如何保证软件设计的质量

软件开发和工程被视为非常年轻的职业；但是，它们得到了广泛应用，并且正以比以往更快的速度增长。在许多国家，软件行业目前通常被视为经济增长的主要支柱之一。软件公司常常面临着提供高质量软件的许多困难挑战，而他们也在竭尽所能地让客户满意。

软件质量不可或缺

随着软件变成日常生活中不可或缺的一部分，对软件的需求也明显增长。相应地，高软件质量目前被视为是 “必须具备的” 而不是 “应该具备的”。让质量保证团队从一开始就参与到项目规划和执行中，这一点至关重要。然而，仍然有一些公司认为软件质量只不过是测试人员在开发生命周期结束时执行的一项任务。

值得注意的是，软件市场充斥着各种选择，有众多免费、开源的软件存在。除此之外，客户和最终用户对他们购买的软件的质量越来越看重。具有糟糕性能或低质量用户体验的应用程序或企业系统将被淘汰，其他产品很容易取代它们的位置。现在，以前所未有的认真态度关注其产品的质量已成为软件公司的使命。

自始至终都要考虑软件质量

对于将软件 QA 浓缩到所有开发任务完成后的测试阶段的方法，它们的问题在于：会给团队带来巨大成本并将整个项目置于高风险之中。在测试阶段，开发人员竭尽全力确保他们的代码具有极少的缺陷。然后测试人员努力揭示软件中每个可能的缺陷，而经理和客户希望他们拥有适合向市场发布的软件。

问题在于：为什么许多软件公司会坚持促使开发团队满足严格的最后期限，完成尽可能多的功能，却毫不关心代码质量有多差，因而忽视了注入代码内的大量缺陷，犯下架构错误，以及忽略文档？

仓促的开发可能会为团队节省片刻的时间，但是，如果有一些重大开发问题没有从一开始就考虑到，最终可能导致需要投入更多的时间。结果是浪费了大量团队资源来修复和重新设计代码，而不是将这些资源投入到更有用的事情上。软件团队人员内心里对整个始末一目了然，但面对着唠叨的客户、严格的销售团队，以及一些自我感觉编写了无缺陷的软件的开发人员，软件团队确实很难将 QA 撇在一边而只顾着完成代码。

软件工程标准和它们的使用

值得一提的是，公司既不需要一定要遵守某种软件开发标准，也不需要有严格的流程。典型的软件开发生命周期 (SLDC) 有着各种不同的标准，比如 IEEE、ISO－12207 或 CMMI。这些标准的目标是确保最终产品符合市场需求，达到最终用户的满意度。

事实上，每天都会许多软件应用程序、移动应用程序，甚至是完整的企业系统销售给各种不同的客户，这些应用程序可能并未使用任何标准来开发。但是，人们还是在购买它们。忽略标准与糟糕的软件质量和对最终产品的更少需求没有直接的关系（只要该软件不是对生命至关重要的软件，比如美国境内需要 FDA 批准和应遵守某种标准的医疗软件）。问题不是遵守标准。真正的问题在于忽略或减弱软件质量的重要性。

本文既不是说要遵守 SDLC 标准，也不是要拥有一个极好的开发和测试流程。首先，重要的是认识到质量是任何软件的一个重要组成部分。公司不一定需要高度专业的 QA 团队和实践，但至少必须接受这种文化本身和拥有相关的实践。

贯穿软件开发生命周期的软件质量实践

这一节提供的实践将对软件质量会产生积极影响，不会给开发团队创造太多负担或麻烦。在开发和测试实践中值得考虑它们。

需求审核

难道您不认为，浪费资源来交付错误的功能确实很可怕？在开始每个新开发阶段之前审核软件需求，这样做能够最大限度地减少缺陷并满足客户的需求。在实现之前审核需求，这样做有助于考虑潜在的变化，克服在项目的整个寿命中可能发生的误解。团队必须与客户一起反复检查所有应实现的业务领域细节。需求审核也可以使用原型和领域模型来完成。当开发团队在开始实际实现之前完成这个小任务时，他们的项目或开发迭代会获得良好的开局。通过确保在实现之前所有利益相关者都达成共识，并且每位团队成员都意见一致，客户和管理人员可确信开发人员将在开发周期结束时交付正确的成果。

代码审核和演练

听起来像很简单，但代码审核是软件开发中最有效的实践之一。它对减少缺陷数量（在错误进入软件之前在窗口中发现它）以及增强代码和软件设计的质量具有直接影响。这消除了在未来的版本中执行重大的代码重构和清理的需求。

依据项目需求和实现细节，团队可能认同简单的编码和设计原则。团队成员应共同遵守这些原则，而且只要开发一项新功能，一个或多个团队成员（除了作者）应审核新代码，并搜索所有编码或设计错误。

这种做法可在许多方面为团队带来帮助，包括提高代码质量和设计，最大限度地减少缺陷，并预防它们。另外，它还使得整个团队能够深入了解彼此的工作，轻松移交工作，并提高团队对不同软件组件和功能的认知。团队协作验证和证明代码的质量和设计的实现方法。它们从同事那里获得直接反馈。这么做可谓一举两得：代码质量增加了，团队的认知和项目责任也增加了。

基于会议的测试

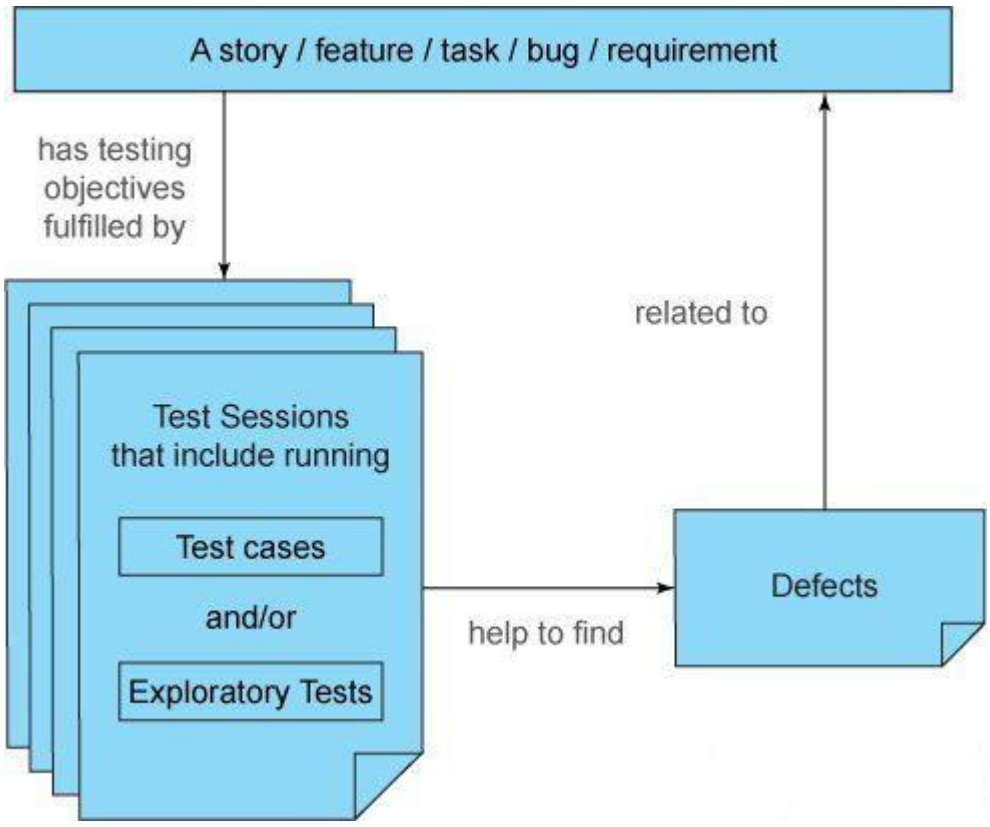
基于会议的测试（James Bach 发明的一种方法）表示将测试负载分解为会议，每个会议有一个任务（一种希望从测试会议获得的明确规定的结果）。每个会议有一个既定的时间范围（从 20 到 40 分钟），测试人员在执行测试会议期间不应中断。

这就像将测试人员放在一个测试房间一段时间，让测试人员专注于查找特定软件特性或功能的缺陷。在会议期间，测试由一组测试案例引导执行，测试人员也可以执行探索性测试。因此，基于会议的测试是正式测试方法与测试创新的一种组合，因为它提供了测试人员房间来进行探索和获得直觉思维，留出了时间和自由空间来发现不常见的缺陷，或者通过折腾软件来进一步了解它。

会议期间，测试人员应将软件的行为记录在案，获取快照，以及写下软件在特定输入和设置下的行为。会议结束时，将与团队领导或技术经理讨论会议脚本。从他们的讨论中，他们找出所认为的正常行为和不正常行为，然后基于讨论创建缺陷报告。

图 1 简短描述了基于会议的测试。对于软件中的任何新更改，计划不同的测试会议，每个会议都有一个指定的目标和任务。测试会议期间，测试人员使用测试案例和/或执行探索性测试。测试会议结束时，会报告在会议期间找到的缺陷。

图 1. 基于会议的测试工作流



基于风险的测试

因为在开发流程中进行了一些更改（主要的或次要的更改），开发团队通常拥有同一个软件的许多常用版本。一种重要的 QA 实践是在每个主要版本之后彻底测试软件。另一方面，在每个版本中都对整个软件运行全面的回归测试既耗时又很难实现。但是，仅测试更改的功能或笨拙地删减测试案例套件是不安全的。一段代码可能解决了一个缺陷，但也可能破坏了代码中的其他内容。

基于风险的测试方法采用了折中方式。它的基本理念是按降序对软件功能和失败模式排序，从最重要或风险最高到值得拥有的功能和简单的风险（一个类似工具是 FMEA：失败模式和影响分析）。如果测试人员在严格的时间限制下测试某个新版本时手头有这个列表，他就可以集中精力确保新引入的更改不会破坏其他任何内容。然后就可以轻松地确保更改不会破坏软件中的任何最重要的功能，因而不会发生任何最严重的风险。

结束语

每家公司都希望在充满竞争的 IT 市场中飞得更高，而且在努力创造人们喜欢的优秀软件。不幸的是，有时来自客户或不耐烦的管理层的压力可能导致团队绕开软件质量实践，最终实现一个低于预期的产品。糟糕的软件质量只有在它发生故障时才会被注意到。

本文中讨论的实践和方法贯穿整个开发生命周期。它们涵盖需求分析、设计和开发，以及测试阶段。而且它们表明缺陷预防比在项目结束时再解决问题要简单得多，在项目结束时，更改带来的技术人情债和成本将会很高。

本文突出了软件质量角色和重要性，以确证忽略软件质量可能严重影响公司实现其业务目标。另外，本文还为读者介绍了一些更简单、更高效的实践，它们不但为团队节省了时间、金钱和精力，还提升了产品的质量。