

软件绿色联盟应用体验标准 3.0

稳定性标准



编制单位：软件绿色联盟·技术与标准工作组

2019年7月

目 录

前 言	4
标准名称.....	5
1 概述.....	5
1.1 背景.....	5
1.2 目的.....	5
1.3 适用范围.....	5
1.4 总体框架.....	5
1.5 规范性引用文件.....	6
1.6 术语及定义.....	6
1.6.1 应用稳定性.....	6
1.6.2 故障率.....	6
1.6.3 自恢复.....	6
1.6.4 稳定性故障类型定义.....	7
2 稳定性标准规范要求.....	7
2.1 稳定性衡量标准.....	7
2.1.1 标准衡量的前置条件.....	7
2.1.2 故障率.....	8
2.1.3 资源过载.....	9
2.1.4 故障自恢复率.....	11
2.2 测试方法与活动.....	13
2.2.1 方法总体介绍和策略条件.....	13
2.2.2 应用稳定性测试时长要求.....	13
2.2.3 AI菜单遍历测试方法.....	14
2.2.4 Monkey测试方法.....	14
2.2.5 资源泄漏测试方法.....	15
2.2.6 踩内存测试方法.....	16
3 附录.....	16

3.1	附录一 不同类型应用故障率.....	16
3.2	附录二 故障率等级定义.....	17
3.3	附录三 稳定性常见问题.....	17
4	修订记录.....	19

前 言

本标准由软件绿色联盟技术与标准工作组提出并归档

本标准起草单位：软件绿色联盟

本标准主要起草人：阿里巴巴、百度、华为、腾讯、网易、360、中国泰尔实验室、新浪微博

标准名称

1 概述

1.1 背景

应用稳定性质量严重影响着应用的用户体验，为构筑极致用户体验，须建立一套应用稳定性质量管控体系。

本标准当前版本 (V3.0) 中对于应用质量只有单一的崩溃率指标，未覆盖影响用户体验的应用冻屏、踩内存、内存泄漏等故障，测试活动也仅有单一的Monkey测试，未能包含应用冻屏、踩内存、内存泄漏等，导致相应故障严重影响用户体验，亟需增强稳定性衡量标准和测试能力，提升应用上架前后相应故障有效拦截和质量评估能力，提升用户体验。

1.2 目的

1、通过增强稳定性衡量标准和测试能力，提升应用市场应用上架前质量，促进应用上架后的质量保障。

2、牵引生态内所有应用的稳定性质量改进，构建共同的稳定和体验良好的应用生态。

1.3 适用范围

本标准规定了应用的质量、应用体验标准。

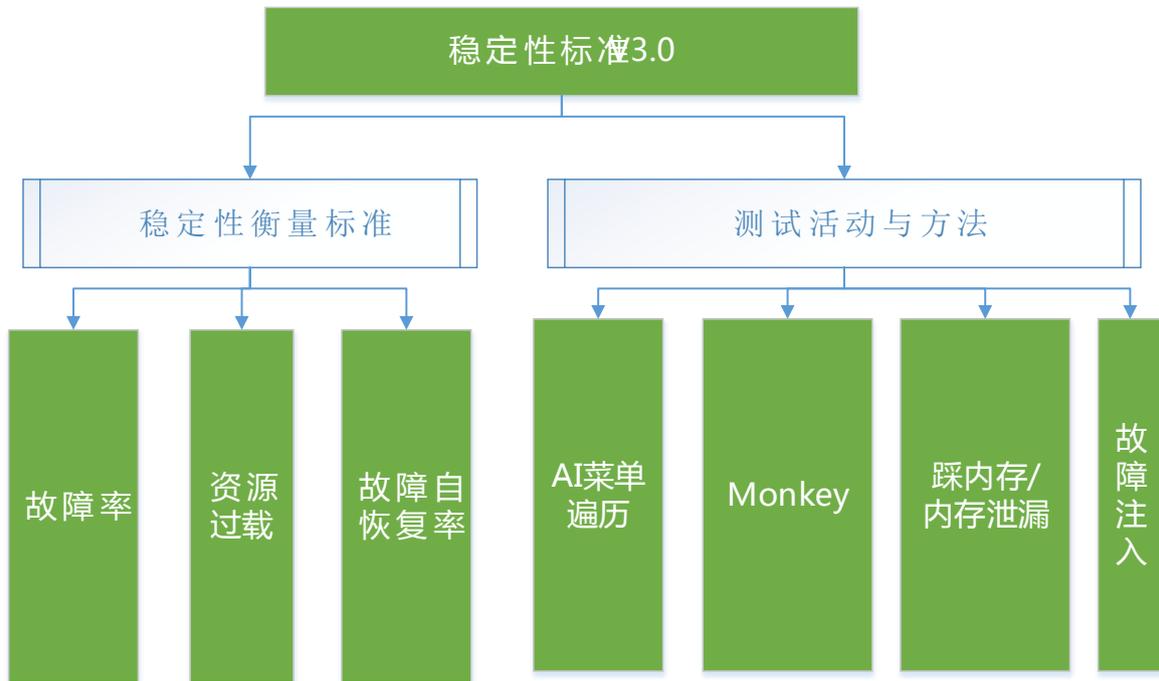
本标准适用于应用软件的设计、开发及检测。

1.4 总体框架

本标准包括稳定性衡量标准和测试活动与方法两部分构成。

稳定性衡量标准由故障率、资源过载、故障自恢复率三个维度组成。

测试方法与活动包括AI菜单遍历、Monkey、踩内存/内存泄漏、故障注入测试，给出支撑本标准的测试手段，用于完成稳定性标准验收和稳定性质量评估。



1.5 规范性引用文件

规范性引用《安卓绿色联盟应用体验标准 2.0 - 稳定性标准》。

1.6 术语及定义

1.6.1 应用稳定性

应用软件在规定的条件下和规定的时间内完成规定功能的能力(源于国际标准ISO-9126定义)。

1.6.2 故障率

当期用户使用过程中本应用发生的故障总次数与当期用户使用本应用总时长的比值。

1.6.3 自恢复

软件系统自我恢复机制，包含了确保系统稳定和平衡的自我恢复、调节机制。

1.6.4 稳定性故障类型定义

1.6.4.1 应用崩溃

在用户正常操作的情况下，应用突然出现应用强行退出、异常停止运行等完全不可用的情况。

1.6.4.2 应用冻屏

整个系统内核和应用系统是正常的，只是某个应用或者某几个应用卡住屏幕不动或突然出现应用程序在一段时间内未能及时响应的故障，即是用户俗称的应用死机、卡死、卡屏、应用无响应ANR问题。

1.6.4.3 应用资源异常(过载、泄漏、踩内存)

资源过载：在用户正常操作的情况下，应用滥用系统资源引起系统软件稳定性死机重启故障。

资源泄漏(包括内存泄漏)：在用户正常操作的情况下，因应用对内存、文件和线程使用不当，有限的资源超上限申请或使用完不释放会导致资源泄漏，进而引起应用崩溃、应用冻屏稳定性故障。

踩内存：在用户正常操作的情况下，程序指令非法访问内存地址，会造成应用崩溃、应用冻屏稳定性故障。

2 稳定性标准规范要求

2.1 稳定性衡量标准

2.1.1 标准衡量的前置条件

- 1、维测日志的完整性和有效性：故障日志结构和内容完整有效。
- 2、维测日志最小化：故障日志输出量有效控制。

2.1.1.1故障日志结构和内容完整有效

【规则】：原则上不允许修改原生稳定性日志内容和结构。

原始报错日志目录如下：

目录	介绍
/data/system/dropbox	1、报错Crash、ANR、Tombstone 等原生稳定性

	异常日志
/data/anr/traces.txt	1、抓取ANR、bugreport等日志各进程堆栈的中间过程文件 2、命令“kill -3 pid” 抓取Java进程堆栈记录文件
/data/tombstone/tombstone_x	1、记录C/C++崩溃日志

2.1.1.2故障日志输出量有效控制

【规则】：

1、日志设计规格：大于 24 小时运行日志可在当前的文件大小中存储。以 Log 为例，1 个文件 4MB，55 个文件，共 220MB， $220 \times 1024 / (24 \times 3600) = 2.61\text{KBps}$ 。

2、日志打印最小化原则：平均日志输出率不应大于 2610 字符/秒，考虑到 24 小时有休眠时间，和原生系统的日志量抵消，因此各模块的日志量要低于 2610 字符/秒。

2.1.2 故障率

2.1.2.1 故障率定义

故障率(λ)是单位时间内发生故障的概率，相比于平均故障间隔时间(MTBF)，故障率可以更直观地呈现应用软件稳定性表现。

2.1.2.2 故障率计算方法和测试活动要求

故障率指标通过实验室稳定性各测试活动获得，计算公式为：故障率=F(对应类型故障总和)/T(对应类型稳定性专项测试周期(单位h))。

各故障类型故障率需满足如下要求：

故障大类	故障小类	故障率	测试活动
应用崩溃	Crash	应用崩溃&应用冻屏故障率： I 级：< 1 次/h	AI 菜单遍历、 Monkey 测试
	Tombstone		
应用冻屏	应用卡死	II 级：< 0.5 次/h	
	应用无响应	III 级：0	
应用资源	资源过载	见 2.1.3 章节资源过载要求	AI 菜单遍历测试

	资源泄漏	0	资源泄漏测试
	踩内存	0	踩内存测试

为持续监控真实用户长时间使用过程中应用稳定性质量，特定义线上故障率指标要求如下：

线上故障率的计算方式为：在某个观测期间，应用的故障发生总数与应用总累积工作时间之比，即： $\lambda = \frac{\sum f_{t_i}}{\sum t_i}$ ，故障率的通常单位为 FITs (Failures in Time)，1FIT = 10⁻⁹/h (h: 小时)。

因手机应用面向消费者，为了更清晰直观呈现手机应用稳定性，将手机应用故障率的单位定义为“次/月”，即手机应用软件故障率的计算方式为：当期用户使用过程中本应用发生的故障总次数/当期用户使用本应用总时长(次/月)。

线上故障率稳定性等级要求如下：

I 级： < 10次/月

II 级： < 5次/月

III 级： < 1次/月

注1：线上故障率等级I级定义来源为业界持平现状(数据详见3.1章节附录一)、II级定义来源为业界领先要求、III级定义为业界最佳水平要求。

注2：线上故障率统计依赖于完善的线上异常上报及处理机制，通过收集线上用户真实场景下发生的稳定性故障日志获取，需要各应用稳定性维测能力符合本标准稳定性日志维测规范要求(详见第2.1章节)。

注3：线上故障率作为故障率的延伸，旨在让应用在使用一段时间内质量还保持在应用稳定性质量要求之上。当前仅作为牵引指标，暂不在绿标测试中度量。

2.1.3 资源过载

2.1.3.1 资源过载定义

在用户正常操作的情况下，由于对内存、文件、线程使用不当，有限的资源超出上限申请或使用完不释放等资源资源过载场景的系统管控能力。

2.1.3.2 资源过载计算方法和测试要求

【资源过载计算方法】：

根据对应用稳定性影响程度，选择TOP资源过载故障作为重点统计度量对象：

资源名称	序号	资源含义	过载门限	意义	告警和管控
NOTIFICATION	1	应用发送 Notification 速率(个/秒)	第三方应用：20个/秒	防护第三方应用发送 NOTIFICATION过速引起的手机变慢、黑屏、应用ANR类问题；	告警 不管控
BROADCAST	2	应用发送 Broadcast速率(个/秒)	第三方应用：30个/秒，连续3个周期(1s/每周期)	监控第三方应用发送 BROADCAST过速引起的引起手机变慢、黑屏、应用ANR类问题；	告警 不管控
RECEIVER	3	应用注册的 Broadcast Receiver 个数(个)	第三方应用：1000个	防护第三方应用注册 BROADCAST Receiver泄漏引起的应用资源异常故障；google系统总共1000个，基本不会触发，用于异常场景的定位	告警+管控-杀后台
ALARM	4	应用注册的 ALARM个数(个)	第三方应用：50个	防护第三方应用注册 ALARM(定时器)太多引起的变慢、黑屏、应用ANR类问题；	告警+管控-杀后台
APPOPS	5	应用注册的 APPOPS个数	第三方应用：200	监控第三方应用注册APPOPS太多引起的变慢、黑屏、应	告警+管控-杀后

		(个)	个	用anr类问题，主要为异常申请不释放，	台
PIDS/T hread	6	应用fork的子进程和线程个数 (个)	第三方应用：3000个	防护第三方应用fork子进程和线程太多引起的变慢、黑屏、应用anr类问题	告警+管控-不给申请
CURSOR	7	应用注册的CURSOR个数 (个)	第三方应用/华为应用/系统应用：300个	防护第三方应用打开Cursor太多引起的变慢、黑屏、应用ANR类问题	告警+管控-杀后台
BINDER	8	应用注册的同一系统接口的BINDER个数 (个)	第三方应用：14个	监控第三方应用同一系统接口的注册BINDER的情况，防止占满导致的系统卡死和黑屏	告警+管控-排队不丢
ACTIVITY	9	应用Activity注册windows个数(个)	第三方应用：20个	监控第三方应用同个Activity注册多个窗口的情况	监控+管控(杀后台)

【资源过载测试】：

标准编号	2.1.3.2	资源过载测试
标准描述	稳定性资源过载测试	
测试方法和用例	稳定性故障注入测试	
测试工具	OpenLab DevEco平台	
判定标准	资源过载满足要求，超标有异常告警	

2.1.4 故障自恢复率

2.1.4.1 故障自恢复率定义

应用各故障发生后，能通过自动恢复机制恢复故障的比例(%)。

2.1.4.2 故障自恢复率计算方法和测试要求

【故障自恢复率计算方式及标准要求】

1、计算公式：

各类故障自动恢复数/故障总数*100%

稳定性故障中，应用资源异常主要依赖设计优化或系统恢复方案，故本方案当前只针对应用崩溃、应用冻屏故障能力进行验收。

2、标准要求：

I 级：≥ 70%

II 级：≥ 80%

III 级：≥ 90%

3、应用崩溃、应用冻屏故障恢复要求：

故障类型	故障名	自恢复方案	自恢复效果	耗时
应用崩溃	应用单次 Crash	1、单次 Crash，提示重启，拉起到 crash 界面	应用恢复到退出界面	立即恢复
	应用屡次退出	1、提示清除应用缓存 2、提示卸载并重新安装 3、提示版本不兼容，建议更新版本	应用使用正常	立即恢复
	Tombstone 异常退出	2、弹框提示用户，应用退出	应用退出，系统正常	立即恢复
应用冻屏	应用频繁 ANR	3、检测系统资源状态和核心服务状态，执行对应的恢复动作		立即恢复
	应用单次 ANR	4、弹框提示用户，选择等待或者退出	5S 内应用弹框，系统正常	立即恢复

【故障自恢复率测试】

应用故障自恢复率通过应用稳定性故障自恢复注入测试专项获得：

标准编号	2.1.4.2	自恢复率测试
标准描述	故障自恢复率测试	
测试方法和用例	稳定性故障自恢复率注入测试	

测试工具	OpenLab DevEco平台
判定标准	故障自恢复率满足II级要求，不达标有异常告警

2.2 测试方法与活动

2.2.1 方法总体介绍和策略条件

根据应用上架测试、绿标测试的时长、能力成熟度不同要求，定义了各阶段测试活动策略如下：

序号	触发方法	应用上架测试	绿标测试
1	AI菜单遍历	必选	必选
2	Monkey	可选	必选
3	内存泄漏验收测试	必选	必选
4	踩内存验收测试	必选	必选

2.2.2 应用稳定性测试时长要求

【应用稳定性测试时长定义】：应用稳定性各专项测试周期统称为应用测试时长(单位h)。

【应用稳定性测试时长要求】：应用稳定性测试时长为4h。

【应用稳定性测试时长推导】：应用稳定性测试是在实验室中进行，测试时长是受限的，无法像真实用户那样真正长时间运行，但是我们可以通过加大使用频率来缩短测试时长，当前TOP应用类型中，单应用人均使用时长为12小时/月，单应用每个页面停留平均时间为161秒，那么实验室测试可以将页面停留时间缩短3倍到54秒，在大约4小时时间内完成用户1个月同样的应用体验时间和页面覆盖。

针对不同类型的应用，根据大数据可以获取到用户各类使用频度，我们可以设置一个相对合理的压缩测试时长，达到类似用户长时间使用的实验室测试效果。不同类型应用的压缩测试时长如下：

应用类型	页面停留时间(秒)	页面停留时间(秒), 等比缩放 3 倍	月使用时长(小时)	月使用时长(小时), 等比缩放 3 倍
影音娱乐	106.03	35.34	16.24	5.41
社交通讯	53.76	17.92	52.95	17.65

游戏	846.32	282.11	1.31	0.44
购物比价	30.14	10.05	5.49	1.83
金融理财	34.56	11.52	6.52	2.17
实用工具	63.60	21.20	4.61	1.54
出行导航	201.47	67.16	3.89	1.30
新闻阅读	108.56	36.19	17.34	5.78
教育	12.05	4.02	0.64	0.21
平均值	161.00	53.67	12.00	4.03

我们取各类应用的平均压缩测试时长4H为基准，作为统一的应用测试时长要求。

2.2.3 AI 菜单遍历测试方法

AI菜单遍历测试是基于AI窗口识别技术和深度遍历各应用页面有效控件算法的自动化测试专项：

标准编号	2.2.3	AI 菜单遍历测试
标准描述	AI菜单遍历测试	
测试方法和用例	AI菜单遍历	
测试工具	OpenLab DevEco平台	
判定标准	a. 稳定性测试要覆盖80%用户的主流机型 b. 安装、启动等场景必须覆盖 c. 完成2小时内满足故障率标准要求	
需考虑的特殊事项	覆盖机型：至少覆盖应用目标用户机型分布中top10，保证80%的机型被覆盖到。	

2.2.4 Monkey 测试方法

Android原生自带的通过随机输入按键键值来测试整机稳定性的小程序：

标准编号	2.2.4	Monkey测试
标准描述	Monkey测试	
测试工具	OpenLab DevEco平台	

判定标准	<ul style="list-style-type: none"> a. 稳定性测试要覆盖80%用户的主流机型 b. 安装、启动等场景必须覆盖 c. Monkey确保覆盖页面>70% d. 完成2小时Monkey不出现异常
需考虑的特殊事项	<p>覆盖机型：至少覆盖应用目标用户机型分布中top10，保证80%的机型被覆盖到。</p> <p>覆盖场景：全新安装、覆盖安装、冷启动、热启动及6小时Monkey，Monkey过程要能覆盖登录与非登录状态，通过优化执行路径、多机运行累加结果等方式，保证页面覆盖率>70%。</p>

2.2.5 资源泄漏测试方法

【内存泄漏测试方法】：

标准编号	2.2.5.1	内存泄露拦截测试
标准描述	内存泄漏拦截测试	
预置条件	<ul style="list-style-type: none"> a. 软件版本具备内存泄露维测功能且维测开关必须开启 b. 覆盖应用返回桌面、热启动流程(不小于4次) c. 覆盖应用关闭、冷启动流程(不小于4次) 	
判定标准	<ul style="list-style-type: none"> a. 退出、关闭、热启动、冷启动等场景必须覆盖 b. 完成1小时AI菜单遍历不出现内存泄露问题 	
需考虑的特殊事项	覆盖应用：上架应用市场应用	

【线程/FD资源泄漏测试方法】：

标准编号	2.2.5.2	线程/FD资源泄露拦截测试
标准描述	线程/FD资源泄露拦截测试	
原理和测试能力描述	<ul style="list-style-type: none"> a. FD泄露(文件描述符耗尽探测)：当某个进程fd消耗超过水位的时候，生成告警信息，并采集该进程当前的fd消耗情况，最大采集次数可以配置。 b. 子线程泄露(子线程创建过多)：监控系统中任意进程 	

	的子线程数目,当某个进程的线程数消耗超过水位的时候,生成告警信息,并采集该进程当前的线程消耗情况,最大采集次数可以配置。
前置条件	a. 软件版本具备资源泄漏维测功能且维测开关必须开启 b. 覆盖应用返回桌面、热启动流程(不小于4次) c. 覆盖应用关闭、冷启动流程(不小于4次)
判定标准	a. 退出、关闭、热启动、冷启动等场景必须覆盖 b. 完成1小时AI菜单遍历不出现线程/FD资源泄露问题
需考虑的特殊事项	覆盖应用: 上架应用市场应用

2.2.6 踩内存测试方法

标准编号	2.2.6	踩内存测试
标准描述	踩内存拦截测试的标准	
前置条件	a. 提供被测应用ASAN版本(被测应用源码编译了ASAN)。 b. 整机软件版本具备ASAN检测能力且已配置被测应用的ASAN检测开关。 c. 覆盖应用返回桌面、热启动流程(不小于4次)。 d. 覆盖应用关闭、冷启动流程(不小于4次)。	
判定标准	a. 退出、关闭、热启动、冷启动等场景必须覆盖。 b. 完成2小时AI遍历测试不出现地址越界问题。	
需考虑的特殊事项	覆盖应用: 上架应用市场应用	

3 附录

3.1 附录一 不同类型应用故障率

应用类型	故障率(次/月)
影音娱乐	2.268
社交通讯	15.12

游戏	7.3008
购物比价	18.072
金融理财	9.5976
实用工具	8.424
出行导航	7.0272
新闻阅读	11.592
教育	11.2392
平均故障率	10.07

3.2 附录二 故障率等级定义

线上故障率等级I级定义来源为业界持平现状、II级定义来源为业界领先要求、III级定义为业界最佳水平要求。从不同类型应用故障率统计数据来看，当前现状数据为10.07次/月，故将线上故障率I级定义为10次/月。

用户人均月度应用使用时长为12小时，实验室测试周期等比压缩为4h，故线下故障率按照等比减少3倍为3.3/4h，即线下故障率I级约为1次/h。

3.3 附录三 稳定性常见问题

故障类型	故障子类	故障检测
应用冻屏	延时类故障 主线程耗时/ 消息循环阻 塞、IPC 超 时、应用卡 死	ANR：事件分发开始处设置 Timeout 以及结束处移除 Timeout。 触发机制为在主线程执行的所有组件 1、Input 事件 2、广播回调 3、服务回调
应用崩溃	Java 运行异常	1、故障(比如 runtime_exception)发生时，安卓 art 虚拟机捕获到 unchecked exception 后，退出故障所在进程。 2、在/data/system/dropbox 目录下生成名为 system_app_crash@xxx.txt 或。

		data_app_crash@xxx.txt 的日志，同时在流水日志中记录该 crash。
	空指针	<p>1、故障(比如 native 代码非法地址访问)发生时，内核发出信号 SIGSEGV(其它故障信号可能不同)，导致故障所在进程退出。</p> <p>2、在/data/system/dropbox 目录下生成名为 SYSTEM_TOMBSTONE@xxx.txt 以及。</p>
资源泄漏	内存泄漏	<p>1、lmk: 内核每隔一段时间检查一次，当系统剩余可用内存较低时，便会触发杀进程的策略，根据不同的剩余内存档位来选择杀不同优先级的进程。内核流水里面可以看到“Killing XXX”</p> <p>内核日志中有 lowmemkiller。</p> <p>2、kernel oom: 分配内存不足时触发，将得分最高的进程杀掉，内核日志里面可以看到 Out of memory 流水。</p> <p>3、java oom: Java 分配堆内存不足时触发，抛出 OOM 异常，产生 crash 日志。</p>
	线程泄漏	线程数超过 17092 个无法创建新线程，本身不产生日志，线程泄漏也往往导致内存泄漏，可能导致 OOM。
	通知过载	应用通知发送超过 20 个/秒。
	全局引用泄漏	global reference 超过 51200 个。
	FD 泄漏	fd 数超过 32768 无法创建新 fd，本身不产生日志。

4 修订记录

日期	修订内容
2017年5月	安卓绿色联盟应用体验标准1.0发布
2018年7月	无修订跟随安卓绿色联盟应用体验标准2.0发布
2019年7月	更名为《软件绿色联盟应用体验标准3.0_稳定性标准》： 1、衡量标准更新：单一的应用崩溃率优化为故障率、资源过载、故障自恢复三项质量和能力要求。 2、测试活动增加：单一Monkey测试基础上增加AI菜单遍历测试、踩内存、内存泄漏测试、故障注入测试。