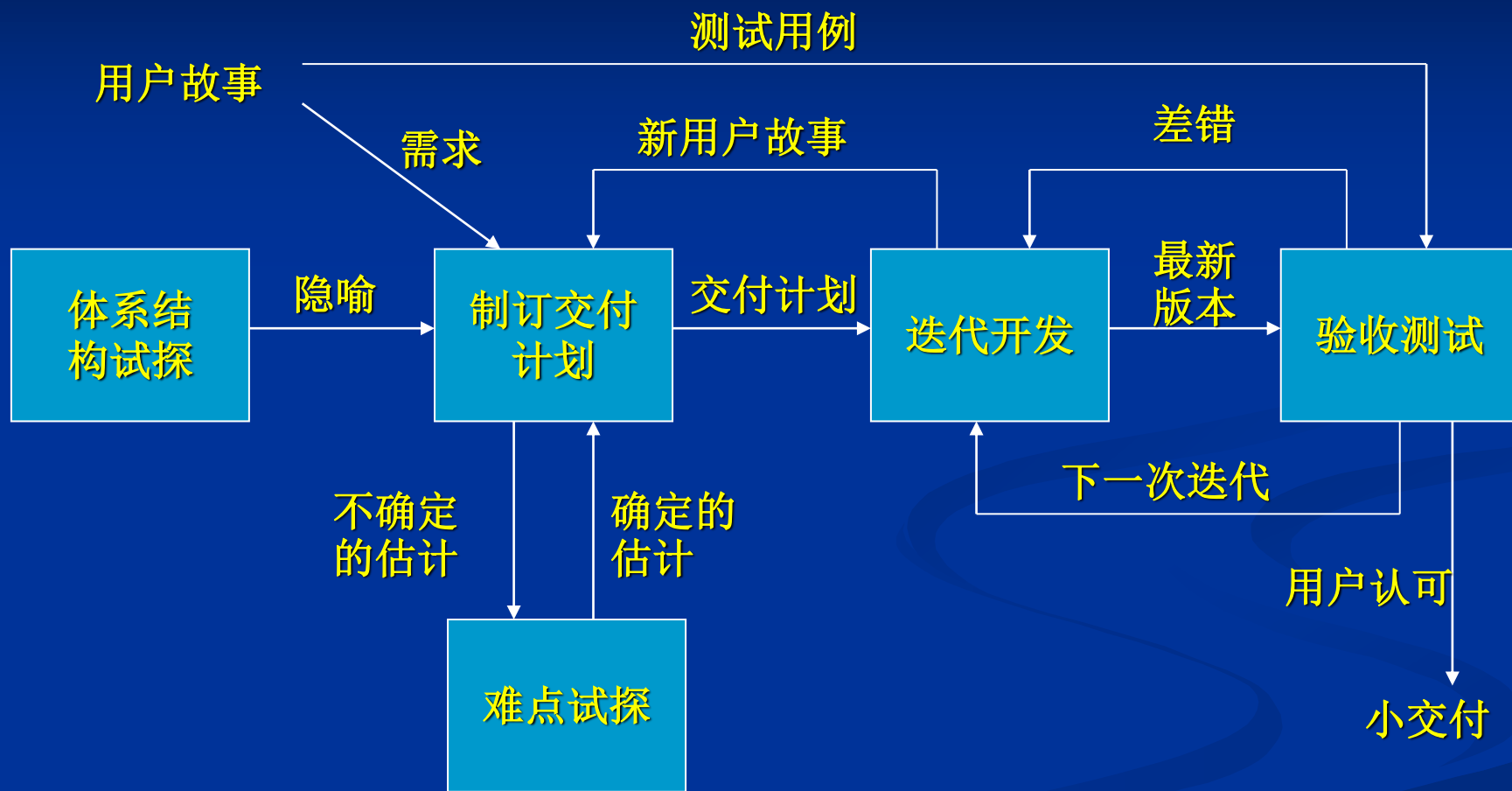

使用TEMPLATE METHOD重构的一个例子



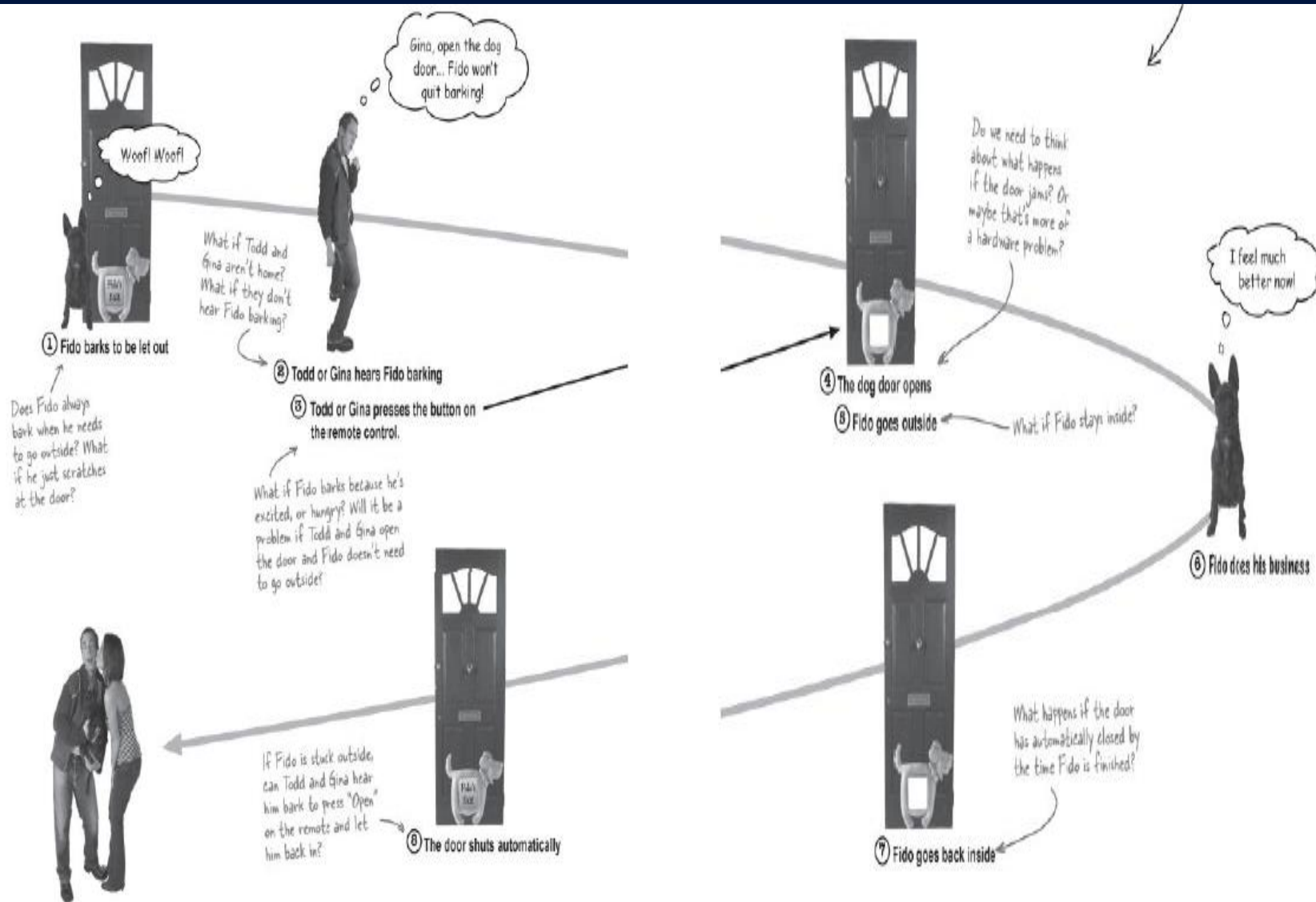
使用TEMPLATE METHOD重构的一个例子



极限编程的整体开发过程



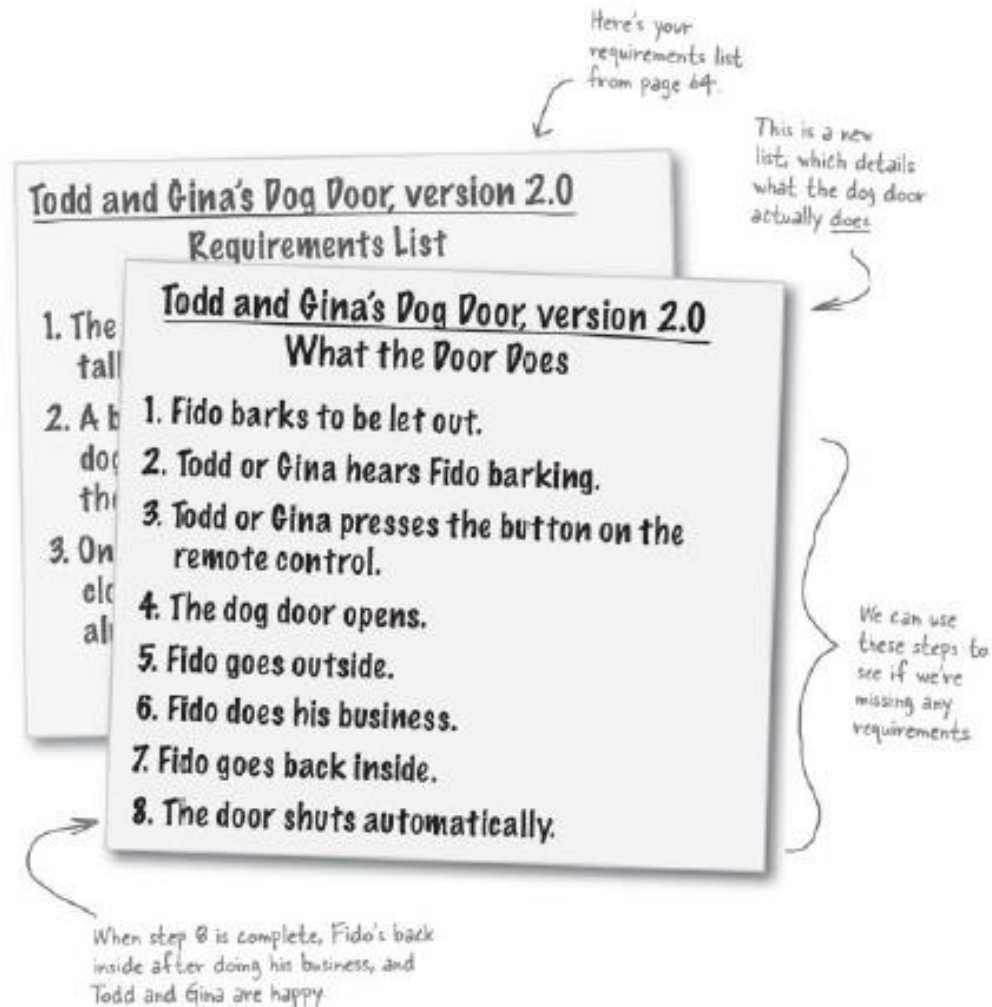
Tips: Use Case & Scenario



Todd and Gina's Dog Door, version 2.0

What the Door Does

1. Fido barks to be let out.
2. Todd or Gina hears Fido barking.
3. Todd or Gina presses the button on the remote control.
4. The dog door opens.
5. Fido goes outside.
6. Fido does his business.
 - 6.1 The door shuts automatically.
 - 6.2 Fido barks to be let back inside.
 - 6.3 Todd or Gina hears Fido barking (again).
 - 6.4 Todd or Gina presses the button on the remote control.
 - 6.5 The dog door opens (again).
7. Fido goes back inside.
8. The door shuts automatically.



极限编程的迭代过程

