（1）可行性分析   
可行性分析是系统在正式立项之前必须进行的一项工作，它的目的不是为了分析软件开发过程中的问题，也不是为了解决软件开发过程中可能存在的问题，而是确定软件系统是否有价值做、是否能够以尽可能小的代价在尽可能短的时间内解决问题。

具体而言，在可行性分析阶段，要确定软件的开发目标与总的要求，所以在做可行性分析的时候，一般需要考虑技术是否可行、经济效益是否可行、用户操作是否可行、法律与社会是否可行等。例如，对于一个超市商品价格查询系统而言，就需要调查顾客是否希望使用这样的软件，超市商品价格来源是哪里？技术上是否能够实现等？

可行性分析一般都由战略专家执行，该阶段的文档成果为《可行性分析报告》

（2）需求分析   
需求分析是软件项目成败的关键，它是回答客户做什么的问题，是一个对用户的需求进行正确加工、正确理解，然后把它用软件工程开发语言表达出来的过程。需求分析不仅仅是用户需求，也应该死开发中遇到的所有的需求。例如，在做需求分析时，首先需要弄清楚该项目的目的是为了解决什么问题；其次弄清楚测试案例中应该输入什么数据，最后就是弄清楚哪些人需要使用本系统等。

本阶段的基本任务是和用户一起确定要解决的问题，建立软件的逻辑结构，编写需求规格说明书文档并最终得到用户的认可。需求分析的主要方法有结构化分析方法、数据流程图和数据字典等方法。本阶段一般能形成软件需求说明书、数据要求说明书以及初步的用户手册。

需求分析需要领域专家与系统架构师都参与进来，阶段性文档成果为《需求分析说明书》等。

（3）设计   
软件设计的主要任务是将软件分解成模块，即实现某个功能的数据和程序说明、可执行程序的程序单元。这个模块可以是一个函数、过程、子程序、一段带有程序说明的独立的程序和数据，也可以是可组合、可分解以及可更换的功能单元。

软件设计可以分为概要设计和详细设计两个阶段。   
概要设计就是结构设计，其主要目标就是给出软件的模块结构。在详细设计中，首先就是要设计出模块的程序流程、算法和数据结构，其次是设计数据库，常用方法还是结构化程序设计方法。

该阶段的文档成果有《概要设计说明书》、《业务用例文档》、《详细设计说明书》、《技术用例文档》等

（4）编码与实现   
编码是指把软件设计转换成计算机可以接受的程序，即写成以某一程序设计语言表示的“源程序清单”。程序设计语言可以是C、C++、C#或Java等。 当前软件开发中除在专用场合，已经很少使用20世纪80年代的高级语言了，取而代之的是面向对象程序设计语言，而且面向对象的开发语言和开发环境大都合为一体，大大提高了开发的速度。由面向对象的开发语言开发的项目，系统的可扩展性与可维护性都大大增强。

该阶段的文档成果有《接口文档》、《关键算法文档》等。

（5）测试   
软件测试贯穿于软件开发的整个过程，它的目的是以较小的代价发现尽可能多的错误。要实现这个目标关键在于根据软件开发各阶段的规格说明和程序的内部结构精心设计一套出色的测试用例（测试数据和预期的输出结果组成了测试用例），利用这些测试用例进行测试，从而发现程序中存在的错误和bug。不同的测试方法有不同的测试用例设计方法，常用的测试方法有白盒测试和黑盒测试。

白盒测试是一种测试单元内部如何工作的方法，目的是通过检查软件内部的逻辑结构，对软件中逻辑路径进行覆盖的测试，而黑盒测试不考虑程序的内部结构与特性，只根据程序功能或程序的外部特性设计测试用例，这两种测试方法都是依据软件的功能或对软件的行为描述，发现软件的接口、功能和结构错误。   
该阶段的文档成果有《单元测试报告》、《集成测试报告》、《系统测试报告》等。

（6）运行与维护   
虽然系统交付给用户，安装运行了，但是任何一个系统都不是一开始就能完全满足实际的应用需求，一般在交付使用后，还需要不断地进行再开发。而维护是指在已完成对软件的研制（分析、设计、编码和测试）工作并交付使用以后，对软件产品所进行的一些软件工程的活动。即根据软件运行的情况，对软件进行适当修改与维护，如完善性维护、适应性维护，以适应新的要求，以及纠正运行中发现的错误，编写软件问题报告、软件修改报告。一个系统的质量的高低和系统的分析、设计有很大的关系，和系统的维护也有很大的关系。   
需要注意的是，软件工程比较强调文档的重要性，所以每个阶段最好能够有文档保存，因为每个阶段建立在上一个阶段的基础之上，如果基础出了问题，后续阶段都可能会出现相应的问题，文档正好可以备查。