
目录

测试移动应用程序的易用性.....	01
网银浏览器兼容性自动化测试实施项目总结.....	08
【搜狗测试】使用 LR 进行手机浏览器 get 请求压力测试的难点和总结.....	38
【搜狗测试】Python 判断 Unicode 字符串类型.....	47
“紫牛”方法论在测试团队中的应用.....	52
测试女巫之石头变宝石篇之四.....	55
如何做好软件验收测试工作.....	71
我了解的的自动化.....	77
敏捷测试的测试设计篇.....	80
盘点那些必不可少的“用例集”	85
心所向，驰以恒.....	88

测试移动应用程序的易用性

◆译者：于芳

概述：移动端易用性在增强终端用户程序的认可度上大有帮助。但是易用性是从用户开始的，而用户在知识、兴趣和目标等方面又各有不同。这篇文章讨论了一些对用户而言有关系的核心理易用性特征，以及测试工程师是怎样理解并实现它们的。

自从 iPhone 的问世以来，智能手机应用程序处于蓬勃发展的状态。在 2015 年 10 月，iTunes APP 商店有近两百万可供下载的 APP。如果我们把这个数据跟安卓和 windows Phone 的提交者加在一起的话，这个数字更让人脑袋打怵。

当一小部分 APP 风靡被下载成千上万次的时候，很多 APP 还没有产生哪怕一部分用户。应用程序【这里指移动端的应用程序】可能会遭遇冷遇因为其内容不有趣，但是另外一个失败的常见（且可避免的）的原因是：他们太难于在移动终端使用。

易用性从用户开始，而用户在知识、兴趣、目标等方面又各有不同。作为测试工程师，我们必须问跟 APP 用户有关的问题，他们的移动任务，他们工作的环境，他们使用的终端的类型，以及他们对技术的理解有多少。

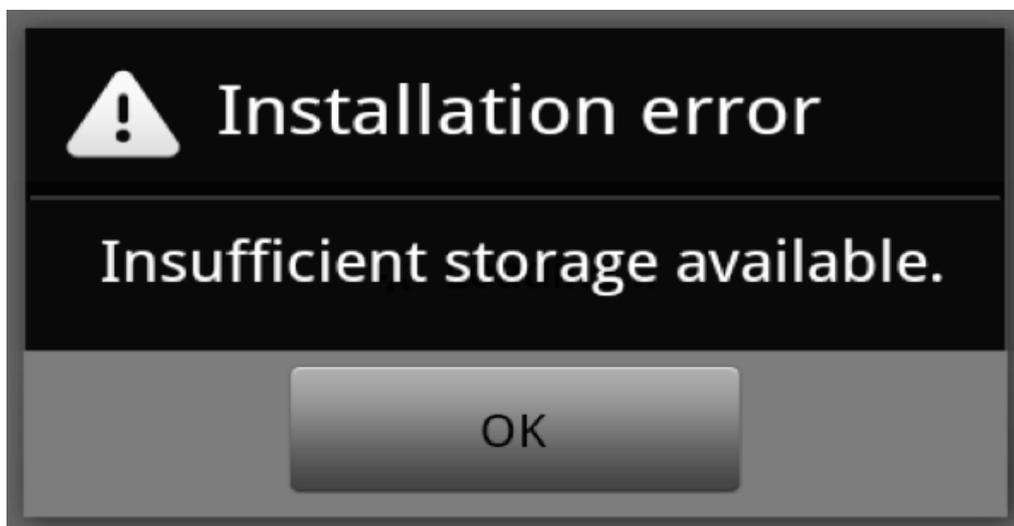
某软件在 2012 年就用户会对较差的移动 APP 体验如何反应做了一份调查。百分之 48 的人汇报说他们会不太可能去使用那个移动 APP，百分之 34 的人说他们会转向竞争者的 APP，百分之 31 的用户会向其他用户说他们的糟糕经理，而百分之 31 的用户会说他们会不再从该 APP 公司购买东西。

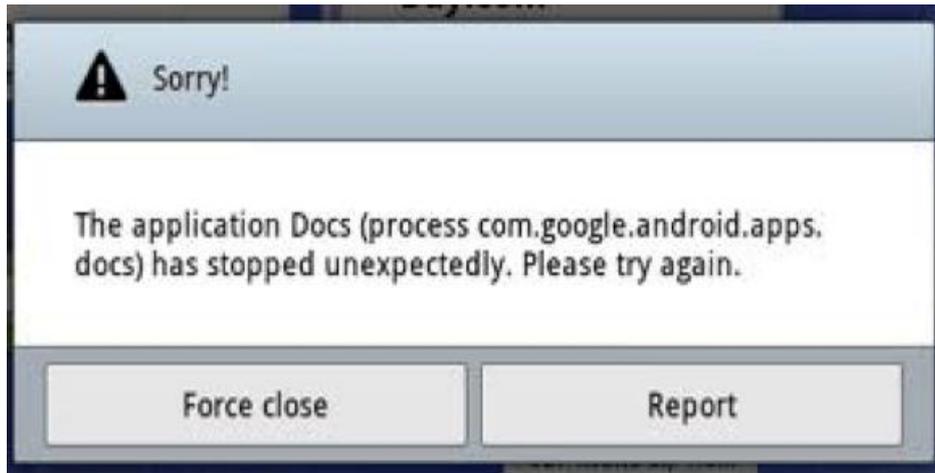
显然，易用性是需要考虑的一项重要的特征。这篇文章讨论了一些跟用户有关的核心理易用性的特征，及测试工程师怎样理解并实现这些易用性特征的。

整体 APP 的简单性：印度铁路移动端 APP IRCTC 被广泛使用，而一件值得称赞的事是它的用户界面的简单易用。它所有的元素对不同技能水平的用户都是陈列良好地放在那里。它的主页，如下图，直接并且容易导航。

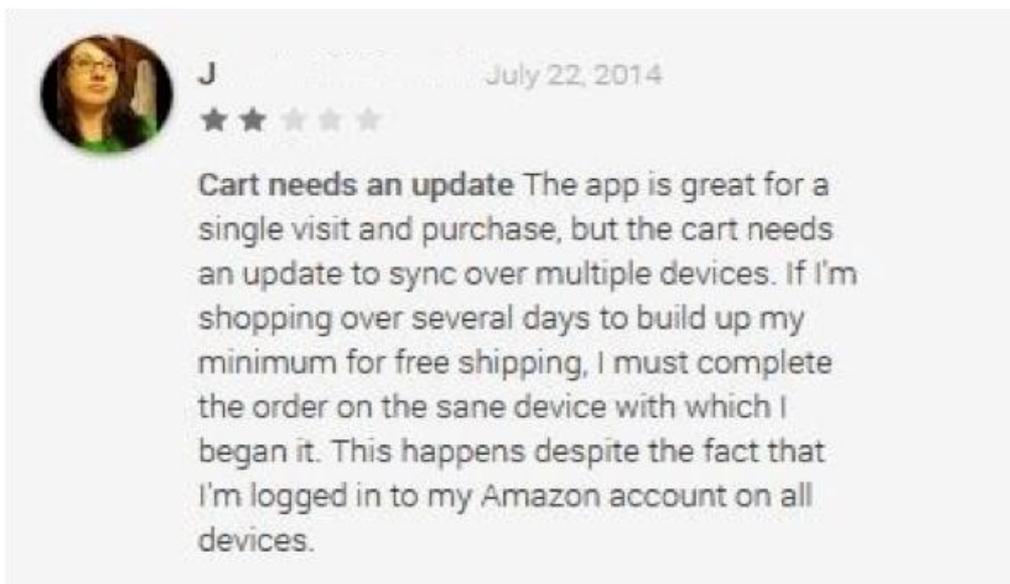


帮助性的错误场景：错误应当易于理解，首先，其次，易于恢复。下面的第一个错误对话框，用户应当能够理解没有足够空间供该程序安装。但是在第二个错误消息里，问题并不清晰，而用户将会有一段比较困难的时候用来排解问题所在。





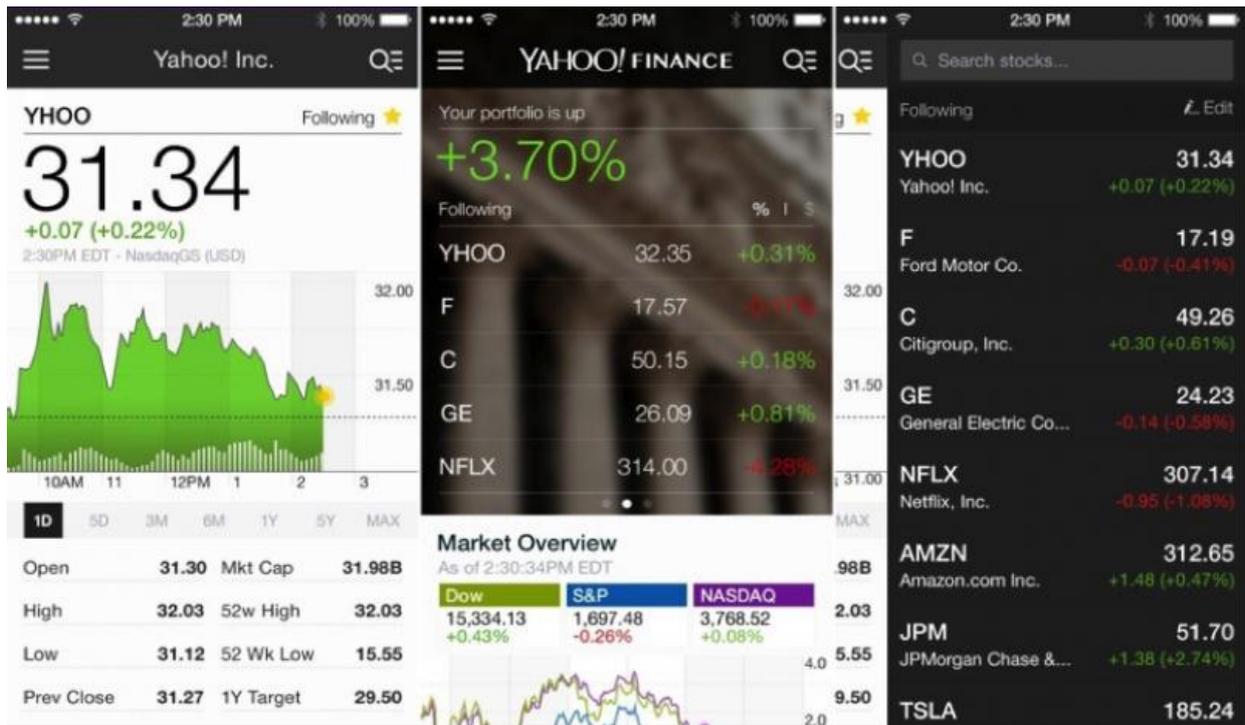
工作效率和同步化：用户使用一个 APP 是为了将一项特定的工作完成。下面的例子中，亚马逊的 APP 用户在多个终端同步了她的 APP。但是，当她在一个终端上建造自己的购物车然后在另一台终端访问，她的购物车是空的。因此，这种对单次访问是好的，对跨终端访问却行不通。这种挫败感让任何人都很丧气。



端到端用户满意度：我公司最近开发了一个 IOS 程序，通过互动游戏教孩子基本的数学概念。如你在下图截图中看到的，这个 APP 很简单，学生能从每页上学到小的片段的信息。作为我们用户测试的一部分，我们实际上会带一些孩子进来测试以测量整体的用户满意度。



另外一方面，考虑下雅虎的金融 APP。创建者知道这个 APP 会被繁忙的专业人士使用，而且他们想要在一个页面上获取所有可能的细节信息，即使这样看起来页面乱糟糟。



用户类型以及他们如何跟你的产品交互必须在对每个 APP 进行测试的时候考虑进去。

作为测试工程师，我们必须理解移动端应用程序跟传统的桌面和基于 web 的 APP 有很多不同之处。相应地，测试集中区域也需要有所不同。很多区域需要特别注意，但是让我们将讨论限定在仅 5 个元素上：功能性，背景，终端范围类别，数据输入方法和多峰行性。

我们用 skype 移动端 APP 作为第一个例子。典型的移动终端通常有两个话筒：一个在前端用作常规电话，一个在机器后端做外放。Skype APP 在将常规电话模式转换为外放模式当用户将终端设备从手掌移到耳朵处时。这不仅不舒服，而且在公共场合也是尴尬的一件事。这是一个功能不好导致用户体验不好的例子。

背景可以定义为一个用户的移动体验的总和。这包括用户形势上的、态度上和偏好的习惯。移动背景被如屏幕尺寸，用户手指，位置，姿势和网络链接这样的因素限制住。移动端背景还包括不同的功能如自动定位检测，动态位置检测，指南针，和加速器。我们那移动背景有关定位功能举例。加入你让你的程序去列出一个中国餐馆的清单，一个好的程序会定位你的地理位置然后给你一些附近去处的单子，包括同一位置的其他用户的平稳，还有依据一天的不同时间的特定饮食选项。结果应当是目的导向，有上下文并且有价值的。

另一个关注区域是该程序的不同支持终端模式。一个程序在一台终端上也许运行良好，但是在另一台可能就会崩溃或者失败。例如，Zomato 是一个餐馆导航 APP，拥有很棒的功能和用户节目。然而，这个程序在安卓 4.3 版本上却是崩溃的。测试者在验证这些讨厌的地方有一个主要的责任，需要在头脑中有一个优化的测试矩阵。

接下来的是数据录入方法。我们认为移动终端，尤其是智能设备，支持很多数据录入方法，包括输入笔，触摸，声音识别，姿势识别等。测试工程师必须确保这个程序对每个可能的输入点都响应良好。

多峰行性将声音和触摸组在一起作为输入，将可视化屏幕和语音响应作为输出--举个例子，一个提供语音响应给司机做音频帮助，将可视化图面做方向的导航程序。测试者需要检查输入和输出来确保音频和视频组件功能良好。将很多需要从移动端易用性角度考虑的变量考虑进去，下面的惯例在要创建一个测试策略时会很有用。

早些开始。测试应当在需求分析阶段就开始。大多数易用性问题与设计有关，而如果你能在评审设计做线框或者模拟的时候早点将这些问题捕捉起来，这将意味着在测试的路上会少很多麻烦。

迭代性地工作。有规律的测试能够让在程序整个生命周期上支持小的增量的变化容易得多。易用性应当在整个开发生命周期的所有阶段被考虑进去，因为每个设计上的变化和每次对产品原型的交互都有影响。

与其他测试类型一起使用易用性。你需要在易用性测试上考虑组合的元素。举个例子，功能性和易用性可能会造成冲突；功能设计者可能想要产品功能齐全，而易用行工程师的关注在保持产品的简单和直观性。易用性研究需要共同合作完成，留意程序的性能，安全，本地化，功能和可得性方面。

发布之后检查。易用性研究应当在产品发布后仍然继续。检查 APP 商店，研究用户的评分和评论，分析竞争都是在增强程序的易用性方面有价值的做法。

使用有生产力的工具。移动 APP 的易用性测试仍然大部分是要手工完成的，有使用有效率的工具来让测试工作更有效率的空间。有很多可用的便捷的有效的工具能测试很多易用性因素，包括测试程序跨终端设备的最佳屏幕分辨率。最后，你得到一个对每个选中的终端的截图的报告，指示屏幕分辨率效率。另外一个工具创造热量地图来帮助查找用户注意力关注最多的区域。这些可视的指示器帮助重新对齐用户界面，如果有需

要的话。而录制软件展示了参与者在哪儿敲打并卷动以及他们怎样从一个页面导到另一个页面，帮助测量工作流的直观性水平。

回到源头去发现用户期望。读有关易用性的东西是一回事，但是跟有特殊可得性需求的人互动是无价的。它打开你的眼睛去观察他们的需求，揭露一些你可能有的覆盖空白。

像普通人那样说话，而不是程序员。情绪智能因素正在引起注意。当你谈论终端用户时，不要把他们说成是实体，而是要把他们看作有情绪的实实在在的人。如果你向在你的测试策略中将情绪角度加入进来而又不知道从哪里开始，易用性是一个供考虑的好地方。

测试，测试，再测试。有一个良好打磨的策略固然是好的，腾出时间来做探索性测试会更好。大多数时候，易用性问题在你使用探索式方式操作产品的时候浮出水面。邀请不同组来测试你的 APP 然后想一些有创意的方式来最大化覆盖测试范围从而确保测试完整全面。

本质上来讲，移动端易用性在增强终端用户的 APP 认可度上大有帮助。这个行业仍然在日新月异地演变着，但是上面讨论的惯例方法会让你在建造可用性强的移动程序上有一个良好的开端。

网银浏览器兼容性测试实施项目总结

◆ 作者：姜林斌

2013年8月底进入浏览器兼容性项目组，历时10个月(原计划3个月)，项目终于全部交付，期间感慨颇多，感谢那些日子陪伴我一起敲代码，一起加班，一起兴奋且幸福前行的兄弟们，谨以此文纪念在甬城的那些美丽的日子和我们放荡不羁青春。

一：银行业务知识总结

商业银行业务：

一、资产业务

资产业务，是商业银行的主要收入来源。

1、放款业务--商业银行最主要的资产业务

1) 信用放款：

信用放款，指单凭借款人的信誉，而不需提供任何抵押品的放款，是一种资本放款。

(1) 普通借款限额：

企业与银行订立一种非正式协议，以确定一个贷款，在限额内，企业可随时得到银行的贷款支持，限额的有效期限一般不超过90天。普通贷款限额内的贷款，利率是浮动的，与银行的优惠利率挂钩。

(2) 透支放款：

银行通过允许客户在其帐户上透支的方式向客户提供贷款。提供这种便利被视为银行对客户所承担的合同之外的“附加义务”。

(3) 备用贷款承诺：

备用贷款承诺，是一种比较正式和具有法律约束的协议。银行与企业签订正式合同，在合同中银行承诺在指定期限和限额内向企业提供相应贷款，企业要为银行的承诺提供费用。

(4) 消费者放款：

消费者放款是对消费个人发放的用于购买耐用消费品或支付其他费用的放款，商业银行向客户提供这种贷款时，要进行多方面的审查。

(5) 票据贴现放款：

票据贴现放款，是顾客将未到期的票据提交银行，由银行扣除自贴现日起至到期日止的利息而取得现款。

2) 抵押放款：

抵押贷款有以下几种类型

(1) 存货贷款。存货贷款也称商品贷款，是一种以企业的存货或商品作为抵押品的短期贷款。

(2) 客帐贷款。银行发放的以应收帐款作为抵押的短期贷款，称为“客帐贷款”。这种贷款一般都为一种持续性的信贷协定。

(3) 证券贷款。银行发放的企业借款，除以应收款和存货作为抵押外，也有不少是用各种证券特别是公司企业发行的股票和债券作押的。这类贷款称为“证券贷款”。

(4) 不动产抵押贷款。通常是指以房地产或企业设备抵押品的贷款。

3) 保证书担保放款：

保证书担保放款，是指由经第三者出具保证书担保的放款。保证书是保证为借款人作贷款担保，与银行的契约性文件，其中规定了银行和保证人的权利和义务。

银行只要取得经保证人签字的银行拟定的标准格式保证书，即可向借款人发放贷款。所以，保证书是银行可以接受的最简单的担保形式。

4) 贷款证券化：

贷款证券化是指商业银行通过一定程序将贷款转化为证券发行的总理资过程。具体做法是：商业银行将所持有的各种流动性较差的贷款，组合成若干个资产库（Assets

Pool), 出售给专业性的融资公司 (Special Purpose Corporation), 再由融资公司以这些资产库为担保, 发行资产抵押证券。这种资产抵押证券同样可以通过证券发行市场发行或私募的方式推销给投资者。出售证券所收回的资金则可做为商业银行新的资金来源再用于发放其它贷款。

2、投资业务:

商业银行的投资业务是指银行购买有价证券的活动。投资是商业银行一项重要的资产业务, 是银行收入的主要来源之一。

商业银行的投资业务, 按照对象的不同, 可分为国内证券投资和国际证券投资。国内证券投资大体可分为三种类型, 即政府证券投资、地方政府证券投资和公司证券投资。

国家政府发行的证券, 按照销售方式的不同, 可以分为两种, 一种叫作公开销售的证券, 一种叫作不公开销售的证券。

商业银行购买的政府证券, 包括国库券、中期债券和长期债券三种。

1) 国库券。国库券是政府短期债券, 期限在一年以下。

2) 中长期债券。中长期债券是国家为了基建投资的资金需要而发行的一种债券, 其利率一般较高, 期限也较长, 是商业银行较好的投资对象。

二、负债业务

负债是银行由于受信而承担的将以资产或资本偿付的能以货币计量的债务。存款、派生存款是银行的主要负债, 约占资金来源的 80% 以上, 另外联行存款、同业存款、借入或拆入款项或发行债券等, 也构成银行的负债。

1、活期存款:

活期存款是相对于定期存款而言的, 是不需预先通知可随时提取或支付的存款。

活期存款构成了商业银行的重要资金来源, 也是商业银行创造信用的重要条件。但成本较高。商业银行只向客户免费或低费提供服务, 一般不支付或较少支付利息。

2、定期存款:

定期存款是相对于活期存款而言的, 是一种由存户预先约定期限的存款。定期存款

占银行存款比重较高。因为定期存款固定而且比较长，从而为商业银行提供了稳定的资金来源，对商业银行长期放款与投资具有重要意义。

3、 储蓄存款：

储蓄存款是个人为积蓄货币和取得利息收入而开立的存款帐户，储蓄存款又可分为活期和定期。

储蓄存款的活期存款，或者称为活期储蓄存款，存取无一定定期期限，只凭存折便可提现。存折一般不能转让流通，存户不能透支款项。

4、 可转让定期存单（CDs）：

可转让定期存单存款是定期存款的一种主要形式，但与前述定期存款又有所区别。可转让存单存款的明显特点是：存单面额固定，不记姓名，利率有固定也有浮动，存期为3个月、6个月、9个月和12个月不等。存单能够流通转让，以能够满足流动性和盈利性的双重要求。

5、 可转让支付命令存款帐户：

它实际上是一种不使用支票的支票帐户。它以支付命令书取代了支票。通过此帐户，商业银行既可以提供支付上的便利，又可以支付利息，从而吸引储户，扩大存款。

开立这种存款帐户，存户可以随时开出支付命令书，或直接提现，或直接向第三者支付，其存款余额可取得利息收入。由此满足了支付上的便利要求，同时也满足了收益上的要求。

6、 自动转帐服务存款帐户：

这一帐户与可转让支付命令存款帐户类似，是在电话转帐服务基础上发展而来。发展到自动转帐服务时，存户可以同时同时在银行开立两个帐户：储蓄帐户和活期存款帐户。银行收到存户所开出的支票需要付款时，可随即将支付款项从储蓄帐户上转到活期存款帐户上，自动转帐，即时支付支票上的款项。

7、 掉期存款：

掉期存款指的是顾客在存款时把手上的由名义上兑换成其所选择的外币，作为外币定期存款存入银行。到期满时顾客先将外币存款连本带息兑回本币后才提取。存款期限由一个月至一年不等。

三、中间业务

中间业务又称表外业务，其收入不列入银行资产负债表。

1、 结算业务

结算业务是由商业银行的存款业务衍生出来的一种业务。

1) 结算工具:

结算工具就是商业银行用于结算的各种票据。目前可选择使用的票据结算工具主要包括银行汇票、商业汇票、银行本票和支票等。

主要的票据结算工具:

(1)、银行汇票: 由企业单位或个人将款项交存开户银行, 由银行签发给其持往异地采购商品时办理结算或支配现金的票据。

(2)、商业汇票: 由企业签发的一种票据, 适用于企业单位先发货后收款或双方约定延期付款的商品交易。

(3)、银行本票: 申请人将款项交存银行, 由银行签发给其凭以办理转帐或支取现金的票据。可分为不定额本票和定额本票。

(4)、支票: 由企业单位或个人签发的, 委托其开户银行付款的票据, 是我国传统的票据结算工具, 可用于支取现金和转帐。

2) 结算方式:

(1) 同城结算方式:

A、支票结算: 商业银行最主要或大量的同城结算方式是支票结算。支票结算就是银行顾客根据其在银行的存款和透支限额开出支票, 命令银行从其帐户中支付一定款项给收款人, 从而实现资金调拨, 了结债权债务关系的一种过程。

B、帐单支票与划拨制度。这是不用开支票, 通过直接记帐而实现资金结算的方式。

C、直接贷记转帐和直接借记转帐。这两种结算方式是在自动交换所的基础上发展起来的。自动交换所交换的是磁带而不是支票。它通过电子计算机对各行送交的磁带进行处理, 实现不同银行资金结算。

D、票据交换所自动转帐系统。这是一种进行同城同业资金调拨的系统。参加这种系统的银行之间，所有同业拆借、外汇买卖、汇划款项等将有关数据输入到自动转帐系统的终端机，这样收款银行立即可以收到有关信息，交换所同时借记付款银行帐户，贷记收款银行帐户。

(2) 异地结算方式:

A、汇款结算：汇款结算（Remittance），是付款人委托银行将款项汇给外地某收款人的一种结算方式。汇款结算又分为电汇、信汇和票汇三种形式。

B、托收结算：托收结算（Collection），是指债权人或售货人为向外地债务人或购货人收取款项而向其开出汇票，并委托银行代为收取款项的一种结算方式。托收业务主要有光票托收和跟单托收两类。

C、信用证结算：信用证（Letter of Credit），一种有条件的银行付款承诺，即开证银行根据进口商的指示，向出口商开立的，授权其签发以进口商或银行为付款人的汇票，保证在条款规定条件下必定付款或承兑的文件。

D、电子资金划拨系统

随着电子计算机等新技术投入银行运用，电子计算机的大型化和网络化改变了商业银行异地资金结算的传统处理方式。通过电子资金结算系统进行异地结算，使资金周转大大加快，业务费用大大降低。

2、 信用证业务:

信用证（Letter of Credit），作为商业贸易的手段之一，银行信用证是进口商的代理银行为进口商提供自身的信用，保证在一定的条件下承付出口商开给进口商的票据，即将所开票据当作是开给本行的票据。所谓信用证即是保证承付这些票据的证书。

信用证的种类:

1) 银行信用证

汇票的接受人是银行，开证行或受其委托的保兑银行承兑开给自己的汇票，这种信用证是银行信用证（Bank Credit）。

2) 不可撤销信用证与可撤销信用证

不可撤销信用证（Irrevocable Credit），是指开证行一旦开立了信用证并将之通知了

受益人，在其有效期间，如若没有开证委托人、受益人或已依据此信用证贴现汇票的银行的同意，不可单方面地撤销此信用证，也不可变更其条件。

3) 保兑信用证与不保兑信用证

开证行以外的银行对卖方开出的汇票保证兑付，这种信用证称为保兑信用证 (Confirmed Credit)，而无此保证者则称为不保兑信用证 (Unconfirmed Credit)。

4) 一般信用证和特定信用证

信用证的开证行特别指定某一银行贴现根据此信用证开出的汇票，这种信用证称为特定信用证 (Special or Restricted Credit)，而不限定贴现银行者称为普通信用证 (General or Open Credit)。

3、 信托业务

信托 (Trust)，可以从两方面考察，从委托人来说，信托是为自己或为第三者的利益，把自己的财产委托别人管理或处理的一种行为；从受托人来说，信托是受委托人的委托，为了受益人的利益，代为管理、营运或处理信托人托管财产的一种过程。广义的信托还包括代理业务，如受托代办有价证券的签证、发行、收回、掉换、转让、还本付息以及代客保管物品等。

信托与代理的主要区别在于财产权是否转移，如果财产权从委托人转移到受托人，则是信托关系，而代理则不涉及财产权转移。

信托关系 (Fiduciary Relationship) 是一种包括委托人、受托人和受益人在内的多边关系。

信托行为、财产转移、本金

委托人 - - - - - > 受托人 - - - - - > 受益人

有关管理指示 受益

按信托方式划分：投资信托、融资信托、公益信托、职工福利信托。

(1) 证券投资信托

证券投资信托是以投资有价证券，获取投资收益为目的信托。它是由信托部门将个

人或企业、团体的投资资金集中起来，代替投资者进行投资，最后将投资收益和本金偿还给受益人。

(2) 动产或不动产信托

动产或不动产信托是由大型设备或财产的所有者提出的、以融通资金为目的信托。

(3) 公益信托业务

公益信托是一种由个人或团体捐赠或募集的基金，以用于公益事业为目的的信托。

4、租赁业务

1) 融资性租赁 (Financial Lease) 是以融通资金为目的租赁。一般先由承租人自行从供货处选好所需设备，并谈妥交易条件，然后找出租人 (金融机构或其附属的专业子公司)，要求后者按谈妥的条件向供货商购买设备，并签订租赁合同，取得设备使用权，并按期交纳租金。这时出租人支付了全部资金，等于提供了百分之百的信贷，因此又叫融资性租赁或资本性租赁。

2) 操作性租赁 (Operating Lease) 又叫服务性租赁，是由出租人向承租人提供一种特殊服务的租赁。这种特殊服务主要是指设备的短期使用或利用服务，如出租人买下库房、车船、电子计算机等，然后出租给承租人。操作性租赁通常适用于一些需要专门技术进行保养、技术更新、使用频度不高的设备。

3) 出售与返租式租赁 (Sale and Lease back) 是财产所有人将其财产出售以后又租回使用的一种租赁方式。这种租赁的后半段与一般租赁完全相同，只是增加了前半段的出售过程，财产所有者又变成了财产使用者。

4) 转租赁 (Sublease)，是将设备或财产进行两次重复租赁的方式。国际租赁中通常采用这种租赁方式。

5、代理业务

代理融通 (Factoring) 又叫代收帐款或收买应收帐款，是由商业银行或专业代理融通公司代顾客收取应收款项，并向顾客提供资金融通的一种业务方式。

代理融通业务一般涉及三方面当事人，一是商业银行或经营代理融通业务的公司，二是出售应收帐款、取得资金融通的工商企业，三是取得商业信用及赊欠工商企业贷款的顾客。三者的关系是，工商企业对顾客赊销货物或劳务，然后把应收的赊销帐款转

让给银行或代理融通公司，由后者向企业提供资金并到期向顾客收帐。

6、银行卡业务

1) 信用卡

信用卡是消费信贷的一种工具和形式，具有“先消费”、方便消费者的特点。

信用卡的种类很多，除银行发行的信用卡外，还有商业和其他服务业发行的零信用卡、旅游娱乐卡等。

2) 支票卡

支票卡又叫保证卡，供客户开发支票时证明其身份的发卡。卡片载明客户的帐户、签名和有效期限。

3) 自动出纳机卡和记帐卡

自动出纳机卡是一种印有磁带、专供在自动出纳机上使用的塑料卡。卡上除标明发行银行、卡片号码外，磁带上还记录有客户的存款户帐号、密码和余额。

4) 灵光卡和激光卡

灵光卡又叫记帐卡、方便卡，是一种带微型集成电路的塑料卡片，具有自动、数据处理和储存的功能，卡片可以记录客户每笔收支和存款余额。

7、咨询业务：

在现代社会，信息已成为社会发展的主要支柱之一。商业银行通过资金运动的记录，以及与资金运动相关资料的收集整理，可以为企业提供丰富实用的经济信息。其主要内容有：企业财务资料资信评价；商品市场供需结构变化趋势介绍；金融市场动态分析。

四、商业银行的国际业务

1、国际结算业务

国际间进行贸易和非贸易往来而发生的债权债务，要用货币收付，在一定的形式和条件下结清，这样就产生了国际结算业务。

国际结算方式是从简单的现金结算方式，发展到比较完善的银行信用证方式，货币的收付形成资金流动，而资金的流动又须通过各种结算工具的传送来实现。

1) 汇款结算业务

汇款是付款人把应付款项交给自己的往来银行，请求银行代替自己通过邮寄的方法，把款项支付给收款人的一种结算方式。银行接到付款人的请求后，收下款项，然后以某种方式通知收款人所在地的代理行，请它向收款人支付相同金额的款项。最后，两个银行通过事先的办法，结清两者之间的债权债务。

汇款结算方式一般涉及四个当事人，即汇款人、收款人、汇出行和汇入行。

国际汇款结算业务基本上分为三大类，即电汇、信汇和票汇。

2) 托收结算业务

托收是债权人向国外债务人收取款项而向其开发汇票，委托银行代收的一种结算方式。

一笔托收结算业务通常有四个当事人，即委托人、托收银行、代收银行和付款人。西方商业银行办理的国际托收结算业务为两大类，一类为光票托收，另一类为跟单托收。

3) 信用证结算业务

信用证结算方式是指进出口双方签订买卖合同后，进口商主动请示进口地银行向出口商开立信用证，对自己的付款责任作出保证。当出口商按照信用证的条款履行了自己的责任后，进口商将货款通过银行交付给出口商。

一笔信用证结算业务所涉及的基本当事人有三个，即开证申请人、受益人和开证银行。

4) 担保业务

在国际结算过程中，银行还经常以本身的信誉为进出口商提供担保，以促进结算过程的顺利进行。目前为进出口结算提供的担保主要有两种形式，即银行保证书和备用信用证。

(1) 银行保证书 (Letter of Guarantee)

银行保证书又称保函，是银行应委托人的请求，向受益人开出的担保被保证人履行职责的一种文件。

(2) 备用信用证 (Stand-by Letter of Credit)

备用信用证是一种银行保证书性质的凭证。它是开证行对受益人开出的担保文件。保证开证申请人履行自己的职责，否则银行负责清偿所欠受益的款项。

2、国际信贷与投资

国际信贷与投资是商业银行国际业务中的资产业务。国际信贷与投资与国内资产业务有所不同。这种业务的对象绝大部分是国外借款者。

1) 进出口融资

商业银行国际信贷活动的一个重要方面，是为国际贸易提供资金融通。这种资金融通的对象，包括本国和外国的进出口商人。

商业银行为进出口贸易提供资金融通的形式很多，主要有以下几种：

进口押汇，是指进出口双方签订买卖合同后，进口方请求进口地的某个银行（一般为自己的往来银行），向出口方开立保证付款文件，大多为信用证。然后，开证行将此文件寄送给出口商，出口商见证后，将货物发运给进口商。银行为进口商开立信用保证文件的这一过程。

出口押汇，出口商根据买卖合同的规定向进口商发出货物后，取得了各种单据，同时，根据有关条款，向进口商开发汇票。

另外，提供资金融通的方式还有打包放款，票据承兑，出口贷款等。

2) 国际放款

国际放款由于超越了国界，在放款的对象、放款的风险、放款的方式等方面，都与国内放款具有不同之处。

商业银行国际放款的类型，可以从不同的角度进行划分。

(1)根据放款对象的不同，可以划分为个人放款、企业放款、银行间放款以及对外国政府和中央银行的放款。

(2)根据放款期限的不同，可以划分为短期放款、中期放款和长期放款，这种期限的划分与国内放款形式大致相同。

(3)根据放款银行的不同，可以划分为单一银行放款和多银行放款。单一银行放款是

指放款资金仅由一个银行提供。一般来说，单一银行放款一般数额较小，期限较短。多银行放款是指一笔放款由几家银行共同提供，这种放款主要有两种类型：第一，参与制放款。第二，辛迪加放款。

3) 国际投资

根据证券投资对象的不同。商业银行国际投资可以分为外国债券投资和欧洲债券投资两种。

(1)外国债券投资。外国债券（Foreign Bond）是指由外国债务人在投资人所在国发行的，以投资国货币标价的借款凭证。外国债券的发行人包括外国公司、外国政府和国际组织。外国债券的购买人为 债权国的工商企业、金融机构以及个人等，其中，商业银行是重要的投资者。

(2)欧洲债券投资。欧洲债券（Europe-Bond）是指债务人在欧洲金融市场上发行的，以销售国以外的货币标价的借款凭证。

欧洲债券是目前国际债券的最主要形式。

欧洲债券有很多形式：

①普通债券（Straight Bond）。这是欧洲债券的基本形式，其特点是：债券的利息固定，有明确的到期日，由于所支付的利息不随金融市场上利率的变化而升降，因此，当市场利率波动剧烈时，就会影响其发行量。

②复合货币债券（Multiple-Currency Bond）。债务人发行债券时，以几种货币表示债券的面值。投资人购买债券时，以其中的一种货币付款。

③浮动利率债券（Floating Rate Bond）。浮动利率债券是指债券的票面利率随金融市场利率水平的变化而调整的债券。

④可转换为股票的债券（Convertible Euro-Bond）。这种债券的特点是：债务人在发行债券时事先授权，投资者可以根据自己的愿望，将此种债券转换为发行公司的股票，成为该公司的股东。

商业银行的国际业务中，外汇交易业务也是很重要的一部分，它包括：外汇头寸、即期外汇买卖、远期外汇买卖、期权交易、套汇与套利、以及投机等。

五、商业银行联行往来业务

1、联行往来的基本概念

社会资金往来运动最终要体现在银行间的划拨上，当资金结算业务发生时，必然要通过两个或两个以上的银行机构往来才能完成，如果往来双方同属一个银行系统，即同属一个总行的各个分支机构间的资金帐务往来，则称其为联行往来：

1) 全国联行往来。全国联行往来适用于总行与所属各级分支之间以及不同省、自治区、直辖市各机构之间的资金帐务往来。全国联行往业帐务由总行负责监督管理。

2) 分行辖内往来。分行辖内往来适用于省、自治区、直辖市分行与所辖各分支机构之间以及同一省、自治区、直辖市辖内各银行机构之间的资金帐务往来。分行辖内联行在往来帐务由分行负责监督管理。

3) 支行辖内往来。支行辖内往来适用于县（市）支行与所属各机构之间以及同一县（市）支行内各机构之间的资金帐务往来。其所涉及的帐务由县（市）支行管理监督。

2、联行往来账务核算

1) 发报行核算

发报行是联行往来帐务的发生行，是保证联行帐务正确进行的基础，对整个联行工作质量，起着重要作用。包括：报单的编制；报单的审查与寄发；联行往帐报告表的编制。

2) 收报行核算

收报行是联行往帐的受理者，它对发报行寄来的联行报单及所附凭证，必须进行认真审核和再复核，并应准确、迅速办理转帐和对帐，以保证全国联行往来核算工作的正确进行。

3) 总行电子计算中心

总行电子计算中心是对全国联行往来帐务进行逐笔集中监督的部门，它根据联行往帐报告表监督联行往帐；接收报行行号编制对帐表，寄收报行对帐，监督联行来帐，以保证全国联行往帐与来帐双方一致。

二：浏览器兼容性自动化测试实施

2.1 VisualStudio+SeleniumWebDriver 开发环境搭建

2.1.1 下载 IEDriverServer

先到官网先下载 IEDriverServer (32 位或者 64 位), 解压文件将 IEDriverServer.exe 存放到一个指定路径, 要记住这个路径, 配置过程中要用到。(http:

//docs.seleniumhq.org/download/)

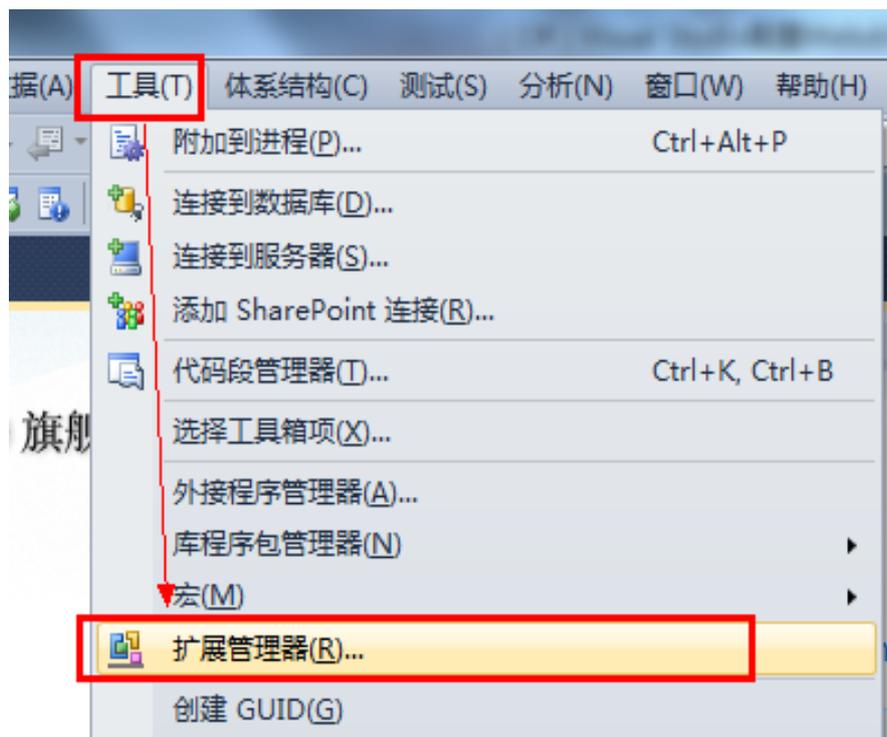
The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your \$PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.

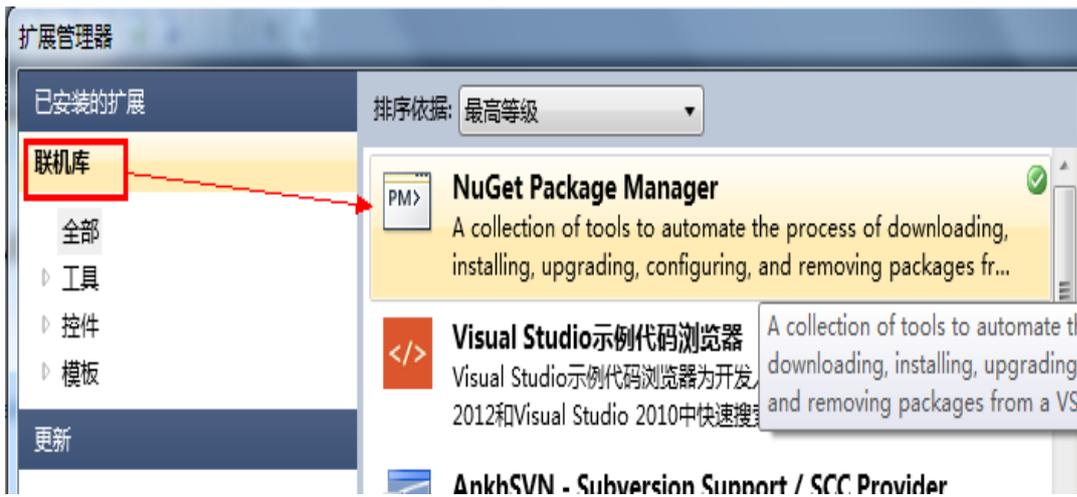
Download version 2.37.0 for (recommended) [32 bit Windows IE](#) or [64 bit Windows IE](#)
[CHANGELOG](#)

2.1.2 配置 VisualStudio

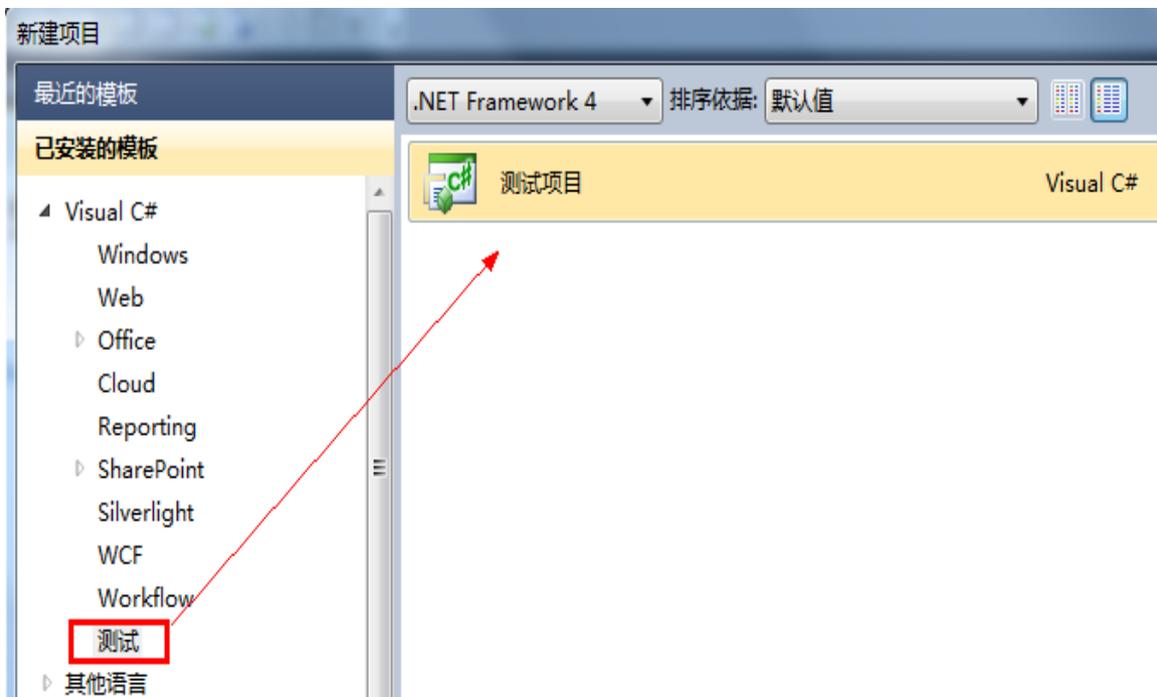
打开 VS (为了便于查看, 我用的是中文版的, 英文版的请自己对照位置), 打开“工具”菜单下的“扩展管理器”:



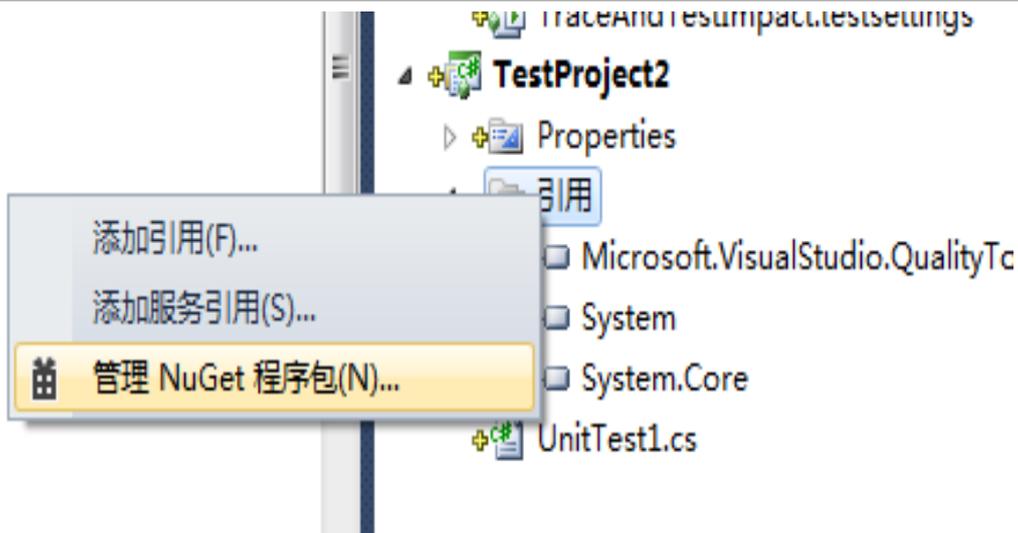
我们需要在“扩展管理器”中安装“NuGet”, 单击下载安装, 然后重启你的 VS



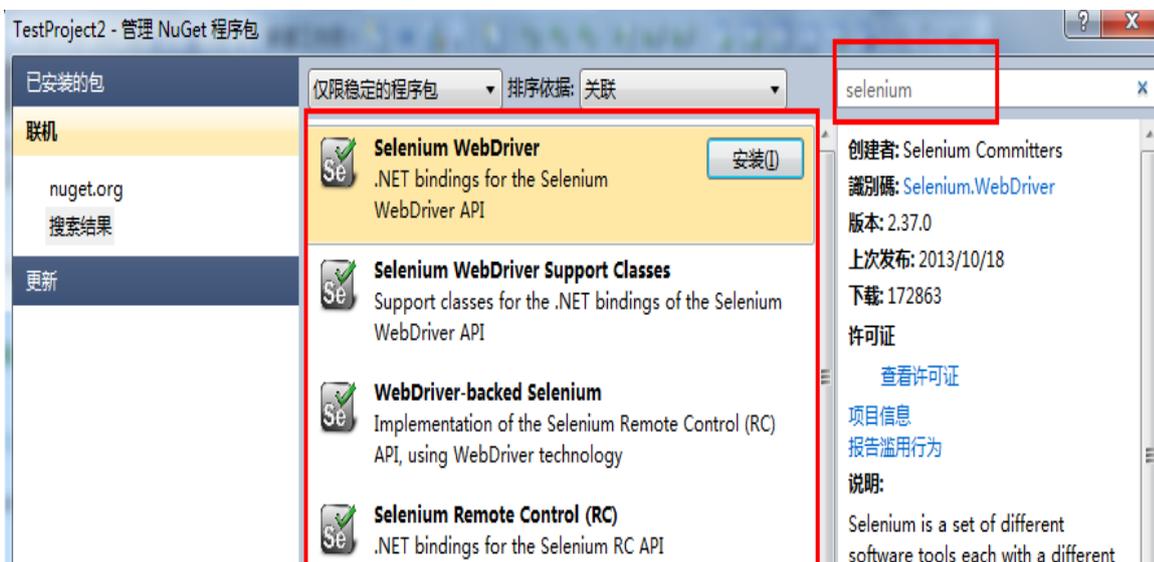
创建一个测试项进行测试



右键“引用”，选择“NuGet 程序包”。



在这里选择“联机”，搜索“selenium”。然后安装就行了。



安装 Selenium 包后项目引用里可看到如下动态库

- > Microsoft.VisualStudio.QualityTools.UnitTestFramework
- > Selenium.WebDriverBackedSelenium
- > System
- > System.Core
- > System.Drawing
- > ThoughtWorks.Selenium.Core
- > WebDriver
- > WebDriver.Support

创建一个测试类进行环境验证(注意到“C: \IEDriverServer”了吗? 这就是存放IEDriverServer.exe 的路径)

```

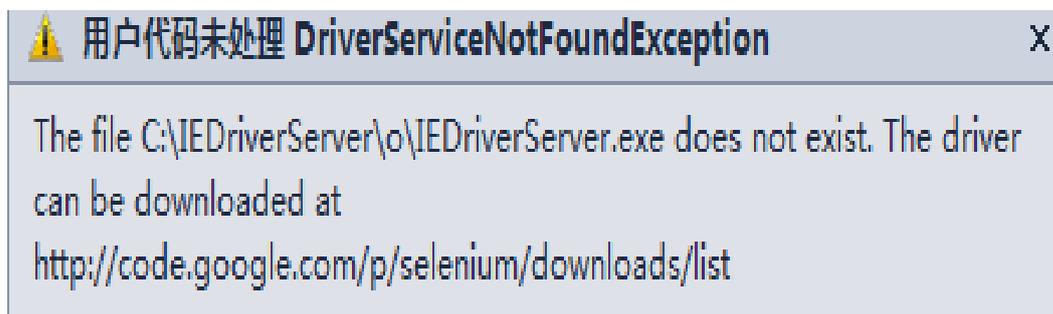
UnitTest1.cs x
TestProject4.UnitTest1
TestMethod1()
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.IE;
using OpenQA.Selenium.Support.UI;

namespace TestProject4
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            IWebDriver driver = new InternetExplorerDriver(@"C:\IEDriverServer");
            driver.Navigate().GoToUrl("www.baidu.com");
            driver.FindElement(By.Id("kw")).Clear();
            driver.FindElement(By.Id("kw")).SendKeys("Selenium");
            driver.FindElement(By.Id("su")).Click();
        }
    }
}

```

可能遇到的问题:

1: 未找到驱动服务



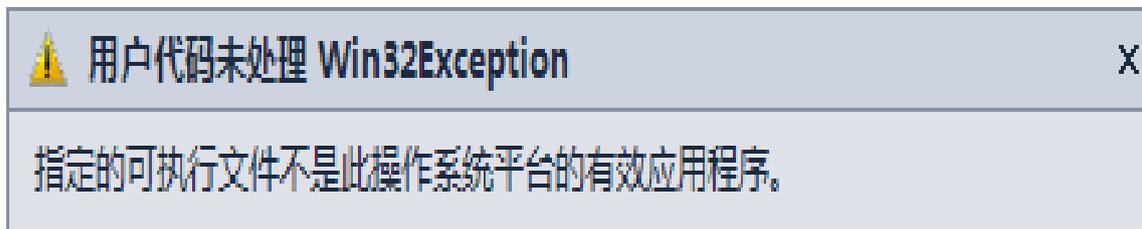
参考方案:

就看看 IEDriverServer.exe 的路径是不是错了。

另外: 浏览器的“启动保护模式”我给取消勾选了, 如果勾选的话偶尔也会报异常。

可能遇到的问题:

1: 用户代码未处理



参考方案: 看看是不是浏览器的驱动用错了 (32 位用了 64 位的, 或者反过来)。

2.2 C#版的 SeleniumAPI

1: Selenium 中对浏览器的操作

首先生成一个 Web 对象

```
IWebDriver driver = new FirefoxDriver();
```

```
//打开指定的 URL 地址
```

```
driver.Navigate().GoToUrl(@"http: //12.99.102.196: 9080/corporbank/logon_pro.html");
```

```
//关闭浏览器
```

```
Driver.quit();
```

网银浏览器兼容性测试过程中, 关闭浏览器后会有对话框, 此问题解决方法如下:

```
public void logout()
{
    System.Diagnostics.Process[] myProcesses;
    myProcesses = System.Diagnostics.Process.GetProcessesByName("IEXPLORE");
    foreach (System.Diagnostics.Process instance in myProcesses)
    {
        instance.Kill();
    }
}
```

2: Selenium 中执行 JS 脚本

```
//需要将 driver 强制转换成 JS 执行器类型
```

```
((IJavaScriptExecutor) driver).ExecuteScript("js 文件名");
```

3: Selenium 中定位页面元素

```
driver.FindElement(By.Id("cp1_btnModify")).click();  
By.ClassName(className);  
By.CssSelector(selector) ;  
By.Id(id);  
By.LinkText(linkText);  
By.Name(name);  
By.PartialLinkText(linkText);  
By.TagName(name);  
By.XPath(xpathExpression);
```

3.1 根据元素 id 定位并操作

```
//向指定的文本框输入字符串 500001  
Driver.FindElement(By.Id("amount")).SendKeys("500001");
```

3.2 根据元素 classname 定位并操作

```
//点击 classname 为指定值的按钮  
Driver.FindElement(By.ClassName("WotherDay")).click();
```

3.3 根据元素的 linktext 定位并操作

```
Driver.FindElement(By.LinkText("选择账号")).click();
```

3.4 根据元素的 Name 定位并操作

```
Driver.FindElement(By.Name("quality")).perform();
```

3.5 使用 CssSelector 定位并操作

```
string order = "#resultTable.result_table tbody tr.bg1 td.center a";  
driver.FindElement (By.CssSelector(order)).click();
```

3.6 使用 XPath 定位并元素并操作

```
//使用多个属性定位元素  
Driver.FindElement(By.XPath("//input[@id='submit' and @value='下一步']")).click();
```

```
//使用绝对路径定位元素  
string path = "/html/body/div[4]/div/div/div[2]/table/tbody/tr/td/a";
```

```
Driver.FindElement(By.XPath(path)).click();
```

各方法使用优先原则:

优先使用 id, name, classname, link; 次之使用 CssSelector(); 最后使用 XPath();

因为 Xpath()方法的性能和效率最低下。

4: Selenium 中清空文本框中的默认内容

```
//清空文本框 clear()  
Driver.FindElement(By.Id("tranAmtText")).clear();
```

5: Selenium 中在指定的文本框中输入指定的字符串

```
//在文本框中输入指定的字符串 sendkeys()  
Driver.FindElement(By.Id("tranAmtText")).SendKeys("123456");
```

6: Selenium 中移动光标到指定的元素上

```
//移动光标到指定的元素上 perform  
Actions action=new Actions(driver);  
action.MoveToElement(Find(By.XPath("//input[@id='submit' and @value='确定']"))).Perform();
```

7: Selenium 中点击按钮/链接

```
//点击按钮/链接 click()  
Driver.FindElement(By.XPath("//input[@id='submit' and @value='下一步']")).click();
```

8: Selenium 中等待页面上的元素加载完成

```
//等待页面元素加载完成  
//默认等待 100 秒  
WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(100));  
//等待页面上 ID 属性值为 submitButton 的元素加载完成  
wait.Until((d) => { return WaitForObject(By.Id("submitButton")); });
```

9: Selenium 中模拟鼠标晃动

```
//模拟光标晃动 movebyoffset()  
Actions action = new Actions(driver);  
action.MoveByOffset(2, 4);
```

10: Selenium 中对 iframe 中元素的定位

5.1: 切换焦点到 id 为固定值的 iframe 上

进入页面后，光标默认焦点在 DefaultContent 中，若想要定位到 iframe 需要转换焦点

```
driver.SwitchTo().DefaultContent();
```

```
//切换焦点到 mainFrame
```

```
driver.SwitchTo().Frame("mainFrame");
```

需要注意的是：切换焦点之后若想切换焦点到其他 iframe 上 需要先返回到 defaultcontent，再切换焦点到指定的 iframe 上。

5.2 切换焦点到 id 值为动态值的 iframe 上

有时候 页面上浮出层的 id 为动态值，此时需要先获取所有符合记录的 iframe 放置在数组中，然后遍历数组切换焦点到目标 iframe 上。

如下方法：

```
protected string bizFrameId = string.Empty;
```

```
protected string bizId = string.Empty;
```

```
//获取动态 iframe 的 id 值
```

```
protected void SetIframeId()
```

```
{
```

```
    ReadOnlyCollection<IWebElement> els = driver.FindElements(By.TagName("iframe"));
```

```
    foreach (var e in driver.FindElements(By.TagName("iframe")))
```

```
    {
```

```
        string s1 = e.GetAttribute("id");
```

```
        if (s1.IndexOf("window") >= 0 && s1.IndexOf("content") >= 0)
```

```
        {
```

```
            bizFrameId = e.GetAttribute("id");
```

```
            string[] ss = s1.Split(new char[] { '_' });
```

```
            bizId = ss[1];
```

```
        }
```

```
    }
```

```
}
```

11: Selenium 中关闭多个子 Browser 窗口

```
//获取所有的 WindowHandle，关闭所有子窗口
```

```
string oldwin = driver.CurrentWindowHandle;
```

```
ReadOnlyCollection<string> windows = driver.WindowHandles;
```

```

foreach (var win in windows)
{
    if (win != oldwin)
    {
        driver.SwitchTo().Window(win).Close();
    }
}

```

```
driver.SwitchTo().Window(oldwin);
```

12: Selenium 中对下拉框的操作

//选择下拉框

```

protected void SelectUsage(string selectid, string text)
{
    IWebElement select = Find(By.Id(selectid));
    IList<IWebElement> AllOptions = select.FindElements(By.TagName("option"));
    foreach (IWebElement option in select.FindElements(By.TagName("option")))
    {
        if (option.GetAttribute("value").Equals(text))
            option.Click();
    }
}

```

13: Selenium 中对 confirm , alert , prompt 的操作

//在本次浏览器兼容性测试项目中遇到的只有 confirm 和 alert

//下面举例说明 confirm 和 alert 的代码, prompt 类似

//confirm 的操作

```
IAlert confirm = driver.SwitchTo().Alert();
```

```

confirm.Accept();

//Alert 的操作

//个人网银中同样的业务有时候不会弹对 alert，此时需要判断 alert 是否存在

//对 Alert 提示框作确定操作，默认等待 50 毫秒
protected void AlertAccept()

```

```

{
    AlertAccept(0.05);
}

```

//等待几秒，可以为小数，单位为秒

```

protected void AlertAccept(double waitseSonds)
{
    double nsleepMillon = waitseSonds * 1000;
    int k=0;
    int split=50;
    IAlert alert = null;
    do
    {
        k++;
        Thread.Sleep(split);
        alert = driver.SwitchTo().Alert();
    } while (k * split <= nsleepMillon || alert==null);
        if (alert != null)
        {
            alert.Accept();
        }
    }
}

```

14: Selenium WebDriver 的截图功能

//WebDriver 中自带截图功能

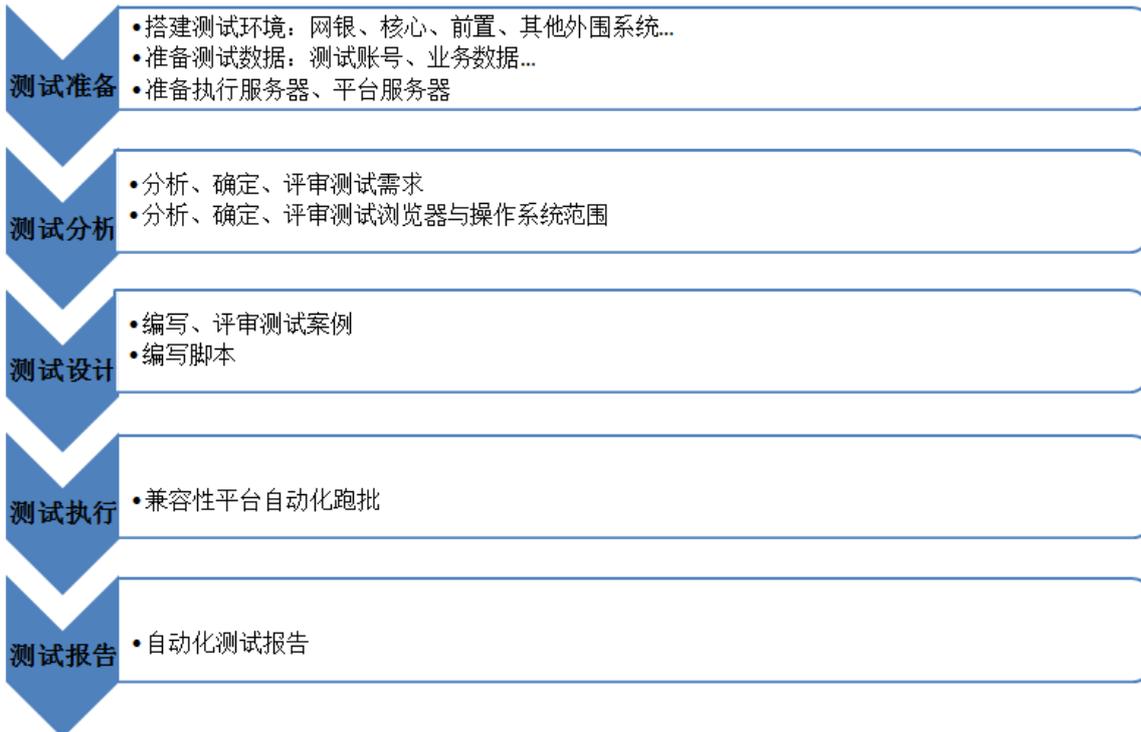
```

Screenshot screenShotFile = ((ITakesScreenshot)driver).GetScreenshot();
screenShotFile.SaveAsFile("test", ImageFormat.Jpeg);

```

2.3 浏览器兼容性测试平台介绍

平台工作流程介绍



2.3.2 搭建平台服务

1: 把最新版本的兼容性平台代码放置到 8.99.9.207 服务器的如下目录:

C: \apache-tomcat-7.0.42\webAPPs\CompatibilityTest

2: 启动 tomcat 服务

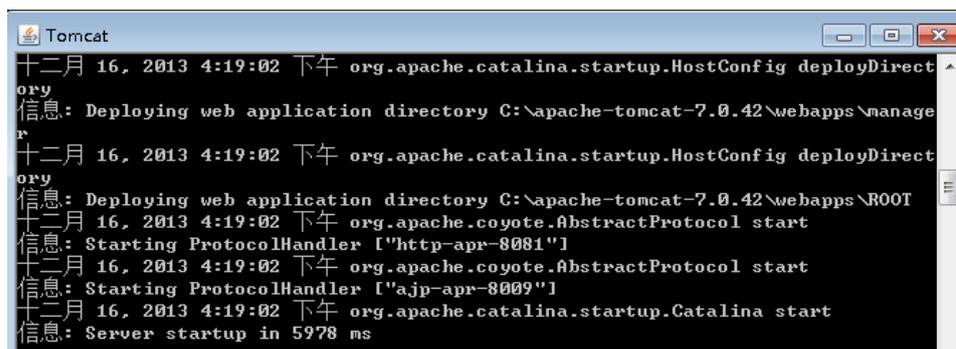
点击桌面上的 “ startup - 快捷方式” 如下图:

3: 访问服务对兼容性平台进行验证 [http://8.99.9.207:](http://8.99.9.207:8081/CompatibilityTest/_index.html)

8081/CompatibilityTest/_index.html

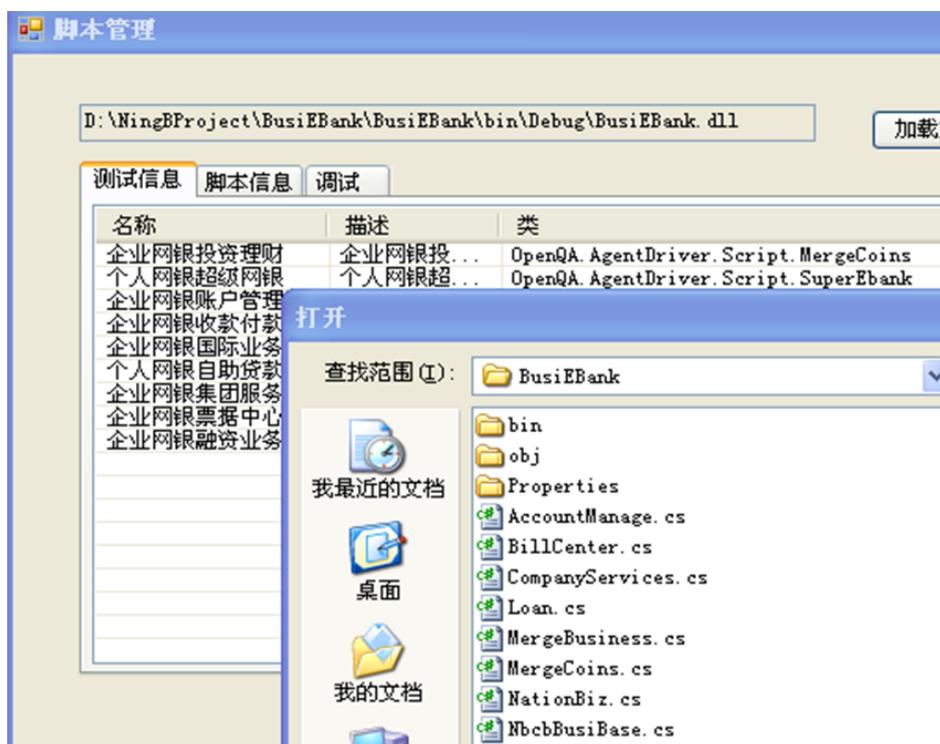
4: 启动 Test Agent

5: 启动 Analyse Agent



2.3.3 上传自动化脚本

- 1: 双击项目解决方案 bin\debug 目录下的 ScriptTool.exe 文件，如下图
- 2: 点击“加载文件”选择 bin\debug 目录下的对应的解决方案文件名
(即：编译后的 dll 文件)
- 3: 双击每条测试案例信息选择脚本的类文件进行关联。
- 4: 点击：“上传更新”（上传前请参考清空 test_run, test_unit, test_plan, test_plan_case 表）



2.3.4 创建测试计划

- 1: 点击兼容性平台的 图标
- 2: 填写测试计划属性：计划名称、计划描述、URL 起始地址(必填项)
- 3: 切换到“计划设计”主页面，选择测试用例
- 4: 点击 图标 保存当前测试计划的测试用例
- 5: 启动各主机的 AgentMain 进程
- 6: 点击 图标开始执行任务

2.3.5 测试监控

切换到执行监控 Tab 页可以查看哪些服务器正在执行自动化脚本以及服务器的性能。

测试环境				
计划设计		执行监控		结果分析
用户信息	网段IP	连接状态	执行状态	Action
nbc-b-3	8.99.9.217	Yes	Busy	
nbc-b-e102624408	8.99.9.121	Yes	Idle	
nbc-b-e102624408	8.99.9.124	Yes	Idle	

2.3.6 前端代码抓取后和 W3C 比对平台展示效果

运行结果
 W3C标准分析
 图像分析比对...

W3C只针对运行结束的用例进行分析
 刷新结果
 生成任务

用例	步骤	错误	告警	URL
企业网银投资理财	账户信息查询主页面	0	8	<ul style="list-style-type: none"> http://12.99.102.196:9080/corporbank/CB0402_queryAccountInfor.do?EMP_SID=DOEPIaccountNo=60010122000402627&tStep=2 源文件 结果
企业网银投资	开立指令明细浮出层	0	7	<ul style="list-style-type: none"> http://12.99.102.196:9080/corporbank/CB0402_queryOpenCommand.do?EMP_SID=DOorderFlowNo=SG0000355202&tStep=1 源文件

W3C 比对结果 csv 文件内容:

序号	行	列	类型	内容
1	2	1	Warning	<meta> element not empty or not closed
2	2	51	Warning	<link> element not empty or not closed
3	3	1	Warning	<title> isn't allowed in <body> elements
4	44	33	Warning	<input> element not empty or not closed
5	45	33	Warning	<input> element not empty or not closed
6	2	235	Warning	<object> proprietary attribute "size"
7	2	373	Warning	<object> proprietary attribute "size"
8	13	17	Warning	<table> lacks "summary" attribute

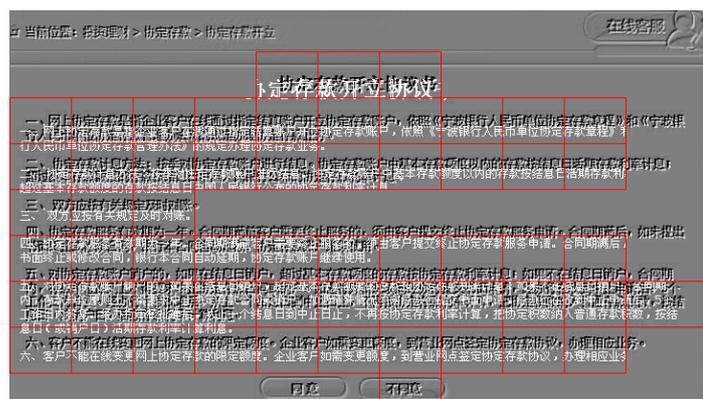
2.3.7 页面截图比对结果在平台的展示

运行结果
W3C标准分析
图像分析比对...

兼容性图像比对分析
刷新结果
生成任务

用例	步骤	环境	相似度	URL
企业网银投资理财	同意协议	Win8 IE(10) 1024*768	0.898	<ul style="list-style-type: none"> • http://12.99.102.196:9080/corporbank/logonJump.do 📎 原图 📎 标准图 📎 叠加图
企业网银投资理财	取指令明细浮出层	Win8 IE(10) 1024*768	0.926	<ul style="list-style-type: none"> • http://12.99.102.196:9080/corporbank/logonJump.do 📎 原图 📎 标准图 📎 叠加图

图片比对叠加效果:



三：实施过程中的团队合作+客户协调

1: 团队协调

现场实施一共 4 个人，老高和海涛负责个人网银，我和老胡在鼓捣企业网银。

作为现场的负责人，其实对于管理之道，我个人的想法是重在理，而不是管。

所谓“管”——制定赏罚措施，以身作则。

“理”是管理人员最重要的技能。管理者作为一团队的带头人，不管在技术，责任，沟通，协调，合作，态度等等软硬实力上都需要有拿得出手的东西。

好在我们4个IT男年龄相仿，爱好类似(要不是dota，要不是篮球K歌大餐)，在团队内部的沟通团结上我们是没有任何问题滴~哈哈~在此谢谢兄弟们的配合和努力！

对团队内部技术和业务上的东西~我自己的想法是 Share everything with my partners! 不管是技术还是业务知识~只要我自己掌握的都会尽力在团队内部作分享，绝对不会在技术上和业务知识上有任何保留，我相信真诚地对待周围的人~教学相长~在项目中的成长是最快的！

2: 客户沟通

说实话，我不善于和女人打交道，尤其是不懂技术不懂管理死钻牛角尖的门外女汉子！

咱和她们谈技术，她们扯原理；咱和她们谈原理，她们表示不关心；咱和她们谈沟通，她们扯流程；咱和她们谈流程，她们表示遵从现状听候领导意见；咱跟进领导意见，她们催着赶紧干活别延期。只能说：在银行里混的，你若是在工作上打不好太极拳，那就别想混的好！

这一点，在这个项目中对后来的交付验收有不小的影响。

项目之初，本着为客户服务的心态，银行项目负责人提出的几乎所有要求，我都会一一答应：挡板开发，测试环境和测试数据维护，这些本来应由行发开发支持的事情占据了现场实施兄弟们的一部分时间，挡板开发把老胡搞的头大眼花(还好数据库备份对我来说是小 case，在此龇龇滴窃喜一把)。

项目里，让我最后悔的事儿是：由于我们的项目在中标时并未配备有成熟的兼容性自动化测试产品，实施总监在标书中写到的产品功能点相对来说比较宽泛，也点也是从商业上来考虑的，后期可以优化或者继续拉新项目。但是，我错误滴答应行方的负责人给他们详细的描述了我对产品验收的理解，邮件发出，双发碰头，项目验收所谓的21条功能点，为后来项目的一再延期埋下了缕缕伏笔。

项目一直延期到 14 年的 6 月份，在后来和实施总监 Dong 哥的交流中，我才真正意识到自己的错误，是多么滴愚蠢！

不好意思了，兄弟们！这个项目的延期，哥们暂且记下一笔！

3: 周报

对这个项目中的各种报告，哥们不得不极力吐槽一把。

我个人认为一份项目进度周报告包括以下内容就完事儿了：

- 1: 投入的工时
- 2: 上周遗留问题解决和本周计划完成情况。
- 3: 下周计划安排。
- 4: 本周问题汇总，是否需要其他部门支持。
- 5: 项目总体情况，是否有延期风险。
- 6: 项目在该阶段中的风险和规避措施。

内容详实，格式不限，这是我的风格，可单单在周报和月报的格式上，我花费的时间是写报告内容的 N 倍(N>=5 呀，我嘞个去！)，印象最深刻的一次是写月报，光是格式就修改了 8 次之少！

终于体会到了国企体制中这种狗屁不通的报告格式文化！

唉~~懒得吐槽！只能用高端大气上档次，低调奢华有内涵的“呵呵”一词来表示自己对格式的看法了！

4: 借款报销和备案

兵马未动，粮草先行，是兵家之大事。

在这个项目中，哥们终于体会了一把粮草的重要性！兄弟们喝着西北风，别说干活了，TM 立即辞职的心都有了！

外包公司可能就是比较扯淡！据说 SM, WS, RT, DR, ZR 这几家外包巨头都差不多的。

兄弟们切记啦，珍惜银子，控制风险，远离外包！

5: IT 有风险，且码且珍惜

对外包工作的看法

1: 工作态度

性格决定命运，气度影响格局，态度改变未来。

这句话我个人一直铭记于心，也是我一直奉为圭臬的。生活就像一面镜子，你怎样对它，它就怎样对你。而工作又何尝不是呢？只是我们大多数人并没有把自己目前所从事的工作看作自己的事业，“混”日子是最常见的态度。以一名打工者的心态去面对工作，我觉的这样的心态是极其可悲的，你所作的一切都将影响日后你的资源和实力。

尽自己最大努力把当下的做完美，拥抱变化 快速学习新的知识并使之转换为实际工作中的生产力。我只想说：多年以后，我们肯定会感谢曾经如此努力拼搏的自己！

2: 工作中的责任心

外包其实是服务业的一种，但是我们大多数的服务员(IT 外包人员)有多少人去正确定位自己的服务价值呢？

我个人觉得不管是外包抑或非外包的正式成员，都应该勇敢去承担自己应尽的职责。

出问题勇于担当积极寻找问题的解决方案，从不推三阻四 拖拖拉拉。

做服务就应该有服务行业的样，品牌的力量是无从评估的！

未雨绸缪，在变化来临之前努力改变自己去适应变化，因为永远不变只有变化。

使用 L R 进行手机浏览器 get 请求压力测试的难点和总结

◆ 作者：搜狗测试 . 闫彩凤

关于 loadrunner 压力测试在知识库中已经有多篇文章。我这篇不再讲怎么使用 loadrunner 进行压力测试，而是总结一些实际测试中遇到的难点和问题，以及解决这些问题的思路。

手机浏览器美图笑话功能协议如下：

笑话协议：

```
pread.ie.sogou.com/infolist?count=20&lastindex=&b=%E7%AC%91%E8%AF%9D&mode=up&t=1445931068206&h=00000000-4379-c005-d445-0540728befc5&r=0000&v=4.1.0&hv=GT-I9500&pv=ANDROID4.2.2
```

美图协议：

```
pread.ie.sogou.com/infolist?count=20&lastindex=&b=%E7%BE%8E%E5%A5%B3&mode=up&t=1445932269394&h=00000000-4379-c005-d445-0540728befc5&r=0000&v=4.1.0&hv=GT-I9500&pv=ANDROID4.2.2
```

测试过程中的难点和解决方案、思路归纳如下：

首先是六个难点：

第一个难点：

由于是两个请求，如何测试两个请求同时存在情况下，服务端承受压力情况？即如何考虑两个请求的压力配比？

这个测试场景涉及到两个请求，为了模拟实际的使用环境，我们必须考虑到两个请求的配比情况，一般设计配比有两种情况：

一种情况是已经上线过这个功能，就像我这个版本，很久之前有过相关功能，只是这次做一个接口重构，所以有历史数据可以参照。

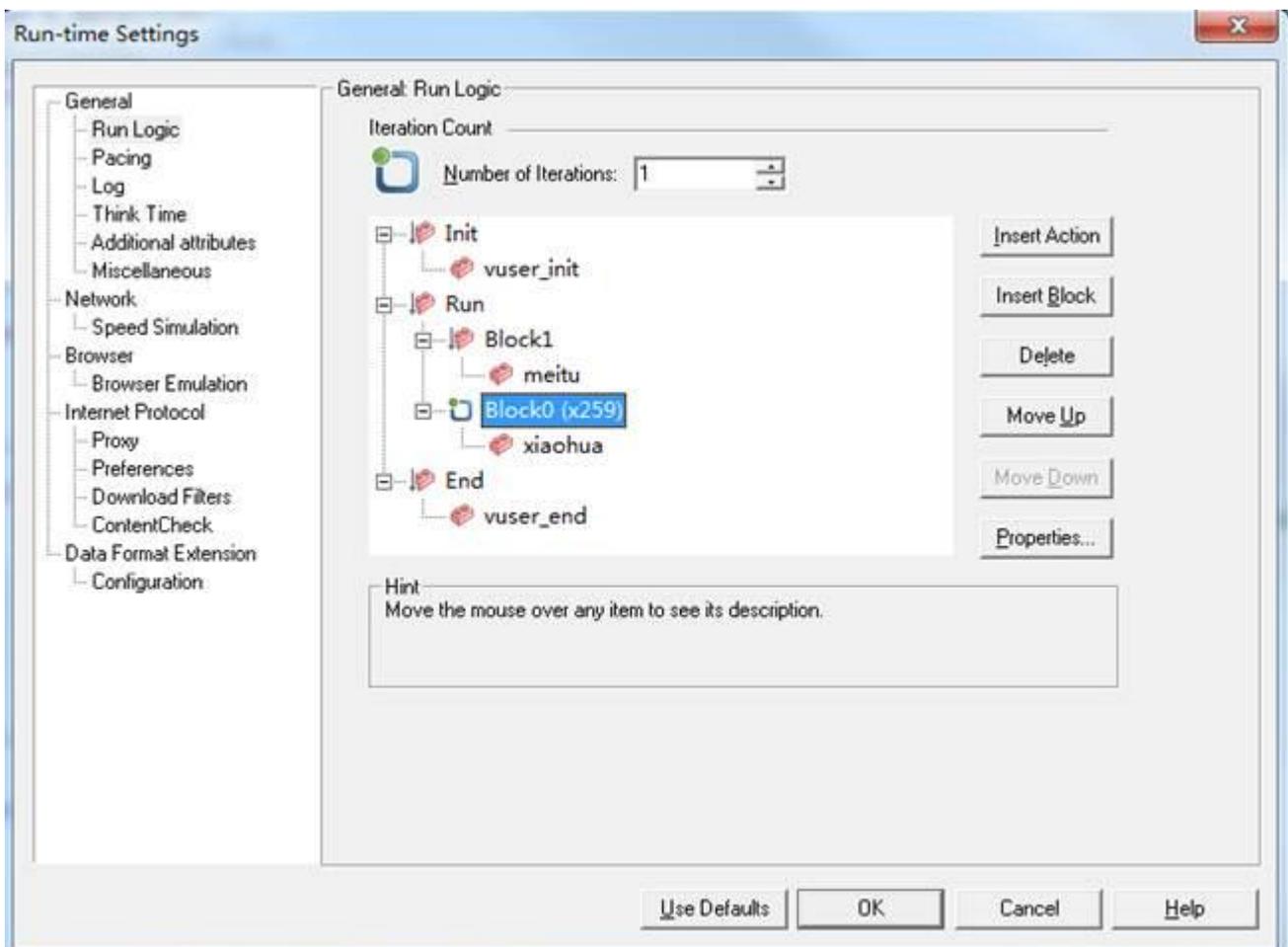
我这边主要是与产品要了笑话、美图的 UV 数据，分别利用笑话美图一段时间的 UV 和 PV 最大值，算出相应的比例，从而用这个数据设置为相应的打点比例。

另一种情况是从未上线过该内容，此时就需要依赖于一个小型灰度，灰度前要设置相应的请求 pingback 数据，从而能通过灰度数据计算出相应的比例。

第二个难点：

两个请求一起打压，如何设计脚本和添加到 loadrunner 中？

首先要在 Run logic 中设置好 Init、Run、End，以美图笑话为例具体设置截图如下：



我们从图中看出，笑话是美图的 259 倍。

脚本展示如下：



美图脚本:

```
meitu()
{
int itera_num, rand_num, i;
char StrTable[]="abcdefghijklmnopqrstuvwxy1234567890____";
char i_data[36]="";
//itera_num=rand()%16;
itera_num=36;
for(i=0;i<=itera_num;i++)
{
rand_num=rand()%39;
strncat(i_data, StrTable+rand_num, 1);
}
lr_save_string(i_data, "i_data_value");
web_reg_find("Text=index",
LAST);
lr_start_transaction("meitu");
web_url("xiahuameitu",
"URL=http:
//10.134.73.228/infolist?count=20&lastindex=&b=%E7%BE%8E%E5%A5%B3&mode=up&t=1468726249
158&h={i_data_value}&r=0000&v=4.1.0&hv=GT-I9500&pv=ANDROID4.2.2",
"Resource=0",
"RecContentType=text/html",
"Referer=",
"Snapshot=t15.inf",
"Mode=HTTP",
LAST);
lr_end_transaction("meitu", LR_AUTO);
```

```
return 0;  
}
```

脚本中需要注意的是:

(1) `i_data_value` 为一个随机数, 从而保证模拟到不同手机请求的结果, 也能防止服务端的缓存机制。

(2) `h={i_data_value}`, 其中大括号的意思为参数化, 一定要进行参数化, 否则 loadrunner 不认为这是一个变量。

参数化的方法: <http://www.jquery001.com/loadrunner-script-parameter-set.html>

第三个难点:

承受最大打点数通过 loadrunner 检测出来, 如何估算出能够承载的用户数?

通过请教 loadrunner 专家小曹和我的 leader 立人, 我了解到, 估算这个承载用户数有多种方法, 罗列如下 (以案例的形式):

1、如何建设一个能承受 500 万 PV/每天的网站吗?

计算模型:

每台服务器每秒处理请求的数量 = $((80\% * \text{总 PV 量}) / (24 \text{ 小时} * 60 \text{ 分} * 60 \text{ 秒} * 40\%)) / \text{服务器数量}$

其中关键的参数是 80%、40%。表示一天中有 80% 的请求发生在 40% 的时间内。24 小时的 40% 是 9.6 小时, 有 80% 的请求发生一天的 9.6 个小时当中 (很适合互联网的应用)。

$((80\% * 500 \text{ 万}) / (24 \text{ 小时} * 60 \text{ 分} * 60 \text{ 秒} * 40\%)) / 1 = 115.7 \text{ 个请求/秒}$

2、做过线上灰度, 有一个请求数与用户日活的比例, 可以直接计算得出。

本次笑话我算出最大承受打点数为 200QPS, 主要是利用第一种算法计算

$(200\text{QPS} * (24 * 40\%) * 3600) / 80\% = 864\text{w}$

鉴于线上的 2 台机器做负载均衡, 整体可以承受至少 400QPS 并保持性能稳定。

根据以上经验公式计算评估, 400QPS 可以支撑线上至少每天 1600W 用户的访问量。由于该服务端同时还要支撑 PC 端的 150w 用户, 故在笑话美图这边实际支持

1400w 用户的访问。

第四个难点:

测试中涉及到的是两个服务器，一个中转服务器，一个提供数据的服务器，如何同时进行两个服务端的压力测试？

两台服务端直接进行压力测试，根据响应结果，观察是哪台服务端成为瓶颈，当然，有些时候是带宽成为瓶颈，参考文章：<http://venus.sogou-inc.com/APPStore/Wiki/Pages/WikiArticles.html?id=5327>

第五个难点:

Loadrunner 只有打点端服务器的数据监控，如何监控被测服务端的参数情况？

需要在被测服务端上安装 nmon_analyser_34a 这个软件

这里有一个特别重要的命令:

`./nmon_x86_rhel52 -f -c 1`（执行次数）`-s 1`（时间间隔）的意思是形成分析表格，其中第一个 1 是表示执行次数，第二个 1 是表示记录数据间隔，如果是 1，就是代表时间间隔为 1s。

具体第一个 1 的数据是由总共执行的时间决定的，其中总共的时间是由 loadrunner 设置决定的，如果 loadrunner 总设定时间为 10 分钟，那么如果命令中时间间隔设置为 1s，那么执行次数应该就是 600 次。

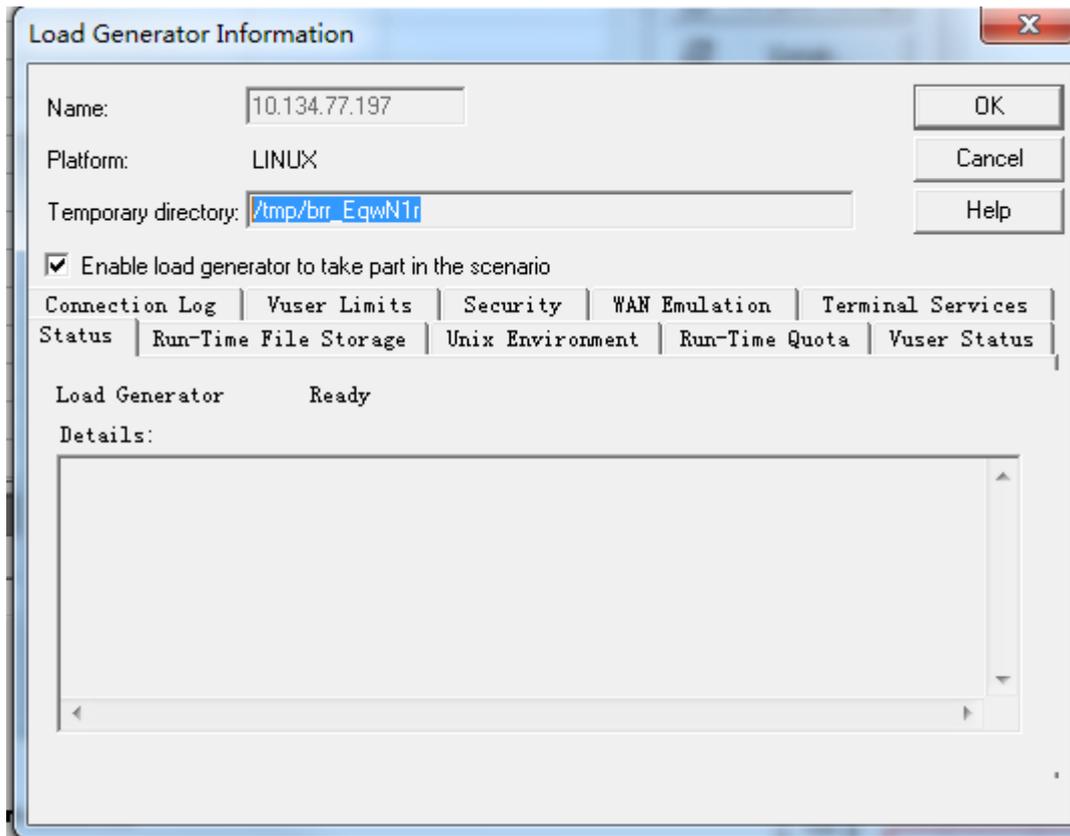
将结果形成表格后，再通过命令将该表格导出即可。

详细使用方法参考文档：<http://venus.sogou-inc.com/APPStore/Wiki/Pages/WikiArticles.html?id=5256>

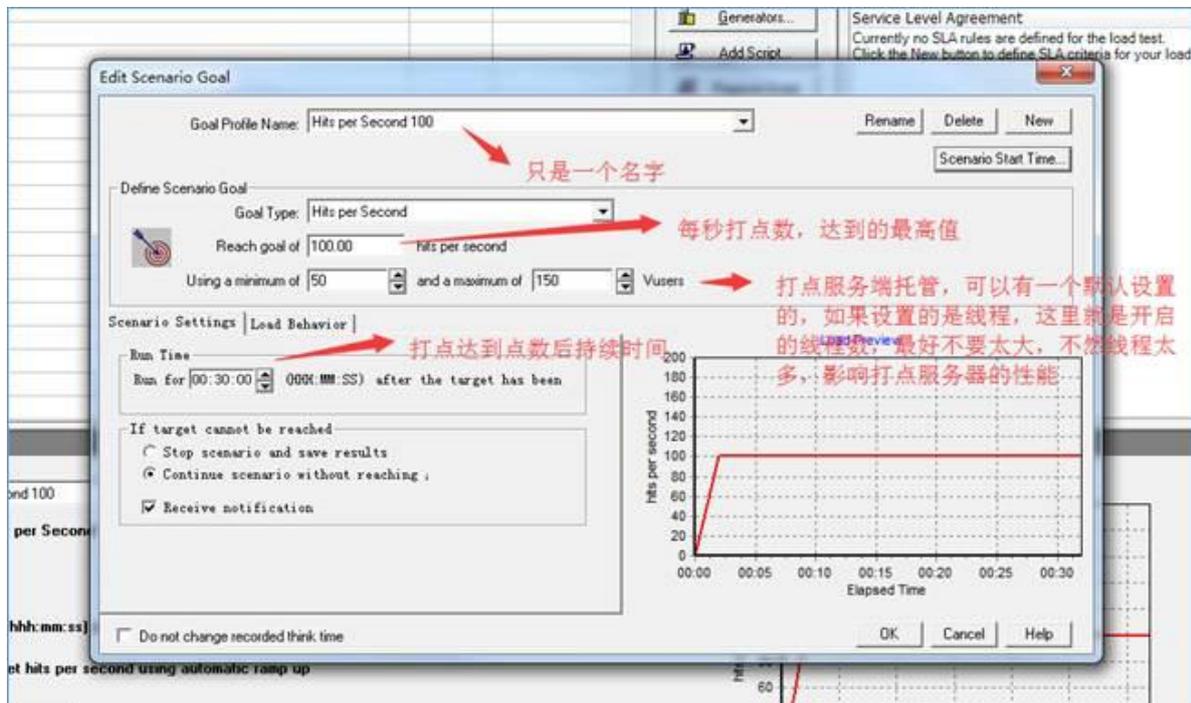
第六个难点:

最后打点过程中的一些参数和选项配置的细节问题。

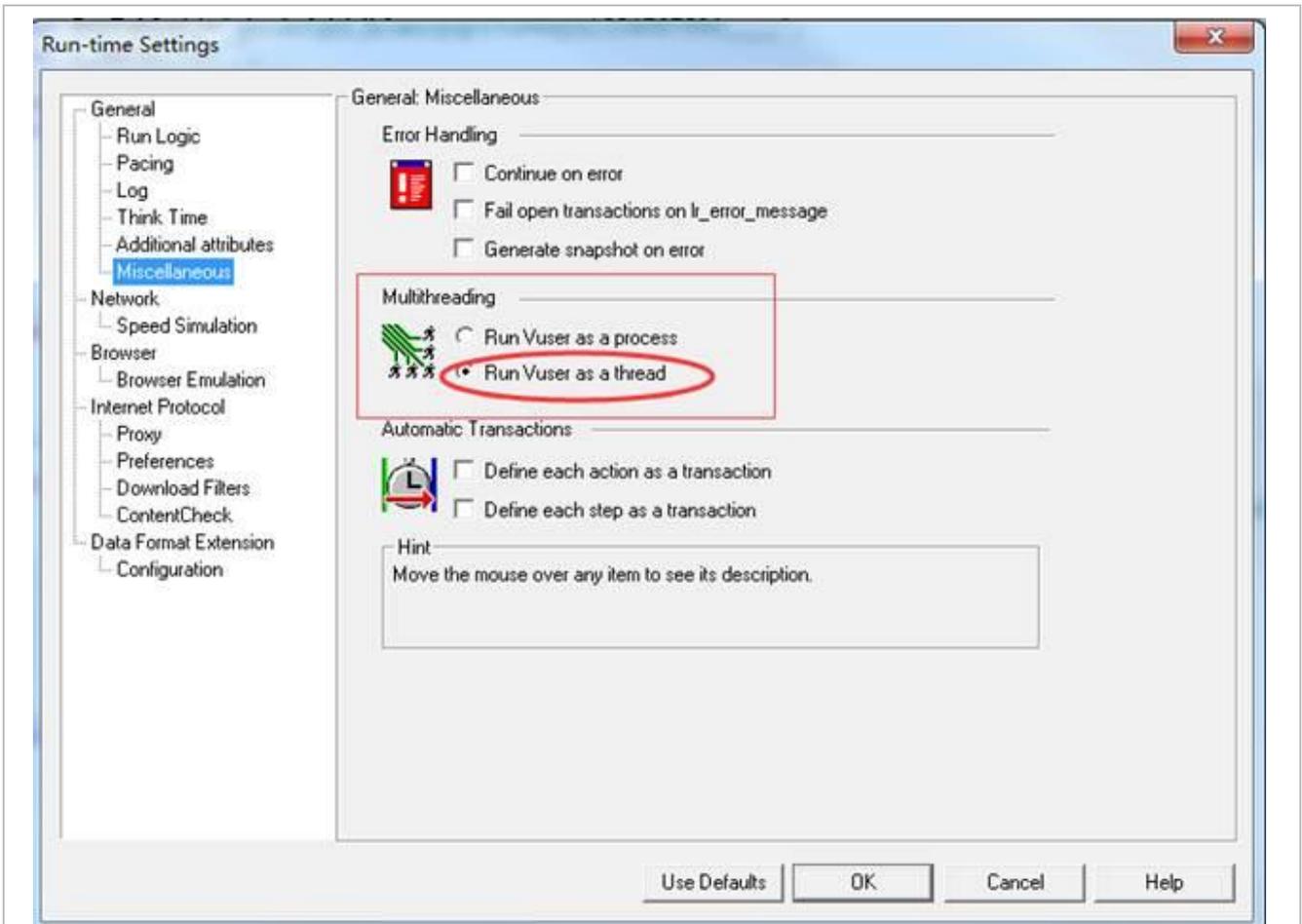
1、压力测试 log 存放位置：Temporary directory，截图如下：



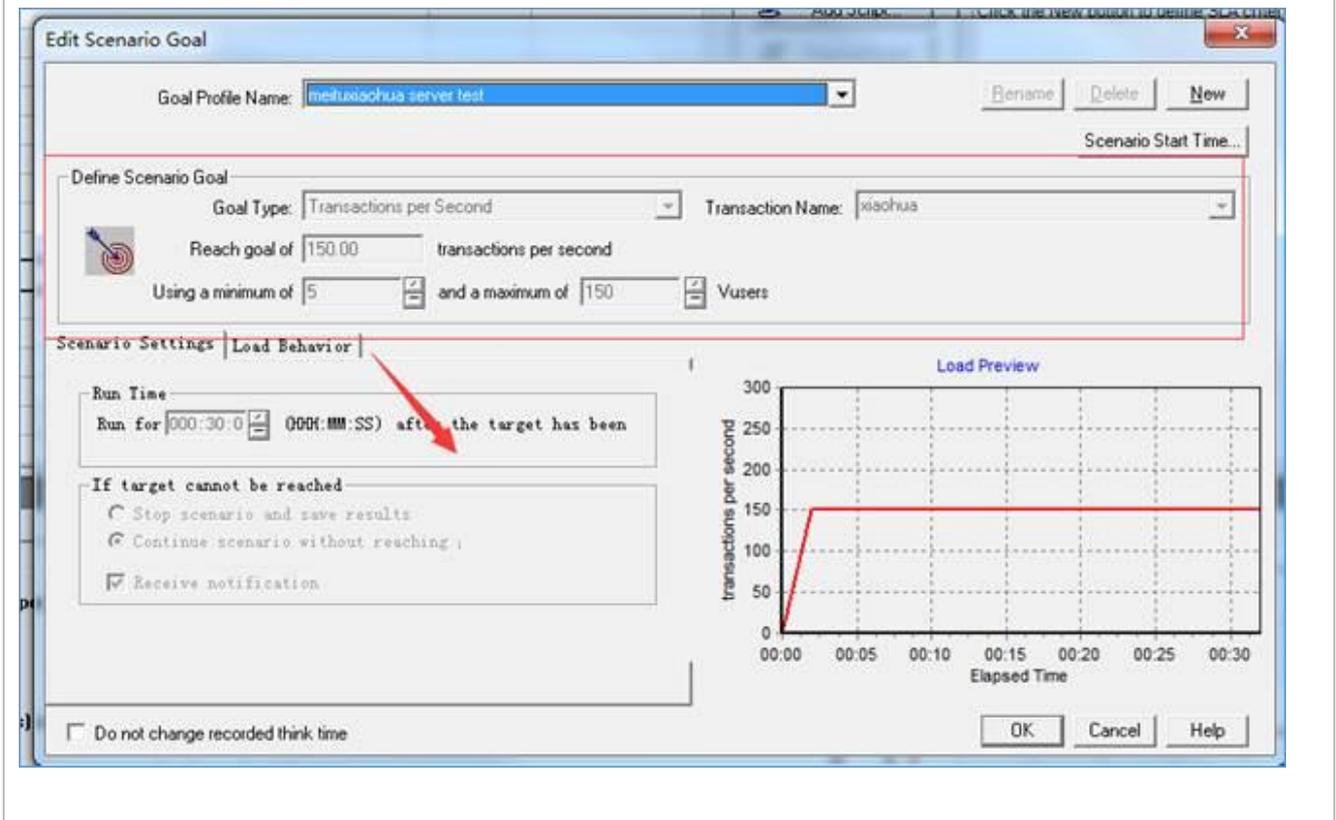
2、设置打点参数的细节:



3、线程托管一般选择 5-150Vusers，选择是线程还是什么的位置如下:

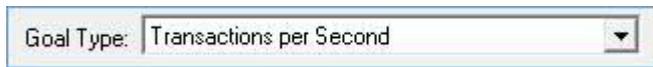


4、其他设置项:



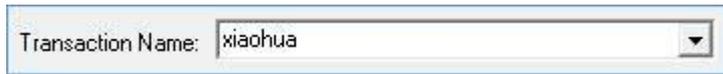
以上要注意这些设置项:

Goal Type 选择的是



Goal Type: Transactions per Second

Transaction 选择的是



Transaction Name: xiaohua

我这次测试中选择的是 xiaohua 这个时间的事务数。

两个总结:

第一个总结:

压力测试中到底是在关注什么?

- 1、CPU 占用: 比较稳定, 不要超过 70% 这个是平均到每一个核的
- 2、内存占用: 只要有空余内存即可, 一般不会成为瓶颈, 具体可以参考 Linux 内存管理知识
- 3、平均响应时间: 3、5、8 原则
- 4、带宽限制: 如果带宽成为限制, 就要与产品沟通, 在这种带宽下给出极限值, 要采取限流措施。
- 5、测试出最高的打压值, 之后, 需要降低打压值, 一直到服务端的 CPU 占用、内存占用、平均响应时间均为正常值情况。这时的指标值才是真正想要的值。

第二个总结:

如何写出一个压力测试的测试报告? 应该公示哪些内容?

一个基本的压力测试报告包含以下几个部分:

- 1、测试结论总结: 主要包含能够承受的访问量, 在这个访问量下内存占用情况。
- 2、具体测试结果:
 - (1) CPU 占用情况
 - (2) 每秒事务数

(3) Free 内存

(4) 平均事务响应时间

以上就是我整个压力测试过程的一些总结，希望对一些同学有所帮助，最后感谢整个测试过程 PC 浏览器 loadrunner 专家小曹的热情帮助。

Python 判断 Unicode 字符串类型

◆ 作者：搜狗测试 . 王玉

问题引出：

入口自动化脚本执行过程中，发现输入法的英文补全模块最近一直是失败的。（输入法正常候选长成这个样子：

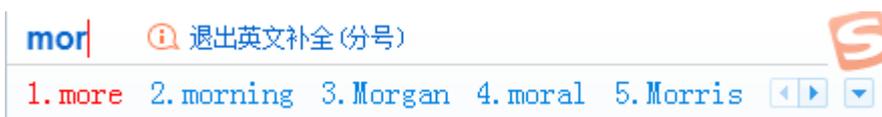


还有这个样子：



->点击更多英文补全或者快捷键

“分号+E”，进入英文补全模式后是这个样子：



问题分析：

检查脚本的步骤，判断是否正常进入英文补全模式，是通过上屏二选，判断上屏的文字与预期的文字是否一致。由于正常模式下，英文候选要么在首位，要么在五选，如果未能正常进入英文补全模式的话，上屏的二选必然是汉字，因此通过二选来判断还是很靠谱的~但是通过对比上屏的字符和预期的是否一致是否靠谱呢？

脚本运行相当一段时间内，其几乎没有失败过，但是最近却一直失败。尝试手动操

作，发现完全可以正常进入英文补全模式，那是哪里出错了呢？对比上屏的文字和预期结果的文字，发现候选的词序一旦发生变化，上屏的字符串和预期结果就不一样，脚本就会执行失败。

既然正常模式和英文补全模式的区别是候选中是否全部都是英文，那是是否可以通过判断上屏的字符是否是英文来判断呢？

解决方法：

定义一个函数，判断上屏的字符是否是英文字母，通过转码和编码区间来判断：

```
def is_alphabet(uchar): """判断一个 unicode 是否是英文字母"""
    if (uchar >= u'\u0041' and uchar <= u'\u005a') or (uchar >= u'\u0061' and uchar <= u'\u007a'):
        return True
    else:
        return False
```

.....

```
txt = ImeCommon.GetEditHandle()
print 'txt = ', txt
flag = is_alphabet(txt)
```

首次运行，成功！开心了一分钟后，意识到：上屏的是字符串，函数判断的是字符，函数取得是字符串的首个字符吧？肿么办？——>加循环嘛！

```
def checktxt(ss):
    flag = False
    for i in range(len(ss)):
        if is_alphabet(ss[i]):
            flag = True
    else:
        return False
    return flag
```

搞定！

加上其他几种字符类型的判断，备用：

```
def is_chinese(uchar):
    """判断一个 unicode 是否是汉字"""
    if uchar >= u'\u4e00' and uchar <= u'\u9fa5':
        return True
    else:
```

```
return False
```

```
def is_number(uchar):
```

```
    """判断一个 unicode 是否是数字"""
```

```
    if uchar >= u'\u0030' and uchar<=u'\u0039':
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def is_alphabet(uchar):
```

```
    """判断一个 unicode 是否是英文字母"""
```

```
    if (uchar >= u'\u0041' and uchar<=u'\u005a') or (uchar >= u'\u0061' and uchar<=u'\u007a'):
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def is_other(uchar):
```

```
    """判断是否非汉字，数字和英文字符"""
```

```
    if not (is_chinese(uchar) or is_number(uchar) or is_alphabet(uchar)):
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def B2Q(uchar):
```

```
    """半角转全角"""
```

```
    inside_code=ord(uchar)
```

```
    if inside_code<0x7e:
```

```
        #不是半角字符就返回原来的字符
```

```
        return uchar
```

```
    if inside_code==0x0020:
```

```
        #除了空格其他的全角半角的公式为：半角=全角
```

```
        -0xfee0 inside_code=0x3000
```

```
    else:
```

```
inside_code+=0xffe0 return unichr(inside_code)

def Q2B(uchar):
    """全角转半角"""
    inside_code=ord(uchar)
    if inside_code==0x3000:
        inside_code=0x0020
    else:    inside_code-=0xffe0
    if inside_code<0x7e:
        #转完之后不是半角字符返回原来的字符
        return uchar
    return unichr(inside_code)

def stringQ2B(ustring):
    """把字符串全角转半角"""
    return "".join([Q2B(uchar) for uchar in ustring])

def uniform(ustring):
    """格式化字符串，完成全角转半角，大写转小写的工作"""
    return stringQ2B(ustring).lower()

def string2List(ustring):
    """将 ustring 按照中文，字母，数字分开"""
    retList=[]
    utmp=[]
    for uchar in ustring:
        if is_other(uchar):
            if len(utmp)==0:
                continue
            else:
                retList.APPend("".join(utmp))
                utmp=[]
        else: utmp.APPend(uchar)
```

```
if len(utmp)!=0:
    retList.APPend("".join(utmp))

return retList

if __name__=="__main__":
    #test Q2B and B2Q
    for i in range(0x0020, 0x007F):
        print Q2B(B2Q(unichr(i))), B2Q(unichr(i))

    #test uniform
    ustring=u'中国人名 a 高频 A'
    ustring=uniform(ustring)
    ret=string2List(ustring)
```

“紫牛”方法论在测试团队中的应用

◆作者：琦少

“紫牛”是最近炒的挺火的一个词。关注互联网的朋友早已耳熟能详。通俗讲，可以用三个核心词来概括其特征：“差异化”，“极简目标”，“全力以赴”。今天想和大家探讨下这个曾指导“小米”发家，“猎豹移动”上市的方法论，如何在 QA 团队建设领域助力成功。

【差异化】

古人云：“随人作计终人后，自成一家是逼真”。何谓差异化？顾名思义，即是那些与众不同的事情，那么先来看看几种典型的反例：

- 1) 疯狂加班，杀到眼红，才可能会有一点点机会的事情。
- 2) 等待对手犯错误，才能弯道超车的事情。
- 3) 大多数团队（人）每天都在重复，并不能拿来作为“亮点”的事情。

包括但不限于以上几种情况，都是不妙的。为何？姑且抛开互联网的大圈子，聚焦于公司内部团队管理。相信有一个词，大家耳熟能详--“KPI”。大家一定问过自己一个问题“我怎样才能做的优秀？”或者“我怎么做的比别人好？”如果主管告诉你，要比别人努力，这等于没说。因为互联网不养闲人，这是个定律。如果主管仅告诉你“要拼工作强度，拼细致程度，拼技术实力”。你做了，最终的绩效结果却差强人意，那么我可以告诉你，这不是你的错。是你老大太平庸。

仔细想想不难发现，规范的用例设计，严谨的用例执行，甚至必要的加班赶项目进度。这些都是作为一个 QA 人员的基本要求。即便你做的再好，绩效无非是个达标。你老大有什么理由说你做的“优秀”呢？丧失了结果导向，大家只能靠拼加班来向 leader 证明你比别人做的更好更负责。如果你现在的处境是这样的，千万警惕四个字“虚度光阴”。

正如平庸的 leader 总喜欢让员工做些平庸的事情，久了就打造了一支平庸的团队。

我想这是谁都不愿意看到的。那么拒绝平庸，首先要做到差异化。去做那些"即使你的实力没那么强依然有可能成功的事情"。所谓："一个好的目标等于成功了一半"。分享一些差异化的思路，姑且算作抛砖引玉。

客户端负责功能测试的同学。建议建立起"量化的质量体系"。诸如需求评审，用例设计，用例执行，沟通反馈。这些大家张口就来的工作。自然是缺乏"亮点"。基于差异化的理论考虑，我们要做一件有意义且不平庸的事情。QA 是质量保证的中坚力量。别人希望我们回答一个问题"这款 APP 的质量如何？"并且希望回答是有理有据，且客观公正。回答这个问题就是一件很有意义的事情。但是很多团队，未必能迅速回答这个问题，或者回答起来太过于主观。这就又要说到我提到的"量化质量体系"。篇幅关系，简单来说（之后会有单独文章对此进行阐述），就是有一套打分机制，以一个迭代周期为单位，来客观的评估的从项目立项，到发布上线这一过程中，包括 APP 功能测试，适配情况，稳定性，反馈情况，以及项目流程监控等环节的综合质量情况。总分为 100 分，每项都逐级分配权重。体系运行后，不仅可以快速回答出综合质量情况。而且可以客观的评估质量短板之所在。通过不断的暴露问题，追溯问题，提高得分来不断的优化产品质量。通过一定时间的积累，还可以掌握 APP 质量的客观规律。比如在需求评审的时候，可以说基于量化数据，需求变更的程度会对 APP 质量造成多少影响。从而起到风险把控的作用。

至于性能测试，差异化在于，逐步形成某一领域的"标准化"。例如对于安全或者工具类软件。启动速度，垃圾扫描速度，垃圾清理速度，多快才算达标？电量消耗，cpu 消耗如何才算正常？这当然需要不断分析积累才有说服力。但更重要的是，你是否已经开始做了？

至于测试平台的建设，差异化在于，学会换位思考。因为你在做一款产品，那么怎么才能让用户喜欢买单？自然是突出性价比！那么平台的性价比在于，能让使用者通过尽可能简单的接入步骤，得到尽可能强大的功能（包括但不限于数据分析，可视化处理，智能控制，自动报告）。以此为出发点，才能做出有差异化的测试平台，不然只能是闭门造车。用之者甚少。

【极简目标】

这里只强调一点，如果一个目标用一个词甚至是一句话都不能描述清楚的话，这个目标必须重新思考。如果整个工程相对复杂，建议拆解目标。例如上面提到的"量化质

量体系”。可以拆解几个阶段的目标：

- 1) 定监控范围
- 2) 定打分准则
- 3) 人工运行
- 4) 体系自动化

每个阶段目标明确，让执行者明白自己要做什么。一口吃个胖子的情况不多。但是一个好的 leader 应该有很强的概括能力，或者目标拆解能力。如果只用一个目标来概括该体系。就是：通过测试过程中关注项的加权打分，量化的回答整体质量情况。

【全力以赴】

这个不必说太多，归根到底，事情还是要有人来做的。只想马儿跑得快，又让马儿不吃草是不可能的。认定的事情，要投入人去做，投入精英去做，即便不能有立竿见影的效果。也要敢于决策，聚焦来做。不放啰嗦几句，很多公司很多团队，之所以成就甚微。不见得是想不到，也未必是做不到。更多的是因为，有没有决心聚焦来做。大家都知道压力和压强的概念。我今天想用压力来比如团队的规模和基础实力。用压强来比如实际的效果和成就。就如压强等于压力除以受力面积。这个关键的受力面积就是你是否聚焦。这就是一个 leader 好坏的本质分别。这就是为啥一些百八十人的大团队反而成就有限，一些十几人的小团队往往可以业绩斐然的原因。

以上，阐述了，差异化，极简目标，全力以赴这三个核心词的含义。只是希望大家有所启发，有所思考。如果看完这篇文章，你可以开始做些真正有意义的事情，那就是，个人（发展）大幸，团队（成就）大幸，QA（职业化）大幸。

测试女巫之石头变宝石篇之四

◆ 作者：王平平

一、前言：

测试女巫系列陪伴着大家度过了 2015 年，转眼到了 2015 年年底了，到年底了，女巫要开始放大招了！今天教大家一个非常简单有用的方法，用于去怎么去说服那些有着十万个为什么的老板和客户。准备好了吗？我们要开始 2015 年最后一次奇幻之旅了~~~

这次我们学习新的“魔法”是什么呢？在我们实际工作中如何根据目前有限的测试数据去说服客户相信：这个数据能说明整个产品的状态呢？具体一点：例如我们生产某个产品，希望它的标准尺寸是 10mm；一共生产 100000 台这样的产品，我们抽样测试 100 台，测试它的平均尺寸为 9.99mm。我们可以拿着这个数据直接上报老板：老板没问题！因为我随机抽样都是好的，说明没问题。这样可以吗？

当然不可以，老板会准备 10 个以上的问题，把你问得体无完肤：你怎么保证这 100 台的测试数据能说明 100000 台的实际情况？你抽样 100 台的测试结果，对于你自己对这个产品的信心程度是多少？为什么会有这样的信心程度？你抽样的方法是什么？为什么要使用这样的抽样方法？

如果没有学习统计学的知识，可预计结果就是被狠做事太肤浅，不用心，此工作打回重做！好头疼是吗？别怕，测试女巫来了，带着魔法来了，带着魔法轻轻的来了，你学或不学，测试女巫就在这里，在这里等着你，成为做事深刻，走心的职场达人^_^。

此期我们集中火力，学习的魔法就是信赖区间，以及学习信赖区间所需要的背景知识，将我们即将学习的魔法罗列如下：

标准差和平均值的概念（此观念在第 39 期杂志上做了详细说明），趋中定律（此观念在第 39 期杂志上做了详细说明，这里主要需要抽样的方法），最后重中之重：信赖区

间的概念，以及介绍根据不同的需求有 3 种计算信赖区间的方法。

对于上述已经讲过的知识点，这里不再详细说明，亲如果忘记了，翻开对应期刊杂志看看吧，温故而知新，不亦乐乎嘛。

闲话不多说，我们开始正式学习吧！

二、6 sigma 常用工具基础知识介绍

1、标准差以及平均数的基本概念

[在第 39 期杂志](#)已经有详细的介绍，这里就不再赘述

2、趋中定理 (Central Limit Theorem)

[在第 39 期杂志](#)已经有详细的介绍，这里就不再赘述

这一期主要用到的概念是如何抽样，上一期讲的是概念，这里我用通俗的语言给大家总结如下：每一次抽样不是某一个具体的数据，而是多个数据(至少 30 个以上的数据)的平均值，抽样的次数并没有强制规定，当然越多，越好，如果受制于成本考量，可以只抽样 10 次。所谓的“样本”就是：10 个数据，但是要注意，每一个数据是 30 个数据的平均值。

只有这样抽样，才能得到这样的公式：样本的平均值等于母体的平均值；样本的标准差= σ/\sqrt{n} ，其中 σ 是母体的标准差。

综上：只要我们按照上述抽样方法进行抽样，就可以根据抽样的样本的平均数和标准差得到母体的平均数和标准差。

3、统计学推论统计的定义

1) 推论的两个对象：

a. 母体：是我们所要研究的问题中，所有研究对象组成多的组合。

b. 样本：是我们从母体中挑选出部分元素的一个子集合。

2) 推论的意义目的：

推论是通过样本中所包含的讯息对母体的一种说明。

3) 推论的类型：

a. 估计（点估计、信赖区间）

点估计：这种推论方法就是在前言中提到的被老板质疑的推论方法，这种方法不科学，有很大的漏洞，不建议使用。

信赖区间：会在下面详细说明。

b. 假设检定

在第 38 期杂志有较为详细的说明，有遗忘的童鞋可以翻看 38 期的杂志，温故而知新不亦乐乎^_^。

4、信赖区间(Confidence Intervals)

1) 定义：

对于未知的母体参数（平均数或者标准差）估计出一个区间，并指出该区间可以代表母体参数的信赖程度（有时也可以成为信心度）

a. 区间的分类

可以分为三类：

- 上界信赖区间 $(-\infty, b]$ 表示 $P(\mu \leq b) = 100 * (1 - \alpha) \%$
- 下界信赖区间 $[a, \infty)$ 表示 $P(a \leq \mu) = 100 * (1 - \alpha) \%$
- 双边信赖区间 $P(a \leq \mu \leq b) = 100 * (1 - \alpha) \%$

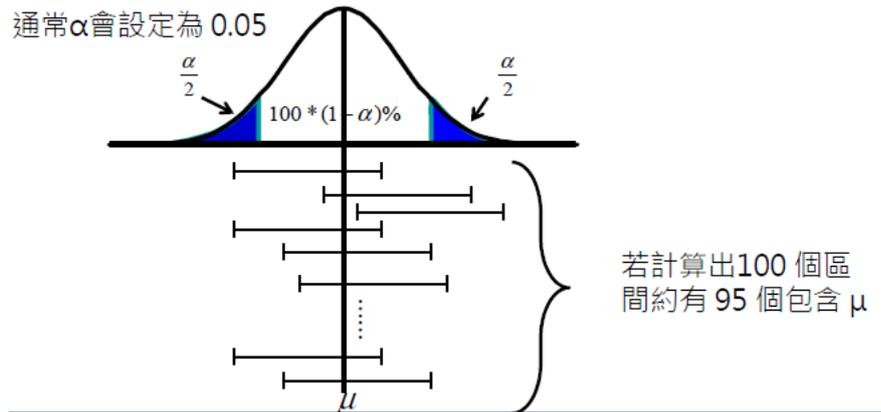
其中 μ 指的是母体的平均数；a, b 指的是预估区间的两个边界值；

b. 信赖程度（又称信心水准）

α 是一个固定的指标，对于工业上固定值为 0.05；这个数值在第 38 期杂志上讲解“假设检定”时讲解过此指标，也可以将其理解为型 I 错误，具体的内容可以翻看 [38 期杂志](#)。

对于信赖区间， $100 * (1 - \alpha) \%$ 可以理解为信心水准，表示我们有多少的信心，计算出的信赖区间包含母体真实的参数。它表示的意义

可以参看【图 1】



【图 1】

2) 根据推论母体参数的不同，选择 Minitab 不同的菜单

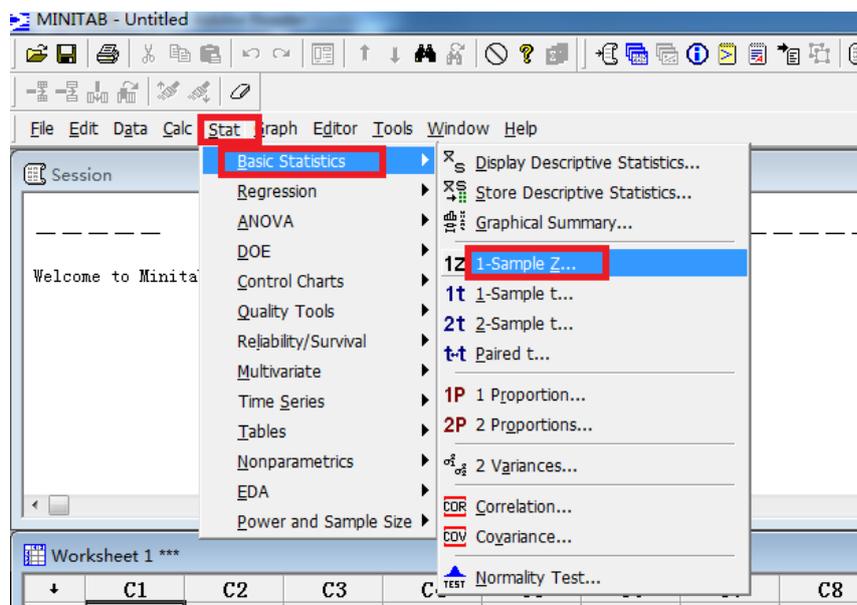
有三种母体参数：

a. 希望推论母体的平均数（实际运用中比较常见）

对于样本大于 30 的使用 Z 分配

例如：某个产品抽样 100 个样本，此样本的平均数为 390.3，标准差是 122，需要计算母体的平均数是多少，信心程度是 95%。

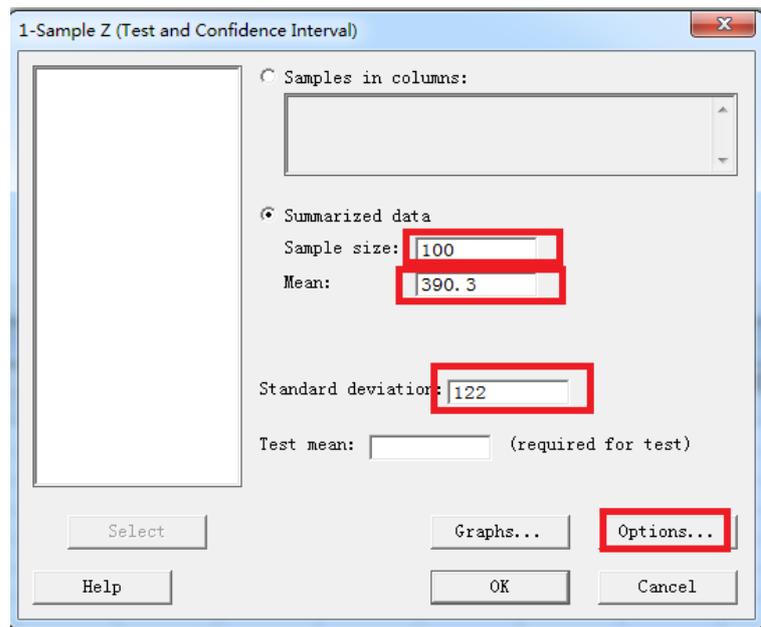
(1) 选择 Stat->basic statistic->1-Sample Z...如【图 2】



【图 2】

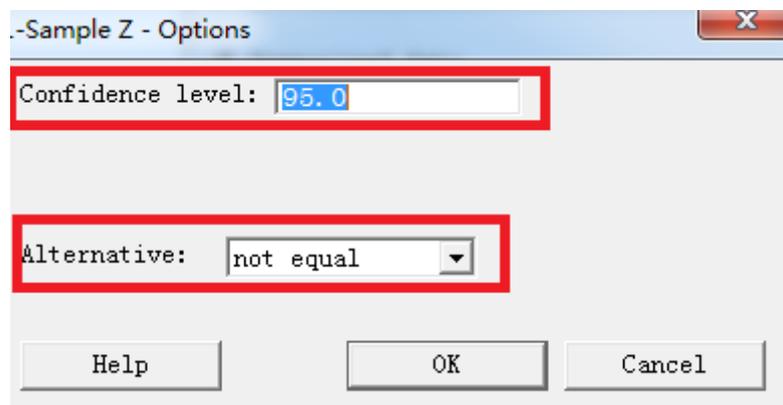
(2)选择 Summarized Date，输入样本大小，样本的平均值，标准差，点击 Option 如

【图 3】



【图 3】

(3)设置信心水准，默认是 95%，如【图 4】



【图 4】

(4) 点击 OK 后就可以看到结果，如【图 5】

即可以得到有 95%的信心程度：母体的平均数在 366.388 到 414.212 之间。

One-Sample Z

The assumed standard deviation = 122

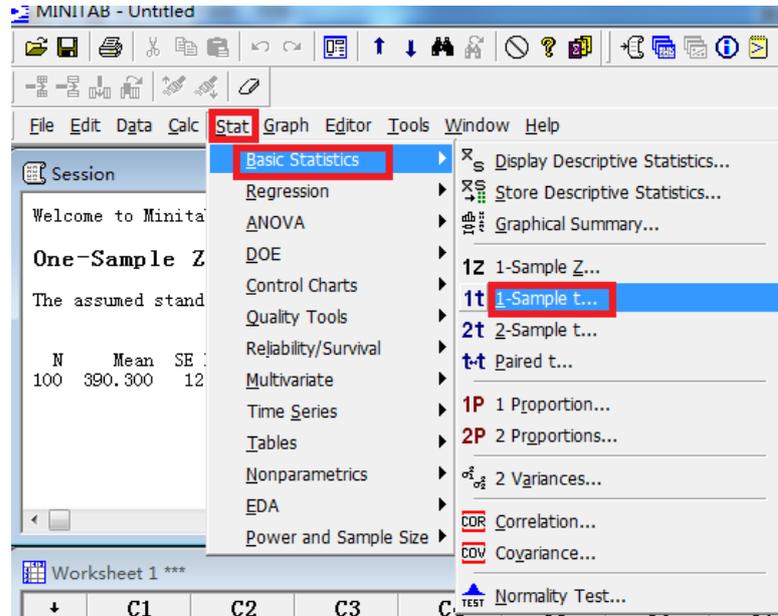
N	Mean	SE Mean	95% CI
100	390.300	12.200	(366.388, 414.212)

【图 5】

对于样本小于 30 的使用 T 分配

还是用上面的例子，样本大小为 10，其它都一样

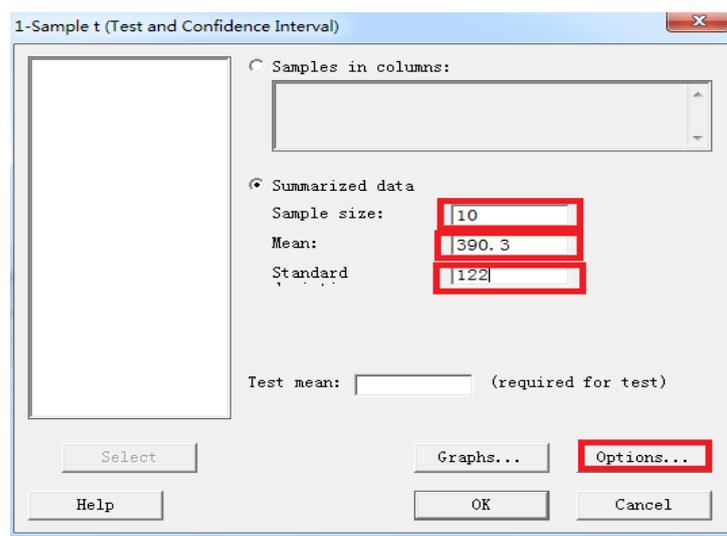
(1)选择 Stat->Basic Statistics->1-Sample t...如【图 6】



【图 6】

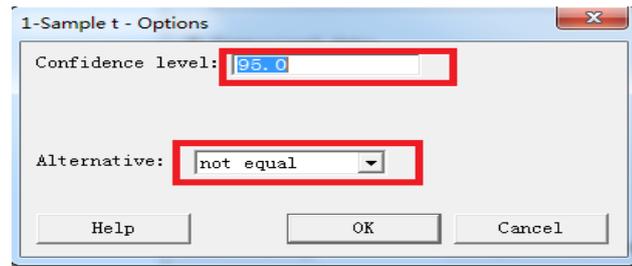
(2)输入 Sample size: 10; 样本平均数 390.2; 样本标准差: 122

如【图 7】



【图 7】

(3)点击 Option，输入信心水准，注意 Not equal 是双边信赖区间



【图 8】

(4)点击【图 8】【图 7】的 OK 后，会出现结果如【图 9】

对比 One Sample Z 检定的结果就可以发现：样本的数量越多，得到的结果越精确：

有 30 个 sample 的，使用 One sample Z 检定的结果：366-414

有 10 个 Sample 的，使用 One sample T 检定的结果：303-477

所以，大家一定要记住：一分耕耘一分收获，抽样数据还是尽量多。这样得到结果才会比较精准!!

One-Sample T

N	Mean	StDev	SE Mean	95% CI
10	390.300	122.000	38.580	(303.026, 477.574)

【图 9】

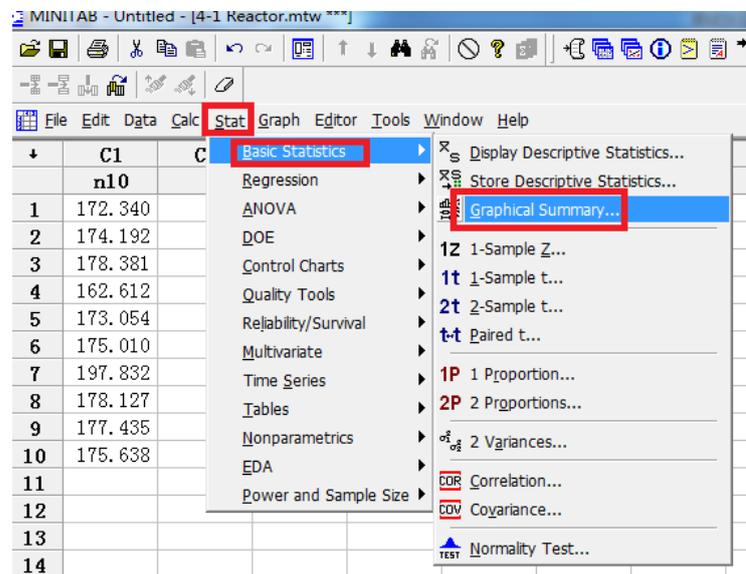
b. 希望推论母体的标准差（希望知道母体数据变异程度）使用 χ^2 分配

例子：有如【图 10】数据：即 sample 有 10 个，因为已经有了数据所以可以计算这组数据的平均数和标准差

	C1	C2
	n10	
1	172.340	
2	174.192	
3	178.381	
4	162.612	
5	173.054	
6	175.010	
7	197.832	
8	178.127	
9	177.435	
10	175.638	

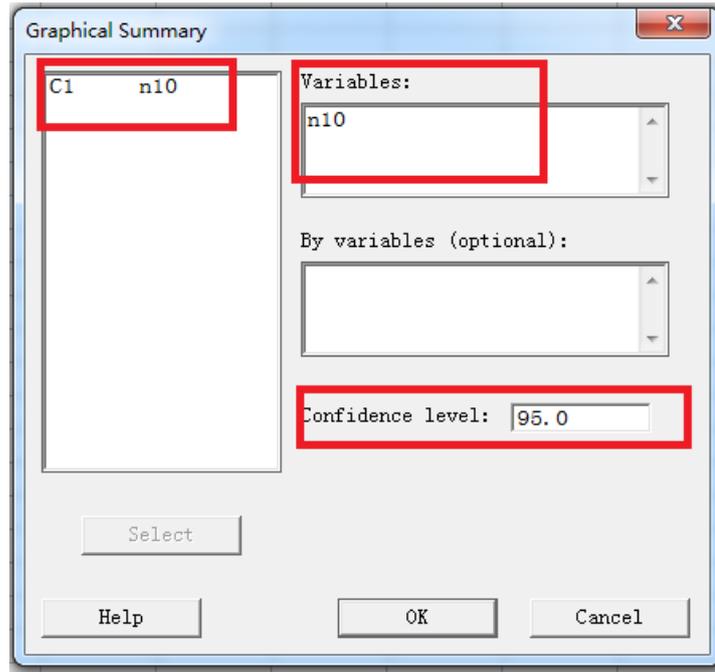
【图 11】

(1)选择 Stat->Basic Statistics->Graphical Summary, 如【图 12】



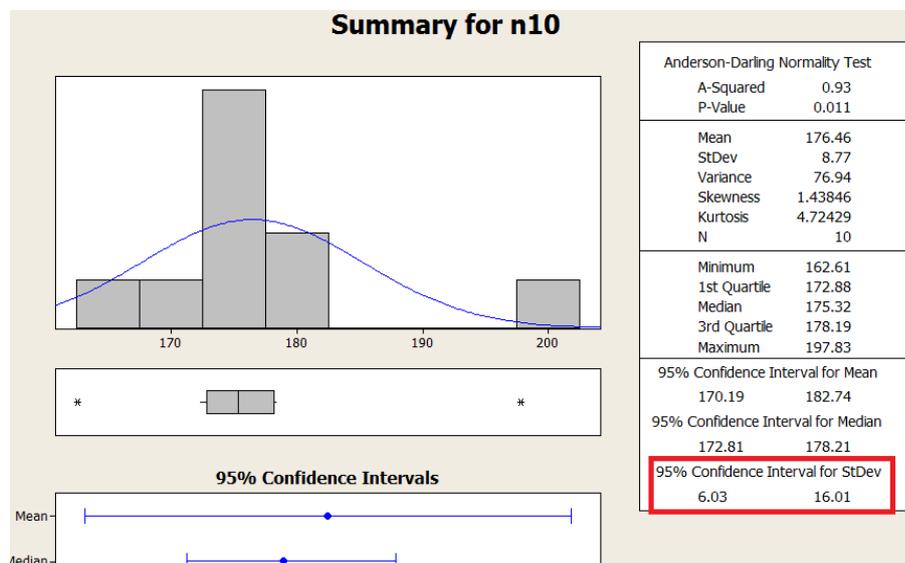
【图 12】

(2)选择变量，并设置信心水准，默认是 95%如【图 13】



【图 13】

(3) 点击 OK 后出现【14】可以看到母体标准差如红框标出

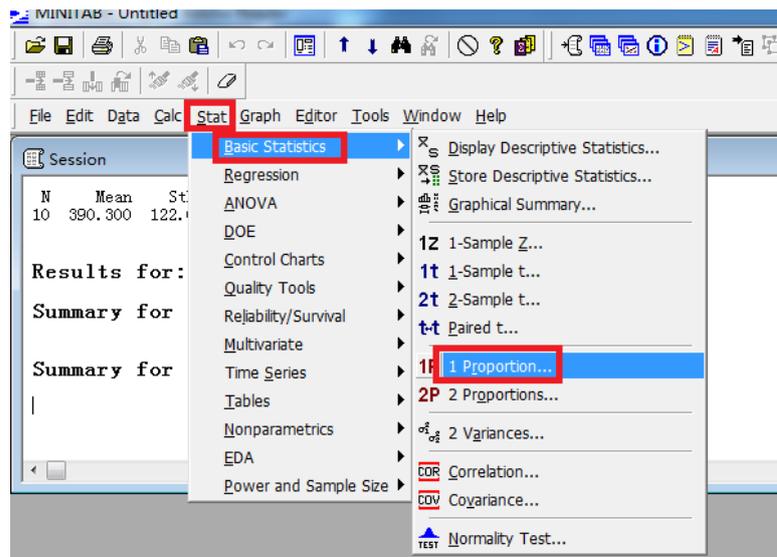


【图 14】

c. 希望推论母体的百分比即比率，使用二项式分配

例如：抽样 200 个样本，发现有 5 个有问题，计算信心水准是 95% 的母体的信赖区间

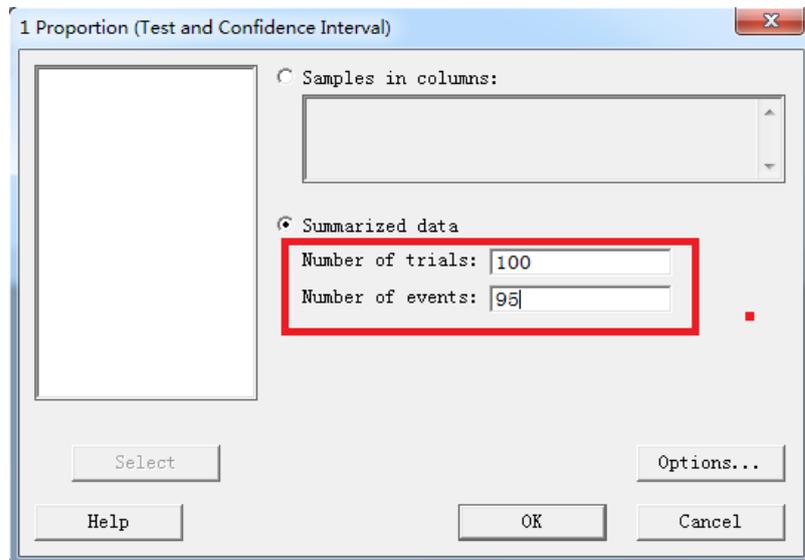
(1) 选择 Stat->Basic Statistics->1 Proportion... 如【图 15】



【图 15】

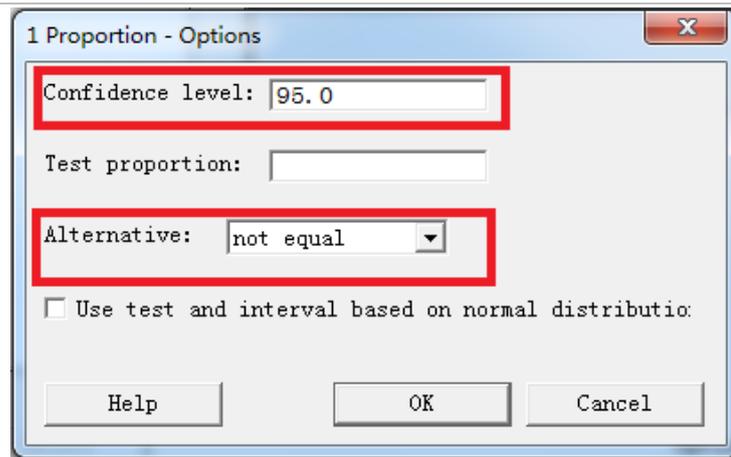
(2) 选择 Summarized Data->Number of trial 输入抽样数量

Number of events 输入测试 Pass 的次数，如【图 16】



【图 16】

(3) 点击【图 16】中的 Option，出现【图 17】，设置信心水准为 95%，信赖区间是双边，所以选择 Not equal



【图 17】

(4) 点击【图 17】和【图 16】的 OK 按键出现【图 18】

即测试结果，即有 95% 的信心成功比率是此区间：88.72%-98.36%。

Test and CI for One Proportion

Test of $p = 0.5$ vs $p \text{ not } = 0.5$

Sample	X	N	Sample p	95% CI	Exact P-Value
1	95	100	0.950000	(0.887165, 0.983568)	0.000

Summary for n10

【图 18】

应用到实际工作之根据送样测试结果进行推论

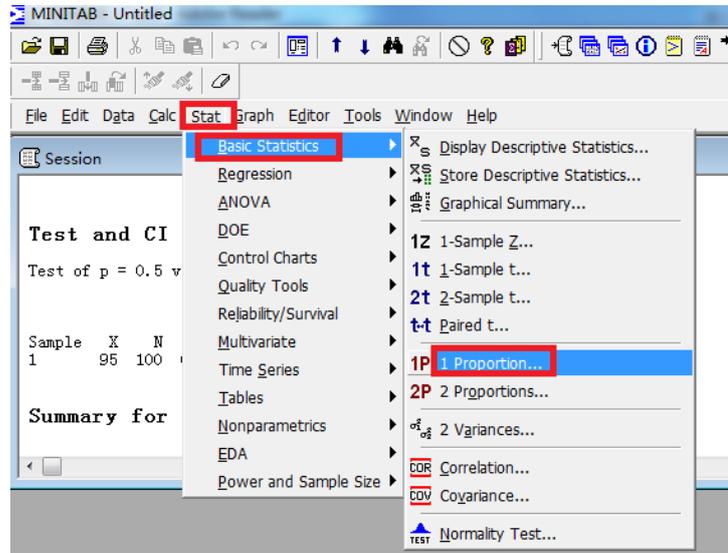
对于生产终端的通讯公司在实际工作中会有这样的事情发生：在项目每一个阶段会需要给客户送样的需求。例如某一个阶段试量产 1000 台，客户需要 100 台。我们如何根据这抽取的 100 台状态，推论出这 1000 台的产品质量呢？对于主管品质的 Q 部门，我们该如何做？该如何产生一个经得起老板和客户质疑的结论？

分析步骤

1) 首先我们先确定对于此产品判断成功和失败的标准，已经确定好标准后就可以开始抽样了：

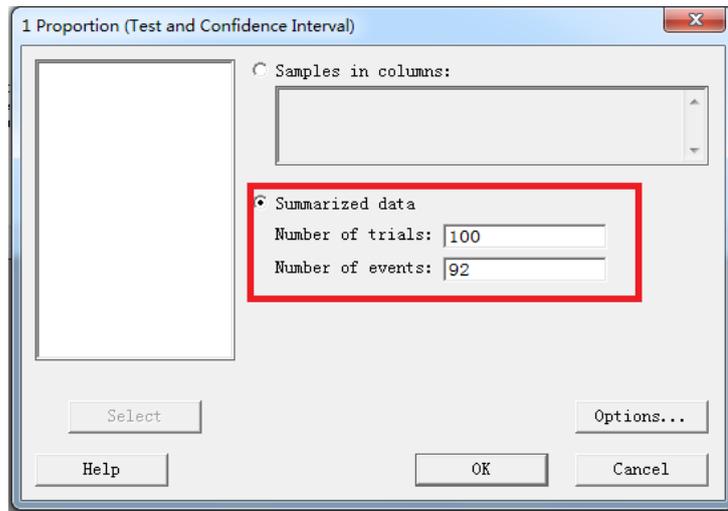
例如抽样 100 台产品，发现 8 台有缺陷，需要计算这 1000 台产品的合格品比率的 95% 的信赖区间。

2) 使用 Minitab 中的 1 Proportion 此工具如【图 19】



【图 19】

输入相应的数值如【图 20】



【图 20】

3) 结论如【图 21】: 从此 100 台测试结果来推论此 1000 台产品: 有 95%的把握其合格率为 84.8%到 96.5%,

Test and CI for One Proportion

Test of $p = 0.5$ vs $p \text{ not} = 0.5$

Sample	X	N	Sample p	95% CI	Exact P-Value
1	92	100	0.920000	(0.848442, 0.964828)	0.000

【图 21】

根据结论我们就可以告诉老板，根据这次抽样测试结果：这批试量产的合格率在 84.8%-96.5% 之间，最差的合格率是 84.8%，最好的合格率是 96.5%

如果老板觉得这个范围太大，不足以说明问题，就要提高抽样的数据，例如如果将样本数改为 200，同样的数据得到【图 22】的结果。

Test and CI for One Proportion

Test of $p = 0.5$ vs $p \text{ not} = 0.5$

Sample	X	N	Sample p	95% CI	Exact P-Value
1	192	200	0.960000	(0.922708, 0.982575)	0.000

【图 22】

总结方法

1) 如果确认我们需要推论母体数据的合格率。

首先我们要确认判断抽样样本测试结果是 Pass 还是 Fail 的标准。

2) 进行抽样，并根据第一步的标准得出每次抽样的测试结果并记录。

3) 使用 Minitab 中的 1 Proportion 根据样本的数据(需要输入一共抽样的数目和测试结果为 Pass 的数目)。

4) 得出未知母体的合格率的信赖区间。

应用到实际工作之关键指标的专业分析

对于路由器这一类的产品，网络的吞吐量是非常重要的指标，我们在向客户或者老板汇报此项的测试结果时如何更好的展现我们的专业性呢？（当然对于其它类型的产品，确认对于此类产品的关键性指标，然后再按照这个思路进行分析）。

分析步骤

1) 抽样方法

如果是使用工具测试，不需要花费太多人力，建议测试每次数据严格按照统计学的抽样方法：每一次的数据是 30 个数据的平均值，要有 30 个以上的这样的数据。如

【图 23】

Throughput
57.9283
54.4820
53.1512
43.2619
49.3745
48.7820
50.1388
51.2045
57.7894
61.1255
50.1885
53.3328
59.3882
51.5822
60.2248
66.7055
47.5869
46.1350
54.6304
59.9724
41.7097
56.9466

【图 23】

2) 对第一步严格抽样的数据使用 1 sample Z 检定

One-Sample Z

The assumed standard deviation = 5.99

N	Mean	SE Mean	95% CI
30	53.4500	1.0936	(51.3065, 55.5935)

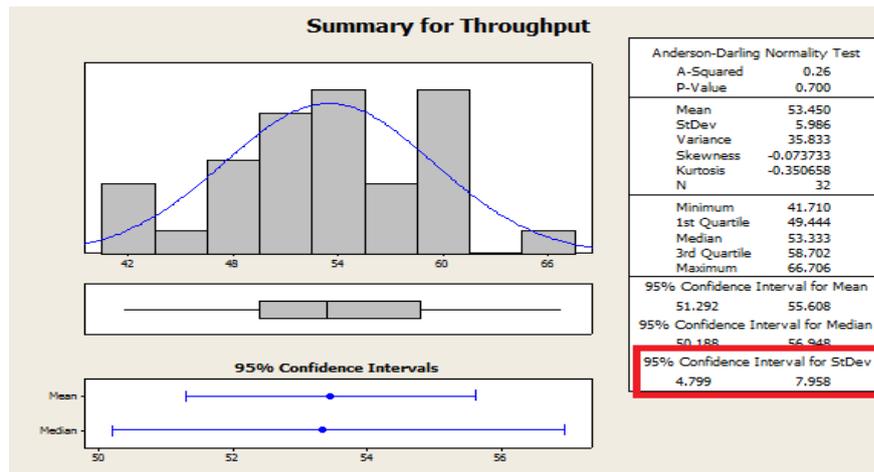
【图 24】

3) 结论

综上所述我们得到结果如上面【图 24】，即有 95% 的信心水准其 Throughput 的测试信赖区间为【51.30-55.59】：翻译为大家能理解的说法是：我们根据目前的测试结果，可以得出对于此产品“网络吞吐量”此指标，有 95% 的信心程度测试数据在【51.30-55.59】之间。

4) 扩大战果

利用现有的 Sample 数据计算母体标准差的信赖区间，如【图 25】得到 95% 的信心水中其 Throughput 的标准差信赖区间为【4.799-7.959】：翻译为大家能理解的说法是：我们根据目前的测试结果，可以得出对于此产品“网络吞吐量”此指标的变化区域是：【4.799-7.959】。



【图 25】

总结方法

1) 因为搜集样本的方法还是比较严苛的，先确认这个功能适不适合以及值不值得这样做即：这个功能的测试数据是否需要花很多人力？测试的结果产生的结论会有多大影响？如果测试数据不需要花很多人力，测试结果产生的结论非常重要，例如根据此测试结果需要给客户一定的承诺，就需要使用此方法。

2) 搜集之前项目有关这个标准的数据，这个数据就是样本，所以至少要搜集 30 笔资料

3) 使用 Minitab 1 sample Z 检定，得出相应的结论，注意要把统计学的结论尽量口语化得转成非统计学人士也能听得懂的结论。

总结

我们在这一期中主要介绍了统计学中推论学上的重要的知识点：信赖区间

计算信赖区间有三个类型：第一是计算平均数的信赖区间；第二是计算标准差的信赖区间；第三是比率的信赖区间。可以根据实际工作的需要选择使用哪一个计算方式。

第一个是使用最多最常见的类型：根据抽样的样本计算情况，推论母体数据的平均值区间，注意需要知道样本的平均数，标准差，以及样本大小。

第二个是使用不多：根据抽样的样本计算情况，推论母体数据的标准差的区间，但是还是建议大家使用这个方法，因为一旦收集了样本的数据，在把平均值区间算出来后，计算这个数值几乎不需要花人力物力；且这个数值非常有价值：告诉老板或者客户，这个母体数据的变异范围是什么。因为在第 39 期杂志给大家讲解了，如果需要全面了解一组数据，不仅要了解其平均数，也要了解其标准差。

第三个是使用也比较多：根据抽样数据的合格率去推估母体数据的合格率的区间

注意：在使用这三个计算方式前，都要先制定好：你的信心水准是什么：即既然是推论，就具有一定的风险，有 95% 的信心程度的意义就是，你必须承担 5% 的风险。所以对于统计学来说：没有什么事情是 100% 的，都是存在一定的机率，换言之：你必须承担一定的风险，这一点与现实生活十分类似，没有什么事情是 100% 必然会发生的。

这次使用的工具也不困难，但是我还是要强调，“简单”的工具下隐藏了“不简单”的统计学原理，如果希望后续在大家自己的工作中灵活的运营这些工具，必须要理解这些统计学的原理。

还是那句话：对于 6 sigma 可以带来的奇妙旅程，我将不懈的坚持探索，怀着赤子之心去探索如何使用这些工具去改善我的工作，反之在不断的使用这些工具的过程中又大大加深了我对统计学原理的认识，所以“路漫漫其修远兮，吾将充满欢喜的上下而求索”！

参考文献：

1、有关标准差基本概念介绍的网站：http://baike.baidu.com/link?url=4v_m3jJhHUKllt11A0tRe_I-pVurV1tNNrztJ6_PbZf0Me5rJr-bxIdp4fRCdsrsSIRd-hSFBglCEyZCkunk6K

2、有关介绍统计过程控制的网站：

http://baike.baidu.com/link?url=M9BLOK736USXqiqCHa2uhW624zYJpD-UXRAotTwo9PyTn-AWZv3GDHofg2Gz_uOqC42RK1GoKdLf-Wr30fWJR0_sgWYIj_IItG-JSdB1Amm

如何做好软件验收测试工作

◆作者：linda

日常验收：

现实生活中的项目验收是公司乃至每个项目成员都想要的结果，一旦验收对公司来说就是，可以收验收阶段的款了，不需要再投入那么多人力到项目当中，项目终于可以告一段落，大家都可以轻松一下了。项目验收是一系列细致工作完成到位的结果，而不是某一点的成功或某个人能力就可以促成的事情。怎么做好验收工作，合理的收尾都是我们每个人必须要关注到的内容和工作的重中之重。

软件验收：

标准定义：验收测试是部署软件之前的最后一个测试操作。在软件产品完成了单元测试、集成测试和系统测试之后，产品发布之前所进行的软件测试活动。它是技术测试的最后一个阶段，也称为交付测试。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

验收测试原则：

在测试方法上，由于验收阶段的特殊性，一般以黑盒测试和配置复审为主，以自动化测试和特殊性能测试为辅，项目实施方会同最终用户在项目专家组的领导与协调下共同参与。

当然，作为一个大的综合性的信息化项目，验收测试一定要慎之又慎。参与人员务必要本着认真负责的态度。验收时必须下注意以下几个原则问题：一是验收测试始终要以双方确认的需求规格说明和技术合同为依据，确认各项需求是否得到满足，各项合同

条款是否得到贯彻执行。二是验收测试和单元测试、集成测试不同。它是以验证软件的正确性为主，而不是以发现软件错误为主。三是对验收测试中发现的软件错误要分级分类处理，直到通过验收为止。四是验收测试中的用例设计要综合全面，能以最少的时间在最大程度上确认软件的功能和性能是否满足要求。

通过综合测试之后，软件已完全组装起来，接口方面的错误也已排除，软件测试的最后一步--验收测试即可开始。验收测试应检查软件能否按合同要求进行工作，即是否满足软件需求说明书中的确认标准。

常用策略:

实施验收测试的常用策略有三种，

它们分别是：正式验收、非正式验收测试、Beta 测试

我们选择的测试策略通常都是建立在合同之上和现场工作具体情况而定

今在此举例 Beta 测试为讲解和测试说明

Beta 测试需要的控制是最少的。在 Beta 测试中，采用的细节多少、数据和方法完全由各测试员决定。各测试员负责创建自己的环境、选择数据，并决定要研究的功能、特性或任务。各测试员负责确定自己对于系统当前状态的接受标准。

Beta 测试由最终用户实施，通常开发(或其他非最终用户)组织对其的管理很少或不进行管理。Beta 测试是所有验收测试策略中最主观的。

β 测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，Beta 测试不能由程序员或测试员完成。

当开发和测试根本完成时所做的测试，而最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其他人员完成，不能由程序员或测试员完成。

Beta 测试形式优缺点:

优点：测试由最终用户实施。大量的潜在测试资源。提高客户对参与人员的满意程度。与正式或非正式验收测试相比，可以发现更多由于主观原因造成的缺陷。

缺点：未对所有功能和/或特性进行测试。测试流程难以评测。终用户可能沿用系统工作的方式，并可能没有发现或没有报告缺陷。最终用户可能专注于比较新系统与遗留系统，而不是专注于查找缺陷。用于验收测试的资源不受项目的控制，并且可能受到压

缩。可接受性标准是未知的。您需要更多辅助性资源来管理 Beta 测试员。

测试的总体思路:

用户验收测试是软件开发结束后,用户对软件产品投入实际应用以前进行最后一次质量检验活动。它要回答开发的软件产品是否符合预期的各项要求,以及用户能否接受的问题。由于它不只是检验软件某个方面的质量,而是要进行全面的质量检验,并且要决定软件是否合格,因此验收测试是一项严格的正式测试活动。需要根据事先制订的计划,进行软件配置评审、功能测试、性能测试等多方面检测。

用户验收测试可以分为两个大的部分:软件配置审核和可执行程序测试,其大致顺序可分为:文档审核、源代码审核、配置脚本审核、测试程序或脚本审核、可执行程序测试。

要注意的是,在开发方将软件提交用户方进行验收测试之前,必须保证开发方本身已经对软件的各方面进行了足够的正式测试(当然,这里的"足够",本身是很难准确量化的)。

用户在按照合同接收并清点开发方的提交物时(包括以前已经提交的),要查看开发方提供的各种审核报告和测试报告内容是否齐全,再加上平时对开发方工作情况的了解,基本可以初步判断开发方是否已经进行了足够的正式测试。

用户验收测试的每一个相对独立的部分,都应该有目标(本步骤的目的)、启动标准(着手本步骤必须满足的条件)、活动(构成本步骤的具体活动)、完成标准(完成本步骤要满足的条件)和度量(应该收集的产品与过程数据)。在实际验收测试过程中,收集度量数据,不是一件容易的事情。

验收测试关注的内容:

1.安装测试

安装测试的目的在于验证软件能否在不同的配置情况下完成安装,并确认能否正常运行。

2.功能测试

功能测试亦即业务测试。是验收测试中的核心内容,不单单是软件功能的测试,同时也是对企业业务流程梳理的测试。

项目功能测试验收的主要内容应该是由系统中不同模块决定的，包括系统运行情况、业务处理情况、各种单据及报表出具情况，主要涉及财务模块、销售管理模块、库存管理模块、采购模块、生产计划模块等。在验收过程中，可以以部门为单位进行，便于集中精力来处理主要问题。

在库存管理方面，重点是物料收发的流程是否合理，物料收发的效率是否有所提高。各种单据是否按照要求及时处理。

在生产管理方面。主要是考核生产计划的执行率是否有所提高。生产效率是否有所提高。包括产能的最大发挥、生产排程的合理性、生产工艺的优化等。

3.界面测试

界面要符合现行标准和用户习惯。界面测试要从友好性、易操作性、美观性、布局合理性、分类科学性、标题描述准确性等方面入手。

4.性能测试

性能测试主要测试软件的运行速度和对资源的消耗。通过调整所依赖的软硬件配置、网络拓扑结构、工作站点数、数据量和服务请求数来测试软件的移植性、运行速率、稳定性和可靠性。一般借助 WinRunner 之类的企业级自动化测试工具来辅助测试，通过极限测试来分析评估软件性能。

5.文档测试

文档是软件的重要组成部分。也是软件质量保证和系统配置管理的重要内容。作为一个大规模软件，覆盖了企业的各种业务。它至少要具备需求定义、开发设计、测试评估、项目管理、用户应用等 5 类文档。文档测试主要通过评审的方式检查文档的完整性、准确性、一致性、可追溯性和可理解性。

6.其他测试

除了上述的测试外，还有必要对系统的其他特性和需求加以测试。

第一，负载压力测试。主要包括并发性能测试、疲劳强度测试、大数据量测试和速度测试。一般采用自动化技术分别在客户端、服务器端和网络上进行测试。用例设计时，要以真实的业务为依据，选择有代表性的、关键的业务操作作为测试对象。

第二，恢复测试。通过模拟硬件故障或故意造成软件出错，检测系统对数据的破坏

程度和可恢复的程度。

第三，安全性测试。通过非法登录、漏洞扫描、模拟攻击等方式检测系统的认证机制、加密机制、防病毒功能等安全防护策略的健壮性。

项目验收是对项目在整个实施阶段产生的效果的一个检验的过程，也是对项目在整个实施阶段的一个终结，它为系统在今后应用中的顺畅运行奠定了坚实的基础。

过程：

1. 软件需求分析：了解软件功能和性能要求、软硬件环境要求等，并特别要了解软件的质量要求和验收要求。

2. 编制《验收测试计划》和《项目验收准则》：根据软件需求和验收要求编制测试计划，制定需测试的测试项，制定测试策略及验收通过准则，并经过客户参与的计划评审。

3. 测试设计和测试用例设计：根据《验收测试计划》和《项目验收准则》编制测试用例，并经过评审。

4. 测试环境搭建：建立测试的硬件环境、软件环境等。（可在委托客户提供的环境中进行测试）

5. 测试实施：测试并记录测试结果。

6. 测试结果分析：根据验收通过准则分析测试结果，作出验收是否通过及测试评价。

7. 测试报告：根据测试结果编制缺陷报告和验收测试报告，并提交给客户。

注意事项：

验收测试业务洽谈 双方就测试项目及合同进行洽谈

相关流程图
签订测试合同
委托方提交测试样品及相关资料，委托方需提交的文档有：
基本文档：（验收测试必需的文档）用户手册 安装手册 维护手册 软件样品（可刻录在光盘）
·特殊文档：（根据测试内容不同，委托方所需提交下列相应的文档）
软件产品开发过程中的测试记录 软件产品源代码。
编制测试计划并通过评审进行项目相关知识培训
测试设计，评测中心编制测试方案和设计测试用例集。

方案评审：

评测中心测试组成员、委托方代表一起对测试方案进行评审。

结语：怎么做好验收测试是一个学问，也是测试验收人员对质量最后一关的把握，产品的交付和试用直接决定客户对产品的好坏程度的感知，还请大家不要放过任何一道质量把控的阶段。

我了解的的自动化

◆ 作者：邱王鼎

最近没事看了一些自动化的文章，有些地方我同意，有些地方有些看法。所以想投一下稿，第一次写文章，请见谅。本文章不涉及具体的操作，只是讲讲心得。

先说说我，我做原来是做自动化外包的，用的 QTP，做了差不多 5 年的测试。和我其他人解释自动化的时候，最直接的方式就是：你们看到的自动化就是按键精灵，不过他比按键精灵仔细，可以做我们想要做的东西，而且还可以出文档。说道 QTP 我想说的就是，录制是 QTP 的精华，不过要做自动化项目。只用录制是远远不够的。

为什么要引入自动化，网上有很多说法，具体我就不说了。我自己说一下我做过的自动化项目的结构，系统对大家有帮助。

我们一个自动化的包有这些：对象库、代码、函数库、数据、组装好的流程。如果项目经费充足的话，可以买一套框架。

对象库就是放对象库的，我们放对象库一般放得很少，如果是 web 一般就会放两次，浏览器+第一次页面。其他的都不会用到，如果第一次页面有变动量很大的话，可能对象库都不要用，都是用描述性编程。写的时候可能会难写，但是后期维护很好。

代码，重点说说代码。我看过有些人做的代码有一类是分得很细，会把每个控件都封装，交给业务人员来组装，我不同意这样的观点。我们一般封装是一个大的模块进行封装。例如：登陆模块，购物模块，支付方式模块、发货模块、退换货发起模块、审批模块等等，差不多就这几个模块，不要过于细分，细分可能会造成代码的冗余。

函数库，主要是放一些对控件特殊的处理，数据库处理，一些通用的逻辑处理。特别要说明的时候，我会把很多逻辑处理存放于函数库。让我的代码里面尽量少出现 if。

数据，就不用多说了，我的做法是，吧所有用到是数据都放在表格中，而且这边表

格数据一般情况下是不用修改的，如果是用户名之类的，没错都要修改的数据，我会放在 TXT 中，这种数据要尽量少一般不会多过 5 个数据，这些数据是每次运行脚本的时候都要修改的数据。

组装好的流程，也可以说成是案例。不过不像案例那要细。手工测试可能会把一次点击当成一个案例，自动化案例不需要那要细分，流程中就包含了这些，举例说购物的流程中包含这几个部分：一件商品的流程、多件商品流程、支付宝支付、自己的钱包支付、后台不出库、后台出库。把这几个部分做一个矩阵就有 8 条流程，足够覆盖购物流程。

前面都是铺垫，这一次重点说说代码。我们写代码一般都是通用性很强。例如：选择支付方式模块中有支付宝、自己的钱包、货到付款。这三种方式，应该吧三种方式都写着一个文件里面。而不是分开写，这样有助于后期的维护。只要通过输入的数据对他进行区分就好。例如:excel 中支付方式为：支付宝，我的脚步就会运行支付宝部分的脚本。

而且在脚本中会放入很多的校验和动态等待的脚本，感兴趣以后可以讲讲这部分。

现在很多人不喜欢这样做，他们喜欢把每个模块更加的细化，脚步有自动化人员进行编写。然后由业务人员进行组装。我不同意这样的做法，原来我们也做过这样的细分，不过效果不佳。业务人员只关心他们的业务，他们在来模块的时候，经常忘记某个模块，而且在填写数据的时候，如果没有下拉数据，他们很难在没有页面的情况下填写数据。我们的做法是自动化还是由专门的自动化人员进行操作。这样可能人力方面增加，不过这样做可以有效的利用自动化的模式对系统进行测试。而且做自动化的人都知道，自动化最麻烦的事情就是数据、控件、校验。特别是控件，我们在运行自动化的时候，遇到最多的就是找不到控件，而让业务人员进行组装，这种问题就更加多了。而且我碰到过很多业务人员，直接说，你们自动化操作比我们手工还麻烦。原来组装好的，系统有变动，我又要组装。这个就是业务人员组装造成的问题。

我来说说专门自动化测试人员的好处。对个人而言，编写自动化脚本本身就是对自己的一个提高。业务方面，你要比所有的人员都了解系统，不需要了解过细，但是起码每个流程你都必须了解，你必须了解每个流程的重要性，每个流程中那些校验点是必须校验的。就像手工测试一样。重要的校验点必须校验。还有就是代码的提高，自己编写脚本一次，比你看别人编写的脚本一百次都强。去面试的时候，你可以和别人说，我是

怎么细化整个系统，怎么样对他进行组装。对项目而且，自动化的案例都是从手工案例中提取。但是也必须通过项目的评审，保证案例的覆盖率。而且每次项目的修改，我们只需要修改代码，组装好的案例一般是不需要修改的，这样可以减少组装带来的麻烦。组装代码并不比编写代码简单，特别是数据，填写错误是常有的事情。

第一次写文章想写的东西很多，所以写得很乱，请见谅如果可以以后慢慢的细分每个部分都写下去。

敏捷测试的测试设计篇

◆译者：王宏瑜

增量测试设计

关于测试设计，国际软件测试资格委员会在去年追加的敏捷测试基础级别扩展教学大纲里敏捷测试实践篇里，有有趣的记述。

增量测试：测试用例和测试纲领是在用户故事和其他测试基础上逐步建立的，从简单变向复杂的测试。

下面列举了作为敏捷测试的测试设计的特征。

- 用户故事作为测试基础
- 从追加简单的测试用例开始，向更复杂的东西进行变化
- 是增量的。

关于敏捷测试的温习

敏捷可以给客户带来价值的最小单位需求被叫做用户故事。在被叫做迭代的时间内，进行进行多个用户故事的开发。

敏捷测试是把用户故事作为测试用例来使用。测试者把用户故事记载的从顾客视角的系统的期望动作变换成测试可执行的故事。使用 **Example** 和 **BDD** 等标记法，通过沟通来协商顾客，开发者，测试者三者间期待的系统样式。而且，从简单的路径覆盖测试开始，测试就已经向着复杂的测试用例前进了。

虽然不是严密的敏捷测试，敏捷测试的原型是测试驱动开发。测试之前先进行反复评审来和重构来改善测试设计是有可能的。

敏捷测试还有一个敏捷测试四象限的重要概念。敏捷测试把测试分类放在四个象限里进行管理，来制定执行测试计划的。

研发者的测试=内部品质

顾客的测试=利用时的品质

像上面那样分类，推荐制定何时如何执行哪些测试的计划。

敏捷测试的测试设计特征

基于上面这些内容，敏捷测试的测试设计有以下特征。

参加面向顾客的测试设计会议

敏捷测试中，重要的测试者的作用是把顾客的商业性的系统需求变成非工程师的顾客也能理解的例子和测试。因此，面向测试设计的顾客也可能会参加。

开发前的测试设计执行

敏捷测试是迭代开始前，就是开发开始前，与执行测试设计的客户，开发者达成一致。这里所说的测试设计不是一般意义上的详尽无遗的设计

- 测试观点的筛选
- 基本路径(或关键路径)的例子

也许不只包含上面的，还硬要在这里的测试设计。

从测试流程的角度看的话，通过测试设计能在早期就对样式的遗漏和错误进行评审。

用户故事和测试的可跟踪性

为了把用户故事作为测试基础进行测试设计，要保证用户故事和测试的可跟踪性。这样，由于用户故事的需求变化进而引起的测试脚本的变化，或者用户故事级别的测试设计的变化更容易就是优点。而且，在用户故事的范围内，可以进行单体测试和组件测试持续性重构。

从基本路径到极端用例

这是被叫做增量设计的理由之一。敏捷测试使用冒烟测试来分阶段执行 hAPPy path 的测试和极端用例。实际上不止测试执行，测试设计业是分阶段进行设计的比较多。

(感觉就在身边一样)

在执行测试的优先顺序上，越早发现风险大的问题，越能建立更敏捷稳定的开发，测试过程。

每个迭代特别的测试设计的追加

测试设计是在迭代开始前被作为测试用例被执行的用户故事。这样，测试设计会在每个迭代中都有追加。

敏捷测试的测试设计不擅长的点

如果这样写的话，感觉都是好的敏捷测试的测试设计，也是有下面这些不擅长的事情存在的。

不擅长跨迭代的功能测试设计

因为把用户故事作为测试用例进行测试设计，所以无论怎么样在对跨迭代开发的功能的测试都是不擅长的。这个缺点通常不算问题，优秀的产品业主和设计师应该会很好的把系统要求切割成故事的。

例如，功能 B 和功能 C 是依存于功能 A 的情况

- 迭代 1: 功能 A 的开发和测试
- 迭代 2: 功能 B 的开发以及功能 A 和功能 B 的综合测试
- 迭代 3: 功能 C 的开发以及功能 A 和功能 C 的综合测试

像上面这样，但是

- 迭代 1: 功能 A 和功能 C 的一部分开发
- 迭代 2: 功能 B 和功能 C 的一部分开发
- 迭代 3: 功能 A 和功能 B 的一部分开发

像上面这样，如果用户故事没有好好的整理的话，什么时候开始综合测试？综合测试的范围是什么？等等问题就会产生。（如果说不是那样的迭代增量开发过程就到那里为止吧，如果忽略非增量的话，就会失掉测试设计重要性，迷失在没效率的设计中。）

不擅长跨迭代的测试设计优化

测试设计的重要作用，是保证测试覆盖，并且考虑测试经济性减少测试用例数。但是敏捷测试在跨迭代的测试设计优化上是不擅长的。

为什么呢？第一对于用户故事 100%覆盖的前提下，其次为了体现对用户故事追踪下的重视，对整个系统考虑如何测试以及测试多大规模进行可视化是这件事不擅长的原因。

例如，我认为根据这次迭代追加的 10 件测试用例，即使可以做到把上次迭代做成的 100 件测试用例删除了，也可以保证做到同样的覆盖的情况，一般来说测试用例的优化也没被执行的吧。

基于这样的考虑，就知道从把数百亿计的测试通过自动化执行的工程师那里听到「实际上不知道测试用例减少了没」的话的原因了。在公司里，测试执行的优先顺序进行测试优化就是很有名的话了。

另外，实践敏捷测试的作者 Janet Gregory 的标题名为“可追溯的”的博客很有意思。对于用户故事，可追溯性本身很重要，可追溯性矩阵一样的俯瞰全体的表是不需要说的。

不擅长架构的测试设计

我认为褒贬兼有，尽管敏捷测试四象限的第四象限定义了非功能性需求的测试，敏捷测试还是被指出来了不擅长非功能性需求的测试。

这个问题根源很深，架构如果能够自律，持续进化的话，当然这反映系统的品质，例如演示或运用性上的品质也有持续测试，被监测的必要。

这是由于敏捷测试的架构相关经验积累少，所以架构反应系统品质往往容易被忽略。作为这个问题的解决方法，演示和可用性等非功能性的需求也需要作为用户故事被追加提案。

总结

敏捷测试是把用户故事放在重点的。

- 顾客参加测试设计

- 通过测试设计，提前样式评审
- 增量测试设计

做上面的事情之外，还列举了敏捷测试设计的另一方面不擅长的事情。

- 跨迭代测试设计优化
- 和用户故事关联的最低限度架构的品质的测试

引进敏捷测试时，其优点当然是要考虑的，其不擅长的部分也是必须要考虑的。如果无视不擅长的部分，会导致测试过火或者测试遗漏。那些能好好处理的话，才可能迅速稳定的建立测试流程。

另一方面，这些敏捷测试的测试设计不擅长的部分，可以作为今后敏捷测试系统化的课题。到现在为止敏捷测试说起来还是给人开发者方面的东西的印象，解开这些课题也有必要考虑测试和品质的技术。

盘点那些必不可少的“用例集”

◆ 作者：琦 少

“用例集”是所有 QA 同学都不陌生的一个名词。多条有某种关系的用例，组合在一起就是一个用例集。工作中，接触过的用例集可谓千千万。但是在 QA 工作中有这么几个用例集，是质量保证环节必不可少的。接下来和大家逐一探讨一下这几个用例集的价值，意义，以及组织技巧。

1、冒烟测试用例集

【时机】: beforeTest（真正开始测试之前）

【意义】: 用来快速的验证提测产品是否具备可测性。

【经验分享】:

此用例集，虽叫冒烟测试用例集。但更准确的称谓应该是“测试打回标准”（testable checkList）。QA 同学重点工作应放在定义清晰明确且可操作性强的打回标准。并且监督标准执行。定义哪些情况是不接受提测申请，必须打回的。比如：新功能未实现，已实现功能和既定需求不符合，新功能模块存在严重的崩溃，影响后续的测试。

此用例集，不一定要 QA 人员亲自执行，可以作为研发同学自测的标准。这样可以充分发挥该用例集的约束性。用时也就意味着，该标准必须得到一致性认可，并且必须明确，易操作（RD 不是专业的 QA 要避免一些过于复杂的步骤，或者冗余的描述）

此用例集，不应过长（建议控制在 10 条左右）。且不应过多关注具体功能，更多关注整体的可测性。同时，应该具备充分的约束力，本着执法必严的精神，不让其成为空摆设。

2、功能测试用例集

【时机】: onTesting（功能，测试进行中）

【意义】: 用来保证新功能符合既定需求，找出 bug，并推动改进。

【经验分享】:

此用例集，是大家最为常见的用例集。但需注意，其子用例集需架构清晰，子用例集之间尽量低耦合，以便在资源允许的情况下由多位同学并行执行，以提高测试效率。

此用例集，非常容易也最为忌讳变成对产品需求的简单复述。当然这也是区分高级测试工程师和初级测试工程师的环节。多用些专业的测试技巧。如边界值，等价类等。注重相关测试经验的运用，以保证测试维度。

在组织此用例集的用例时，建议用极简的标题，来表明用例的验证内容。并且多用动宾短语来使得表述清晰。此外还需要关注粒度的控制。让步骤，和验证内容都控制在合理的层级。以谋求测试覆盖范围和执行效率的平衡点。

3、回归测试用例集

【时机】: afterTest（功能测试完成后，发布之前）

【意义】: 保证常用，重要，或三级界面中的易见功能，不出问题。

【经验分享】: 测试过程中可能会因为研发改动过某一个模块，导致其他模块不可预知的问题，也可能因合入某个较新的 SDK 导致奇葩问题。此时非常有必要，从用户角度出发思考质量保证的方案。也就是需要关注用户最常用的功能，以及自己产品主打的重点功能，或者那些显而易见（通常是三级界面之内）的功能。此时无疑需要针对这三个方面设计回归测试用例集。

该用例集，是建立在功能测试用例集的基础之上，前提是基本信任功能的质量。力求快速的保证产品的整体质量。在移动互联网测试领域，因为快速迭代的风格，通常回归测试的时间不会超过 2 个小时。这就要求用例数在 50 个左右为宜（最多不要超过 100 个）。太少了难以保证覆盖的面积，太多了在有限的时间内，执行情况必然大打折扣。所以回归测试用例也是考察用例设计水平，用有限的用例，系统的保证 APP 的核心功能的安全。在概括能力，系统化设计能力等方面都有较高的要求。可以说，如果能设计出优秀的回归测试用例集，就可以称为高级测试工程师。

当然，选择执行的人也非常关键，务必选择那些谨慎认真，踏实，细致的同学去执

行。以便让回归测试用例真正发挥其应有的作用。

如上，这些用例集，都是一款高质量的 APP 必不可少的。也是成长到高级测试工程师所必须经历的。作为一个 leader 你该盘点自己的团队是否已经组织好这些用例集。作为一个优秀的测试工程师，你该想如何用自己的能力，来不断的优化这些重要的用例集。

心所向，驰以恒

◆作者：琦少

“心所向，驰以恒”是奔驰的宣传口号。听闻之后，非常震撼。仔细揣摩，其实 QA 之路也是铸心之路。在测试团队建设过程中，何尝不需树立这几心呢？

1) 匠造之心

“工匠精神”所阐述的是仔细雕琢每一个细节，本着精益求精的态度来对待自己的工作。优秀的 QA 团队，会自觉将这种精神贯彻到日常工作。围绕简单明确的目标，努力做到更好，努力超越自我，这是 QA 同学，细心，耐心，责任心的最好体现。比如：

➤ 设计手工 case 时，能精确的拿捏 case 的粒度。

可体系化的考虑各模块的用例设计，尽量做到不重不漏。

➤ 设计自动化 case 时，能多考虑 case 的覆盖面。

通常 1 条自动化 case 具有覆盖多条手工 case 的特性。可充分发挥自动化测试的高效性，和便捷性。逐步从单纯复述手工 case，转变为体系化的重构 case。

➤ 平台设计，可多从使用者的角度出发，换位思考。常问自己几个问题：

哪里用的不爽？可以如何改进？是否做到简单可依赖？

2) 挑战之心

QA 团队成员应用于承担自己的责任，明确 KPI，并努力实现。不仅如此，更要有好强的态度，朝着更高的目标努力。

挑战目标，超越目标。

一个优秀的团队应做好正态分布，并且以结果为导向。将绩效作为标准。作为激励大家的工具。不仅要勇于挑战，更要善于挑战。懂得以结果向导，做好合理的拆分。给每周，每天都设定个小的挑战目标。且要建立快速反馈机制，注重效果的验收，做好必要的指导。

3) 追梦之心

工作过程是追求认同感的过程：leader 对自己的认同，同事对自己的认同，自己对自己的认同。若把工作当做一种事业来经营，将业绩和梦想结合，是一种最佳的状态。工作中应不断鼓励 team 成员，发掘并成就自己的亮点（让团队为你骄傲，让自己感到自豪）让梦想成为奋斗的动力，用结果来证明自己的价值。

4) 敬畏之心

last but not important! QA 同学应该是敏感的，甚至是敬畏的。敬畏神圣的质量保证工作。敬畏测试过程中风险。敬畏团队关系。只有你足够重视一件事。才能真正的全身心投入。首先，质量保证不是儿戏。绝不允许团队中存在“随测测”的风气。其次，风险通常是不能完全避免，但把控风险无疑是 QA 团队的核心价值。如此，又怎能不谨小慎微？再次，在中国大多数公司，QA 团队相对弱势。这就注定你不能以非常霸道的方式和其他团队打交道。需要谦虚谨慎，懂得阐述问题的技巧。另自己的言辞具有说服力。QA 团队和 RD 团队的关系非常微妙。但是绝不可站在完全对立的层面。不然只会妨碍整个项目的效率。一个好的 QA 团队，不会以“良药苦口，忠言逆耳”，来安慰自己。而是要以简单可依赖来要求自己。

一个团队，技术沉淀固然重要，但团队建设也同样重要。TB 不是简单的吃吃喝喝，它是一门学问，更是一门艺术。所谓，路漫漫其修远兮，吾将上下而求索。和各位朋友共勉之。

原创测试文章系列（四十）上篇 精彩预览

- APP 测试之找回密码
- Appium+Robotframework 实现手机应用的自动化测试-iOS 篇
- LoadRunner 下载的疑难问题一例
- 如何进行测试范围确认
- 关于移动终端之 WiFi 测试点的总结
- 从业务骨干到管理精英
- 史上最全的网银转账测试分析与设计
- 系统性能工程师指南-译
- 界面测试的定义和测试关注点
- 软件测试之需求分析
- 不想当兔子的猫不是好猫
- 胚界传说-测试公主之拯救天蝎星

● 马上阅读 ●