

# 读后感软件有哪些(精选5篇)

读后感，就是看了一部影片，连续剧或参观展览等后，把具体感受和得到的启示写成的文章。这时候最关键的读后感不能忘了。那要怎么写好读后感呢？这里我整理了一些优秀的读后感范文，希望对大家有所帮助，下面我们就来了解一下吧。

## 读后感软件有哪些篇一

这几天来，涂老师没有像以前上课那样有详细的讲解和细心的提示，只能靠我们自己对photoshop学习的基础来自己完成制作。这次实训不仅仅是对我们的考察，也是一个自我总结，查漏补缺的难得的机会。经过这五天的学习，巩固和强化了我们的基本知识和基本技能，尤其是photoshop的基本知识和操作技能。

photoshop作品要做的好就必须有一个好的创意。我觉得学习photoshop不是在于把所有的工具都要深入去了解，只要把自己常用的会用就可以了，因为只有有创意，才可以做出一副好的作品来。色彩应用是图形图像处理 and 制作的一个重要环节，色彩应用搭配的好能让人产生一种舒适的感觉，作品的美感也由此而生。相反如色彩应用搭配不当，则会让人产生不想看的心理，作品也就谈不上什么感染力。色彩的应用搭配不仅要平时留心观察身体的事物，还在于多练习。留心观察才会知道什么地方用什么色彩能达到最好效果。因此我们在设计的过程中非常重视色彩的搭配。可色彩应用搭配不可能一下子学好，而是一个比较慢长的过程。所以在实训的过程中，我们也遇到了很多的困难。

但我们在实习的过程中也很开心，当遇到问题时，我们都会相互帮助，当有困难时，大家一起解决，那种解决困难后的成功的喜悦无法用言语来形容。其实，不管最后的结果会怎

样，我们都是成功的，因为我相信过程比结果更重要，在实习的过程中，我们就一直在收获。所以，我们没有白白浪费时间，我们都是成功的。

另外，我也明白了，无论做什么事，我们都得做一个有心人，有信心，有耐心，有恒心；同时，我们也要培养独立分析问题，解决问题的能力。

最后给本次实训提一些小建议：希望老师对我们的作品进行点评，让我们知道自己的作品好在哪里，哪里还有不足，这样才能不断的进步。

## 读后感软件有哪些篇二

在设计阶段采用bim技术，可以对建筑设计进行分析与优化，确保设计的可施工性。首先要建立相关建筑项目的3d设计模型，包括建筑、结构及建筑设备等；其次，基于建立的3d设计模型，可进行设计检测、协同修改。设计检测可根据需要设定相关参数，确定检测范围，从而检测设计冲突问题，可施工性问题。对发现的问题及时进行分析和沟通，从而及时、有效地解决问题，得出合理的施工图。通过建立的三维设计模型，实现工程的三维设计。能够根据3d模型自动生成各种图形和文档，而且始终与模型逻辑相关。当模型发生变化时，与之关联的图形和文档将自动更新。设计过程中所创建的对象存在着内在的逻辑关联关系，当某个对象发生变化时，与之关联的对象随之变化。通过建立模型，实现不同专业设计之间的信息共享。各专业cad系统可从信息模型中获取所需的设计参数和相关信息，不需要重复录入数据。避免数据冗余、歧义和错误。实现各专业的之该对象会随之更新。通过建立模型，实现虚拟设计和智能设计，实现设计碰撞检测、能耗分析、成本预测等。通过对结构上的分析，利用工具软件建立3d模型，完成结构条件图，对结构进行分析得出合理的结构施工图；通过对能耗进行分析，可以对建筑物的能效进行

分析和计算，从而对节能、经济、绿色进行更优化的设计。

在施工中，采用**bim**技术对建筑项目进行虚拟施工（虚拟施工就是：先试后建），对项目施工方案进行模拟、分析和优化，从而发现施工中可能出现的问题，在施工前就采取预防措施，直至获得最佳的施工方案，尽最大可能实现“零碰撞、零冲突、零返工”，从而大大降低返工成本，减少资源浪费与冲突及安全问题，指导真实的施工。随着项目复杂性的增大，如规模的增大以及建筑系数数量和复杂性的增加，传统的2d不能直观、准确的表达建筑物。bim技术的应用，使人们将建筑物由2d变化到nd的表达模式，将2d的传统建模发展到4d、5d的信息建模方式（增加了发展时间上、造价控制上的模拟），解决了对缩短工期的要求和控制造价的压力。

在施工中通过采用**bim**技术，对现场施工等进行模拟分析。通过对工程造价的分析，运用“零库存”的生产管理方式，限额领料施工，最大程度发挥业主资金的效益；通过工序的分析、bim模型和进度计划软件的数据集成，实时监控工程进度，实时调整对环境的影响等全面的可建性模拟分析；通过对冲突碰撞检查分析，建造前期对个专业的碰撞问题进行模拟，生成与提供可整体化协调的数据，解决传统的二维图纸会审耗时长、效率低、发现问题难的问题。

以上是本人参加此次**bim**培训后的心得和体会，有些地方说的可能不是很深入，希望大家能够与我多多交流，共同探讨**bim**的未来。

### 读后感软件有哪些篇三

我在一家国企做了三年软开，我是本科学数学的，编程全是自学的，参与了4-5个项目都是核心开发人员，有的项目一个人都是独立做的。也做过项目负责人。工作算是得心应手。可是我们软件部门开发过程不正规，基本是作坊式开发。感觉的到一定

程度很难再提高.

技术方面, 我一开始用c++c#做windows点net客户端的开发, 后来做了一年的android开发,sdkndk都用过. 自己私下做一些ios的小作品. 我是比较踏实肯干的那种, 做开发基本是用到哪里学到哪里. 有些基本还不扎实. 不过自己一直在学, 买了不少书有的书还没翻过. 我的c++c#java都掌握得还可以c++用的最多应该是最强的, 那些经典书看一些(c++primerc++深思路thinkinginc++ )最近每天看一点stl想把c++学精. 设计模式和重构技术也都用过一些. 学设计模式感觉对编程的帮助最大, 能站在另一个高度。

## 软件工程实习心得

转眼之间, 两个月的实习期即将结束, 回顾这两个月的实习工作, 感触很深, 收获颇丰。这两个月, 在领导和同事们的悉心关怀和指导下, 通过我自身的不懈努力, 我学到了人生难得的工作经验和社会见识。我将从以下几个方面总结软件工程岗位工作实习这段时间自己体会和心得:

### 一、努力学习, 理论结合实践, 不断提高自身工作能力。

在软件工程岗位工作的实习过程中, 我始终把学习作为获得新知识、掌握方法、提高能力、解决问题的一条重要途径和方法, 切实做到用理论武装头脑、指导实践、推动工作。思想上积极进取, 积极的把自己现有的知识用于社会实践中, 在实践中也才能检验知识的有用性。在这两个月的实习工作中给我最大的感触就是: 我们在学校学到了很多理论知识, 但很少用于社会实践中, 这样理论和实践就大大的脱节了, 以至于在以后的学习和生活中找不到方向, 无法学以致用。同时, 在工作中不断的学习也是弥补自己的不足的有效方式。信息时代, 瞬息万变, 社会在变化, 人也在变化, 所以你不学习, 你就会落伍。通过这两个月的实习, 并结合软件

工程岗位工作的实际情况，认真学习的软件工程岗位工作各项政策制度、管理制度和工作条例，使工作中的困难有了最有力地解决武器。通过这些工作条例的学习使我进一步加深了对各项工作的理解，可以求真务实的开展各项工作。

## 二、围绕工作，突出重点，尽心尽力履行职责。

在软件工程岗位工作中我都本着认真负责的态度去对待每项工作。虽然开始由于经验不足和认识不够，觉得在软件工程岗位工作中找不到事情做，不能得到锻炼的目的，但我迅速从自身出发寻找原因，和同事交流，认识到自己的不足，以至于迅速的转变自己的角色和工作定位。为使自己尽快熟悉工作，进入角色，我一方面抓紧时间查看相关资料，熟悉自己的工作职责，另一方面我虚心向领导、同事请教使自己对软件工程岗位工作的情况有了一个比较系统、全面的认知和了解。根据软件工程岗位工作的实际情况，结合自身的优势，把握工作的重点和难点，尽心尽力完成软件工程岗位工作的任务。两个月的实习工作，我经常得到了同事的好评和领导的赞许。

## 三、转变角色，以极大的热情投入到工作中。

从大学校门跨入到软件工程岗位工作岗位，一开始我难以适应角色的转变，不能发现问题，从而解决问题，认为没有多少事情可以做，我就有一点失望，开始的热情有点消退，完全找不到方向。但我还是尽量保持当初的那份热情，想干有用的事的态度，不断的做好一些杂事，同时也勇于协助同事做好各项工作，慢慢的就找到了自己的角色，明白自己该干什么，这这就是一个热情的的问题，只要我保持极大的热情，相信自己一定会得到认可，没有不会做，没有做不好，只有你愿不愿意做。转变自己的角色，从一位学生到一位工作人员的转变，不仅仅是角色的变化，更是思想观念的转变。

## 四、发扬团队精神，在完成本职工作的同时协同其他同事。

在工作间能得到领导的充分信任，并在按时完成上级分配给我的各项工作的同时，还能积极主动地协助其他同事处理一些内务工作。个人的能力只有融入团队，才能实现最大的价值。实习期的工作，让我充分认识到团队精神的重要性。

团队的精髓是共同进步。没有共同进步，相互合作，团队如同一盘散沙。相互合作，团队就会齐心协力，成为一个强有力的集体。很多人经常把团队和工作团体混为一谈，其实两者之间存在本质上的区别。优秀的工作团体与团队一样，具有能够一起分享信息、观点和创意，共同决策以帮助每个成员能够更好地工作，同时强化个人工作标准的特点。但工作团体主要是把工作目标分解到个人，其本质上是注重个人目标和责任，工作团体目标只是个人目标的简单总和，工作团体的成员不会为超出自己义务范围的结果负责，也不会尝试那种因为多名成员共同工作而带来的增值效应。

## 五、存在的问题。

几个月来，我虽然努力做了一些工作，但距离领导的要求还有不小差距，如理论水平、工作能力上还有待进一步提高，对软件工程岗位工作岗位还不够熟悉等等，这些问题，我决心实习报告在今后的工作和学习中努力加以改进和解决，使自己更好地做好本职工作。

## 读后感软件有哪些篇四

对于学习软件工程这门课程，我认为有许多东西要学习。其实在我看来学习这门课程的精髓是学习一种方法。是一个如何去分析和处理问题的过程，应该说其范畴已经远远不止局限于该门课程，成为了一个综合的一个能够解决问题的思想集合。读完软件工程案例教程这本书，我觉得自己受益匪浅。

整本书的内容逻辑很清晰明了，由浅入深循序渐进，首先我就大概描述下我们所学的内容，第一章是从整体分析软件工

程这门学科的发展和所处的社会环境，接着后面的几章深入分析了软件开放过程和模式、软件项目管理、计算机工程、需求分析、结构化分析建模以及基于uml面向对象分析建模和测试等。对于这本书我主要对需求分析和测试比较感兴趣，在这我要着重的谈一些自己的心得体会以及自己的看法。

## 需求分析的重要性

一款成功的软件是建立在成功的需求分析之上的，而高质量的需求来源于用户与开发人员之间有效的沟通与合作。当用户有一个问题可以用计算机系统来解决，而开发人员开始帮助用户解决这个问题，沟通就开始了。由此我们可以看出需求分析的重要性。

需求获取可能是最困难、最关键、最易出错及最需要沟通交流的活动。对需求的获取往往有错误的认识：用户知道需求是什么，我们所要做的就是和他们交谈从他们那里得到需求，只要问用户系统的目标特征，什么是要完成的，什么样的系统能适合商业需要就可以了，但是实际上需求获取并不是想象的这么简单，这条沟通之路布满了荆棘。首先需求获取要定义问题范围，系统的边界往往是很难明确的，用户不了解技术实现的细节，这样造成了系统目标的混淆。

其次是对问题的理解，用户对计算机系统的能力和限制缺乏了解，任何一个系统都会有很多的用户或者不同类型的用户，每个用户只知道自己需要的系统，而不知道系统的整体情况，他们不知道系统作为一个整体怎么样工作效率更好，也不太清楚那些工作可以交给软件完成，他们不清楚需求是什么，或者说如何以一种精确的方式来描述需求，他们需要开发人员的协助和指导，但是用户与开发人员之间的交流很容易出现障碍，忽略了那些被认为是xxx很明显xxx的信息。最后是需求的确认，因为需求的不稳定性往往随着时间的推移产生变动，使之难以确认。为了克服以上的问题，必须有组织的执行需求的获取活动。

## 需求分析的原则

(1) 需求分析必须能够表达和理解问题的数据域和功能域。数据域包括数据流、数据内容和数据结构，而功能域反映上述 3 方面的控制信息。

(2) 需求分析要把一个复杂问题按功能进行分解并逐层细化。通常，软件系统要处理的问题如果太大、太复杂就很难理解，若划分成几部分，并确定各部分间的接口，就可完成整体的功能。在需求分析过程中，软件系统的用户需求中的数据、功能和行为都应细化。

(3) 需求建模。模型可以帮助系统分析人员更好地理解软件系统的数据、功能和行为，这些模型是软件工程中下一阶段进行系统设计的基础。

## 需求分析的注意事项

(1) 确定详细的需求，否则经费就算不准。经费估计错误的原因多为：用户需求频繁变动、遗漏重要需求、与用户交流不够、需求规格说明书质量低劣、需求分析不充分等。

(2) 在编写需求规格说明书之前，应明确要解决的问题。在试图解决问题之前，要保证已考察了全部可替代的方案。要搞清哪地方有问题，真正的问题出在哪里。这样，在编写需求规格说明书时做到有的放矢，把存在的问题暴露出来。

(3) 立即确定需求，并记录下该需求的背景。没有明确问题，就进行下一步的设计，想回避矛盾，可能会带来更大的问题。用户不确定需求，软件设计人员自己决定需求，将会带来严重的问题。为了避免将来可能出现的问题和软件工程项目能够尽快地进入到下一个阶段的系统设计中，要尽可能迅速地把用户需求确定下来。任何决定总比没有决定要好。



(4) 一旦在需求规格说明书中发现问题，立即改正。如果把存在的问题拖延到系统设计阶段去改正，就可能要花数倍的时间和精力才能纠正同一错误。

(5) 在众多用户需求中确定各个需求的优先顺序，并确定可能存在的子集，以便为软件设计、实施和项目管理等后续阶段提供有利条件。

(6) 需求分析时，不要进行系统设计的工作。需求分析的主要目的是确定软件系统的外部特征，充分反映软件系统应有的面貌，便于让软件设计人员根据用户需求，去全面地考虑软件系统的体系结构、算法等。在需求分析阶段要集中精力解决用户需求存在的问题，尽可能避免产生遗留问题。

(7) 对于复杂的软件系统，要从多种视角进行需求分析。根据软件系统的本质，切合实际地组织多种视角的需求。例如，可从根据用户的类型，或根据响应的类型，或根据对象的软件工程案例教程类型，或根据系统的模式等视角来组织用户需求。通过多个视角来研究用户需求问题，把可得到的不同的“投影”组合起来形成完整系统的描述。当试图从整体观点来描述软件系统发生困难，或者有可能发生错误，或者很有可能遗失软件系统的某些特性。而从不同的视角来描述软件系统，因为每个视角限制了研究的范围并能够将注意力集中于此，所以很容易保证所研究的问题是真正完整的。

(8) 重视形式化方法，但不放弃自然语言。为了用户需求表达的精确性和方便用户的可理解性，一个好方法是把自然语言的表达与形式化规格说明并立，互相对照，而且在一般情况下，先用自然语言写出，再给出它的形式模型。

(9) 用户需求中不应存在“待确定”的条款。如若有这种需要，应同时说明：何时由谁来解决该问题。

用户需求的类型

需求分析是从用户最初的非形式化需求到满足用户要求的软件产品的映射过程。它实际上是一个对用户意图不断进行揭示和判断的过程，其目的在于细化、精化软件的作用范围，确定拟开发软件的功能和性能、约束、环境等。可将用户的需求分为两大类：功能性需求和非功能性需求。

(1) 功能性需求。功能性需求主要说明了系统各功能部件与环境之间的相互作用的本质，即拟开发软件在职能上实际应做到什么。一般来说，它是用户最主要的需求，通常包括系统的输入、系统能完成的功能、系统的输出以及其他反应。在功能性需求中还应包括备选功能的定义识别。

(2) 非功能性需求。非功能性要求主要从各个角度对所考虑的可能的解决方案起约束和限制作用。

## 需求分析的方法

在软件工程中，常用的需求分析方法有面向数据流的结构化分析方法（简称 sa）和面向对象的分析方法（简称 ooa）此外，还有以用户为中心的需求分析方法。这些方法都采用图文结合的方式，可以直观地描述软件的逻辑模型。这里仅介绍结构化分析方法和以用户为中心的需求分析方法。

## 软件测试概述

软件本身无形态，它是复杂的知识高度密集的逻辑产品，其中不可能没有错误。软件实施工程过程中必须伴随着软件质量保证的活动，而软件测试是主要活动之一。在开发软件的过程中，人们使用了许多保证软件质量的方法分析、设计和实现软件，但难免还会在工作中犯错误。这样，在软件产品中就会隐藏许多错误和缺陷。对于规模大、复杂性高的软件更是如此。在这些错误中，有些是致命的错误，如果不排除，就会导致生命与财产的重大损失。

## 软件测试的目的

测试的目的是“说明程序能正确地执行应有的功能”，还是“表明程序没有错误”？基于不同的立场，存在着两种完全不同的测试目的。从用户的角度出发，普遍希望通过软件测试暴露软件中隐藏的 errors 和缺陷，以考虑是否可以接受该产品。而从软件开发者的角度出发，则希望测试成为表明软件产品中不存在错误的过程，验证该软件已正确地实现了用户的要求，确立人们对软件质量的信心。因此，他们会选择那些导致程效概率小的测试用例，回避那些易于暴露程序错误的测试用例。同时，也不会刻意去检测、排除程序中可能包含的副作用。显然，这样的测试对完善和提高软件质量毫无价值。因为在程序中往往存在着许多预料不到的问题，可能会被疏漏，许多隐藏的 errors 只有在特定的环境下才可能暴露出来。如果不把着眼点放在尽可能查找错误这样一个基础上，这些隐藏的 errors 和缺陷就查不出来，会遗留到运行阶段中去。如果站在用户的角度，替他们设想，就应当把测试活动的目标对准揭露程序中存在的 errors。在选取测试用例时，考虑那些易于发现程序错误的 data。

## 软件测试的原则

而应当把它贯穿到软件开发的各个阶段中。在需求分析阶段就应该制订测试计划，以保证每个需求，每个设计单元都是可测试的，便于测试。坚持在软件开发的各个阶段的技术评审，这样才能在开发过程中尽早发现和预防 errors，把出现的 errors 克服在早期，杜绝某些隐患，提高软件质量。

(2) 测试用例应由测试输入 data 和与之对应的预期输出结果这两部分组成。测试以前应当根据测试的要求，选择在测试过程中使用的测试用例 [test case] 测试用例主要用来检验程序员编制的程序，因此不但需要测试的输入 data，而且需要针对这些输入 data 的预期输出结果。如果对测试输入 data 没有给出预期的程序输出结果，那么就缺少了检验实测结果的

基准，就有可能把一个似是而非的错误结果当成正确结果。

(3) 程序员应避免检查自己的程序。测试工作需要严格的作风、客观的态度和冷静的情绪。自己测试自己的软件不容易发现错误，程序员应避免测试自己的程序。测试是一种“挑剔性”的行为，人们常常由于各种原因具有一种不愿否定自己工作的心理，认为揭露自己程序中的问题总不是一件愉快的事，这一心理状态就成为测试自己程序的障碍。心理状态和思维定式是测试自己程序的两大障碍，应由别人或另外的机构来测试程序员编写的程序。另外，程序员对软件规格说明理解错误而引入的错误则更难发现。如果由别人来测试程序员编写的程序，可能会更客观、更有效，并更容易取得成功。要注意的是，这点不能与程序的调试[debugging]互相混淆，调试由程序员自己来做可能更有效。

(4) 在设计测试用例时，应当包括合理的输入条件和不合理的输入条件。合理的输入条件是指能验证程序正确的输入条件，而不合理的输入条件是指异常的、临界的、可能引起问题变异的输入条件。在测试程序时，人们常常倾向于过多地考虑合法的和期望的输入条件，以检查程序是否做了它应该做的事情，而忽视了不合法的和预想不到的输入条件。事实上，软件在投入运行以后，用户的使用往往不遵循事先的约定，使用了一些意外的输入，如用户软件工程案例教程 在键盘上按错了键或打入了非法的命令。如果开发的软件遇到这种情况时不能做出适当的反应，给出相应的信息，那么就容易产生故障，轻则给出错误的结果，重则导致软件失效。因此，软件系统处理非法命令的能力也必须在测试时受到检验。

## 读后感软件有哪些篇五

接触广联达安装算量软件算起来也快一年了，这一年来通过广联达客服人员及同事的帮助基本上对广联达算量软件有一些心得体会，和大家分享：

## 经验和方法

### 1、cad基础很重要

实践证明cad绘图能力是广联达安装算量的基础，要学习广联达安装算量软件首先需要一定的autocad知识和能力，需要一定作图能力，一般来说，如果对autocad有一定的基础安装算量软件学起来较容易些，效果较好！

### 2、循序渐进

整个学习过程应采用循序渐进的方式，首先需要先了解软件算量的基本知识，如软件如何对工程量进行计算、如何进行分类、如何进行汇总、如何进行系统设置，通过反复观看教学视频，反复练习使自己能由浅入深，由简到繁地掌握安装算量软件。

### 3、学以致用

在学习安装工算量命令时始终要与实际工程应用相结合，不要把主要精力花费在各个命令孤立地学习上；要把学以致用的原则贯穿整个学习过程，以使自己对命令有深刻和形象的理解，有利于培养自己应用软件独立完成算量的能力。

### 4、熟能生巧

### 5、常见问题要弄懂

在学习软件中对于我们经常出现的问题，要对照教学视频、对照实际的工程进行练习，总之要通过各种途径对软件中一些通用的设置、操作要熟记于心，作为初学者弄懂这些问题，很有必要，对提高工程量的准确的和效率很有帮助。

学习过程中，要特别注意纠正自己的一些平时不要的习惯。

严格做到层次分明，规范识别步骤。我的体会是：养成良好的习惯，受益匪浅。

## 软件学习心得体会五 access软件的学习心得体会(492字)

不知不觉间，已经上了十次计算机课了，我对研究生的计算机课程—access20xx很感兴趣，并且通过老师的讲解已经取得了很大的进步。虽然作为一个初学者，在学习过程中会遇到一些难题，但是我都努力去克服，多问老师多练习，争取领会老师所讲的内容。

原来我对“access20xx”并不了解，但是通过这几个月的学习，我开始对这门课程有所了解。首先，它的功能非常强大，它是当今应用最广泛的计算机技术之一，可以组织、储存和管理任何类型和任意数量的信息。它在很多地方得到广泛使用，不仅在医疗部门，还包括学校、公司等等，例如很多企业用数据库来制作处理数据的桌面系统。它也常被用来开发简单的web应用程序。

其次，对于一名中医学研究生来说“access20xx”课程也是非常实用的，因为数据库技术可以应用于中医药领域的很多方面，如中医文献、中医证候、中药有效成分、方剂的配伍规律研究等，这些领域都与我们中医学生密切相关的，所以对以后的工作非常有帮助。

总之，我一定要好好学习这门课程，虽然有点儿难，但是，我相信功夫不负有心人，通过自身的努力一定能把它学好。最后还要感谢老师平时耐心的讲解让我理解起来更加容易！