

手机软件测试

目录

1 手机知识	3
1.1 手机的主要功能	3
1.1.1 通话功能	3
1.1.2 消息功能	3
1.1.3 电话本	3
1.1.4 增值服务	3
1.1.5 其他功能	3
1.1.6 为特定语言定做的功能	4
1.1.7 附件	4
1.2 手机的软件结构	4
1.3 手机的硬件结构	5
1.4 Nokia手机相关知识	6
1.4.1 Project Line of Nokia	6
2 测试基础	7
2.1 测试与开发	7
2.1.1 软件开发的一般流程	7
2.1.2 测试在软件开发中的作用	7
2.1.3 测试与开发对应图	8
2.1.4 Nokia手机软件测试介入开发的时间	9
2.1.5 Nokia手机的开发流程	9
2.2 测试的流程	10
2.2.1 制定测试计划	10
2.2.2 测试准备	10
2.2.3 测试执行	10
2.2.4 测试评估	10
2.2.5 文档收集	11
2.2.6 测试总结报告	11
2.3 测试的分类	11
2.3.1 按测试的手段分	11
2.3.2 按测试发生的时间和目标分	12
2.3.3 按测试的任务分	12
2.3.4 其他测试	12
2.4 黑盒测试详细介绍	12
2.4.1 Release Test	12
2.4.2 System Test	12
2.4.3 Focus Test	13
2.4.4 Stress Test	13
2.4.5 Free Test	14
2.5 测试相关文档说明	14

2.5.1 测试计划.....	14
2.5.2 测试用例.....	14
2.5.3 错误报告.....	15
2.5.4 进度报告.....	16
2.5.5 总结报告.....	17
3 手机相关.....	17
3.1 GSM.....	17
3.2 GPRS.....	17
3.3 CDMA.....	17
3.4 3G.....	17
4 手机软件测试工程师必备素质.....	17
4.1 Team Leader的任务和必备能力.....	17
4.2 QA的工作内容.....	17
4.3 测试工程师应有素质.....	18

1 手机知识

1.1 手机的主要功能

1.1.1 通话功能

- 对拨入拨出电话的管理
- 对通话记录的管理
- 呼叫转接、呼叫等待、通话计时计费等方便用户使用的功能

1.1.2 消息功能

- 文字短消息（SMS）的编辑、发送、接收、转发和存储等；
- 多媒体短消息（MMS）的编辑、发送、接收、转发、存储和配置；

1.1.3 电话本

- 名片的管理
- 存在 SIM 卡上的名片
- 存在手机内存中的名片
- 一个名字多项内容（如传真、固话、手机、Email 等）
- 名片的新建、修改、拷贝、转存、删除
- 名片以红外或短消息形式发送给其它手机
- 单键拨号（Speed Dialing）
- 号码分组（Caller Groups）

1.1.4 增值服务

- Business cards 的管理（如发送和接收：通过红外线或 SMS。）
- 书签的管理（如发送和接收：通过红外线或 SMS 或书签形式，以及编辑、存储、新增和进入。）
- 服务信箱（自动存储服务信息，服务信息有点播铃声，下载彩色图片和 COD 文件等。）
- 服务设置（GPRS 上网设置，WAP 服务设置。）
- 多模式浏览器（GPRS 上网，WAP 服务。）
- OTA 待机图片（通过无线下载待机图片）
- OTA 铃声
- V-calendar

- XHML
- 移动梦网
- 动感地带

1.1.5 其他功能

- 闹钟 (Alarm)
- 日历 (Calendar)
- 计算器 (Calculator)
- 定时器 (Count Down Timer)
- 屏保 (Screen Saver)
- 待办事项 (To-Do List)
- 游戏 (Games)

1.1.6 为特定语言定做的功能

- 中文输入 (拼音/笔划)
- 中文菜单
- 农历 (Lunar Calendar)

1.1.7 附件

- 充电器 (Charger)
- 耳机 (Headset)
- 车载免提 (Car Kit)
- 照相头 (Camera)

1.1.8 数据连通

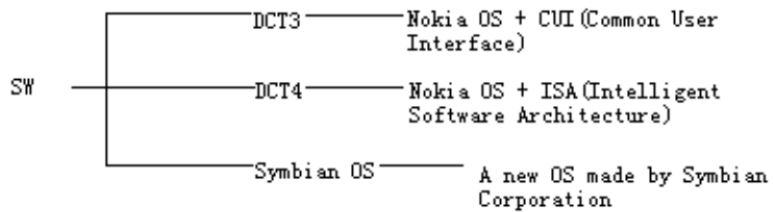
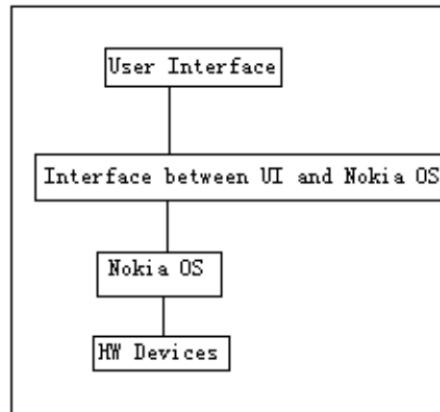
- GPRS 应用程序
- 同步应用程序 (同步应用, 同步设置)
- 红外线应用程序
- 数据线
- 手机 Flash 程序
- Trace Log

1.1.9 JAVA 程序

- 电子邮件

- QQ 程序
- 应用程序
- 游戏程序

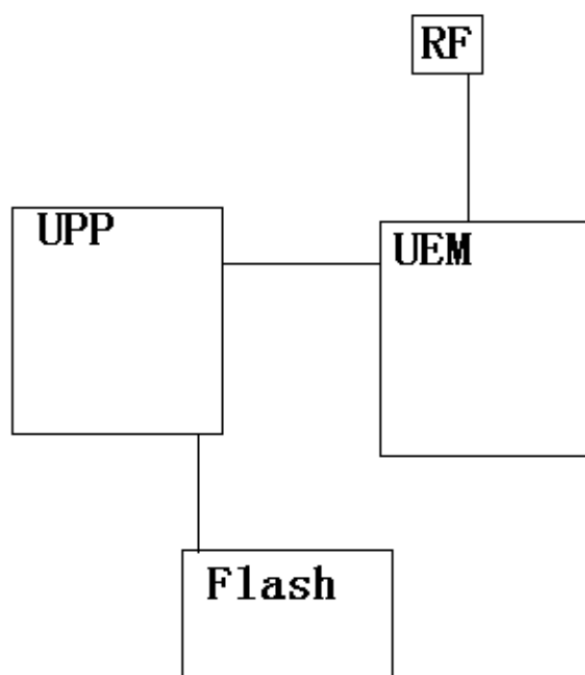
1.2 手机的软件结构



DCT = Digital Core Technology

DCT3 和 DCT4 的区别主要在硬件结构和 UI 上

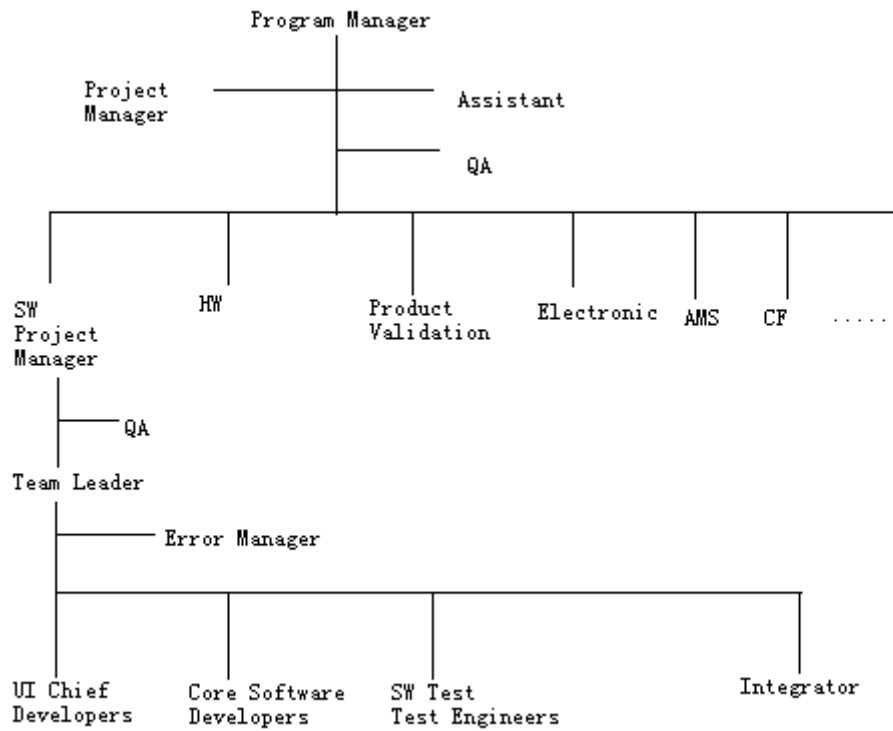
1.3 手机的硬件结构



- **RF** 是手机的射频接受和发送设备，是手机本机与无线网络的接口
- **UEM** 是手机对外设备联接的接口，包括 SIM 卡，充电器，电池，RF，数据线借口，红外接口等，并提供对上述数据和信号的处理。无线信号在这里进行调制和解调。
- **UPP** 是手机的核心处理模块。其中，DSP 负责数字信号和模拟信号的转换。UPP 和 UEM 之间存在着接口，两者之间的通信是通过接口进行的。UPP 除了负责整个手机的操作系统外，还负责手机本身设备的运作，如耳机，麦克风，显示屏等。除此之外，还负责缓存和 Flash(相当于硬盘)的运作。
- **Flash** 是手机的数据存储区，非临时数据都存在这里。一般包括以下几个方面：Core Code, DSP Code, HW data, PPM, PMM。

1.4 Nokia 手机相关知识

1.4.1 Project Line of Nokia



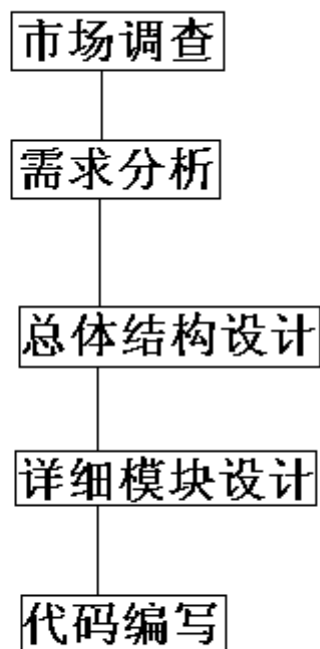
1.4.2 手机类型

- MEP = Mobile Entry Phone
- MP = Mobile Phone
- MSW = Mobile Software
- SWEP = software engineer product

2 测试基础

2.1 测试与开发

2.1.1 软件开发的一般流程



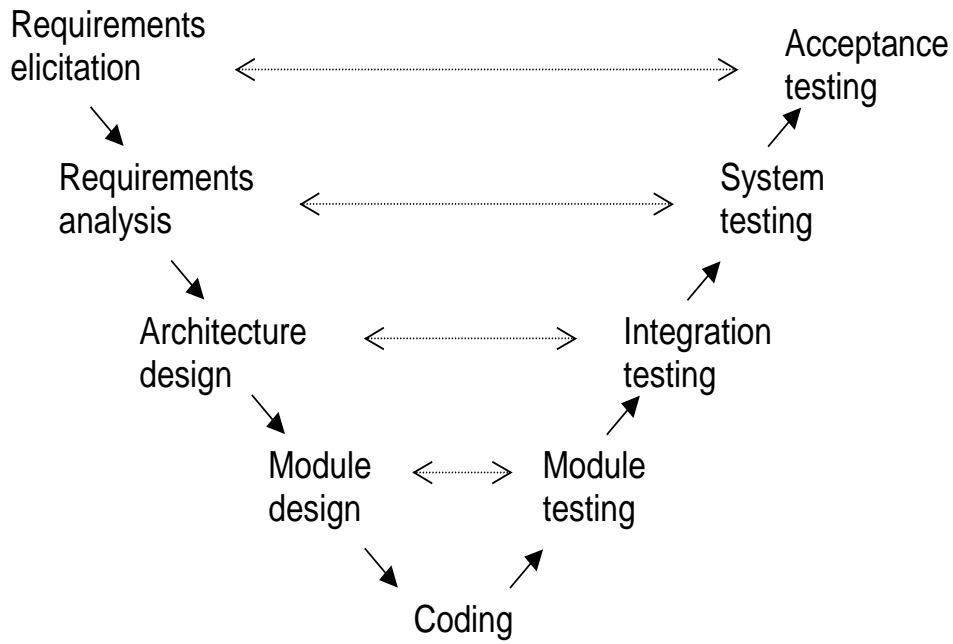
- Marketing
- Requirement Analysis
- High Level Design
- Low Level Design
- Coding

2.1.2 测试在软件开发中的作用

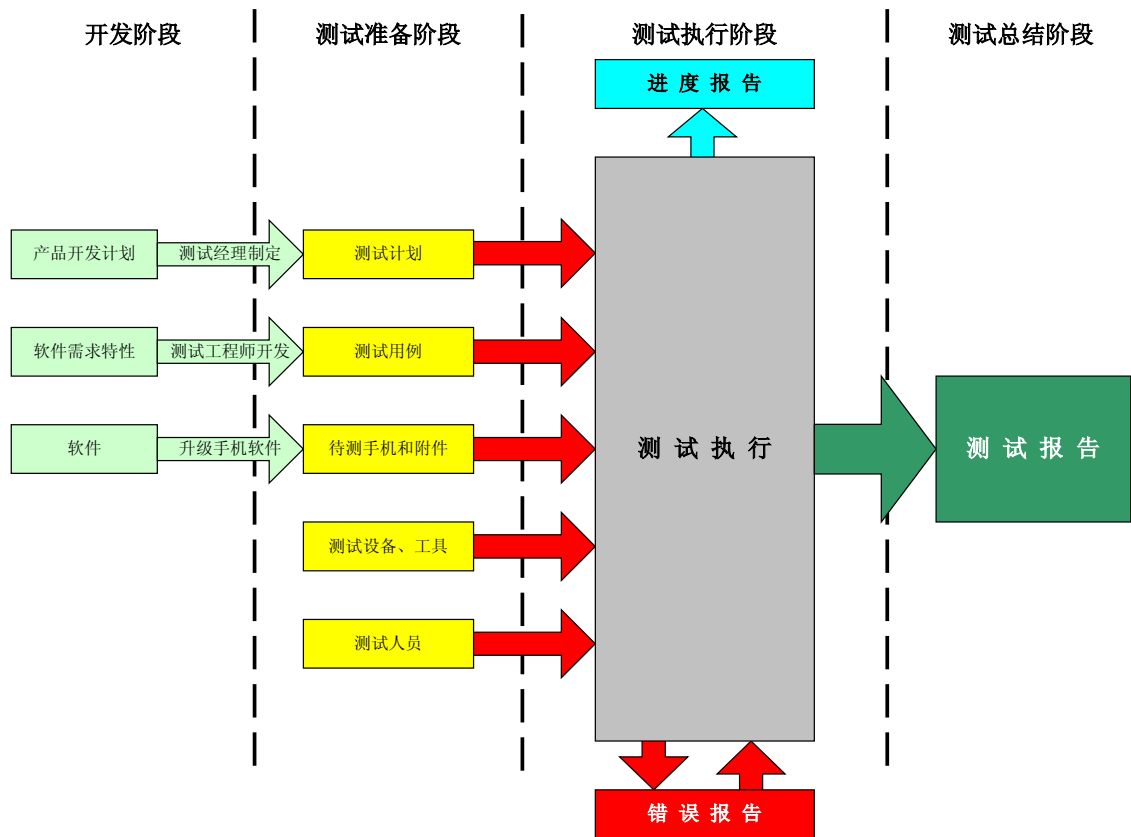
- 由于现在软件的规模越来越大，一个人或者少数几个人已经不可能在一定的时间内完成一个软件，所以软件开发的过程越来越复杂，层次越来越深。这就导致开发人员之间的沟通有了一定的隔阂。所以，软件测试越来越有单立出来的必要和重要性。
- 由于软件开发的过程的复杂性，软件必然存在着无数的 **Bug**。而且大多数是在软件上市前必须解决的，而开发者有不定能发现这些问题，故而测试就显得非常必要。测试是开发成功的必要保障。

- 由于软件开发的层次性，所以开发的结果很可能与初衷不一样，这就需要测试者去发现这些差异。因此，测试是软件成功的重要保证。
- 软件不仅要实现一些功能，更要完善它的性能。这就需要测试人员对软件进行评测，从而不断地完善软件的性能。

2.1.3 测试与开发对应图

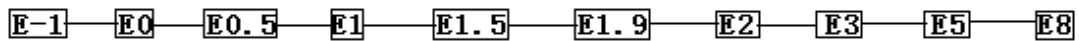


2.1.4 Nokia 手机软件测试介入开发的时间



- 在制定开发计划的同时就要制定测试计划
- 测试在进行结构设计时就已经进行了

2.1.5 Nokia 手机的开发流程



- E-1
During this period, an idea box will appear. The ideas in the idea box are collected from Region Marketing and have a certain priority (The lower the priority number is, the higher the priority is). For example:0, 1, 2.
- E0
During this period, the HW designer must make out the B0-HW version.
That is to say, B0 must be put out after E0 period.
- E0.5
综合考虑 HW, SW and Cost
- E1
From E0 to E1, Design and Test Plan, Risk, Organization, Schedule must be discussed and

made out.

- E1.5
全体讨论 Design and Test Specification
- E1.9
From E1 to E2, Design and Test Specification must be made out.
During E1.9, Last version of Specification should be made out and be approved.
- E2
During E2, The formal draft SW should be made out.
- E3
From E2 to E3, 对 SW 进行精美化、完美化测试和改良
Purpose: No fatal error (市场部可以接受的 Fatal Error 不算)
- E5
From E3 to E5, 进行 LCD 以及其他 HW 的改动

During E5, 可以让生产工厂进行大批量生产

- Before E5, the test stays in the CE (concurrent engineer)

After E5, the test goes into PE (production engineer)

2.2 测试的流程

2.2.1 制定测试计划

- 开启测试项目
- 在接了一个测试项目后，要在一定的期限内制定好测试的详细计划以及日程安排表

2.2.2 测试准备

- 在计划制定好之后，在执行之前，必须将测试所需的人力资源，硬件资源，软件资源，文档资源以及环境和人文资源准备充分

2.2.3 测试执行

- 测试组根据测试计划和测试日程安排进行测试，并输出测试结果

2.2.4 测试评估

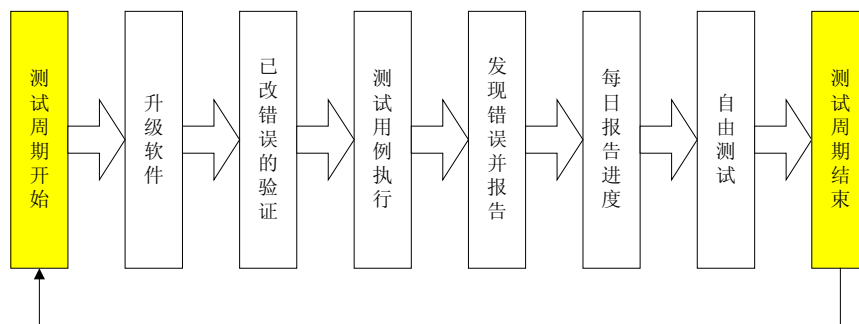
- 有测试结果评估小组或评估人员对测试结果进行评测，分析，并输出分析结果

2.2.5 文档收集

- 将从测试计划开始到评估结束的所有文档进行整理收集。
- 对整个测试过程进行总结，并对测试结果进行总结

2.2.6 测试总结报告

- 提交测试结果
- 归还所借相关资源
- 文档入库
- 关闭测试项目



2.3 测试的方法

2.3.1 正确性测试

- 正确性测试又称功能测试，它检查软件的功能是否符合规格说明。
- 测试基本的方法是构造一些合理输入（在定义域内），检查是否得到期望的输出。
- 由于定义域是一个连续区间，所以不可能枚举所有可能的值，那么等价测试就很必要了（将定义域分成若干个等价区间）。
- 等价区间的概念可表述如下：
记 (A, B) 是命题 $f(x)$ 的一个等价区间，在 (A, B) 中任意取 x_1 进行测试
如果 $f(x_1)$ 错误，那么 $f(x)$ 在整个 (A, B) 区间都将出错。
如果 $f(x_1)$ 正确，那么 $f(x)$ 在整个 (A, B) 区间都将正确。

2.3.2 容错性测试

- 容错性测试是检查软件在异常条件下的行为（输入不同的数据类型或者定义域之外的值进行测试）。

2.3.3 边界性测试

- 因为边界一直是比较敏感的地方，而且是程序员最容易忽略的地方，所以，这种测试也往往最容易奏效。

2.3.4 性能与效率测试

- 性能与效率测试主要是测试软件的运行速度和对资源的利用率。
- 性能与效率测试中很重要的一项是极限测试，因为很多软件系统会在极限测试中崩溃。

2.3.5 易用性测试

- 易用性测试没有一个量化的指标，主观性较强。这主要是从 End User 的角度去考虑软件是否会有一定的使用缺陷。如果对此有任何看法，可以向 Team Leader 反应或者与客户负责人直接交流。

2.3.6 文档测试

- 文档测试主要检查文档的正确性、完备性和可理解性。好多人甚至不知道文档是软件的一个组成部分。
- 我们工作中的文档主要是 UI Spec.和 Test Case。UI Spec 使我们无法改变的，但是 Test Case 是我们测试的对象。Test Case 是我们用来测试手机软件的参考文档，但是它本身也有一定的局限性。所以，在测试的过程中，如果发现 Test Case 不正确或者不充分，可以直接补充，或者和 Team Leader 商议后把不足的地方补充起来。

2.4 测试的分类

2.4.1 按测试的手段分

- 黑盒测试(White-box Test)
 - Release Test
 - (Full Round)SystemTest
 - Focus Test
 - Stress Test---No Test Case
 - Free Test----No Test Case
- 白盒测试(Black-box Test)
 - Module Test
 - Sub-System Test
 - Sub-System Integration Test
 - System Integration Test

- Integration Test

The feature groups for Integration Test are decided by Integrator and provided by SW Component Factory.

2.4.2 按测试发生的时间和目标分

- 单元测试(Module Test/Unit Test)
- 集成测试(Integration Test)
- 系统测试(System Test)

2.4.3 按测试的任务分

- 现场测试(Field Test)
- 互操作测试(Inter-Operatability Test)

2.4.4 其他测试

- 可接受性测试(Acceptance Test)
- α 测试 -----手机研发公司自己做的测试
- β 测试 -----非手机研发公司做的测试

2.5 黑盒测试详细介绍

2.5.1 Release Test

- **Purpose:**
 - 测试手机的基本功能是否实现，是否有进一步测试的必要性
- **Input:**
 - 测试工程师
 - Release Test Cases (较少，一般为 200 左右)
 - 手机以及相关附件
 - 测试环境
- **Output:**
 - Test result of Release Test
 - No Error reports (Optional)
- **Attention:**
 - Release Test 的 Test Case 具有一定的典型性，主要是反映手机最基本功能的 Test Case
 - 本类测试只需要依据 Test Case 进行测试，不需要进一步发挥
 - 如果有发现与 Case 无关的 Error，在测试通过后才可填报 Error Report
 - 此类测试有一门槛值，即 Test Case 的 Pass 率达到一定值（如 95%）才能宣布版本

发布成功，进入进一步的测试，否则此版本无效。

- 除了阈值外，如果重要功能模块的 Test Case 没通过，也会终止这个版本。

2.5.2 System Test

- **Full Round System Test**

- **Purpose**

- ◆ 对手机的所有功能进行全面的测试(所有语言包)
- ◆ 由于 Case 不可能包含所有方面，所以测试时应适度发挥，尽力完成全面测试

- **Input:**

- ◆ 测试工程师
- ◆ Test Cases(较多，一般为 25000 左右)
- ◆ 手机以及相关附件
- ◆ 测试环境
- ◆ Schedule

- **Output:**

- ◆ Daily Report of test cases (number & percent of Pass, Error, NA, NT)
- ◆ Summary Report
- ◆ Error list and Error reports

- **Common System Test (Medium or Minor)**

- **Purpose:**

- ◆ 对手机的一部分的功能进行全面的测试
- ◆ 由于 Case 不可能包含所有方面，所以测试时应适度发挥，尽力完成全面测试

- **Input:**

- ◆ 测试工程师
- ◆ Test Cases(较多，取决于测试的目的和范围)
- ◆ 手机以及相关附件
- ◆ 测试环境
- ◆ Schedule

- **Output:**

- ◆ Daily Report of test cases (number & percent of Pass, Error, NA, NT)
- ◆ Summary Report
- ◆ Error list and Error reports

- **Attention:**

- System Test 一般分为两个部分，“跑 Case”和 Free Test。
- 在测试初期，一般只需要按照 Test Case 测，把一些不可重现的 Error 都记录下来。同时遇到 Test Case 的问题或者不充分，应该立即解决(和 Team Leader 或者 Special List 讨论，补写 Test Case)。在这一阶段结束后，一般要写一个 Summary Report。把这一阶段的测试结果和遇到的问题、自己的见解都写在里面(当然是用 English)。
- 当所有 Test Case 都测完后，就进入 Free Test 期间。这里的 Free Test 具有明确的的目的性和范围。一般来说，这段时间的 Free Test 只需要测自己负责的模块。而且 Free Test 还负责重现前期“跑 Case”是遗留的不可重现的 Error。

2.5.3 Focus Test

- **Purpose:**
 - 集中于一个或几个点进行测试(同 System Test)
- **Input:**
 - 测试工程师
 - Test Cases
 - 手机以及相关附件
 - 测试环境
- **Output:**
 - Test Result
 - Error Reports

2.5.4 Stress Test

- **Purpose:**
 - 为了解决市场上发现的重大 Error，而进行的有针对性的强度测试
 - 主要是利用边缘测试（临界测试）手段
- **Input:**
 - 测试工程师
 - 手机以及相关附件
 - 测试环境
 - Focus List of Phone Features
- **Output:**
 - Expected result: find out the reproducible steps of these errors
 - Decrease the steps as possible as we can
- **Attention:**
 - 压力测试，顾名思义，是给手机施加一定压力，从而找出手机软件上的 Error。一般来说，对手机施加的压力主要有：
 - ◆ 存储压力：由于手机采用的是栈式存储，所以当有一个存储块满了之后，如果程序员不做相应处理或者处理不好的话，很容易造成其他存储区被擦除，从而在 UI 上出现问题（其他功能无法正常使用）。
 - ◆ 边界压力：边界一直是程序员最容易忽略的地方。
 - ◆ 响应能力压力：有时候某个操作可能处理的时间很长，在处理期间如果测试者再不断地进行其他操作的话，很容易出现问题。
 - ◆ 网络流量压力（如在接电话时进行短信服务）等等。
 - 在项目中，Stress Test 有时也会用来重现不可重现的 Error。
 - 由于有不少不可重现的 Error 是由于 Memory Leak(内存泄漏)引起的，所以不停的重复同一个操作是重现一个不可重现的 Error 的一个好方法。

2.5.5 Free Test

- **Purpose:**
 - 测试 System Test 中没有做完的不可重现 Error
 - 寻找平时没有找到的忽略的 Error
- **Input:**
 - 测试工程师
 - 手机以及相关附件
 - 测试环境
 - Error List of System Test (especially the irreproducible errors)
- **Output:**
 - Error List and Error Reports
- **Attention:**
 - 在 System Test 阶段所用的 Free Test 具有明显的目的性和范围
 - 平时的 Free Test 从理论上应该对所测试的范围穷尽所有的测试方法。但是，这是不现实的。在实际项目中，主要有两个方面是 Free Test 所需要重视的。
 - ◆ 一是从 UI Spec 上找灵感。应为 Test Case 是依据 UI Spec 写的，所以从 UI Spec 上突破是一个行之有效的方法。UI Spec 有一定的探索深度，加大探索深度，是一种突破的途径；另外同一个功能用其他不同的方法去实现，也是一种突破途径。
 - ◆ 二是多关注不同 Feature 之间的 Interaction。这是手机软件相对比较容易出问题，而 Test Case 又很少能反映的地方。这是一个很大的 Free Test 空间。

2.6 测试相关文档说明

2.6.1 测试计划

- **测试的任务**

即需要测试什么和不需要测试什么；
- **工作量估算**

需要多少人，测试多少天，测试几个周期；
- **日程表**

每人每天需要做什么；
- **测试方法和流程**

采用什么方法，遵循哪些流程；
- **测试资源**

需要多少人、设备、工具、文档等资源，以及对上述资源都有哪些要求。
- **测试输出**

测试中需完成的错误报告（Error Report）和进度报告（Progress Report），测试完成后需完成的总结报告（Summary Report）。

2.6.2 测试用例

- **Title**

标题一般会描述出当前要执行的 case 是哪个功能模块的，能实现怎样的一个操作。标题下面有当前 case 的 ID 号和软件的版本号，如

Phonebook-Memory Save-Selected memory is Phone and SIM

ID:	EK20010829094907
Version:	1.1.0

- **Description**

整体地描述这个 case 的测试目的，能实现什么功能。例如：

The purpose of this test case is check out that the phone number can be saved to phonebook when selected memory is Phone and SIM.

- **Required test environment and accessories**

必需的测试环境和附件。测试环境包括硬件环境和软件环境。例如：HW, ESIM, Headset.

- **Precondition**

描述执行 case 的前提条件。例如：

Select memory in use to be Phone and SIM.
Return to the Idle State.

- **Action**

详细描述执行 case 时的每一步操作。一般每一步操作都对应着一个期望中的结果。执行时可参照下面的期望结果。例如：

Start the procedure to add a new item to the Phonebook.
Enter some name and press **Ok**.

Enter some number such as **12345** and press **Ok**.

- **Expected result**

描述执行该 case 的期望中的结果，与上面的操作 Action 是相对应的。例如：

Name: query is displayed.
Number: query is displayed.

Saved to phone memory information note is shown. Phone goes to detailed memory screen.

2.6.3 错误报告

- **Title:**

标题是 Error Report 中非常重要的一部分，它要求简单明了地对 Error 作一个整体的描述，让不知道这个 Error 的人看了之后能够很清楚地知道这是个怎样的 Error。记得曾经有人提过“3W1H”的概念。也就是说，标题里面应该包括 What is the error, When will the error appear, Where may the error appear and How to make the error appear. 在 Title 后面，一般要写上 Feature Group 的名字。

例如：

Title: Call register: The phone doesn't remain in the same state after rejecting a call when viewing items under full window choice items in call register.

- **Severity (Fatal/Severe/Minor):**

Severity 用来描述 Error 的严重程度, 有三个级别: 较小的、严重的、致命的。Fatal Error 一般来说是指影响手机系统工作的 Error; Server Error 指的是影响用户操作的或者某些功能实现的 Error; Minor Error 指的是微小的、不影响手机功能正常使用的 Error。一般的 Error 如中文界面中的某个字不正确, 或者是英文界面中的某个单词拼写不正确; 左右功能键显示有误等等都属于 Minor。若手机的某个功能不能实现, 如不能发短信, 不能存电话号码, 不能进行充电等等都属于 Severe; 若手机开不了机, 或经常死机、重启等则是 Fatal。Severe 和 Fatal 两种 Error 对手机来说都是很严重的问题, 这个具体在做项目时可请示项目经理。

例如: Minor

- **Reproducible Error? (Yes/No, if No, how many times?) in English UI or Chinese UI?**

描述 Error 是否可再现, 如果每次操作都能出现, 就是可再现的。如果只是某一次操作才会出现这个 Error, 则是不可再现的。如果是不可再现错误, 要记录一共出现过多少次, 是在英文界面还是在中文界面。每个 Error 都有发生的前提条件和操作步骤。严格的说, 每个 Error 都是可重现的。但是, 发现这个 Error 的人可能没有能够找到这个 error 的完整的前提条件或者完整的操作步骤。所以, 现实中就有了很多不可重现的 Error。对于一个手机而言, 硬件, 软件, 语言包和 SIM 卡都是其重要的组成部分。所以, 在一个手机中用某种 SIM 卡在某种语言的 UI 上发现了某个 Error, 有可能在同样的手机, 同样的 SIM 卡, 不同的语言的 UI 上就没有这个 Error; 也有可能同样得手机上用不同的 SIM 卡也会没有这个 Error; 同样, 在不同的手机上也有可能发现不了这个 Error。总之一句话, 是否可重现, 要考虑手机硬件、软件版本、SIM 卡类型、UI 类型等等相关的影响, 不能简单的说某个 Error 可重现, 有的时候要加上注释。

例如: Yes, both in English UI and Chinese UI

- **Precondition:**

这里写的是在错误发生之前, 手机的状态。为了保证步骤的简洁, 这里要尽可能的详细。当然, 也不要写的很罗嗦。

- **How did you get to the state just before the error:**

详细描述在错误发生之前你是如何到达这个状态的, 要具体到每一步的操作。在这个部分, 步骤一定要清晰、简洁, 让别人能够轻松的理解并完成操作这个可以分成几个步骤来写, 如步骤 1、步骤 2、步骤 3 等。例如:

1. Menu --> Call register --> enter one of full window choice items;
2. Receive a call;
3. Reject it or remote end terminats the call.

- **Description of the error:**

对发生错误的描述, 用通俗易懂的话详细地把这个 Error 描述清楚。注意几个要点: “详细”、“简明”、“清晰易懂”。例如:

After rejecting a call or having a missed call when viewing items under full window choice items in call register. The phone goes back to the full window choice items

under call register.

- **Description of expected result:**

描述期望的操作结果，这个在 case 中一般都有说明，一般情况下，case 的执行结果就是期望的操作结果。这里描述的是，期望情况下，“应该”是什么结果。例如：

The phone **should** remain in the same state just as before receiving a call.

- **SIM card used:**

所用的 SIM 卡是中国移动 (CMCC) 还是中国联通 (CHN-CUGSM)。例如：CMCC

- **SW version and Language package:**

所测手机软件的版本号可通过在待机状态下按“*#0000#”来获得。

我们现在所测的手机语言包大部分都是 C 包，语言包可通过下面的方法来获得：

把手机恢复出厂设置，进入短信的编辑窗口，此时默认的输入法如果是“拼音”，则语言包为 C 包。例如：V5.20C

2.6.4 进度报告

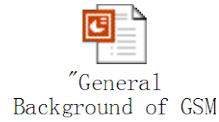
- 工作时间（小时数）；
- 测试用例执行情况：
 - **Total:**已经完成的测试用例数目；
 - **Fail:**其中出错的测试用例数目；
 - **Pass:**通过的测试用例数目；
 - **Not Test:**未测的测试用例数目；
 - **Not Available:**无法测试的测试用例数目；
- 发现的所有错误的列表；
- 执行的所有测试用例及其结果的列表。

2.6.5 总结报告

- 测试活动的时间
- 测试投入的人力
- 测试效果和结论
- 测试用例通过情况列表
- 发现所有错误的列表
- 所有仍未关闭的错误报告列表

3 手机相关

3.1 GSM



3.2 GPRS

见上面文件。

3.3 CDMA



3.4 3G

4 手机软件测试工程师必备素质

4.1 Team Leader 的任务和必备能力

- 熟悉本组成员，包括知识、能力、经验、爱好等等
- 是客户方和本组成员的接口，负责两者之间的沟通
- 负责分配本组的任务，包括制定计划和日程安排
- 总结每天的工作结果，若有重要的 Error 须汇报客户方负责人
- 新进测试工程师的培训
- 回答本组内其他测试人员的问题
- 制作工作进度表，随时报告本组工作进度
- 监督协调本组成员的工作
- 收集本组成员的建议和要求
- 部分测试工作

- 检查测试条件是否满足
- 控制工作质量

4.2 Error Manager 的工作内容

- 确认 Error 的真实性
- 确定 Error 的 Priority 和 Severity
- 遇到问题时需要和客户方负责人商量
- The daily work of an error manager is handling errors and answer questions asked by testers, engineers.

To finish the work well, an error manager must have the following skills.

1. An error manager must be more familiar with the UI spec than the others.
2. An error manager must be familiar with the process of handling errors.

For example:

If there is an error needing to be handled, the error manager should do the following things.

- a. Check whether the error report is complete or not.
 - b. Make sure the error report is clear and clean.
 - c. Using all the related knowledge to judge whether it is a real error.
 - d. If not, set it to be ignored. If yes, search all the errors in the database to find whether it is a known error.
 - e. If yes, set it to be duplicated. If not, try to reproduce it and add more information to the report.
 - f. During the above steps, a manager should keep more communication with the reporter.
 - g. Finally, check who is the SCF that the error should be sent to.
3. An error manager must be familiar with the process of an error's living.
 4. Be familiar with the tools, such as Lotus Notes, PC suite, Phoenix and so on.

4.3 测试工程师职责和应有素质

- Read UI Specification
- 学习数据库里的 Error Report (格式、内容和别人的思路)
- 测试工作
- 写 Error Report 和 Test Case
- 熟练使用相应的软件和工具
- 态度明确、端正, 有责任心, 严谨的工作作风
- 良好的沟通和协调能力, 积极主动地与别人交流
- 技术能力
 - 测试知识和技术
 - 编程知识和技术
 - 无线协议知识
 - 软件知识

- 硬件知识
- 语言水平
 - 英语水平