CASE软件编写工具介绍

XUnit

XUnit是各种代码驱动测试框架的统称，这些框架可以测试软件的不同内容(单元)，比如函数和类。xUnit框架的主要优点是，它提供了-一个自动化测试的解决方案。没有必要多次编写重复的测试代码，也不必记住这个测试的结果应该是怎样的。

单元测试的原则:

1.先编写测试代码，然后编写符合测试的代码。至少做到完成部分代码后，再完成对应的测试代码。

2.测试代码不需要覆盖所有的细节，但应该对所有主要的功能和可能出错的地方有相应的测试用例。

3.发现Bug后，首先应编写对应的测试用例，然后再进行调试。

4.不断总结出现Bug的原因，对其他代码也编写相应的测试用例。

5.每次编写完代码后，运行所有以前的测试用例，验证对以前代码的影响，并把这种影响尽早消除。

6.不断维护测试代码，保证代码变动后能通过所有测试。

XUnit测试框架的四要素:

(1)测试Fixtures

是一组认定被测对象或被测程序单元测试成功的预定条件或预期结果的设定。Fixture就是被测试的目标，可能是一个对象或一组相关的对象，甚至是一个函数。测试人员在测试前就应该清楚对被测对象进行测试的正确结果是什么，这样就可以对测试结果有一个明确的判断。

(2)测试集.

测试集就是一-组测试用例，这些测试用例要求有相同的测试Fixture,以保证这些测试不会出现管理上的混乱。

(3)测试执行

单个单元测试的执行可以按下面的方式进行:

setUp();/\*首先，我们要建立针对被测程序单元的独立测试环境\*//\*然后，编写所有测试用例的测试体或测试程序\*/

tearDown();/\*最后，无论测试成功还是失败，都将环境进行清理，以免影响后继的测试\*/

(4)断言

断言实际上就是验证被测程序在测试中的行为或状态的-一个宏或函数。断言失败实际上就是引发异常，终止测试的执行。

XUnit测试流程

①对Fixture进行初始化，以及其他初始化操作。

②按照要测试的某个功能或流程对Fixture进行操作。

③验证结果是否正确。

④对Fixture的以及其他的资源释放等做清理工作。

XPlanner

XPlanner是为极限编程团队（XP）准备的基于网络的项目规划跟踪工具，她支持XP开发流程，并解决利用XP思想来开发项目所碰到的问题。简单的模型规划，虚拟笔记卡(Virtual note cards),iterations、user stories与工作记录的追踪，未完成stories将自动迭代，工作时间追踪，生成团队效率，个人工时报表，SOAP界面支持。

技术特点：

\*简单模型规划

\*虚拟便笺

\*支持记录和跟踪项目：[迭代](https://baike.baidu.com/item/%E8%BF%AD%E4%BB%A3" \t "https://baike.baidu.com/item/xplanner/_blank)、用户流程和[任务管理](https://baike.baidu.com/item/%E4%BB%BB%E5%8A%A1%E7%AE%A1%E7%90%86" \t "https://baike.baidu.com/item/xplanner/_blank)

\*未完成流程的智能化继续(拷贝未完成任务,拷贝的流程是交联的)

\*分布式整合指令 (以email通知)

\*在个人/组级别，实现在线时间跟踪和时间表单生成

\*度量生成 (组速度, 个人时间, ...)

\*迭代速度的图表, 用Scrum控制管理方法消减任务

\*任务类型的分布, 处理, 及其他

\*迭代估计准确度的[视图](https://baike.baidu.com/item/%E8%A7%86%E5%9B%BE" \t "https://baike.baidu.com/item/xplanner/_blank)

\*项目与[迭代](https://baike.baidu.com/item/%E8%BF%AD%E4%BB%A3" \t "https://baike.baidu.com/item/xplanner/_blank)信息可以导出为XML和MPX格式

\*TWiki-style 文本格式化支持，可以支持外部工具集成和动态wiki word链接

\*综合的、可扩展的验证

\*对高级的XPlanner集成和扩展提供SOAP接口

\*支持英语、西班牙语、法语、德语、意大利语、巴西葡萄牙语和丹麦语