### 文档审查

#### 文档审查

表1‑2文档审查测试项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试项名称 | 文档审查 | 测试项标识 | XQ\_DC\_WDSC | 优先级 | 高 |
| 追踪关系 | 隐含需求 | | | | |
| 需求描述 | 依据相关要求，逐项检查被测文档的完整性、一致性和准确性是否满足要求 | | | | |
| 测试手段 | 静态测试 | | | | |
| 测试项描述 | **1.文档审查（XQ\_DC\_WDSC\_SU01）**  本软件文档审查包括内容如下：  1）软件需求规格说明  2）软件设计文档  3）软件接口需求规格说明  4）软件接口设计说明  5）软件用户手册 | | | | |
| 测试方法 | **1.文档审查（XQ\_DC\_WDSC\_SU01）**  测试人员人工阅读文档，依据文档检查单对软件文档进行审查，文档审查工作内容包括：  1）审查软件文档内容是否完整；  2）审查软件文档描述是否正确；  3）审查软件文档格式是否规范；  4）审查软件文档是否文文一致  按照附录的需求规格说明审查单，对被测软件的需求规格说明进行审查；  按照附录的软件设计文档审查单，对被测软件的设计说明文档进行审查；  按照附录的用户手册审查单，对被测软件的用户手册进行审查 | | | | |
| 充分性要求 | 对所有被测软件文档按照文档检查单逐项进行审查 | | | | |
| 通过准则 | 被测软件文档内容完整，描述准确，格式规范，文档文文一致 | | | | |

### 静态分析

#### 静态分析

表1‑2静态分析测试项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试项名称 | 静态分析 | 测试项标识 | XQ\_SA\_JTFX | 优先级 | 中 |
| 追踪关系 | 隐含需求 | | | | |
| 需求描述 | 根据相关要求，利用静态分析工具对被测软件全部源程序进行控制流分析、数据流分析进行分析，并统计软件质量度量信息，给出软件源代码检查结果 | | | | |
| 测试手段 | 静态测试 | | | | |
| 测试项描述 | **1.静态分析（XQ\_SA\_JTFX\_SU01）**  对被测软件全部源程序进行静态分析，对控制流、数据流进行分析，验证软件是否满足控制流和数据流要求，并依据质量特性需求统计质量度量信息 | | | | |
| 测试方法 | **1.静态分析（XQ\_SA\_JTFX\_SU01）**  使用LDRA TestBed软件和Klocwork软件工具对被测软件全部源程序进行静态分析，依据附录的审查单对源程序进行检查。  1）使用静态分析工具统计软件质量度量信息，包含：  （1）软件总注释率不小于20%（注释行数/软件规模\*100%）；  （2）模块的平均规模不大于200行（模块代码行数之和/模块数）；  （3）模块的平均圈复杂度不大于10（模块圈复杂度之和/模块总数）；  （4） 模块的平均扇出数不大于7（模块扇出数之和/模块总数）。  2）使用静态分析工具结合人工分析对控制流和数据流进行分析，验证软件是否满足控制流和数据流要求。 | | | | |
| 充分性要求 | 对软件全部源程序进行进行质量度量、控制流分析、数据流分析的静态统计信息分析 | | | | |
| 通过准则 | 完成要求的源代码分析，得到软件质量度量信息，软件无违反控制流和数据流检查要求的情况 | | | | |

### 代码审查

#### 代码审查

表1‑2代码审查测试项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试项名称 | 代码审查 | 测试项标识 | XQ\_CR\_DMSC | 优先级 | 中 |
| 追踪关系 | 隐含需求 | | | | |
| 需求描述 | 根据相关要求及软件文档开展针对软件程序代码的代码审查 | | | | |
| 测试手段 | 静态测试 | | | | |
| 测试项描述 | **1.代码审查（XQ\_CR\_DMSC\_SU01）**  通过人工审查及借助工具辅助分析的方式开展代码审查，审查代码编程准则的符合性、代码流程实现的正确性、代码结构的合理性以及代码实现需求的正确性；人工审查中发现的问题，审查人员应及时记录 | | | | |
| 测试方法 | **1.代码审查（XQ\_CR\_DMSC\_SU01）**  人工审查及借助工具辅助分析的方式，和依据附录代码审查单范围内的源代码开展四个方面的审查：  1）编程准则检查：依据编程准则的要求，对程序的编码与编程准则进行符合性检查；  2）代码流程审查：审查程序代码的条件判别、控制流程、数据处理等满足设计要求；  3）软件结构审查：依据设计文档，审查程序代码的结构设计的合理性，包括程序结构设计和数据结构设计；  4）需求实现审查：依据需求文档及其他相关资料，审查程序代码的需求层的功能实现是否正确 | | | | |
| 充分性要求 | 对软件全部源代码/重点模块进行代码审查 | | | | |
| 通过准则 | 代码编码符合编程准则、代码流程实现正确、代码结构设计合理，代码实现需求正确 | | | | |

### 功能测试

#### 连续的功能测试

表1‑2连续的功能测试测试项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试项名称 | 连续的功能测试 | 测试项标识 | XQ\_FT\_FFFF | 优先级 | 高 |
| 追踪关系 | 《PDPU和软件的协议》-2.2.2-设计需求2号 | | | | |
| 需求描述 | 123 | | | | |
| 测试手段 | 动态测试 | | | | |
| 测试项描述 | **1.测试子项名称（XQ\_FT\_FFFF\_SU01）**  测试子项描述 | | | | |
| 测试方法 | **1.测试子项名称（XQ\_FT\_FFFF\_SU01）**  条件1号，操作1号，观察1号，期望1号 | | | | |
| 充分性要求 | 测试用例覆盖测试子项名称子项要求的全部内容。  所有用例执行完毕，对于未执行的用例说明未执行原因。 | | | | |
| 通过准则 | 各测试步骤、测试用例执行结果与预期一致，功能实现正确 | | | | |