

需求规格说明书检查表

| | | |
|--|------------|----|
| Y: 是 TBD: 不确定 N: 不是 NA: 不适用 | | 备注 |
| 检查项 | Y/TBD/N/NA | |
| 清晰性 | | |
| 系统的目标是否已定义？ | | |
| 是否对关键术语和缩略语进行定义和描述？ | | |
| 所使用的术语是否和用户 / 客户使用的一致？ | | |
| 需求的描述是否清晰，不含糊？ | | |
| 是否有对整套系统进行功能概述？ | | |
| 是否已详细说明了软件环境 （共存的软件）和硬件环境（特定的配置）？ | | |
| 如果有会影响实施的假设情况，是否已经声明？ | | |
| 是否已经对每个业务逻辑进行输入、输出以及过程的详细说明？ | | |
| 完整性 | | |
| 是否列出了系统所必须的依赖、假设以及约束？ | | |
| 是否对每个提交物或阶段实施都进行了需求说明？ | | |
| 需求说明书是否已包括了主要的质量属性，例如有效性、高效性、灵活性、完整性、互操作性、可靠性、健壮性、可用性、可维护性、可移植性、可重用性和可测试性。 | | |
| 依从性 | | |
| 该文档是否遵守了该项目的文档编写标准？ | | |
| 一致性 | | |
| 需求说明是否存在直接相互矛盾的条目？ | | |
| 本需求说明书是否与相关需求素材一致？ | | |
| 可行性 | | |
| 所描述的所有功能是否必要并充分地满足了客户 / 系统目标？ | | |
| 需求说明书的描述的详细程度是否足以进行详细的设计？ | | |
| 已知的限制（局限）是否已经详细说明？ | | |
| 是否已确定每个需求的优先级别？ | | |
| 可管理性 | | |
| 是否将需求分别陈述，因此它们是独立的并且是可检查的？ | | |
| 是否所有需求都可以回溯到相应的需求素材，反之亦然？ | | |
| 是否已详细说明需求变更的过程？ | | |

概要设计检查表

| | |
|---|------------|
| Y: 是 TBD: 不确定 N: 不是 NA: 不适用 | |
| 检查项 | Y/TBD/N/NA |
| 清晰性 | |
| 是否所设计的架构，包括数据流，控制流和接口，被清楚地表达了？ | |
| 是否所有的假设、约束、策略及依赖都被记录在本文档了？ | |
| 是否定义了总体设计目标？ | |
| 完整性 | |
| 是否所有的以前的 TBD(待确定条目) 都已经被解决了？ | |
| 是否设计已经可以支持本文档中遗留的 TBD有可能带来的变更？ | |
| 是否所有的 TBD的影响都已经被评估了？ | |
| 是否仍存在可能不可行的设计部分？ | |
| 是否已记录设计时的权衡考虑？ 该文件是否包括了权衡选择的标准和不选择其它方案的原因？ | |
| 依从性 | |
| 是否遵守了项目的文档编写标准？ | |
| 一致性 | |
| 数据元素、流程和对象的命名和使用在整套系统和外部接口之间是否一致？ | |
| 该设计是否反映了实际操作环境（硬件、软件、支持软件）？ | |
| 可行性 | |
| 从进度、预算和技术角度上看该设计是否可行？ | |
| 是否存在错误的、缺少的或不完整的逻辑？ | |
| 数据使用 | |
| 所有复合数据元素、参数以及对象的概念是否都已文档化？ | |
| 是否还有任何需要的但还没有定义的数据结构，反之亦然？ | |
| 是否已描述最低级别数据元素？是否已详细说明取值范围？ | |
| 功能性 | |
| 是否对每一下级模块进行了概要算法说明？ | |
| 所选择的设计和算法能否满足所有的需求？ | |
| 接口 | |
| 操作界面的设计是否有为用户考虑（例如：词汇、使用信息和进入的简易）？ | |
| 是否已描述界面的功能特性？ | |
| 界面将有利于问题解决吗？ | |
| 是否所有界面都互相一致， 与其它模块一致， 以及和更高级别文档中的需求一致？ | |
| 是否所有的界面都提供了所要求的信息？ | |
| 是否已说明内部各界面之间的关系？ | |
| 界面的数量和复杂程度是否已减少到最小？ | |
| 可维护性 | |
| 该设计是否是模块化的？ | |

| | |
|--|--|
| 这些模块具有高内聚度和低耦合度？ | |
| 是否已经对继承设计、代码或先前选择工具的使用进行了详细说明？ | |
| 性能 | |
| 主要性能参数是否已被详细说明（例如：实时、速度要求、磁盘输入 / 输出接口等）？ | |
| 可靠性 | |
| 该设计能够提供错误检测和恢复（例如：输入输出检查）？ | |
| 是否已考虑非正常情况？ | |
| 是否所有的错误情况都被完整和准确地说明？ | |
| 该设计是否满足该系统进行集成时所遵守的约定？ | |
| 易测性 | |
| 是否能够对该套系统进行测试、演示、分析或检查来说明它是满足需求的？ | |
| 该套系统是否能用增量型的方法来集成和测试？ | |
| 可追溯性 | |
| 是否各部分的设计都能追溯到需求说明书的需求？ | |
| 是否所有的设计决策都能追溯到原来确定的权衡因素？ | |
| 所继承设计的已知风险是否已确定和分析？ | |

详细设计检查表

| | |
|---|------------|
| Y: 是 TBD: 不确定 N: 不是 NA: 不适用 | |
| 检查项 | Y/TBD/N/NA |
| 清晰性 | |
| 所有单元或过程的目的是否都已文档化？ | |
| 包括了数据流、控制流和接口的单元设计是否已清晰的说明？ | |
| 完整性 | |
| 是否已定义和初始化所有的变量、指针和常量？ | |
| 是否已描述单元的全部功能？ | |
| 是否已详细说明用来实现该单元的关键算法（例如：用自然语言或 PDL）？ | |
| 是否已列出该单元的调用？ | |
| 依从性 | |
| 该文档是否遵循了该项目已文档化的标准？ | |
| 是否采用了所要求的方法和工具来进行单元设计？ | |
| 一致性 | |
| 数据元素的命名和使用在整个单元和单元接口之间是否一致？ | |
| 所有接口的设计是否互相一致并且和更高级别文档一致？ | |
| 正确性 | |
| 是否处理所有条件 （大于、等于、小于零、switch/case ）? 是否存在处理 “ case not found ” 的条件？ | |
| 是否正确地规定了分支（逻辑没有颠倒）？ | |
| 数据使用 | |
| 是否所有声明的数据都被实际使用到？ | |
| 是否所有该单元的数据结构都被详细说明？ | |
| 是否所有修改共享数据 （或文件）的程序都考虑到了其它程序对该共享数据 （或文件）的存取权限？ | |
| 是否所有逻辑单元、时间标志和同步标志都被定义和初始化？ | |
| 接口 | |
| 接口参数在数量、类型和顺序上是否匹配？ | |
| 是否所有的输入和输出都被正确定义和检查？ | |
| 是否传递参数序列都被清晰的描述？ | |
| 是否所有参数和控制标志由已描述的单元传递或返回？ | |
| 是否详细说明了参数的度量单位、取值范围、正确度和精度？ | |
| 共享数据区域及其存取规定的映射是否一致？ | |
| 可维护性 | |
| 单元是否具有高内聚度和低耦合度 （例如：对该单元的更改不会在该单元有任何无法预料的影响并对其它单元的影响很小）？ | |
| 性能 | |
| 是否该单元的所有约束例（如过程时间和规模）都被详细说明？ | |
| 可靠性 | |

| | |
|--|--|
| 初始化是否使用到缺省值，缺省值是否正确？ | |
| 是否在内存访问的时候执行了边界检查（例如：数组、数据结构、指针等）来确保只是改变了目标存储位置？ | |
| 是否执行输入、输出、接口和结果的错误检查？ | |
| 是否对所有错误情况都发出有意义的信息？ | |
| 对特殊情况返回的代码是否和已规定的全局定义的返回代码相匹配？ | |
| 是否考虑到意外事件？ | |
| 易测性 | |
| 是否能够对每个单元进行测试、演示、分析或检查来说明它们是满足需求的？ | |
| 该设计是否包含检查点来帮助测试（例如：有条件的编译代码和数据声明测试）？ | |
| 是否所有的逻辑都能被测试？ | |
| 是否已描述测试程序、测试数据集和测试结果？ | |
| 可追溯性 | |
| 是否设计的每一部分都能追溯到其它项目文档的需求，也能追溯到更高级别文档的需求？ | |
| 是否所有的设计决定都能追溯到权衡考虑？ | |
| 单元需求是否都能上溯到更高级别的文档？更高级别文档的需求是否已经在单元中体现？ | |

测试计划检查表

| | |
|--|------------|
| Y: 是 TBD: 不确定 N: 不是 NA: 不适用 | |
| 检查项 | Y/TBD/N/NA |
| 完整性 | |
| 该测试计划是否详细说明测试的大体方法和策略？ | |
| 该测试计划是否详细说明所有测试活动的顺序？ | |
| 该测试计划是否描述了将使用的软硬件系统环境 ？ | |
| 该测试计划是否描述了测试活动中断和恢复的条件 / 情形？ | |
| 该测试计划是否为所有测试定义了成功标准？ | |
| 该测试计划是否充分地描述了被测试的功能？ | |
| 该测试计划是否明确地描述了不被测试的功能？ | |
| 该测试计划是否充分地描述了测试基线？ | |
| 对于阶段交付， 该测试计划是否有在每一阶段建立测试基线给下一阶段使用？ | |
| 该测试计划是否定义了足够和正确的衰退测试？ | |
| 依从性 | |
| 该测试计划是否依从了与开发有关的所有说明书、标准和文档？ | |
| 一致性 | |
| 是否已定义了测试顺序来匹配更高级别的文档所指定的集成顺序？ | |
| 该测试计划是否和更高级别的测试计划文档一致？ | |
| 正确性 | |
| 该测试计划的进入和退出条件是否实现？ | |
| 是否所有必须的驱动程序和桩（ stubs ）都已被定义且可利用来测试指定的功能？ | |
| 详细级别 / 程度 | |
| 测试案例是否完整覆盖了所有功能， 是否覆盖了被测试功能的正常执行情况？ | |
| 测试案例集是否覆盖了足够的非法和冲突的输入？ | |
| 测试案例集是否包括了足够的默认输入值的使用？ | |
| 测试案例集是否考虑到了足够数量的程序错误路径？ | |
| 易测性 / 可行性 | |
| 测试方法是否可行？ | |
| 是否所有被认为不可测的需求都被详细说明并说明原因？ | |
| 是否对获得测试软件、方法和工具分配了足够的时间并形成了进度计划？ | |
| 测试所要求的资源是否已经详细说明和估计？ | |
| 对于多次的构建（ builds ），是否已在前一构建的基础上确定所有的需求？ | |
| 测试所包含的所有人员的角色和职责是否都已详细说明？ | |
| 在已计划的测试人员之间是否存在进度冲突？ | |
| 可追溯性 | |
| 测试是否有执行 / 演示在适当级别的文档所说明的需求？ | |
| 测试验收标准是否可追溯到更高级别的文档？ | |

测试计划进程表

| 开发阶段 /测试阶段 | 单元测试 /承建方开发组 | 集成测试 /承建方开发组、测试组 | 确认测试 /承建方测试组 | 系统测试 /业主联合测试组 |
|------------|--------------------------------|-----------------------------------|-------------------------------------|---------------------------------|
| 软件需求分析 | 无 | 无 | 完成确认测试计划 | 完成系统测试计划 |
| 软件概要设计 | 无 | 完成软件集成测试计划 | 开始设计确认测试用例、编写确认测试说明 | 开始设计系统测试用例、编写系统测试说明 |
| 软件详细设计 | 完成软件单元测试计划 | 开始设计集成测试用例、编写集成测试说明 | | |
| 软件编码 | 编写软件单元测试说明、执行软件单元测试、编写软件单元测试报告 | | | |
| 软件测试 | 无 | 完成集成测试说明、执行集成测试、进行测试分析、编写软件集成测试报告 | 完成软件确认测试说明、执行软件确认测试、进行测试分析、编写确认测试报告 | 完成系统测试说明、执行系统测试、进行测试分析、编写系统测试报告 |