在项目测试过程中，测试方案制定的好坏，会直接影响到项目的的质量。因此需要制定一份完善的测试方案，那么如何才能制定一份完善的方案呢?

5W1H原则

不管在任何场景下，制定计划时5W1H原则都是需要适用的。

how：如何去测？用什么资源？依据什么？工具如何选型、案例要执行到什么粒度。

why：为什么要实现这个功能，背景和目的是什么，能给用户或公司带来多大的价值。

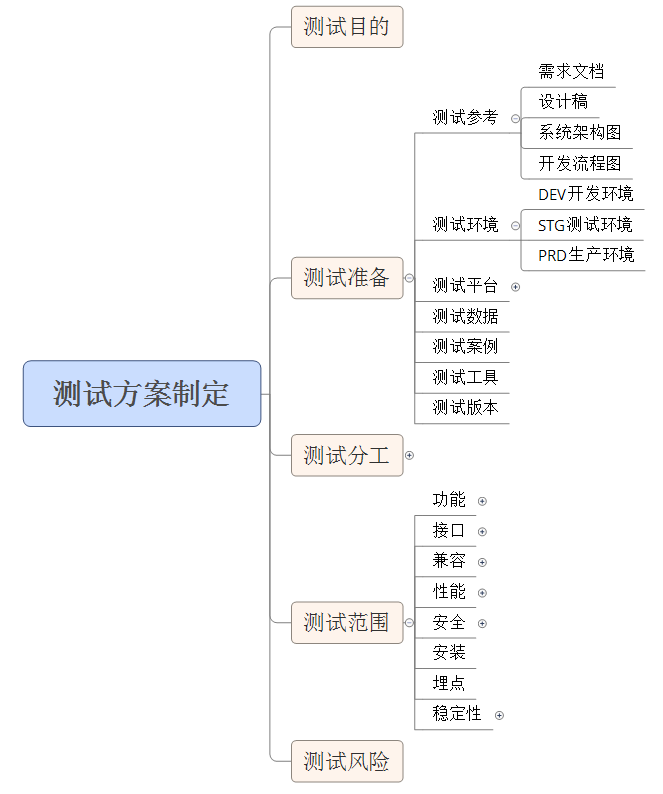
what：我需要做什么？任务的目的是什么？

when：项目周期多长，开发时间和提交测试时间是什么时候？什么时候需要给用户？测试周期需多长？

who：项目各个环节的直接责任人、干系人是谁？谁来主导负责？需要多少人力来参与？

where：相关资源的位置和路径，版本、文档。

那么具体来讲，测试方案一般包含哪些内容呢？



测试目的

通常测试目的包含如下几种：

看测试对象是否满足需求规格说明书，满足目前的要求及未来的发展需求。

看测试对象业务流程的合理性和正确性。

看测试对象的功能、兼容、性能、稳定性、安全测试，是否满足要求。

测试参考

需求文档：需求文档是测试设计和测试的基本依据，从需求文档中挖掘隐含的需求也是测试能力的体现。不过前期需求评审的越仔细，需求的准确性和完整性越高，后期修改和变更的几率就越低。

交互稿：通常体现了整个功能的业务流程，以及页面跳转关系。

设计稿：ui设计稿，通常作为UI界面测试的标准。

系统架构图：

开发流程图：理解开发的交互逻辑，明确代码输入输出规则，才能进行更精准的测试设计和测试，避免出现测试方向的偏差。具体怎么结合开发流程去丰富测试设计可以看这篇文章：https://blog.csdn.net/alice\_tl/article/details/79601697

还可以是一些参考文献和专业术语解释等等。

测试环境

一般项目中至少存在三套环境，开发环境（DEV）、测试环境（STG）、生产环境（PRD），有的项目中还存在预发布环境。

测试前要保证环境的连通性，因此要做一些前期准备，比如确认网络的墙是不是通的，测试的数据是不是提前准备好了。

测试平台

清楚要测试的平台及平台的特性，比如PC有PC的特性，Web和H5有各自的特性，Android和iOS有各自的特性。平台的特性也就决定了各个平台的兼容测试重点是不一样的，问题定位的方法和思路也是不一样的。

测试数据

提前针对要测试的环境准备数据，避免临时造数据造成不必要的测试时间影响。

测试案例

 一般会区分前端、后端案例，后端案例一般还包含接口测试脚本。

测试工具

相关要用到的工具平台，比如缺陷管理平台、案例平台，接口测试工具Jmeter等。

测试版本

 测试的版本号，版本下载的地址。

测试分工

需要多少个测试人力，是按功能模块划分，还是按照前后端测试来划分。

如果涉及到跨项目组的协作，那么各个项目组的研发内容，对应的测试分工。以及最下游端到端验收测试的标准。

测试范围

测试项及指标



功能测试内容

功能测试的方法，不管对于任何平台，软硬件测试，都是通用的。功能测试时除了要覆盖所有的功能清单，所有测试案例以外，也要重点测试核心业务场景和不稳定风险较高的模块。

性能测试内容

系统的CPU，内存，响应时间，流畅度，流量等。至于性能测试的指标可以参考这篇博文：

https://blog.csdn.net/alice\_tl/article/details/79967932

兼容测试内容

主要包含硬件、软件、网络、数据四个方面。 可以参考这篇博文：

https://blog.csdn.net/alice\_tl/article/details/75904056

稳定性测试内容

平均无故障时间达到X小时以上(android 8H,ios 2H)，过程中身边应用无Force close、ANR、Native Crash，无因身边应用导致的手机freeze、shut down或power cycle。

安全测试内容

安全测试主要监测程序漏洞和抗攻击能力、敏感数据泄露等。

安装测试内容

安装测试主要看程序是否能够兼容到各个机型，安装后产生的文件缓存信息等等，以及卸载之后是否有残留文件。

埋点测试内容

看埋点是否符合产品数据统计的要求，以及埋点的准确性。至于埋点的测试方法，可以看这篇博文：

https://blog.csdn.net/alice\_tl/article/details/78072830

测试风险

可能存在哪些风险，比如测试环境由于各种原因导致无法覆盖到的内容，或者测试环境和生产环境有差异的原因。

————————————————

版权声明：本文为CSDN博主「alice\_tl」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/alice\_tl/article/details/80218342