第二部分 软件测试

2.15 静态白盒测试



本节教学目标

- 理解静态白盒测试的背景知识
- 掌握静态白盒测试的方法
- 重难点: 掌握静态白盒测试的方法

目录

- 1 静态白盒测试背景知识
- 2 静态白盒测试怎样做
- 3 对静态白盒测试的总结

静态白盒测试背景知识

- 什么是静态白盒测试?
 - 对系统静态检查,这种检查通常不需要运行被测软件,而是直接对软件形式和结构进行分析
- 为什么进行静态白盒测试
 - 贝尔实验室在其开发中引入审查后,生成率提高14%,质量提高10倍
 - · 某大型电力交换系统,通过使用审查,发现错误的成本降低10倍,成效是测试的20倍
 - · 美国天合汽车集团对大型软件系统的研究发现, 2019个由用户发现的错误 导致代码变更
 - 代码审查可发现62.7%的错误,设计审查可发现57.7%的错误

静态白盒测试怎样做

- 代码检查
- 静态结构分析
- 代码质量度量

代码检查

- 代码检查主要是通过同行评审来发现缺陷;
- 以评审会议为形式,通过多人对软件交付物进行检查,从而发现缺陷,或者获得改进优化的机会
- 同行评审往往需要投入大量时间和人力资源

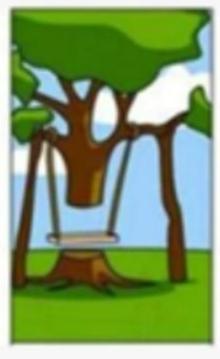
客户真正需要什么?













客户真 正要的

客户口中 的需求

研发经理 所理解的

系统分析师 所理解的

程序员编码实现的

为什么需要评审





为什么需要评审

- 为什么需要评审?
 - 开发早期无法提供可运行对象, 无法执行测试
 - •特定类型的缺陷,通过测试无法发现
- 增加评审的意义:
 - 有助于发现开发早期需求和设计中的缺陷
 - 促使参与者在有监督压力下工作, 提高责任心
 - 有助于程序员发现不足, 提高工作质量
- 同行评审的核心: 缺陷预防
- •目的:发现缺陷,改进开发过程

同行评审的方法

- 审查 (Inspection)
- · 团队评审(Team Review)
- 走查 (Walk Through)
- · 结对编程(Pair Programma)
- 同行桌查 (Peer Desk Check)
- 轮查 (Pass Around)
- 特别检查 (Ad hoc Review)

评审						i	平审过程	L		正式
方法	评审目的	参与人员	评审形式	连用对象	计划	准备	会议	修复	确认	程度
中士	发现缺陷,找到违 反既定标准的问题	不含作者, 3~8 人	专门的会议	软件生命周期 中重要阶段的 产品	是	是	是	是	是	最正式
团队评审	发现缺陷, 达成共 识, 教育参加者	作者可为组长, 3~5人	专门的会议	阶段性产品	是	是	是	是	是	
走查	发现缺陷, 达成共 识, 教育参加者	作者为主导。 2-3人	专门的会议	祭物,蓝图, 源代码	£	否	£	是	否	
结对编程	发现缺陷并立即修复	结对钨程人员 (2人)	两程序员在同一 个工作站上进行	产品模块开发 (包括设计、 算法、代码)	£	委	持续	Ą	Ą	
同行。亲查	发现缺陷	不含作者,单人	独立评审	阶段性产品	杏	是	否	是	否	
轮查	发现缺陷	不含作者,多人 并行同行束查	每人分别独立详 审,最后汇总	阶段性产品	杏	是	可能	£	杏	+
特别 松查	解决当前问题	羊人(程序员)	与作者讨论	需要解决的问 题	否	否	是	是	否	最随意

评审						i	平审过程	L		正式
方法	评审目的	参与人员	评审形式	连用对象	计划	准备	会议	修复	确认	程度
**	发现缺陷,找到违 反既定标准的问题	不含作者,3~8	专门的会议	软件生命周期 中重要阶段的 产品	是	£	是	英	是	最正式
团队评审	发现缺陷, 达成共 识, 教育参加者	作者可为组长, 3~5人	对缺	陷修复的	结果	是召	糯	要	是	1
走查	发现缺陷, 达成共 识, 教育参加者	作者为主导, 2~3人	指派专人						*	
结对编程	发现缺陷并立即修复	结对编程人员 (2人)	进行了正		, 110	且才	云	"	是	
同行业查	发现缺陷	不含作者,单人	独立评审	阶段性产品	*	是	否	£	*	
轮查	发现缺陷	不含作者,多人 并行同行束查	每人分别独立详 审,最后汇总	阶段性产品	*	是	可能	£	杏	+
特别检查	解决当前问题	羊人(程序员)	与作者讨论	需要解决的问 题	杏	否	是	是	否	最随意

评审						i	平审过程			正式
方法	评审目的	参与人员	评审形式	通用对象	计划	准备	会议	修复	确认	程度
辛士	发现缺陷,找到选 反既定标准的问题	不含作者, 3~8 人	专门的会议	软件生命周期 中垂鱼100段的	£	是	是	是	是	最正
团队评审	发现缺陷, 达成共 识, 教育参加者	从本可为相	审过程不限 是从产品模				是	是	是	×
夫士	发现缺陷, 达成共 识, 教育参加者	作者为主号 持 2-3人	续到开发完	成为止			£	£	杏	
结对编程	发现缺陷并立即修复	结对编程人员 (2人)	两程序员在同一 个工作站上进行	产品模块开发 (包括设计、 算法、代码)	£	杏	持续	Ł	是	
同行业查	发现缺陷	不含作者,单人	独立详审	阶段性产品	*	是	杏	是	否	
轮查	发现缺陷	不含作者,多人 并行同行束查	每人分别独立详 审,最后汇总	阶段性产品	否	是	可能	是	*	+
特别检查	解决当前问题	華人 (程序員)	与作者讨论	需要解决的问 题	否	否	£	£	否	最健意

评审		12.00				÷	平审过程	l l		正式
方法	评审目的	参与人员	评审形式	通用对象	计划	准备	会议	修复	确认	程度
审查	发现缺陷,找到递 反既定标准的问题	不含作者,3~8	专门的会议	软件生命周期 中重要阶段的 产品	是	是	是	是	是	最正式
团队评审	发现缺陷, 达成共 识, 教育参加者		不确定是				£	是	£	T
走查	发现缺陷,达成共 识,教育参加者	2-	会议中由参 评审并汇总				是	是	委	
结对编程	发现缺陷并立即修复		別抽时间单		由约	且织	持续	是	是	
同行桌查	发现缺陷	不合作 首 对	意见加以汇	思		*	*	是	否	
轮查	发现缺陷	不含作者,多人 并行同行束查	每人分别独立详 审,最后汇总	阶段性产品	*	£	可能	是	杏	+
特别检查	解决当前问题	羊人(程序员)	与作者讨论	需要解决的问 题	否	否	是	是	否	最随意

评审		143,534					正式			
方法	评审目的	参与人员	评审形式	连用对象	计划	准备	会议	修复	确认	业式 程度
**	发现缺陷,找到选 反既定标准的问题	不含作者, 3~8 人	专门的会议	软件生命周期 中重要阶段的 产品	是	是	是	是	是	最正式
团队评审	发现缺陷, 达成共识, 教育	作者可为组长,	专门的会议	阶段性产品	是	是	是	£	是	
夫士	识, 教育	的是发现缺 程简洁,1~		蓝图, 码	£	委	是	是	*	
结对编程	发现数陷于	(2人)	个工作站上进行	快开发 (包括设计、 算法、代码)	£	杏	持续	是	是	
同行业查	发现缺陷	不含作者,单人	独立详审	阶段性产品	\$	是	*	是	否	
轮查	发现缺陷	不含作者,多人 并行同行来查	每人分别独立详 审,最后汇总	阶段性产品	杏	是	可能	Ļ	杏	*
特别	解决当前问题	单人(程序员)	与作者讨论	需要解决的问 题	否	否	是	是	否	能意

评审						÷	平审过程	l.		正式
方法	评审目的	参与人员	评审形式	通用对象	计划	准备	会议	修复	确认	程度
审查	发现缺陷,找到选 反既定标准的问题	不含作者, 3~8 人	专门的会议	软件生命周期 中重要阶段的 产品	是	是	是	是	是	最正式
团队详审	发现缺陷, 达成共 识, 教育参加者	作者可为组长, 3~5人	专门的会议	阶段性产品	是	是	是	是	是	Т
走查	R, 教育引	正规	200 Arts 807 100	再々与本*	7 40 V	F 200 el		是	否	
结对编程	发现机器并	的不仅是发 术水平、优	化代码质量	、改进开发				是	是	
同行。杂查	● 被评审的对象更关键,流程更复杂 ● 使用最为广泛								否	
轮查	发现缺陷	不含作者,多人 并行同行杂查	每人分别独立详 审,最后汇总	阶段性产品	杏	是	可能	是	杏	*
特别 检查	解决当前问题	单人(程序员)	与作者讨论	需要解决的问 题	*	否	是	是	否	健意

同行评审的5种角色

讲解员

负责讲解被评审的工作产品

记录员



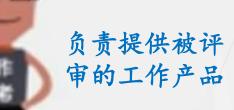






组织者、协调员, 负责组织评审会议







评审员

负责评审工作产品,有时也负责设计测试用例

评审流程





计划评审会议



一般地,设计部门应在评审前3天向项目管理部门提交《设计和开发 评审表》,经批准后,进入计划评审会议阶段

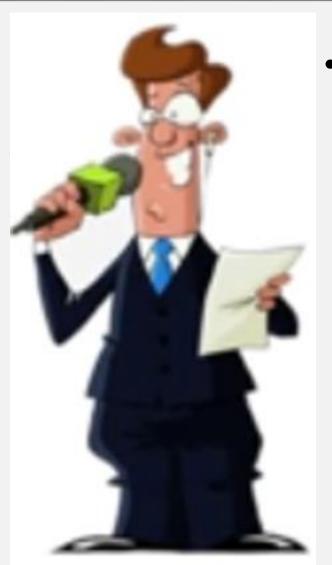
计划评审会议



• 项目经理:根据申请表来指定合适的会议主持人

作者:提供工作产品作为被评审的对象,并提交前检查是否符合相关标准和规范

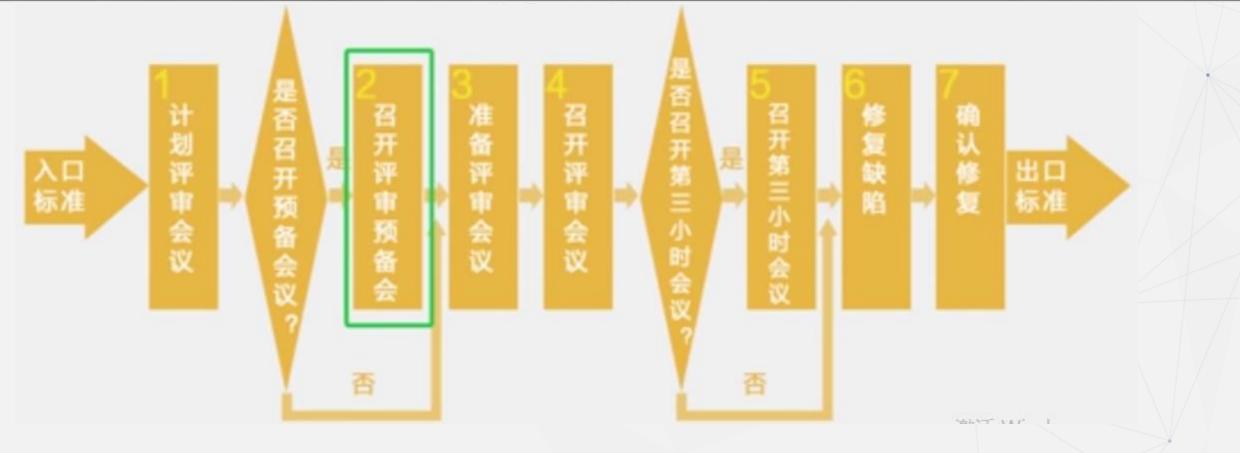
主持人工作



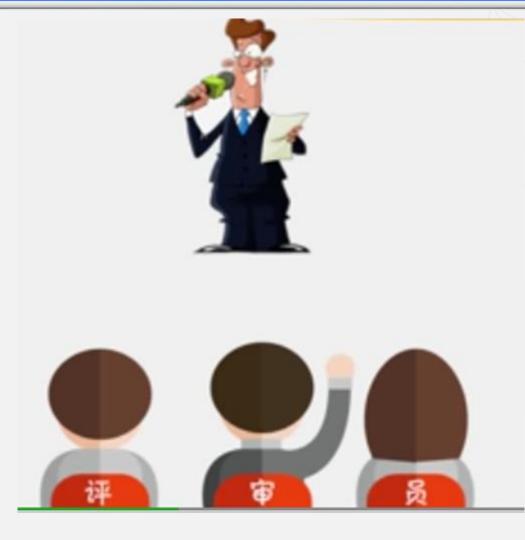
· 主持人:对本次评审会议进行规划。包括:制定评审计划,检查入口标准,准备评审材料包,选择合适的评审员,分发评审材料

召开评审预备会





召开预备评审会议



- 评审员向主持人提出申请,由主持人决定是否需要召开评审预备会
- 时间 < 2小时
- 了解评审流程、目的
- 理解自己的责任
- 评审材料正确无误

召开预备评审会议



向评审员说明评审流程及 相关要求,确定评审重点



介绍工作产品 和相关材料





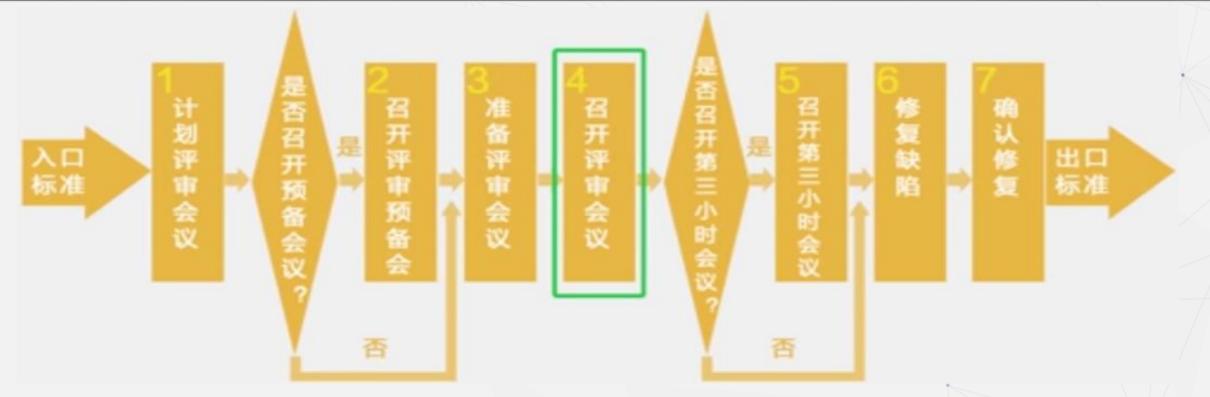
听取作者和主持人的介绍, 查看所有得到的工作产品 正确无误

准备评审会议



- 评审员:检查工作产品,记录发现缺陷,反馈给主持人。提前准备测试用例
- 主持人: 汇总收集的审阅情况记录表, 并判断是否需要增加评审的投入

召开评审会议

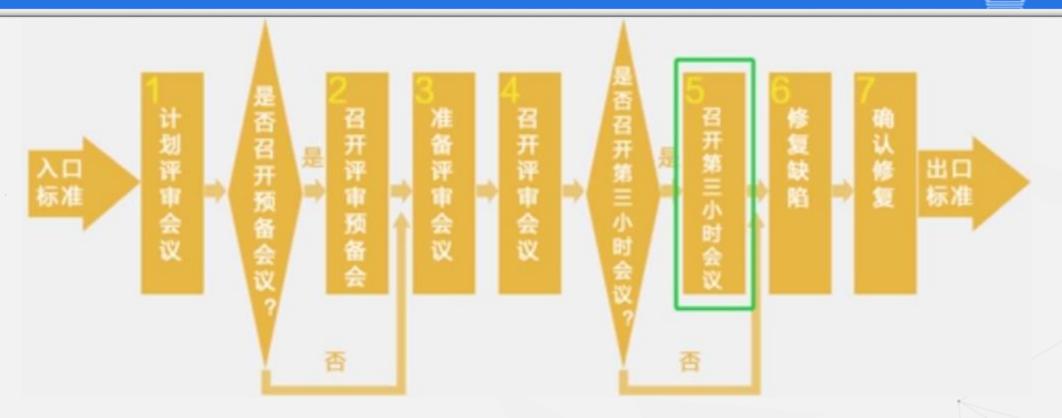


- 参会人员职责:
- 主持人:召开并主持会议,控制会议进度,维持会议程序,决定是否要召开第3小时会议,会后提交评审报告,给出评审结论

召开评审会议

- 讲解员: 讲解工作产品, 引导评审员浏览工作产品
- 评审员: 听取讲解,发表意见,指出问题,与作者确定问题,定义问题的严重程度
- 作者: 倾听讲解和评审员的意见, 回答提问
- •记录员:记录每个达成共识的缺陷,确保评审员同意对问题的记录, 并记录未达成共识的缺陷,标记为TBD,作为第3小时会议评审的对 象。更新审阅情况记录表

召开第3小时会议



• 主持人: 主持并维持会议程序, 控制会议进度。会议结束后一天内负责更新审阅情况记录表, 撰写评审报告, 并给出评审结论

召开第3小时会议

- 评审员:对标记为TBD的问题进行讨论,给出确定意见,并针对 达成共识的缺陷修复方案提出自己的意见
- 作者: 倾听评审员的意见, 提出自己的看法
- •记录员:记录每个达成共识的缺陷及其对应的解决方案

修复缺陷





修复缺陷



- 定位、调试和修复工作产品
- 提交工作产品
- 更新审阅情况记录表
- 分析整理缺陷清单
- 提供给评审员

确认修复



确认修复





确认工作产品,查看记录表,判断是否符合退 出标准





检查工作产品 判断缺陷是否修复 更新记录表 提交给主持人



评审结果

- 正常:评审专家做好了评审准备,评审会议顺利进行,达到了 预期目的,达成明确的评审结论,不需要再次评审
- ·延期: 30%以上的评审专家未做好评审准备,会议无法正常进行, 需要重新安排评审日程
- •取消:初审阶段就发现工作产品中存在太多的问题,需要作者进行修复,然后再进行第二次同行评审

静态白盒测试怎样做

- 代码检查
- 静态结构分析
- 代码质量度量

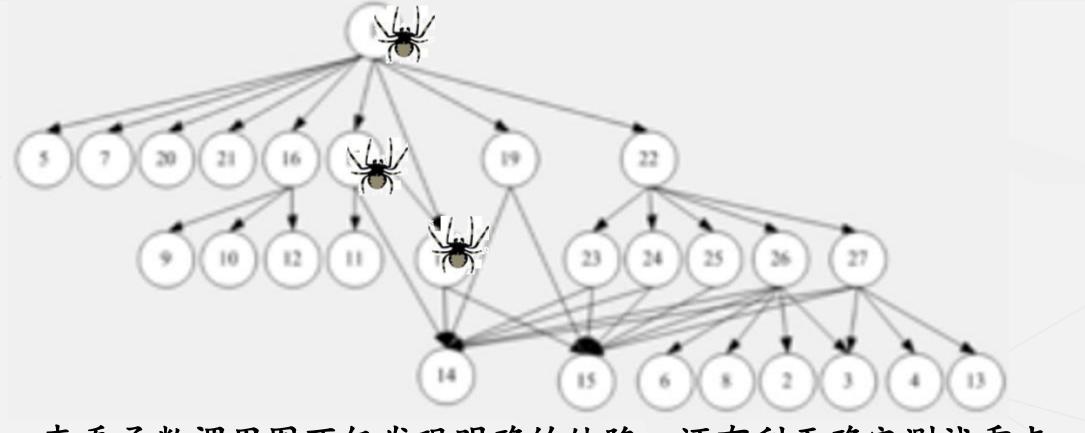
静态结构分析

- 研究表明,程序员38%的时间都花费在对软件系统的理解上。
- · 静态结构分析: 通过引入不同形式的图表, 帮助我们快速了解程序设计和结构, 更好地理解源代码, 有利于找到程序设计的缺陷和代码优化的方向
- 函数调用关系图:
 - 通过树形方式展示被测系统中各函数之间的调用关系

静态结构分析

- 函数之间的调用关系是否符合要求;
- 是否存在递归调用(对内存消耗大,长时间运行容易导致崩溃)
- 函数调用层次是否太深,过深的调用层次容易导致数据和信息 传递错误和遗漏,并增大测试的负担
- 是否存在孤立函数

静态结构分析



- 查看函数调用图不仅发现明确的缺陷,还有利于确定测试重点:
 - 根节点需要优先测试
 - 叶子节点需要优先测试
 - 接口数量多的节点需要优先测试

静态结构分析

- 对函数控制流图进行分析
 - 是否存在多出口情况,多出口容易导致空指针,内存未释放这类缺陷
 - 是否存在孤立语句
 - 环复杂度是否太大
 - 是否存在非结构化设计

静态白盒测试怎样做

- 代码检查
- 静态结构分析
- 代码质量度量

- 代码行度量
- McCabe度量法
 - 计算V(G)
- Halstead软件科学法
 - $N=N1+N2=n_1\log_2(n1)+n_2\log_2(n2)$
 - · (n1=不同运算符的个数 n2=不同运算对象的个数)
- 结构度量
 - 使用扇入、扇出来评价软件设计



• 软件质量模型

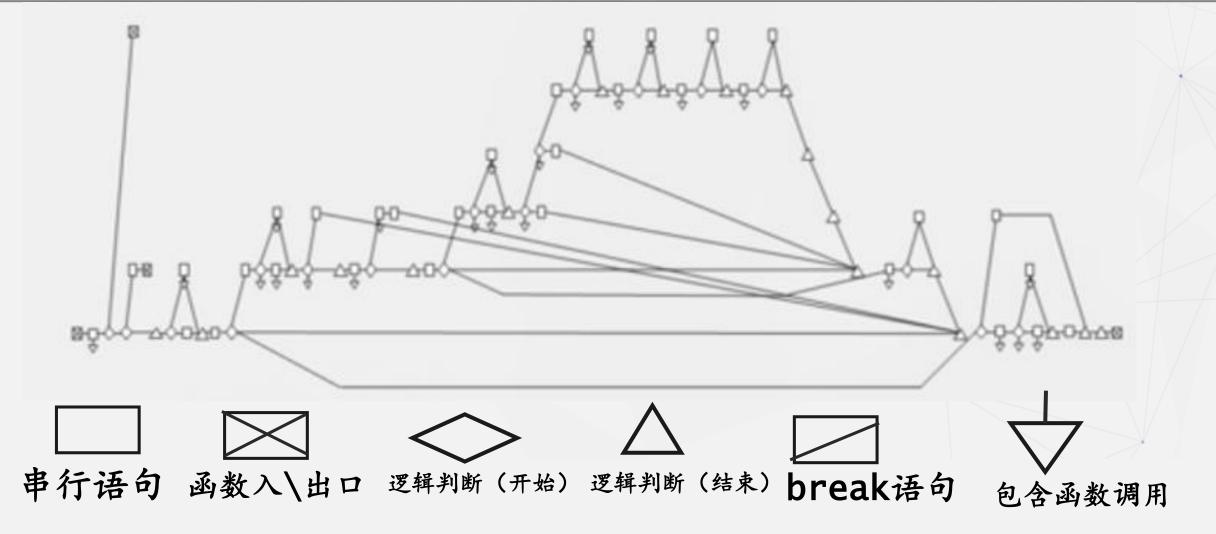


- ·质量因素(Factors):对应ISO9126质量模型的质量特性
- ·质量标准(Criteria):对应ISO9126质量模型的子特性
- 质量度量元 (Metrics): 规范软件的行为属性。每个质量标准由多个质量度量元组成

- 质量度量元定义和计算
 - ·通过对每个度量元规定上、下限,并将其转化为数字,当被测代码关于该度量元的实际取值落在规定的上下限范围内时,就认为被测代码关于该项度量元是合格的,并赋值为"1",否则赋值为"0";
 - ·例如V(G),上下限分别为10和1,若某段代码V(G)=11,则被测代码关于环复杂度不合格
 - 通常质量度量元由各单位自行规定

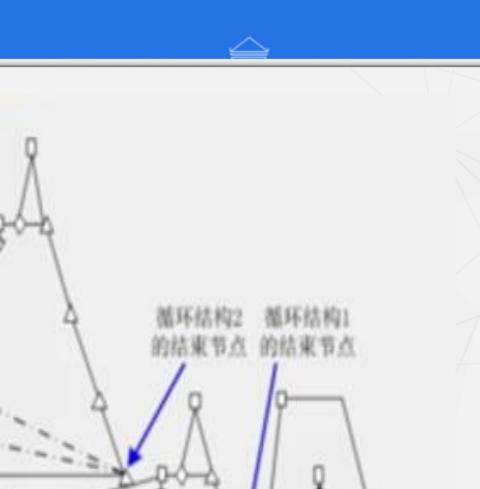
- 质量标准的计算
 - 质量标准由若干质量度量元综合进行评价,必须建立度量元与质量标准的计算公式
 - 例如: 软件可分析性质量化标准的度量元为AVGS(Average size of statements)、COMF(Comments frenquency)、STMT(Number of statements)、VG(Cyclomatic numbe)
 - Analyzability = 1*AVGS+1*COMF+1*STMT+1*VG
 - 计算的等级前三个等表示可以接受, , 最后一个等级表示不可以接受
 - 质量因素的计算也同理

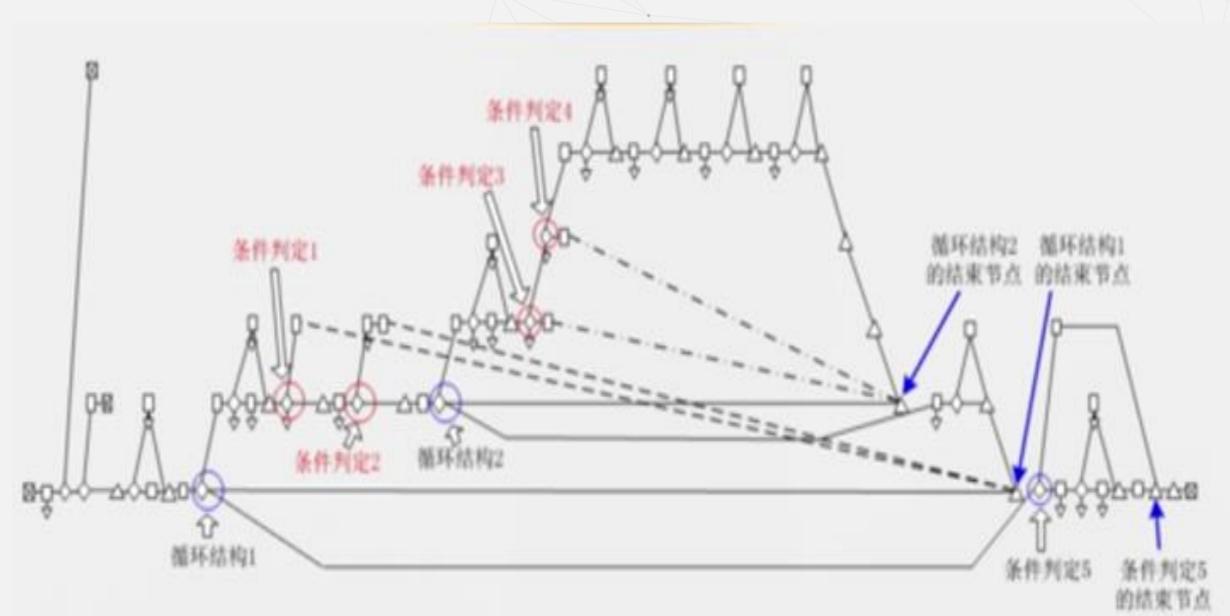
静态结构分析练习



• 分析此程序控制流图, 找找程序设计不合理的地方

静态结构分析练习





静态结构分析练习

- 多出口 (2出口)
- 环复杂度高(14个条件判定节点, 2个循环结构)
- 存在非结构化设计
 - · 结构化: 从该结构的逻辑判断开始节点出发的所有路径(包含嵌套结构的子路径)均是以该结构的同一行内的逻辑判断结束节点终止
 - 条件判定1和2包含的虚线路径都是从循环结构1的结束节点退出的
 - 条件判定3和4包含点划线的路径则是从循环结构2的结束节点退出的

针对不合理的函数改进策略

- ·多出口往往是因多条return语句造成,尽量避免同一函数使用 多个return语句
- ·尽量不使用强制跳转或强制结束语句,如:goto, break等语句
- 将完成单一功能的语句块改为函数调用的方式,降低单个函数的环复杂度

静态结构分析局限性

- 无论是函数调用图还是控制流图, 都是从图的角度, 脱离代码分析
 - 无法看出函数调用接口的复杂性,如包含多少个参数,参数类型是否复杂等
 - 判定节点的复杂度和循环结构的复杂度
 - 如判定表达式包含多少个简单的判定条件,循环次数如何控制
 - 判定节点间是否存在相互关联等
- 局限性解决办法: 通过代码评审和动态白盒测试来对代码进一步测试

静态白盒测试总结

- 静态白盒测试通过对比标准和规范,检查程序逻辑,直接定位 缺陷,从而加快测试进度,降低测试工作量
- 静态白盒测试还基于缺陷预防的思想,通过检查程序的各种图表定位哪些具有高风险的程序代码,并承担部分代码质量度量的工作
- 注意: 评审并非仅针对源代码,这种静态检查适用于从需求阶段到验收测试阶段的所有阶段工作产品

内容总结

- 静态白盒测试
- 评审流程及各个环节中人员的职责
- 代码结构分析
- 代码质量度量

