

绩效考核

1、测试团队绩效考核

绩效评估的的客体：是个体成员还是整个团队。

- ✓ Pascerellayer 认为，团队绩效评价应以成员个人完成工作的状况为基本依据，理由是激励只能作用于个人而不是群体；技能的提高和行为的改进最终必须落实到个人。若仅考核团队绩效，个体的努力得不到充分的肯定，就容易造成社会懒散现象，即个体由于参加团队工作，其工作效率比自己单独工作时的效率反而大大降低。此现象一旦在组织中蔓延开来，不仅会影响组织绩效，还会毒害组织文化。同时，由于绩效考核与薪酬及个人价值的实现相联系，因此，在团队中，能力高的成员倾向于对个人绩效的考核，从而得到更高的认可和报酬。
- ✓ Zingheim 和 Schuster 则认为对个人的考评应考虑团队的整体绩效，因为团队的成功很大程度上依赖于团队成员间的团结合作，理解支持，若评估集中于个体层面，会导致个人主义盛行，忽视团队的协作精神，阻碍信息、技能的共享和绩效的提高，降低团队工作的优势。
- ✓ 因此在实际操作中，企业往往采取一种折中的方法，即按一定比例兼顾团队和个人两个层面的绩效考核。从目前的研究来看，还没有一种很好的办法可以科学地确定这个比例。但是，如果从团队性质的差异、团队所处的阶段等方面来考虑，那么至少可以确定考核的天平是更向个体的一极偏还是更向团体的一极偏。

绩效考核的内容：结果、行为还是能力。对于绩效内涵存在着三种不同的观点，即“绩效是结果”、“绩效是行为”和“绩效是能力”。Bernardin 将绩效定义为“在特定的时间内，由特定的工作职能活动产生的产出记录，工作绩效的总和相当于关键和必要工作职能中等绩效的总和（或平均值）”，这是“绩效是结果”的典型观点。Murphy 等人将绩效定义为“一套与组织或个体所工作的组织单位的目标相关的行为”。近年来，以能力作为绩效的观点得到了广泛的使用，这是以评估个体所拥有的完成某项工作所具备的知识和能力的方式。伴随着这三种观点的诞生和发展，绩效考核大致经历了基于结果、基于行为以及基于能力的三个考核发展过程。虽然这三种观点相互区别，且都是在否定前者的基础之上产生的，但是，如果不带入特定的环境，特定的组织，及组织发展的特定时期，那么三者之间并不存在绝对的优劣。如果组织下达的目标非常清晰，基于结果的绩效考核是最容易实施，也最有效；相反，如果目标模糊，无法准确衡量其结果，这种考核方式就会失效。基于能力的考核方式理论上是从战略管理的角度出发，最具有激励效果和长期效应，最有利于组织不断发展，但在实际操作中却很难达到效果。因为能力是无形的，它依附于个体，既受主观因素的控制，也受各方面客观因素的影响，很难用标准化的方法衡量个体的能力，即使是方法对组织期望成员所具有的能力和特质作出了解释，但这些解释仍是描述性模糊语言，在实际操作中仍然难以做到真正的科学公正。基于行为的绩效考核方法通过考核员工为实现既定的结果所必须做出的行为来实现对结果的控制，由于行为必然是建立在某种能力基础之上的，并且行为比能力更具有外显性和可测性，因此一定程度上，该方法兼顾了组织目标和个人能力。但是，绩效考核中容易出现目标置换的现象，一味对行为测评会导致成员将行为作为目标，进而影响实际目标的实现。因此，无论哪种考核方式，都有其适用的条件和要求，不存在一种绝对对好的方法。

基于项目团队生命周期的绩效考核：

- ✓ 孵化诞生期：这是指团队形成前到团队正式形成的一个阶段，是选择合适的项目成员组成团队的时期。
 - ◇ 考核的客体是个人。团队的首要任务是筛选项目组成员，根据项目目标的要求，选择最为合适的人选组成团队，所以考核的对象是个人。
 - ◇ 考核的重点是能力。从项目团队成立的目的来看，它一般是为了开发一种新产品或者提供一项新的服务，因此对成员的知识技能要求较高，需要成员具有较高的技术水平和知识储备以及不断学习和创新的能力。同时，成立项目团队，意在发挥团队快速响应和凝聚集体智慧的优势，更加需要团队成员间的相互合作相互支持，所以需要较为系统地考核成员的协调合作能力，包括，对团队其它成员工作任务的认识、口头交流、个人成长、问题解决、责任担当、领导技能等等。因此，在选择项目团队成员的时候，通过对被选者专业技能、基本素质当然也包括过去的工作经历和背景等各方面的考核，最终确定较为合适的人选。
- ✓ 成长期：这是团队正式形成之后，团队工作逐渐步入正轨，团队成员开始通过个人努力和彼此的合作共同在所研究的项目上获得初步的成就。
 - ◇ 考核的客体是团队。团队成立之初，成员合作的意识还没有形成，工作的独立性较强，此时的工作重点应该

是营造一种信任、关怀、相互支持的合作氛围。同时，项目也刚刚起步，没有取得实质性的进展，个人的贡献还无法准确衡量，在这种情况下，如果过多地衡量个人绩效，特别是个人产出绩效，不仅不利于合作精神的培养，也会由于准确性不高而使成员产生不公平感，从而对团队工作形成抵触情绪。注重团队整体绩效的考核，可以向整个团队成员传递这样一个信息，即必须注重团队的整体效率，共同开发团队能力。同时，对团队绩效的考核还可以提高团队成员对自己团队自豪感和所有感，并不断提高其认同感和归属感。

- ◇ 考核的重点是行为。刚刚进入一个新的团队，如果此前没有进行过合作，成员之间会由于陌生感而信任度较低，彼此在沟通和交流上存在困难，需要相当一段时间的磨合，工作进度也很缓慢。如果不通过有意识的加强合作意识的培养，难么磨合期就会较长，从而影响目标的实现。因此在项目团队进入成长期时，绩效考核的重点应该放在对团队成员行为的考核之上。绩效考核不仅仅是一种过程的监督和事后的衡量，更是一种对员工行为进行引导的方式。作为一种信息的传播途径，通过评估的本身，反馈以及与薪酬的联系，以直接或间接的方式告诉被考核者，组织鼓励什么样的行为、反对什么样的行为，从而引导和鼓励成员采用更加积极的态度和行为，主动参与团队工作，加强团队成员之间的合作和学习，使项目团队尽快度过磨合期，向着一个良性的方向发展。
- ✓ 成熟期：进入成熟期，团队工作进展顺利，项目取得关键性的突破，团队成员自由沟通，合作意识加强。
 - ◇ 考核的客体是个人。此时应该加大对个人绩效考核的比重。因为项目已经取得一定的突破，目标接近实现，团队成员的成果和贡献相对比较清晰，可以较为准确的衡量，需要对其加以肯定。如果仍然只是停留于对团队绩效的整体考核，并以此为基础进行利益分配，个体会逐渐产生不公平感，因为随着项目工作的深入开展和目标的逐步实现，个人由于态度、能力、技术支持等诸多方面的差异，贡献度的差距会逐渐扩大，客观上会有成员的贡献大于其它人，如果不及及时加以肯定和认可，那么就会挫伤这一部分核心成员的积极性。
 - ◇ 考核的重点是结果。成熟期的团队首要任务是推动工作进展，以保证最终成果的实现。由于既有的工作方式已经基本形成，合作沟通的氛围已经建立，如果仍然强调对个体行为的考核，会使成员将大部分的注意力投入到日常的工作行为和方式之上。事实上，鼓励行为的本身并不是目的，关键是行为带来的结果，合作和交流是团队的基本工作手段，但手段不能代替目的，项目及时高效地完成才是项目团队的存在目的。如果不以任务为导向而长期进行行为考核，容易使个体忽视目标和结果，影响工作的效率，例如，过分的注重沟通和交流，造成决策时议而不决，贻误时机，或者意见趋中，成员过分尊重群体意见，不愿表达自己突破性的想法和思路。
- ✓ 衰退期：项目目标已经基本实现，团队即将解散，此时需要对整个项目团队作一个综合的评估。
 - ◇ 考核的客体兼顾个人和团队。进入衰退期，绩效考核一方面需要通过对项目团队的整体绩效作出评估，以考核项目的完成情况；另一方面，也需要对团队成员绩效作出公正科学的总结，这不仅决定成员能否取得公平的报酬，也是其进入另一个团队的基础。
 - ◇ 考核的重点主要是个人的综合绩效以及团队的产出。项目团队任务明确，业绩是团队成立的最终目的，因此在项目团队解散之际，需要对目标的实现情况作一个综合评估，以此判断项目的成功与否。对个人也需要做一个总体的评价，尤其是产出和能力的评估，组织需要对此进行备案，成为以后的项目团队选择成员的重要根据。

2、测试人员绩效考核

考核基于测试过程进行，因此必须在过程结束之后才能进行。由于工程是分布提交测试的，每月可以根据实际情况进行月考核，工程结束后或任务结束后再统一考核。按照传统测试周期，测试过程分为：测试计划、测试设计和测试执行三个方面进行。测试计划属于测试经理的范畴。测试人员主要是测试设计和测试执行。

测试人员的绩效考核包括多个方面：

- ✓ 工作态度。包括工作责任心和工作积极性。
- ✓ 工作职责与期望达成度(注意:在工作安排前需求明确对应测试工程师的工作职责和对测试工程师的期望值，这里的工作职责一般是和管理相关的工作职责内容)。
- ✓ 工作内容考核。
 - ◇ 参与软件开发过程的工作内容考核，比如参与需求和设计的评审，就需要对需求的理解上，对需求提出问题

的质量上等作出评价。

- ◇ 参与测试文档的准备工作，如测试用例等，需要通过评审测试文档来考核测试人员的能力。如评审测试用例的质量，对需求的覆盖程度，可理解和执行等方面来判定测试人员的能力。
 - ◇ 执行测试的工作，需要从测试人员所发现的问题对测试人员进行评价。包括发现问题是复杂的还是简单的，是隐藏较深的，还是一些表面的问题。包括问题的书写上进行评价，问题的书写是否详细清晰，开发人员可以再现，还是含糊其词，不明所以。一个问题是否写多遍等。
 - ◇ 测试结果缺陷残留，对于已经发布的产品，从用户反馈问题考核测试人员的绩效，但是这个可能需要的时间比较长；对于不同版本的测试，可从版本的漏检进行统计。
 - ◇ 测试人员的沟通能力考核，包括缺陷在开发工程师中沟通的达成率和拒绝率。
- ✓ 工作效率与工作质量考核。
- ◇ 测试设计中工作效率相关指标：
 - 文档产出率:这项指标值主要为测试用例文档页数除以编写文档的有效时间获得。用于考察测试人员测试用例文档的生产率大小。
公式: Σ 测试用例文档页数(页) / Σ 编写测试用例文档有效时间(小时)
参考指标: 根据项目汇总得出平均在 1.14 页 / 小时左右, 高于此值为优, 低于此值为差。
 - 用例产出率:这项指标值主要为上述指标值的补充, 用于考察测试人员测试用例产出率大小。测试文档页数可能包含的冗余信息较多, 因此要查看文档中测试用例的多少。方法是测试用例文档中测试用例编号总和除以编写文档的有效时间。
公式: Σ 测试用例数(个) / Σ 编写测试用例文档有效时间(小时)
参考指标: 平均 4.21 个用例 / 小时
 - ◇ 测试设计中工作质量相关指标：
 - 需求覆盖率:计算测试用例总数之和除以与之一一对应的功能点数之和, 主要查看是否有功能点遗漏测试的情况。
公式: Σ 测试用例数(个) / Σ 功能点(个)
参考指标: 100%。如果连功能指标都不能满足 100 %覆盖, 起码说明测试不充分。这个指标收集起来相当困难, 如果存在需求跟踪矩阵或者测试管理工具能把用例与需求一一对应就容易得多。注意: 有的功能是难于测试的, 那么未能覆盖到的需求要综合分析, 明确是测试人员遗漏? 还是无法测试? 这需要放入问题跟踪表中进行后续跟踪; 另外, 有的功能点包含的信息较多或者有的用例包含几个功能点, 这时只能把重复的功能点或重复用例按一个计, 难于区分的要做说明。
 - 文档质量:测试用例进行评审和同行评审发现的缺陷数, 或者将此缺陷数除以文档页数算出比率。此指标考察测试人员文档编写的质量如何。
公式: Σ 缺陷数(评审和同行评审)(个) / Σ 测试用例文档页数(页)
参考指标: 由于评审是发现的缺陷数是不固定的, 因此, 这个指标没有可供参考的数值。如果缺陷数大小不能直接用于比较就使用缺陷 / 页方式进行横向对比。
 - 文档有效率:使用测试用例文档进行测试时发现的系统测试缺陷数除以此文档页数。用于考察文档是由有效的指导了测试工作。
公式: Σ 缺陷数(系统测试)(个) / Σ 测试用例文档页数(页)
参考指标: 平均 2.18 个缺陷 / 页
注意: 如果存在测试人员在测试时创建新文档用于辅助测试时应包含这一部分。
 - 用例有效率:使用测试用例发现的全部缺陷除以测试用例数总和。这一指标是上一指标的补充指标, 用于考察用例质量是否较高
公式: Σ 缺陷数(系统测试)(个) / Σ 测试用例数(个)
参考指标: 平均 0.59 个缺陷 / 用例, 也就是说, 每执行两个用例才得到 1 个缺陷, 各工程有所不同, 可以自己实践一下

- 评审问题数:是否存在对需求理解、系统架构设计、系统设计等方面引起争议的问题。体现出测试人员发现问题的深入层次,有利于产品质量的提高。

◇ 测试执行中工作效率相关指标:

- 执行效率:利用测试用例文档页数除以此次系统测试执行的时间总和(不包含用例文档编写时间)。补充指标方法是用例的个数除以此次系统测试的时间总和。用于获得工作中测试人员每小时执行测试的速度。

公式: Σ 测试用例文档页数(页) / Σ 执行系统测试的有效时间(小时)

Σ 测试用例数(个) / Σ 执行系统测试的有效时间(小时)

参考指标:平均 0.53 页 / 小时, 1.95 个用例 / 小时。即测试人员每小时执行半页测试用例或者每小时执行 2 个测试用例。通过横向比较,容易知道那位成员的执行效率较高。注意:执行效率高的不代表测试质量也高,甚至执行效率和测试质量成反比,所以后面工作质量的指标会补充这一部分的偏离。实际结果表明,用例执行效率高的成员,其缺陷发现率往往偏低,考核如果不将此纳入进来也可以将其作为测试改进的一项重要数据进行收集。

- 进度偏离度:检查计划时间和实际时间的进度,方法是计划时间差额减去实际时间差额除以实际工时总和,用于考察测试人员进度情况,监控测试是否按照日程进行,是否满足了工程的进度要求。

公式: Σ (计划开始时间 - 实际开始时间) + Σ (计划结束时间 - 实际结束时间) / 总工时

参考指标:15% 进度偏离是个相对的指标,可能偏离了 20 个工作日,但是对于一个长达半年时间的测试而言偏离天数比上整体测试所需天数不足 15%,可能偏离了 3 个工作日,但是对于一个只有 1 星期时间的测试已经超过了整个测试阶段所需天数的 60%。

注意:计算时分子分母要保持一致,即开始或结束时间已经去除了非工作日时间,则总工时也要去除非工作日时间。因为制定计划时是根据每个公司的工作日来制定的,也就是说,考虑了非正常工作日的日程。

测试进度也是考核很重要的一步,如果没有进度保证,所有的测试都存在风险,第一种方法是测试人员可以采用自下而上的方式向测试经理报告计划用时,这种方式风险比较少,个人根据自己能力大小确定,但是缺点是存在测试人员虚报可能性。另一种方法是测试经理进行估算后分配工作日程,这时估算是很重要的前提,除了依赖于测试经理的经验外,对评估结果进行同行评审是很客观可取的方法。

- 缺陷发现率:测试人员各自发现的缺陷数总和除以各自所花费的测试时间总和。由于执行效率不能足够代表测试人员是否认真工作,那么,每小时发现的缺陷数就是重要的考核指标,你的工作可以通过这项指标得到反馈。

公式: Σ 缺陷数(系统测试)(个) / Σ 执行系统测试的有效时间(小时)

参考指标:平均 1.1 个缺陷 / 小时 假使有位测试人员没有达到 1 小时发现 1 个缺陷,那么,除非产品质量高、模块较小,否则,就是他的缺陷发现能力不如其他测试人员。当然,详细分类中可以根据发现重要缺陷的多少来定义缺陷发现能力。

◇ 测试执行中工作质量相关指标:

- 缺陷数:为了更客观度量,考虑到 bug 的严重性、技术难度、产品类型、模块稳定性等因素影响,不是用“所发现的 bug 数量”,而是用“所获得的 bug value (缺陷值)”来度量,公式被定义为:

$$\text{Bug_value} = (\text{P0_Bug_Number} \times 1.6 + \text{P1_Bug_Number} \times 1.4 + \text{P2_Bug_Number} \times 0.7 + \text{P3_Bug_Number} \times 0.3) \times \text{Wd} \times \text{Ws} \times \text{Wt}$$

其中: P0_Bug_Number: 致命的(fatal)缺陷数量; P1_Bug_Number: 严重的(critical)缺陷数量; P2_Bug_Number: 一般的(major/normal)缺陷数量; P3_Bug_Number: 次要的(minor)缺陷数量

Wd: 技术难度系数,如 Database, Enterprise Server, Java 难度系数大,发现 Bug 不容易, Wd 可以定在 1.5 - 5.0

Ws: 稳定性系数,全新模块, Bug 比较多,发现缺陷比较容易;版本越高,越稳定。Ws 可以定在 0.5 - 1.0, 假如以 version 10.0 为 1.0, Version 1.0 = 1/100, Version 2.0 = 4/10, Version 3.0 =

9/100, ..., , Version 8.0 = 64/100, Version 8.0 = 81/100

Wt: 产品类型系数, 可根据实际情况和历史数据来判断。Wt 也可以和 Wd 合并为一个系数。

- 有效缺陷数/率: 被拒绝和删除的缺陷数总和, 或者被拒绝和删除的缺陷数总和除以缺陷总数。这项指标用于考察测试人员发现的、被确认为缺陷的缺陷数高低或者百分比, 数和比率越低测试质量越高。

公式: Σ 缺陷数 (系统测试中被拒绝和删除的) (个)

Σ 缺陷数 (系统测试中被拒绝和删除的) (个) / Σ 缺陷数 (系统测试) (个)

参考指标: 平均 21.9 % (测试人员发现的每 100 个缺陷中平均有 22 个缺陷不被开发组确认、认为不是“缺陷”或者错误录入缺陷)。有效缺陷比率容易给出, 但是有效缺陷数具体数据要根据项目情况, 无法给出可参考的数值。

注意: 这项指标可能有不正确情况, 假使缺陷被拒绝和被删除的原因不是因为测试人员误操作和需求理解等自身错误引起, 而是系统本身不能实现或者数据错误引起的, 那么就要考虑剔除这部分。对于测试人员发现系统框架根本性的、初始化参数设置错误引发的、错误数据、错误环境等而开发人员因无法修正、可以通过改变环境而无需修改程序、重新导入数据、再次发布从而拒绝或删除的缺陷, 应给予此测试人员奖励。

- 严重缺陷率: 这个比例用于弥补缺陷发现率的不足。主要是根据严重程度分类的缺陷数比全部缺陷或者有效缺陷数。一般而言, 每个公司基本把缺陷严重程度分为严重、一般和微小, 或者更细 (通常等级数为奇数)。另外, 可以对缺陷严重程度进行折算 (严重: 一般: 微小 = 1 : 3 : 5) 通过折算可以得出权重, 然后在计算测试人员分值。

公式: Σ 严重 / 一般 / 微小 / Σ 缺陷数

Σ 严重 / 一般 / 微小 / Σ 有效缺陷数

参考指标: 严重 ~10% 一般 ~70% 微小 ~20%。当测试人员发现的缺陷中严重错误比率越高, 说明测试质量相对就好, 通常严重程度缺陷数的分布呈正态分布。

- 模块缺陷率: 这个指标主要是根据一个单独测试模块的缺陷数除以模块本身功能点数得出来的。假使一个模块是单独测试的话, 很容易可以和其他模块进行指标横向对比, 参照对应的测试人员, 得出所测试模块的缺陷数, 可以考察测试人员测试水平, 也为开发考核提供数据。

公式: Σ 缺陷数 (系统测试) (个) / 功能点 (个)

Σ 缺陷数 (系统测试) (个) / 子功能点 (个)

参考指标 平均 3.74 个缺陷 / 功能点 1 个缺陷 / 子功能点

注意: 有些功能点没有子功能点, 计算子功能点时要进行说明。

- 遗漏缺陷率: 发布后的线上故障, 现阶段测试相关的故障主要都是因为测试遗漏, 有遗漏就说明我们的测试还是效率不高, 可以改进。

公式: Σ 遗漏缺陷数 / (Σ 遗漏缺陷数 + Σ 遗漏版本发现缺陷数)

- Bug 发现的时间点, bug 曲线的收敛性, 理想的效率高的模式应该是前多后少, 慢慢收敛的, 如果前期 bug 非常少, 后期却发现大量 bug, 那我们的前期效率就有问题。

- 缺陷定位和可读性: 可读性内容包括 Bug 描述的规范性, 分优秀、良好、普通与不合格, 描述是否清晰, 问题定位的附件是否完备等。如果一个测试人员只会通过页面将现象表达出来, 而无法定位这种现象是有什么引起的, 或者无法定位该缺陷到底错在何处, 那么可以判定测试人员只是做了简单的表面测试, 并没有对所发现问题进行分析定位。

- ✓ 对技术组 (性能自动化和环境) 测试人员效率的度量:

- ◇ 自动化测试的引入和使用是否合理, 不是每个项目都适合做自动化的, 自动化并不能保证效率的提高, 用 5 个小时开发的自动化脚本来替代 3 个小时的手工测试并不合算, 自动化测试需要评审, 按照项目的大小不同, 必要的情况下才引入自动化测试。

- ◇ 自动化测试, 特别是性能测试结束之后, 我们要分析部分测试结果, 测试结果的分析水平, 也可以作为衡量测试效率的一个指标。

- ✓ 对测试项目负责人的效率的度量:
 - ◇ 测试是否提早介入项目，例如 FRD 阶段就介入，越早介入，越有利于测试，使测试人员更加熟悉整个项目，使问题早暴露，提高整体效率。
 - ◇ 开发提交测试的时候，标准是否合理，把关是否严格，如果开发的质量不行，坚决要退回，不然会影响测试的效率和进度。
 - ◇ 测试计划阶段，评价测试计划的合理性，包括任务细化，细化的程度是否合理，任务顺序，资源安排，任务分配合理，风险预估等等。
 - ◇ 项目结束后，评价项目进行阶段中负责人的跟进情况，特殊情况处理，风险触发之后的处理，资源协调，信息收集，共享，沟通，配合等等。
- ✓ 测试管理。
 - ◇ 计划质量:测试计划的评审缺陷数或比率，可以与其他同类型项目或数据库平均指标进行对比。

$$\Sigma \text{缺陷数} (\text{评审和同行评审}) (\text{个}) / \Sigma \text{测试计划文档页数} (\text{页})$$
 - ◇ 成本质量:成本度量主要放在工作量这块。因为无论涉及工资还是奖金，都要和工作量挂上关系。成本质量主要是对测试活动的计划工作量总和比上实际的工作量数值总和。对测试人员考核的进度偏离已经考虑了进度因素，而工作量涉及的是成本因素。
 公式: $\Sigma \text{测试活动计划工作量} (\text{估算人日}) / \Sigma \text{测试活动的实际工作量} (\text{人日})$
 参考指标: 原则上不能偏离计划的 $\pm 15\% \sim \pm 20\%$ 。实际上，这个指标是对成本的一种度量。对于一个大的项目来说，估算值往往差距非常大，阶段统计时可能有 $\pm 500\%$!! 这时调整计划是很必要的，在最终阶段取考虑计算平均估算值。一个测试经理必须对完成任务的成本进行有效控制。这两项指标是相对容易量化的部分，而需要添加其他量化指标需要综合考虑由项目经理和测试部部长给出标准，例如管理用时比率（整个项目测试期间管理时间占整个项目测试总时间）、系统整体缺陷数与其他同类型项目或数据库平均指标进行对比等等。
- ✓ 考核具体方法:
 - ◇ 将各项指标进行汇总分析，得出总和表格，根据测试人员各项指标大小进行排行榜制作，如列出 1、2、3、4 名。
 - ◇ 确定阶段涉及的权重。例如将测试设计和测试执行权重各为 50%。其中，工作效率占 40%（即占所在阶段 20%），工作质量占 60%（即占所在阶段 30%）。
 - ◇ 确定每类指标的分值，然后每类指标达到平均标准给 100%，达不到或者超过根据 80%~120% 比率给分。
 - ◇ 将比分统计出来后进行综合评定，必要的话增加一些调整系数。
 - ◇ 最好将定性分析纳入进来，采用问卷调查和项目经理评分制度给出定性指标分数，建议这部分权重不要超过 10%~15% 以保证测试考核的可度量性。
 - ◇ 当所有考核分数给出之后，提醒一点的是，既然做了考核，就必须公开这些结果，而且考核具有导向型，不要让考核误导了对质量工作的追求才是最重要的。
- ✓ 考核注意事项:
 - ◇ 项目并不是一个月就能完成的，如每月进行，要考虑“可考核部分”为那些，挑选那些指标能够横向对比，然后分阶段、分任务评定。
 - ◇ 参与测试的时间长短也要给予重视，除了上述量化指标外，测试人员整体投入时间长短也是很重要的，加班也要作为特殊考虑因素，也许某个测试人员只参加了测试执行 3 小时，各项指标都是良好的，但是不可能给他比其他参与时间更长的人员更多的分数。这部分就是增加调整系数的原因。
 - ◇ 测试经理的测试设计和执行部分和项目测试人员一起考核，但是测试管理工作要单独考核，作为另外的加分，或者如文章前面所述纳入项目组给予考核。因为测试经理在项目测试中起着管理者和质量保证负责人的角色，不要把他和其他测试工程师平等对待。
 - ◇ 考核前要考虑项目的实际情况，不要盲目的轻易承诺测试组人员考核会和薪金或者淘汰机制挂钩，否则考核会起到反效果。

- ◇ 作为考核者要注意以下比例，也许有些没有列入考核内容，但是如下这些点可以指导测试。
 - 测试团队发现的 bug 和所有 bug 之间的比例
 - spec 设计造成的 bug
 - 重复或者误提交的 bug 所占的比例
 - 每周发现的 bug 的趋势图
 - Bug 严重等级的构成比例
 - Bug 从提交到解决的平均需要时间
 - Bug 从解决到关闭的平均需要时间

项目组测试人员考核的主要目的是在于激励测试组测试人员工作，鼓励能者，鞭策落后；另外，还可以起到发现人才和查找不足的作用。考核中即要体现多劳多得的原则，也要体现公正性和合理性原则，奖罚分明才能有效促使质量管理工作的进步。要想考核得到满意的效果，上述方法的重要的前提条件是：必须要在项目中充分收集相关的数据，包括采集缺陷数，记录工时、提交详细工作日志和进行文档配置管理，没有这些数据，定量分析就无从谈起，测试人员考核也无从谈起。

3、测试人员工作度量

测试度量主要从 3 部分开展工作：一个是缺陷数据的统计分析，第二个是工作量的统计分析，第三个是测试工作量的估算。

- ✓ 缺陷的统计分析。主要是从缺陷严重性、优先级、模块缺陷的分布、缺陷的收敛情况、缺陷的修复情况进行统计，并根据统计结果，进行一定的分析。
- ✓ 工作量的统计分析。
 - ◇ 日常工作量的记录，这个由团队成员自己编写。在填写工作记录时，需要为每个工作记录选择相应的任务类型，并且工作任务持续时间最长不超过 4 小时。
 - ◇ 每星期统计本周团队成员在各个项目中的投入情况。不仅让自己清楚了，也让上司了解测试部对于项目的支持情况。
 - ◇ 每半个月统计整个团队的工作分配情况(但是数据是每周都填写的)统计每个人在各个项目的工作量分配情况。这个和上面那个统计表的侧重点不一样，上面这个统计表侧重在部门整体，现在这个表侧重于个体。
 - ◇ 定期(如每周或半个月)将团队成员在项目中的工作量投入情况记录到项目工作量投入表中。这个表格主要用于统计具体每个项目的测试工作投入情况，及作为后续测试工作量估算的原始数据。
 - ◇ 在项目到达一个阶段后，将项目测试收集的数据进行汇总、统计。收集的数据除项目基本信息外，还包括工作量、测试投入成本、项目规模、项目总成本、项目总工作量。主要分析测试在项目中的投入情况、成本情况、各个测试任务的分配情况等。
- ✓ 最后，根据对几个项目的工作量、成本以及测试任务占项目总测试投入的比例分析后，得到测试团队测试工作量估算的简易公式。可以根据这个简易的公式进行测试的估算，方便测试计划中关于工作量估算部分的编写，避免在估算工作量时缺乏依据。估算内容主要包括：测试总人力成本占项目总人力成本的比例及各项测试任务的工作量分配比例。

测试任务	比例	测试任务	比例	测试任务	比例
熟悉系统需求	5%	测试计划	3.5%	测试需求	7.5%
测试用例	15%	测试执行	40%	测试报告	4%
测试管理	7%	沟通、会议	4%	测试环境搭建	2%
性能测试	9%	验收测试	4%		

4、效率提高

- ✓ 测试负责人与开发负责人共同对项目进度进行商讨分析，作出合理的测试计划，并在测试执行过程中严格按照测试计划的进度和测试策略进行。
- ✓ 测试人员尽早的进入需求理解阶段，充分理解需求文档。
- ✓ 必要时做跟进测试，提高需求理解深度，可间接提高测试执行的效率；跟进测试，即系统测试之前的草稿版测试，需要与开发方沟通，让其协助来执行。跟进测试的目的不是发现 bug，而是熟悉系统环境，助于需求理解和测试设计。

- ✓ 尽量避免失败的接收测试。一次版本无法接收，会浪费很多人力和时间，还会影响测试人员的测试热情。
- ✓ 任务分配合理化。测试负责人应根据项目组成员的经验和能力个人因素，合理的分配测试任务，并将测试任务的模块和时间详细化，这样有助于提高整个项目的测试效率。
- ✓ 测试工作从某种角度看，会很容易掺杂个人主观意见，测试质量也受测试人员的责任感的因素影响，所以，培养良好的测试风格，提高测试人员的责任感，也能间接提高项目的测试效率。
- ✓ 慎重安排员工职务。管理者应该充分了解员工的特点，能力特长，个人气质，以让这些特质和分配的工作最佳配合，从而能够达到更好的效果。这样，员工也容易从高效的工作种找到乐趣，满足感，成就感，得到锻炼，同时很好地完成工作任务。
- ✓ 设定高绩效工作标准：
 - ◇ 给员工制定一个一般人都很容易达到的“低标准”，让员工觉得，你认为他能力不强，即使他努力达到了标准，也很难从工作中得到满足感，因为这个工作标准其它人也很容易就达到。
 - ◇ 低的工作标准，让大家很难打起精神努力工作，因为这很容易让人觉得，这个事情根本不重要，它对公司，对部门重要性不够，所以他们也不会将很大的精力投入到管理层认为不重要的工作中去，即使做好了，也是多余，因为这项工作本身部要求那么好。
 - ◇ 很多员工时间长了，很容易在潜意识中觉得，你分配给他的工作不重要，那么就意味着，他对这个团队，对这个部门也不重要，他大好的青春应该能够创造不俗的成绩，产生希望寻求实现自己价值的团队。本来最求成功的好员工，在这个的低标准工作中被浪费了，可惜。
- ✓ 提供员工自我控制方面的信息。这里的信息是指：让员工知道自己所负责的工作，对公司，对团队成功具有哪方面的意义。如果做砸了，对公司的损失是什么，对团队影响有多坏。这样，很容易让员工对工作产生很强的责任感。还有，这项工作达到何种程度可以认为完成出色，这样，他可以自行按照高要求高效完成工作，以达到高绩效。这些信息确实是非常必要。
- ✓ 提供员工参与机会以培养管理者的愿景。要让员工具有和团队一起成功的自豪感，成就感，这样，他会站在团队集体利益的立场上看待自己的工作。那么如何赋予员工这种自豪感，成就感？当员工确实做了值得骄傲的时期是，他们才会觉得骄傲，否则就是不诚实，反而具有杀伤力。只有当员工确实有成就感时，他们才会有成就感，也只有当他们承担了重要的任务时，才会真正觉得自己对团队重要。真正的自豪感，成就感和受重视感是莫基于积极，负责地参与有关自己工作和团队管理的决策。让员工觉得团队的成功有他重要的贡献，如果他失败了，那么团队就会受牵连，他是团队成功不可缺少的一份子，他和团队是紧密联系在一起，那么他就会对团队产生强烈的责任感，从而高绩效，高要求地完成自己的工作。

5、测试团队评估

测试团队评估目标：高效的提高软件测试用例的覆盖率。高效提高软件测试用例的覆盖度可分解为三个指标：

- ✓ 通过测试用例的复用。
- ✓ 以数据驱动的形式自动化测试用例。
- ✓ 通过有效的测试用例设计方法扩大覆盖率。

软件测试用例覆盖率提高判断条件：

- ✓ 现有测试用例进行了有效的管理，建立测试用例基线。
- ✓ 明确测试用例编写的颗粒度，测试用例颗粒度可以跟代码行数对应，可以跟功能点对应，组织内部测试用例颗粒度要统一，保证所有人员大致是一致的。
- ✓ 单个功能点都进行了有效的规整，确定最小测试范围。保证单一个功能点进行了规整。
- ✓ 提高的方式放在功能点的组合和测试数据的增加上。