

需求评审检查表	
需求特性	检查内容
清晰性 / 无二义性	所有定义、实现方法是否清楚地表达了用户的原要求 在功能实现过程、方法和技术要求的描述上，是否背离了功能的实际要求 是否有不能理解或造成误解的描述
完整性	是否有一个内容表格，该表格包含了所有需求描述 是否所有的图形、表格都进行了标号 是否所有的需求项都进行了标号，并提供了索引 是否所有需求可以被定义的更细致，或简单 对于不清晰的信息是否进行了指出 是否存在有需求让你不舒服 是否所有与需求相关的设计约束都包含了 是否所有与需求相关的性能都包含了 是否所有与需求相关的属性都包含了 是否所有与需求相关的外部接口都包含了 是否所有与需求相关的通信都被包含了 是否所有与需求相关的硬件都被包含了 是否所有与需求相关的数据库都被包含了 是否所有与需求相关的软件都被包含了 是否所有与需求相关的输入和输出都被包含了 是否所有与需求相关的安装特性都被包含了 是否所有与需求相关的维护特性都被包含了 是否所有与需求相关的安全特性都被包含了 所有对其他需求的内部交叉引用是否正确 * 所有需求的编写在细节上是否都一致或者合适 需求是否能为设计提供足够的基础 是否包括了每个需求的实现优先级 需求定义是否包含了有关文件（指质量手册、质量计划以及其他有关文件）中所规定的需求定义所应该包含的所有内容 需求定义是否包含了有关功能、性能、限制、目标、质量等方面的所有需求 功能需求是否覆盖了所有非正式情况的处理 是否对各种操作模式（如正常、非正常、有干扰等）下的环境条件都作了规定 是否对所有功能与时间因素有关的方面都作了考虑 是否标识出了所有与时间因素有关的功能？它们的时间准则是否都说明了？时间准则的最大、最小执行时间是否都定义了 是否标识并定义了在未来可能会变化的需求 是否定义了系统所有的输入 是否标识清楚了系统输入的来源 是否标识出了系统的输出 是否说明了系统输入、输出的值域、单位、格式等 是否说明了如何进行系统输入的合法性检查 是否定义了系统输入、输出的精度 是否定义了系统性能的各个方面 在不同负载情况下，是否规定了系统的生产率 在不同情况下，是否规定了系统的响应时间 是否充分定义了有关人机界面的需求 是否对需求定义进行了可行性分析和相关文件（资料）是否已归档 是否对影响需求实现的因素进行了调查、调查结果是否已归档 是否有商业行为，分析结果是否已归档 是否详细描述了有关硬件、软件、操作人员、操作过程等方面的安全性

	<p>是否评估了本项目对用户、其他系统、环境的影响特性</p> <p>是否按完成时间、重要性对系统功能、外部接口、性能进行了优先排序</p>
兼容性	<p>界面需求是否使软硬件系统具有兼容性</p> <p>需求定义的文档是否满足项目文档编写标准？在矛盾时，是否有适当的标准可供选择</p>
一致性	<p>各个需求之间是否一致？是否有冲突和矛盾</p> <p>所规定的模型、算法和数值方法是否相容</p> <p>是否适用了标准的术语和定义形式</p> <p>需求是否与其软硬件操作环境相容</p> <p>是否说明了软件对其系统和环境的影响</p> <p>是否说明了环境对软件的影响</p> <p>所采用的技术是否与用户要求的技术一致</p>
正确性	<p>需求定义是否满足标准的要求</p> <p>算法和规则是否有科技文献或其他文献作为基础</p> <p>是否定义了对在错误、危险分析中所标识出的各种故障模式和错误类型所需的反应</p> <p>是否参照了有关的标准</p> <p>是否对每一个需求都给出了理由？理由是否充分</p> <p>对设计和实现的限制是否都有论证</p>
可行性	<p>需求定义是否使软件的设计、实现、操作和维护都可行</p> <p>所规定的模型、数值方法和算法是否对待解决问题合适？是否能够在相应的限制条件下实现</p> <p>是否能够达到相关质量的要求</p>
易修改性	<p>对需求定义的描述是否易于修改（如是否采用良好的结构和交叉引用表等）</p> <p>是否有冗余的信息？是否一个需求被定义了多次</p>
健壮性	<p>是否有容错的需求</p>
易跟踪性	<p>是否每个需求都具有惟一性并且可以正确地识别它</p> <p>是否可从上一阶段的文档中找到需求定义中的相应内容</p> <p>需求定义是否明确地表明前阶段中提出的有关需求和设计限制是否都被覆盖了</p> <p>需求定义是否便于向后继开发阶段查找信息</p>
可理解性	<p>最终产品的每个特性是否始终用同一个术语进行了描述</p> <p>是否每一个需求都只有一种解释</p> <p>功能性需求是否以模块方式描述的？是否明确地标识出了其功能</p> <p>是否有术语定义一览表</p> <p>是否使用了形式化或半形式化的语言</p> <p>语言是否有歧义性</p> <p>需求定义中是否只包含了必须的实现细节而不包含不必要的实现细节？是否过分细致了</p> <p>需求定义是否足够清楚和明确使其能够作为开发设计规约和功能性测试数据的基础</p> <p>需求定义的描述是否将对程序的需求和所提供的其他信息分离开来</p>
可测试性	<p>需求是否可以验证（即是否可以检验软件是否满足了需求）</p> <p>是否对每一个需求都指定了验证过程</p> <p>数学函数的定义是否使用了精确定义的语法和语义符号</p>
性能	<p>是否精确地描述了所有的性能需求和可容忍的性能降低程度？对每一个性能应包含两个方面的内容</p> <p>a. 在最坏情况的执行结果</p> <p>b. 本性能失效后，对系统产生的影响</p> <p>是否指定了所有期望的处理时间</p> <p>是否指定了数据传输的速度</p> <p>是否指定了系统的吞吐量</p>
功能	<p>是否清楚、明确的描述了所有的功能</p> <p>所有已描述的功能是否是必须的？是否能满足任务书或系统目标的要求</p>

接口	是否清楚地定义了所有的外部接口 是否清楚地定义了所有的内部接口 所有接口是否必须？各接口间的关系是否一致、正确
数据	在某异常数据（如条件、标志等）下，是否有尚未考虑到的结果 对异常数据产生的结果是否作了精确的描述
硬件	是否指定了最小内存需求 是否指定了最小存储空间需求 是否指定了最大内存需求 是否指定了最大存储空间需求
软件	是否指定了需要的软件环境 / 操作系统 是否指定了需要的所有软件设施 是否指定了所有要与系统一起允许的购买的软件产品
通信	是否指定了目标网络 是否指定了需要网络协议 是否指定了需要的网络能力 是否指定了需要的 / 估计的网络吞吐量 是否指定了估计的网络连接数量 是否指定了最小网路性能需求 是否指定了乐观的网路性能需求
可维护性	需求定义中是否包括了可行的系统维护方法 模块或子程序（过程）间的关系是否是松耦合的（即能否保证对某部分修改后，产生最小的连锁效应）
可靠性	是否为每个需求指定了软件失效的结果 是否指定了特定失效的保护信息 是否指定了特定的错误检测策略 是否指定了错误纠正策略
其他	是否所有的需求都是名副其实的需求而不是设计或实现方案 是否确定了对时间要求很高的功能并且定义了它们的时间标准 是否已经明确地阐述了国际化问题 使用实例是否是独立的分散任务 使用实例的目标或价值度量是否明确 使用实例给操作者带来的益处是否明确 使用实例是否处于抽象级别上，而不具有详细的剧情 使用实例中是否不包含设计和实现的细节 是否记录了所有可能的可选过程 是否记录了所有可能的例外条件 是否存在一些普通的动作序列可以分解成独立的使用实例 是否简明确写、无二义性和完整地记录了每个过程的对话 使用实例中的每个操作和步骤是否都与所执行的任务相关 使用实例中定义的每个过程是否都可行 使用实例中定义的每个过程是否都可验证

要求

方面的所有需求

条件都作了规定

表等)
