

# 2015 软件测试技术 报告

## 目录

行动纲要.....	2
<b>第1节</b> 汇总数据：看看行业的主要趋势.....	2
<b>有待深入探讨的概念</b> .....	3
物联网市场的发展及其影响.....	3
持续集成和持续测试：为何如此重要.....	3
越来越多关于技术债的问题 - 为什么大家都要解决这个问题.....	5
对不同角色进行深入分析.....	9
<b>第2节</b> 管理层会考虑哪些问题？.....	9
<b>第3节</b> 开发工程师所考虑的是什么？.....	13
<b>第4节</b> QA 团队和测试工程师的想法.....	15
总结.....	17
<b>第5节</b> 所有受访对象的总体观点.....	17

## 行动纲要

因为有些要求较高的客户不断要求提高产品的质量，并希望产品的规程越来越严格，再加上严峻的成本压力和紧迫的市场投放时间目标，软件测试行业正处于惊人的转型期。“Vector Software 年度软件测试技术报告”是为了了解软件测试行业的领军者对以下几个问题的看法：

- 他们现在正在经历什么样的潮流？
- 用户想更多地了解哪方面的信息？
- 最近，市场计划完成哪些项目？
- 他们对什么功能比较感兴趣？

简单来说，我们要把现实和激动人心的宣传区分开，了解嵌入式软件测试行业现在所关心的是什么，以后又会关注什么。本报告中汇总的是主要的调查结果。

## 第 1 节 汇总数据：看看行业的主要趋势

### 1.1 人们最熟悉的软件测试行业术语是什么？

“2015 年软件测试技术报告”邀请参与者用 1-5 之间的数字评定一下他们对行业术语的熟悉程度。1 表示不熟悉，5 表示非常熟悉。受访对象拿到的完整的术语列表如下：

- |   |   |
|---|---|
| ■ API 测试<br>(API Testing)                                     | ■ 嵌入式软件测试<br>(Embedded Software Test)     |
| ■ 基于变更的测试<br>(Change-Based Testing)                           | ■ 模糊测试<br>(Fuzz Testing)                  |
| ■ 代码更改影响分析<br>(Change Impact Analysis)                        | ■ 工业物联网<br>(Industrial IoT)               |
| ■ 持续交付<br>(Continuous Delivery)                               | ■ 物联网<br>(IoT)                            |
| ■ 持续集成<br>(Continuous Integration)                            | ■ 大规模并行测试<br>(Massively Parallel Testing) |
| ■ 用 Bamboo 进行持续集成测试<br>(Continuous Integration with Bamboo)   | ■ 安全测试<br>(Security Testing)              |
| ■ 用 Jenkins 进行持续集成测试<br>(Continuous Integration with Jenkins) | ■ 左移测试<br>(Shift Left Testing)            |
| ■ 持续测试<br>(Continuous Testing)                                | ■ 技术债<br>(Technical Debt)                 |
| ■ DevOps  |   |

下表是得分的加权平均数大于 3 的术语，也就是受访对象非常熟悉的一些术语。

术语	加权平均数
嵌入式软件测试	4.04
持续集成	3.28
持续测试	3.22
API 测试	3.16
代码变更影响分析	3.15
物联网 (IoT)	3.15

## 有待深入探讨的概念

### 物联网市场的发展及其影响

物联网 (IoT) 对软件测试行业有着深远的影响。实际上,“物联网”是唯一一个不那么确切的测试术语,但是受访对象表示对该术语很熟悉。通过 IoT,能够利用可互操作的通信技术将物理世界与虚拟世界互联起来,也就是说,每个电子设备都有网络连通性,电子设备的每个生产商都置身于软件业务中。

关于物联网的惊人增长速度已经有了很多报告。有个甘特图报告预测到“2015 年有 49 亿件互联的物体,比 2014 年增长了 30%,预计 2020 年数量会达到 250 亿。”这么多互联物体所涉及的财务分别是 2630 亿美元和 8.9 万亿美元。

从现在再往后看 10 年,麦肯锡全球研究所(McKinsey Global Institute)估计到 2025 年, IoT 的总体经济影响将达到 11 万亿美元。未来物联网系统将有如此大量的互联设备,那么为了保证物联网系统的质量,我们将面临怎样的挑战呢?只要这些预测中有一个成真了,就必须得提高质量——质量是取胜的保障。

### 持续集成和持续测试:为何如此重要

除了宏观的术语“嵌入式软件测试”,受访对象最熟悉的术语就是“持续集成”(CI)和“持续测试”。这非常符合整个行业的重心转向提高质量这个趋势。如果对独立的代码进行修改,并将修改的代码添加到整个代码库中时,进行持续集成之前就需要立刻对该代码进行测试并报告问题。持续集成的主要目的就是通过快速提供反馈来防止出现集成问题,以便在新代码引进了 bug 的时候,能够尽快发现并修复。

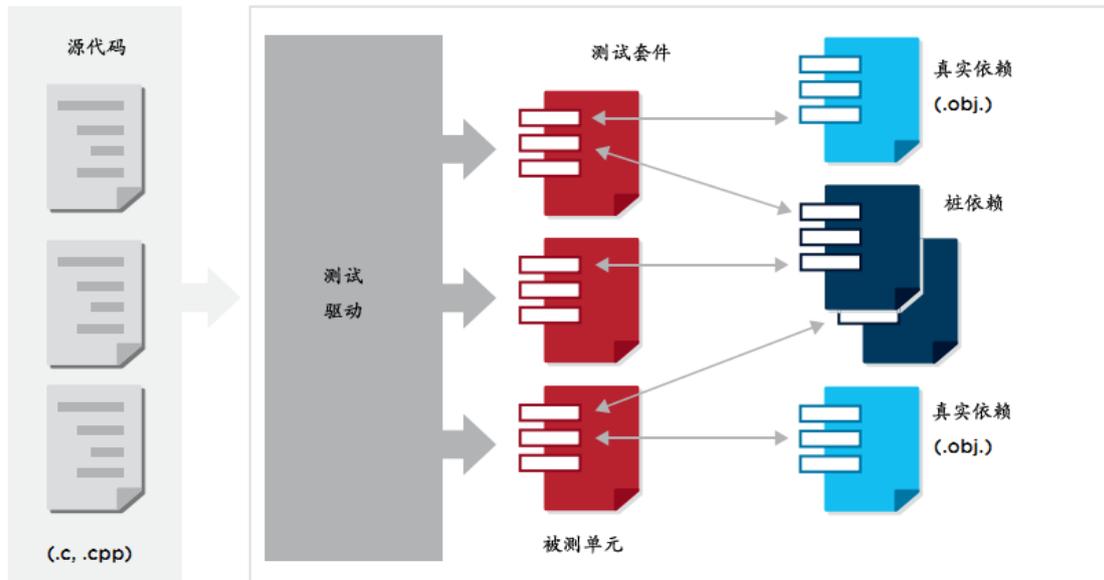


图 1: 如果对独立的代码进行修改, 并将修改的代码添加到整个代码库中时, 进行持续集成之前就需要立刻对该代码进行测试并报告问题。通过快速得到反馈能够防止出现集成问题, 以便在新代码引进了 bug 的时候, 能够尽快发现并修复。

持续集成/持续测试环境的理想状态是: 每次对软件代码做了修改之后, 在进行集成之前, 把该修改可能影响到的测试都执行一遍。但是, 即使是再完善的自动化测试, 要执行完 “所有的测试” 也需要几小时甚至几天的时间。鉴于这种情况, 对于大多数研发测试小组来说, 比较合理的做法是定期执行一次全面测试, 而不是不断执行。

“基于变更的测试” 正好能解决这个问题。它能自动筛选出代码修改会影响到的测试子集, 针对该修改只需要执行这些测试, 从而保证测试量与源代码的修改量相称。对于较小或中等规模的修改, 测试周期只需要几分钟, 而不需要花费几天的时间。

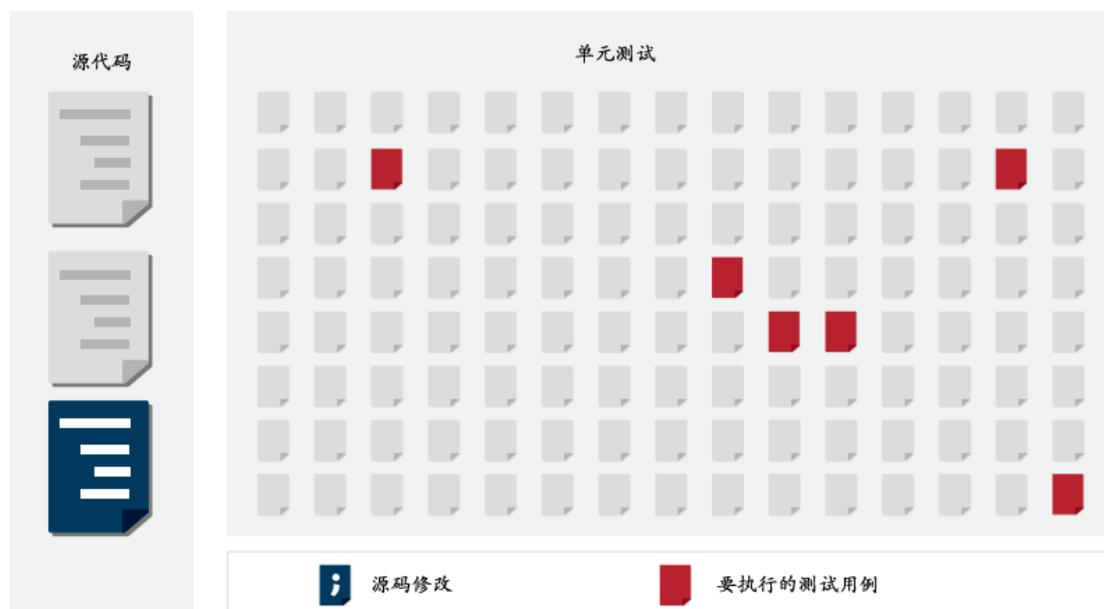


图 2: 基于变更的测试能自动筛选出该修改会影响到的测试子集, 针对该修改只需要执行这些测试, 从而保证测试量与源代码的修改量相称。

## 越来越多关于技术债的问题 - 为什么大家都要解决这个问题

在本次调查中，特别向受访对象询问了他们是否了解“技术债”这个术语。出乎意料的是，受访对象中，45%的人说他们“非常了解”这个术语，该术语的熟悉度评分的加权平均值为2.33/5。

技术债是一种比喻说法，实际是指系统架构、系统设计、系统开发过程中引进的潜在缺陷。该术语是 Ward Cunningham 在 1992 年发明的，指的是组织机构为了达到短期目标，在设计 and 测试方面走捷径而积累下来的不利条件。和金融债务一样，技术债本质上也是一种贷款形式，会产生利息。如果“贷款”未能付清，最终会导致软件无法维护。

技术债越高，软件产品的市场创新和竞争能力就越差——因为随着技术债的增加，开发工程师将花费大部分时间来修复 bug，维护容易出错的代码，而没有很多时间来构建新功能。软件工程师和生产商必须设计新的开发程序，来确保不会增加新的技术债，并且已经偿还了已有的技术债。

那些以前未经充分测试又背负了数十亿美元的技术债的应用程序该怎么办？没有人会回去手动测试这些应用程序。减少旧有应用程序的技术债的关键在于：随着时间的推移不断重构（改善）组件。但是，只有在能够轻松执行测试环境来验证组件可否正常运行的情况下，才可能成功重构（改善）组件。

自动生成测试用例是提高旧有应用程序的测试完整性的关键所在（自动测试旧有代码）。软件经过多年的改进和扩展会产生过时的、不用的或重复的代码，自动化测试能够帮助轻松发现这些代码。既可以通过界面进行测试也可以根据应用程序的逻辑进行测试。自动化界面测试是根据一些参数的类型边界值的组合生成的（如：最小和最大整数）。自动化逻辑测试能够让应用程序中的每个决策点都呈现正确或错误的结果。

自动生成测试用例能够快速提供一套测试“基准”，来获取应用程序中现有的功能，确保重构代码时不会使程序中的功能退化。一旦建立了测试基准，就可以通过“基于变更的测试”来看代码变更对整个系统的完整性产生了怎样的影响。自动化流程能够在“一行代码”发生了改变之后，确定需要执行的测试子集，而不需要花好多天时间将整套测试都执行一遍，这样不仅节省了宝贵的时间和资源，还能让开发工程师可以不断进行测试。

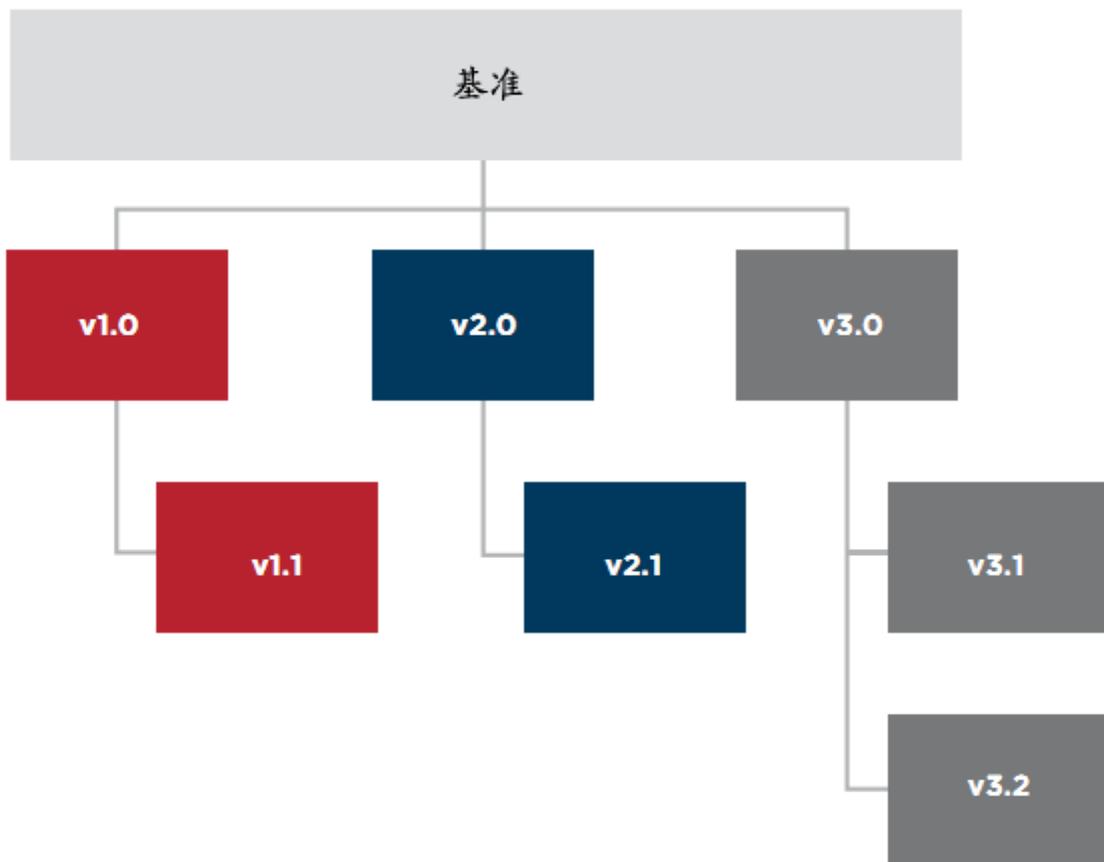


图 3: 可以利用测试“基准”来获取应用程序中现有的功能, 确保重构代码时不会使程序中的功能退化。一旦建立了测试基准, 就可以通过“基于变更的测试”来看代码变更对整个系统的完整性产生了怎样的影响。

已经讲了理解技术债和提高测试完整性的一般做法, 最后要提的是一款报告工具, 该工具具备以下功能的:

- 展现代码库的整体发布就绪情况
- 清楚标明测试未覆盖到的部分
- 显示一段时间内的趋势
- 让管理员能够预知需要哪些资源来降低技术债

## 1.2 软件测试市场比较希望更多地了解哪些信息?

我们询问了受访对象, 他们比较希望了解关于软件测试哪些方面的信息。我们给出的完整术语列表和 1.1 节中的一样。5 个比例最高的回答总结在下表中。这些结果还是与物联网的快速发展对软件测试行业产生的深远影响一致, 也与软件质量的代码变更产生的影响, 以及持续集成和持续测试所需的结果一致。

术语	比例
嵌入式软件测试	49%
持续测试	38%
持续集成	32%
代码变更影响分析	32%
物联网 (IoT)	29%

### 1.3 软件测试行业计划在未来项目中解决哪类问题？

下表总结了受访对象认同的未来项目最应该解决的问题以及他们最熟悉的问题（结果包含 1/3 或更多的受访对象的积极回应）。

问题	比例
Bug 和缺陷的追踪	78%
自动构建单元测试和集成测试的测试框架	55%
向开发流程引进合规标准	51%
在目标上执行基于需求的测试	50%
性能和实时动态测试	45%
代码质量和测试分析平台	40%

大部分受访对象（约 80%）指出他们计划在下一个项目中解决 bug 和缺陷的追踪问题。其实这个结果一点也不出人意料，因为很多研究显示软件成本花费最高的部分往往不是原始开发、测试及生产，而是软件发布之后的维护。

举个例子，剑桥大学的研究发现全球每年解决软件问题的成本已经增加到了\$3120 亿。该研究还发现，开发工程师平均要耗费 50% 的编程时间来发现和修复 bug。很明显，公司在这方面花费了大量的时间和金钱，但都只是治标不治本，无法真正解决根本问题。

企业要提高软件的质量就必须重新定位测试在开发周期中的作用，尤其测试对生命周期的总成本的影响，而不是单单考虑何时发布 1.0 版本。通常，发布周期就像一场零和博弈，在投资方面投资的时间越多，发布的日期也会拖延越多。但是这种想法其实忽略了这些测试对产品生命周期的价值。

另外一个有趣的发现是：超过一半的受访对象表示他们计划在下一个项目中自动为单元和集成测试构建测试框架。如果单元测试和集成测试是测试过程中最关键的两方面，那么这样做就一定会有助于找到合适的解决方案。

在单元测试过程中，开发工程师要单独对各个单元进行评估，以保证各功能可以正常运行，并达到了操作要求。单独测试完各个单元之后，开发工程师就会开始集成测试。集成测试是对构成功能过程的一小部分单元集合进行测试，确保单元之间能够很好地协作。构建自动化测试框架来自动执行单元和集成测试，就可以利用高级功能来简化复杂代码的测试工作。如果像这样软件经过了更加全面的测试，那么可能引进的 bug 就会大大减少。

#### 1.4 人们最感兴趣的实际功能是什么？

受访对象最感兴趣的功能有以下几个：

功能	比例
回归和系统测试	65%
代码覆盖率	65%
C/C++/Ada 的自动化动态/单元测试	63%
自动生成测试用例	57%
静态分析	56%
编码标准，认证和合规标准（包括 MISRA）	53%
目标测试 —— 单芯和多芯	48%
MCDC，语句和分支测试	43%
旧有代码的自动化测试	42%
敏捷/TDD API 测试	36%
代码变更影响分析和校正	34%
敏捷交付中基于变更的测试	33%

我们的受访对象最感兴趣的两个功能是：回归和系统测试、代码覆盖率。在开发复杂的嵌入式软件时，很多开发团队选择进行自动化回归测试。在软件开发过程中，经常会有新 bug 不断出现。这是因为程序员为了修复已发现的问题，会对代码进行修改，但是这样可能又会使一些之前能正常运行的功能出现新问题。

所以，需要对开发过程中所做的每个修改进行额外测试。自动化回归测试工具就是个非常强大的工具，适用于这种情况。回归测试，也叫验证测试，是在代码发生了变化之后对软件进行分析的过程。回归测试除了可以保证已发现的问题已被修复，还能对软件中的已有组件进行重复测试，确保没有引进新 bug。

自动化回归测试能够不断快速地重复执行测试，而开发工程师则可以继续进行项目开发。因为测试是自动进行的，所以当代码发生改变时，它可以对相同的代码重复测试很多遍。对于复杂的嵌入式系统而言，比如：用于航空、军工、医疗、轨道交通、汽车电子和工业控制等行业的软件，自动化回归测试是不可或缺的开发工具。

另外一个大家关注度较高的功能是代码覆盖率测试。人们已经知道在开发过程发生任何改变时，代码覆盖率测试带来的投资回报率是最高的。利用现代化的工具可以很轻松地将代码覆盖率测试当作构建程序的一部分来执行，无需改变测试过程就能够进行覆盖率评估。

但是，人们对于代码覆盖率与软件质量之间的关系有点误解。软件测试的目的不是一定要代码覆盖率达到 100%，而是要保证测试的完整性。也就是说，代码覆盖率分析和需求测试是相辅相成的。判断需求测试的完整性的唯一途径就是评估代码的覆盖率结果。

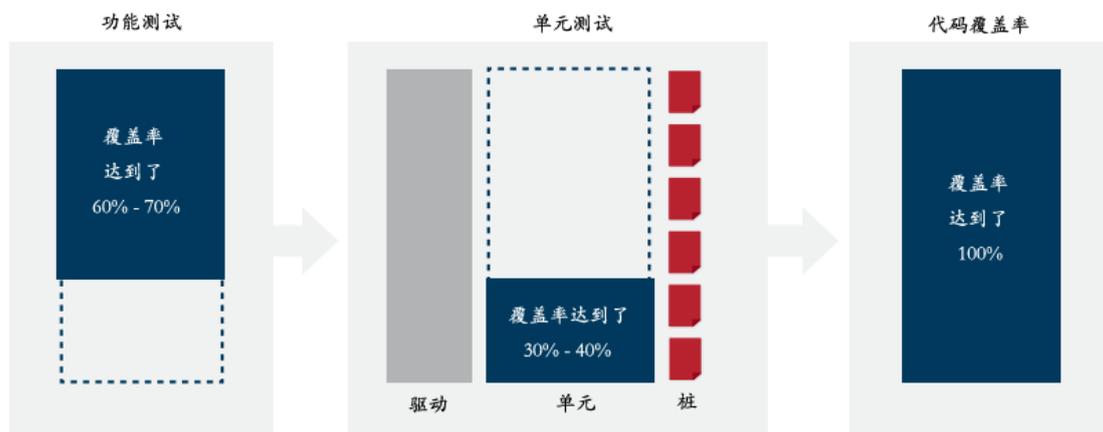


图 4: 人们已经知道在开发过程发生任何改变时, 代码覆盖率测试带来的投资回报率是最高的。完成测试的覆盖率结果应该是 100%。

未被覆盖到的代码可能意味着测试执行得不够充分, 没能完全达到要求, 但是这些没有覆盖到的代码也可能是一些“另加的”代码(如: 需求中说有 100 条信息要测试, 但是开发工程师其实增加了 10 条新信息, 只是一直没有更新需求。)

大家比较感兴趣的另外一个功能(42%的受访对象认同)是旧有代码的自动化测试功能, 这表明人们越来越意识到对旧有代码进行自动化测试的必要性。因为现在的软件越来越复杂了, 大多数新开发都是在旧有代码库的基础上进行的。我们在本报告的 1.1 节中提到了技术债带来的这个问题。旧有代码面临的严峻问题都是由技术债带来的。正如上面提到的一样, 自动生成测试用例是提高旧有应用程序的测试完整性的关键(与旧有代码的自动化测试同义)。

## 对不同角色进行深入分析

### 第 2 节 管理层会考虑哪些问题?

#### 2.1 管理层最熟悉软件测试行业的哪些术语?

下表是受访对象中的经理或项目经理最熟悉的术语, 所列出的术语是熟悉度评分的平均值高于 3 的术语。

术语	平均得分
嵌入式软件测试	4.08
代码变更影响分析	3.43
物联网 (IoT)	3.40
持续集成	3.37
持续测试	3.29
API 测试	3.23
安全测试	3.08
基于变更的测试	3.00

只有处于管理层的受访对象对安全测试这个概念比较熟悉,这说明安全测试是管理层比较重视的领域。

对于很多商界领导人而言,因为物联网的发展,安全问题被提到了一个非常重要的位置。物联网扩展了在新平台和服务领域的职责,这就重新定义了安全要求,而且在保障物联网设备的安全方面的要求也会越来越复杂。

然而,要在像物联网竞争这么激烈的市场中占据优势,使产品率先进入市场将是一个非常重要的竞争优势,所以开发工程师就会面临巨大的压力,要尽快发布产品。在如今的软件开发领域,很多开发组织都面临这样的问题。但是事实一次又一次地证明,这种想法常常会让人们牺牲质量,一味地追求速度。而物联网中的一些产品(如智能汽车、医疗、家庭安全系统)如果只求速度,不求质量,那就更为危险了——所以,这很自然地就成了很多高级主管最关心的问题。如果这类系统出现了故障,真的会危及生命安全。

在软件开发阶段就要解决安全问题,因为如果在这些高级系统发布了之后再重新对其进行设计,成本会非常高。所以管理层仍然会将严格的软件质量评估和测试作为重中之重。

## 2.2 软件测试行业的管理层希望更多地了解哪方面的信息?

我们向处于管理层的受访对象询问了他们比较希望更多地了解软件测试行业中哪些方面的信息。下表中列出了他们最想了解的话题。

话题	比例
嵌入式软件测试	44%
持续测试	41%
持续集成	38%
代码变更影响分析	36%
基于变更的测试	35%

除了广义的嵌入式软件测试,所列出来的这些话题都聚焦在持续测试/集成和代码变更方面。这更加突出了 1.1 节中提出的行业将重心转向提高质量这个观点。

### 2.3 管理层希望在未来的项目中解决哪类问题？

问题	比例
Bug 和缺陷追踪	80%
在目标上执行基于需求的测试	60%
自动构建单元测试和集成测试的测试框架	57%
向开发过程引进合规标准	55%
性能和实时动态测试	50%
代码质量和测试分析平台	44%
鉴定项目风险和技术债	36%
测试基于模型的设计生成的代码 (Simulink, SCADE, Rhapsody)	36%

处于管理层的受访对象中计划在下一个项目中解决 bug 和缺陷追踪的比例也很高,有些则对 bug 和缺陷追踪这一解决方案非常了解 (80%是这种情况)。

受访对象中也只有管理层想更多地了解关于技术债方面的信息,管理层中有 1/3 的人表示,他们可能计划在下一个项目中鉴定项目风险和技术债。考虑到 1.1 节中强调的技术债带来的严重的财务问题,管理层会率先关注到这方面的问题就不足为怪了。

### 2.4 管理层最感兴趣的实际功能是什么？

当处于管理层的受访对象被问及他们最感兴趣的实际功能是什么时,他们回答最多的是:

功能	比例
回归和系统测试	71%
代码覆盖率	64%
自动生成测试用例	61%
C/C++/Ada 的自动化动态/单元测试	54%
静态分析	54%
编码标准, 认证和合规标准 (包括 MISRA)	50%
目标测试——单芯和多芯	47%
代码变更影响分析和校正	45%
旧有代码的自动化测试	44%
MCDC, 语句和分支测试	44%
敏捷/TDD API 测试	41%
敏捷交付中基于变更的测试	40%

管理层最感兴趣的功能是: 回归和系统测试, 代码覆盖率分析, 自动生成测试用例。下面我们详细探讨一下为什么这些功能如此重要:

回归和系统测试: 回归测试的目标是: 能够在代码做了变更之后发现系统中现有领域出现的新问题。这无疑是在开发出高质量的软件产品的关键步骤。

但是，现在软件应用程序越来越大，生命周期越来越长，开发团队分布的地域也越来越广。很多开发团队都在努力寻找进行自动化测试的方法，否则测试所要花费的时间就太长了。测试的费用也占据了软件开发预算中的很大一部分，所以对于管理层来说能将测试用例方面的投资利用在整个开发团队以及整个应用程序生命周期中是至关重要的。

管理层都在找一些测试资源，希望能够方便软件团队执行回归测试，让团队的每个成员无论何时都可以对应用程序的各个版本执行测试。这样不仅能够缩短发布周期，还能使客户接触到的 bug 大大减少。很明显，这对于管理层来说也是重中之重。

代码覆盖率：代码覆盖率分析能够发现测试执行过程中哪些代码已经执行了，从而有效地评估测试的效率。利用代码覆盖率可以立刻对部分应用程序或整个应用程序进行分析。这个功能非常重要，因为它能够：

- 帮助了解测试的完整性
- 提供报告来说明已发布的应用程序是否经过了充分的测试
- 促使开发工程师在建立了测试之后重新对代码进行评估
- 整合从单元测试、回归测试、系统测试中获得的代码覆盖率

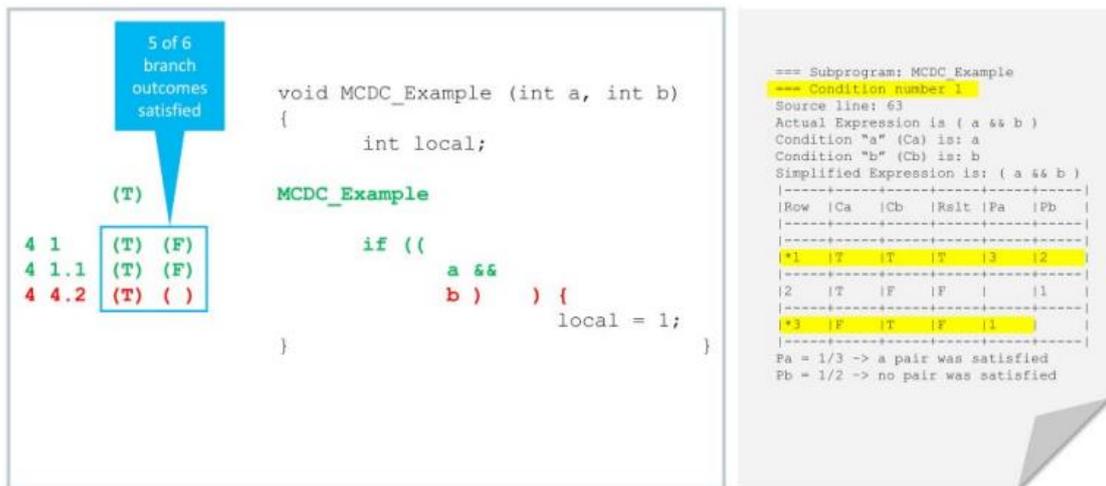


图 5：这是一个修正条件/判定覆盖（MC/DC）的例子，是代码覆盖率分析在高安全软件测试方面的一种应用形式。使用这种特殊形式的代码覆盖率是为了保证运用于航空航天系统中的嵌入式软件的安全性和可靠性。

自动生成测试用例：对于运用于诸如航天军工、医疗、工业、汽车电子、轨道交通等领域的高安全应用程序而言，对需求进行测试从而确定嵌入式软件的功能是至关重要的。自动为这些特定的需求生成测试用例，将使该过程更加紧密、高效、准确。本报告也已经讨论了很多关于技术债带来的严重的影响，以及需要可靠的旧有代码测试解决方案来帮助解决这个问题。2015 年 5 月份的 Gartner 研究中指出，“如果企业没有可重复使用的测试用例，就会对企业展示功能对等的的能力产生客观、严重的影响。”

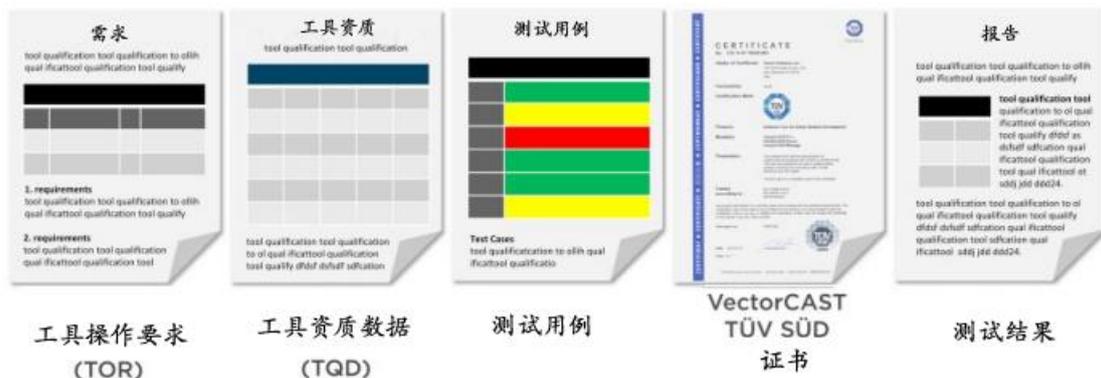


图 6: DO 178B 的测试框架, 机载系统和设备认证的软件考虑。该标准是特定机载系统中的高安全软件的指导方针。

如果很多部署好的/现场应用程序都没有测试用例可执行, 那问题就会更加严重了。正如之前所讨论的, 测试旧有代码时, 先创建一个基准是非常关键的, 因为测试时必须要有标准可对照。自动为旧有代码生成测试用例就使情况大不一样了。实际上, 上面所提到的 Gartner 报告建议企业“使用可解读代码的工具或服务来记录旧有系统中现有的业务规则, 标注出完成项目所需要的阶段。”

### 第 3 节 开发工程师所考虑的是什么?

#### 3.1 开发工程师最熟悉的软件测试行业术语是什么?

下表列出了受访对象中的开发工程师最熟悉的术语, 这些术语的熟悉度评分的平均值高于 3。

术语	加权平均值
嵌入式软件测试	4.02
物联网 (IoT)	3.27
持续集成	3.23
API 测试	3.21
持续测试	3.11

#### 3.2 软件开发工程师想更多地了解哪些信息?

下表是排名前 6 的结果:

话题	比例
嵌入式软件测试	58%
持续测试	36%
物联网 (IoT)	35%
用 Jenkins 进行持续集成	31%
持续集成	29% (TIE)
代码变更影响分析	29% (TIE)

### 3.3 开发工程师计划在未来的项目中解决哪类问题?

问题	比例
Bug 和缺陷追踪	77%
自动为单元测试和集成测试构建测试框架	54%
向开发过程引进合规标准	49%
在目标上执行基于需求的测试	43%
性能和实时动态测试	40%
代码质量和测试分析平台	36%
多芯硬件上的目标测试	34%
测试和管理单一代码库中的多条产品线	34%

开发工程师的期望是什么? 受访对象中超过 1/3 的开发工程师说他们计划在未来的项目中解决质量和测试分析平台的问题/他们对那些解决方案最熟悉, 而其它受访对象对此并不是特别熟悉。

将质量和测试分析平台结合起来意味着使用的不再是那些只能发现潜在问题的软件分析工具。调查结果显示, 越来越多的开发工程师都在寻找一些工具, 希望有工具既能发现软件问题, 又能提供可有效运用于提高软件质量的情报。这也就意味着工具要将分析功能和解决严重问题的功能结合起来, 然后再执行测试来确保问题不会重现。

针对这个问题, 受访对象中超过 1/3 的开发工程师也提到了“测试和管理单个代码库中的多条产品线”, 当然并不是受访对象中的所有人都了解这一点。该想法与物联网不断发展的趋势紧密相关, 比如: 智能汽车的多条生产线可能 95%使用的是相同代码库中的代码。显然, 对于开发工程师而言, 能够在多个配置下自动执行相同的测试用例是极为有利的。

### 3.4 软件开发工程师最感兴趣的实际功能有哪些?

下表中所列的是开发工程师最感兴趣的实际功能:

功能	比例
C/C++/Ada 的自动化动态/单元测试	77%
代码覆盖率	71%
回归和系统测试	61%
静态分析	60%
编码标准, 认证和合规标准 (包括 MISRA)	59%
自动生成测试用例	51%
目标测试 —— 单芯和多芯	43%
MCDC, 语句和分支测试	43%
旧有代码的自动化测试	40%

因为自动化测试工具和代码覆盖率分析工具能简化测试和提高测试的完整性, 所以开发工程师对自动化测试和代码覆盖率测试功能最感兴趣就不足为奇了。有趣的是, 开发工程师也对旧有代码的自动化测试功能感兴趣 —— 这意味着旧有应用程序和技术债产生的问题正引发越来越多的关注/失望。

## 第 4 节 QA 团队和测试工程师的想法

### 4.1 测试团队对哪些软件测试行业术语最熟悉?

下表是 QA 团队或测试工程师认同的他们最熟悉的术语, 这些术语的熟悉度评分的加权平均值大于等于 3:

术语	加权平均值
嵌入式软件测试	3.97
持续测试	3.24
持续集成	3.20
基于变更的分析	3.10
持续交付	3.00

这部分受访对象说他们最熟悉的术语是“持续交付”。采用持续交付方法时要对软件质量进行监控和分析, 同时还要监控并分析软件是否符合一些重要标准的要求, 比如: 行业标准 (ISO 26262, DO-178B/C /DO-278, IEC 61508, and IEC 62304) 或企业的内部质量流程和标准。所以了解该概念对 QA 和测试工程师来说无疑是非常重要的。实际上, 除了广义的嵌入式软件测试这个概念, 大家了解的所有其它概念都聚焦于持续测试/集成和变更。

有趣的是, QA 团队和测试工程师是唯一一组没有提到对“物联网”这个术语比较熟悉的人群。

#### 4.2 测试工程师希望能够更多地了解哪些信息?

下表总结出了 5 个他们最希望了解的信息。

话题	比例
嵌入式软件测试	44%
大规模并行测试	34%
持续测试	33%
安全测试	31%
利用 Jenkins 进行持续集成	28%

#### 4.3 测试团队计划在未来的项目中解决哪类问题?

问题	比例
Bug 和缺陷追踪	77%
自动构建单元测试和集成测试的测试框架	53%
向开发过程引进合规标准	44%
性能和实时动态测试	44%
在目标上执行基于需求的测试	42%
代码质量和测试分析平台	41%
确保交付和评估质量	36%

#### 4.4 测试工程师对哪些实际功能最感兴趣?

下表中显示的是受访对象中的 QA 和测试工程师最感兴趣的实际功能:

功能	比例
回归和系统测试	61%
自动生成测试用例	61%
C/C++/Ada 的自动化动态/单元测试	59%
代码覆盖率	56%
静态分析	56%
编码标准, 认证和合规标准 (包括 MISRA)	51%
目标测试 —— 单芯和多芯	44%
MCDC, 语句和分支测试	43%
旧有代码的自动化测试	39%

这组受访对象最感兴趣的 3 个功能包括: 回归和系统测试, 自动生成测试用例, C/C++/Ada 的自动化动态/单元测试。这与软件测试行业的管理层和开发团队提出的优先级是非常一致的, 这也表明了软件测试行业中不同的人群都非常倾向于使用自动化测试解决方案来提高软件质量。

# 总结

## 第 5 节 所有受访对象的总体观点

受访对象根据要求用 1-5 之间的数字来评定对软件测试行业术语的熟悉程度，通过结果发现在行业中处于不同职位的所有受访对象都认为嵌入式软件测试是他们最熟悉的术语，该术语在所有受访人群中得到的加权平均评分都接近或大于 4。

同时也询问了受访对象他们在下个项目中打算解决哪类问题/他们最熟悉的解决方案是什么。所有的受访对象人群：

- 一致把 bug 和问题追踪这一项排在了第一位
- 大多把自动为单元测试和集成测试构建测试框架排在了第二位(管理层把这一项排在了第三位)
- 都选择了性能和实时动态测试
- 都选择了在目标上执行基于需求的测试
- 都选择了向项目过程引进合规标准
- 都选择了代码质量和测试分析平台

被问及他们最感兴趣的实际功能时，所有的受访对象都一致认为是：

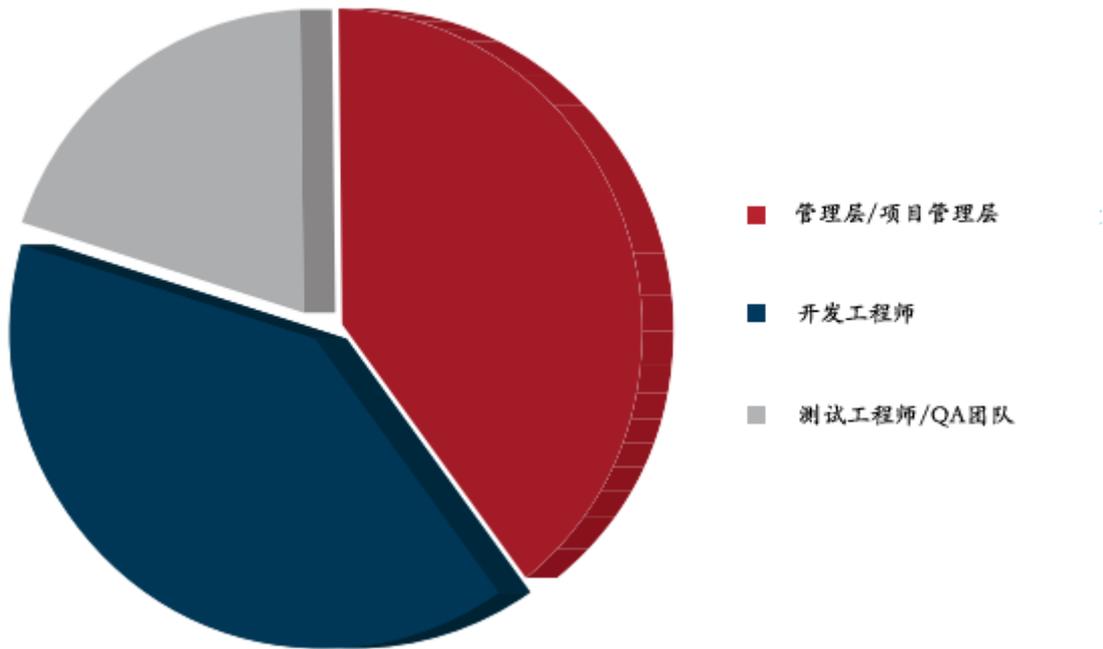
- 静态分析
- 代码覆盖率
- 自动化的 C/C++, Ada 动态/单元测试
- 自动生成测试用例
- 旧有代码的自动化测试
- 回归和系统测试
- 编码标准，认证和合规标准（包括 MISRA）
- MCDC，语句和分支测试
- 目标测试 —— 单芯和多芯

## 调查人口统计

受访对象总人数：285

工作职位：

- 管理层/项目管理层
- 开发工程师
- 测试工程师/QA 团队



2015年10月汇编了2015年Vector Software测试技术报告中从受访对象给出的数据。

### 创提信息科技（上海）有限公司 - Trinity Technologies

专注于嵌入式软件研发质量和自动化测试的方案和咨询服务，提供覆盖软件测试整个流程的完整的解决方案，包括从研发前期的代码级测试到后期的系统级测试，从静态分析到动态测试，从编码检查，单元测试、集成测试到性能测试和测试覆盖率分析等。

公司通过专业的自动化工具（如 DT10, VectorCAST, QAC/C++, SQUORE 等）和服务满足不同客户对软件质量和测试的需求，持续协助客户改进软件研发质量和效率。客户主要集中在高安全和高可靠性领域，如国防和航空航天、轨道交通、汽车电子、医疗器械、工业控制、通讯和电力电子等行业。公司提供的领先的解决方案不仅为数以百计的客户提高产品质量，还协助客户遵循高安全和高可靠性行业的合规性要求，如 DO-178B/C, IEC61508, EN50128, ISO26262, IEC62304 和 MISRA 等行业标准，并获得相关机构认可和认证。

版权声明：本文档版权归创提信息科技（上海）有限公司所有，并保留一切权利。