第五章软件可信性量化评估体系

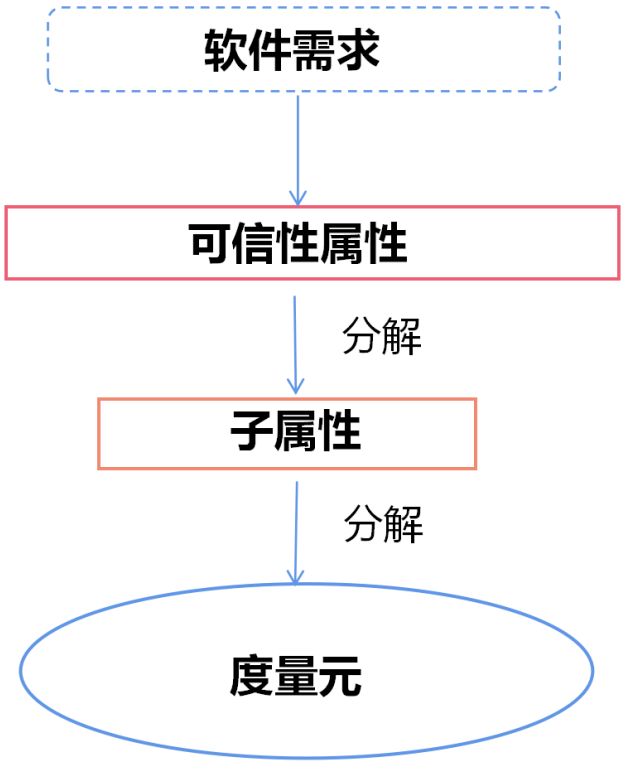
陈仪香

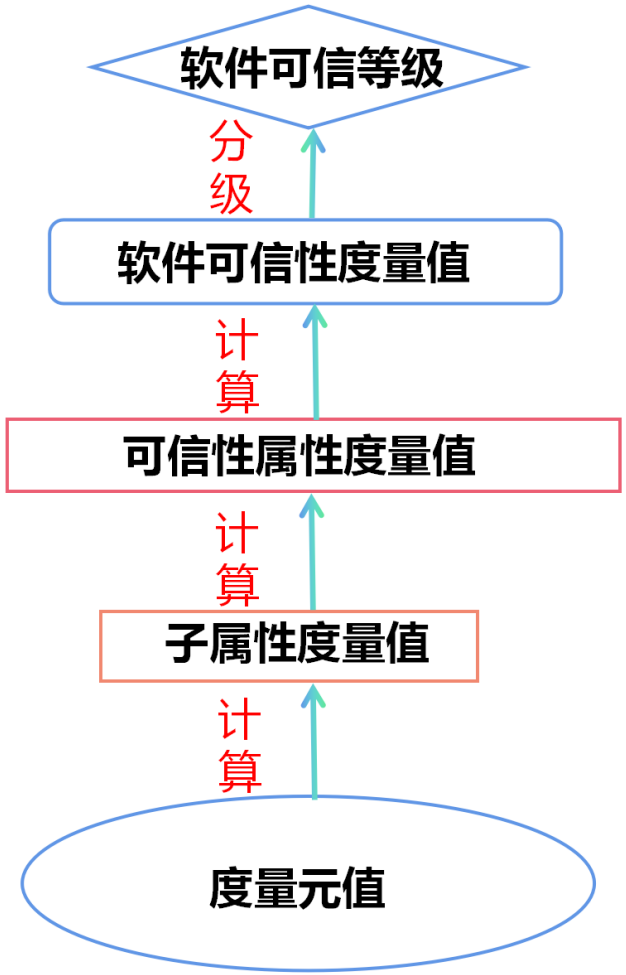
2024年11月07日

Outline

第5章软件可信性量化评估体系

本章介绍软件可信性度量与评估体系，给出软件可信度量方法，包括度量元计算模型、子属性计算模型、属性计算模型和软件可信性计算模型，构建软件可信性分级模型。





软件可信量化评估准则

为使软件可信性度量与评估能够准确清晰地反映软件可信性状况，其需满足以下软件可信量化评估准则：

准则**1**： 评价区分性
评价依据有一定的区分性，更高评定级别要比较低评价级别更难达到。

准则**2**： 依据客观性
尽可能不用模糊的评价依据，级别之间也要有一定的区分度，让评估更客观。

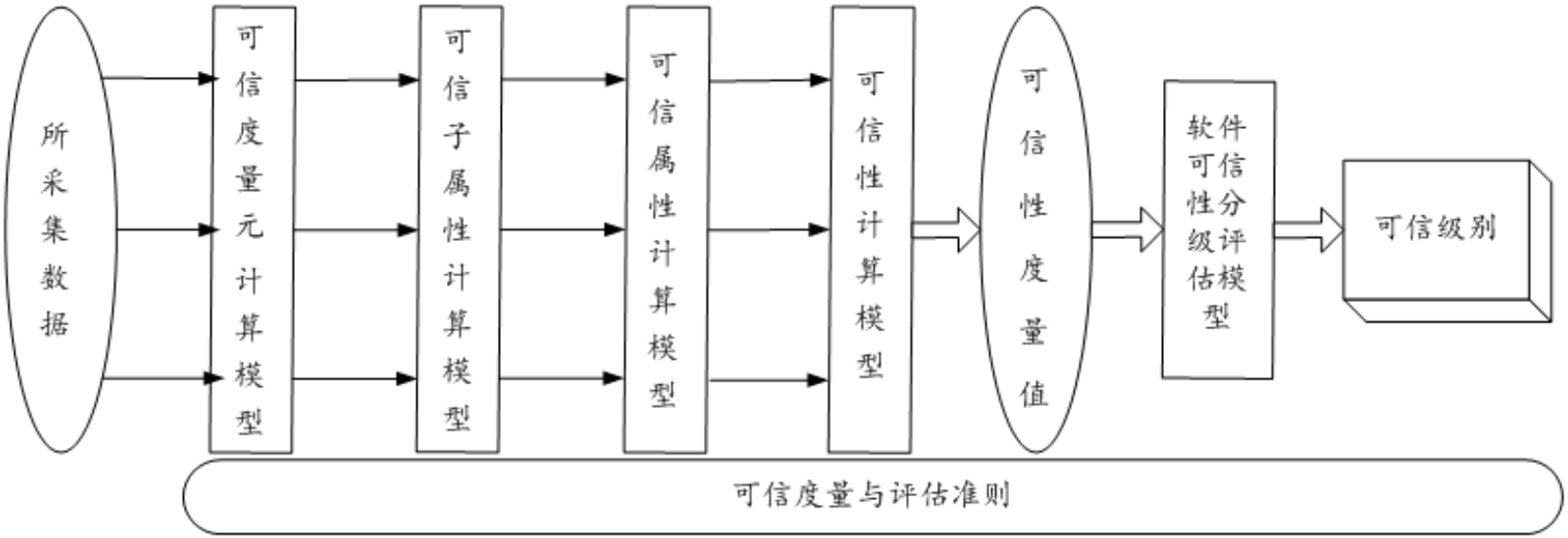
准则**3**： 评估重现性
在相同环境下，相同和不同的评估者对软件产品可信度量所得结果在一定容差范围内。

准则**4**： 评价数值性
以数值的形式表示评价结果。可信度大小体现可信程度高低，数值越大软件可信度越高。

体系结构

基于上述准则，结合软件全生命周期和出厂报告现有评价方法和标准，制定同时体现软件开发过程和软件属性的软件可信性度量与评估体系，其体系结如下图所示。该体系结构由

- 可信度量元计算模型: 度量元计算模型依据所采集数据计算度量元值
- 可信子属性计算模型:子属性计算模型通过度量元值集结得到子属性值
- 可信属性计算模型:属性计算模型根据子属性值计算属性值
- 软件可信性计算模型: 软件可信性计算模型通过融合属性值获得软件可信值
- 软件可信分级模型:在可信度量结果基础上依据软件可信性分级模型对软件可信性进行分级。
- 可以在该体系结构基础上进行定制或者扩充构建所需体系结构，比如，基于全生命周期的软件可信性度量评估体系结构，其添加了阶段可信性计算模型，用于计算各阶段的可信性度量值，并且将数据的采集扩充到软件开发的整个过程。


所采集数据
可信度量元计算模型
可信子属性计算模型
可信属性计算模型
可信性计算模型
可信性度量值
软件可信性分级评估模型
可信级别
可信度量与评估准则

分级模型

软件可信分级模型是根据软件可信度量结果将软件可信性分为若干个等级，我们在表5.1中建立了一个此类模型。

Claim

级别越高可信度就越大： *V*级别最高， *I*级别最低

软件可信性分级模型性质

性质**1.** 门限性

门限性是指当软件可信性要达到某一级别，规定软件各可信属性必须要达到此级别所要求的可信属性最低值。该性质表明软件可信属性度量值不能太低，提高属性值最低的可信属性，很大可能性提高软件可信性级别。

性质**2.** 绝对多数性

绝对多数性是指软件可信性要达到某一级别，规定至少三分之二的可信属性度量值需达到此级别。假设可信属性有n个，需要达到此级别的可信属性个数不少于$\lceil n*2/3 \rceil$，所以低于此级别的可信属性个数不超过$n-\lceil n*2/3 \rceil$个。该性质用于刻画软件可信属性度量值三分之二多数达到某级别才有可能让软件可信性级别达到此级别。

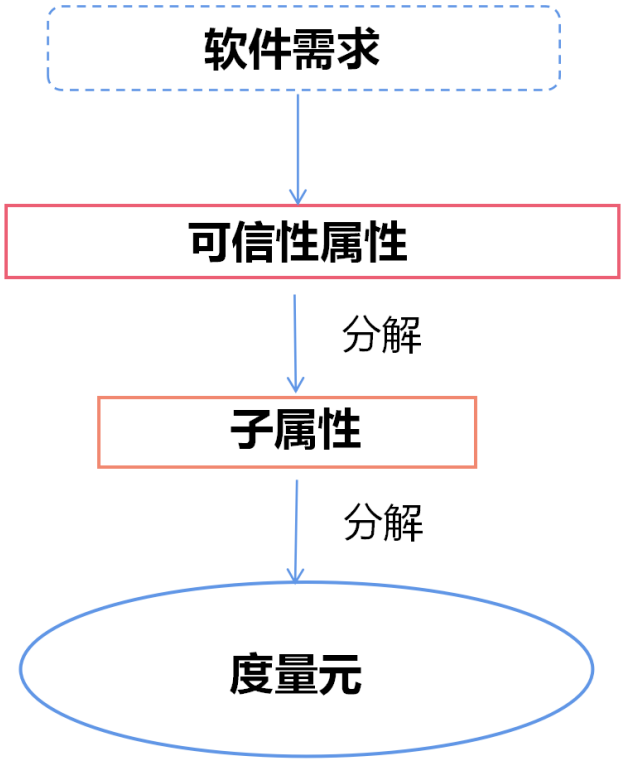
分级模型的基本特性

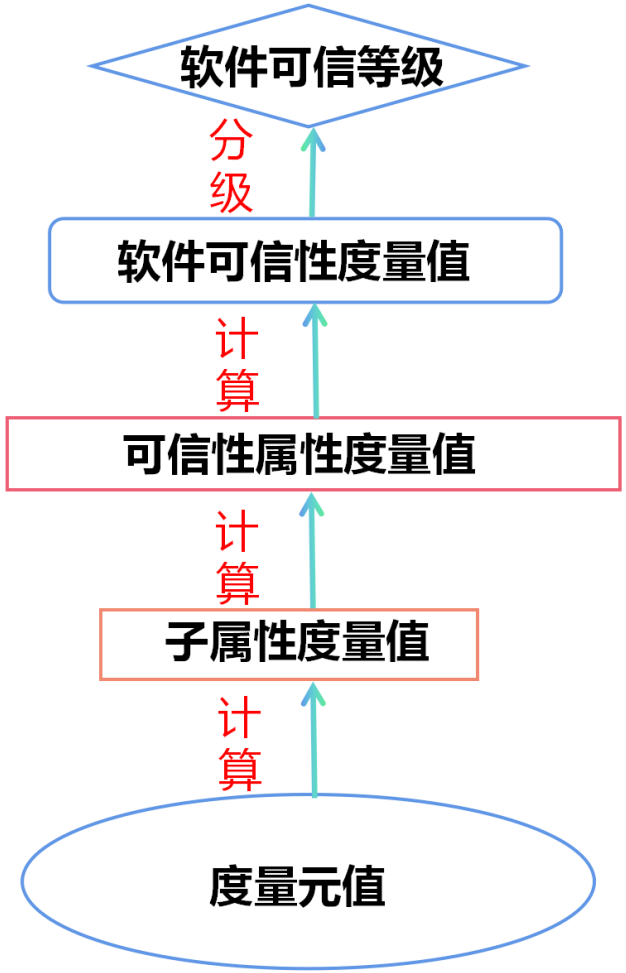
- 该模型评定的级别完全覆盖软件可信性度量值的所有可能分布区间；
- 软件可信性的每一种取值情况都可以且只可以评定为一个级别，保证了其可信等级的唯一性。
- 在该模型中软件可信等级的评定不仅需要软件满足软件可信性度量值要求，而且需要满足可信属性值要求，对于具有相同可信性度量值的两个软件，若其中一个软件的某一属性未达到可信属性值要求，那么它的可信等级就会降低。
- 考虑到软件可信度越高，对其进行提高的难度越大，同时可信级别越高对可信属性度量值要求也越高，所以在该可信分级模型中分级区间非等距，从最低级向上增加可信级别的值区间近似按黄金分割比例递减。

分级模型表

Table: 软件可信性分级模型

软件可信度量值要求（黄金分割）	可信属性要求（多数原则）	可信等级
$9.5 <= T$	1. 低于9.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$个 2. 没有低于8.5分的可信属性	V
$8.5 <= T < 9.5$ 或者$T > 9.5$且不能评为V级别	1. 低于8.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$个 2. 没有低于7.0分的可信属性	IV
$7.0 <= T < 8.5$ 或者$T > 8.5$且不能评为IV级别及以上者	1. 低于7.0分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$个 2. 没有低于4.5分的可信属性	III
$4.5 <= T < 7.0$ 或者$T > 7.0$且不能评为III级别及以上者	1. 低于4.5分的关键属性个数不超过 $n - \lceil n \cdot 2/3 \rceil$个	II
$T < 4.5$ 或者$T > 4.5$且不能评为II级别及以上者	1. 无要求	I


软件需求
可信性属性
分解
子属性
分解
度量元

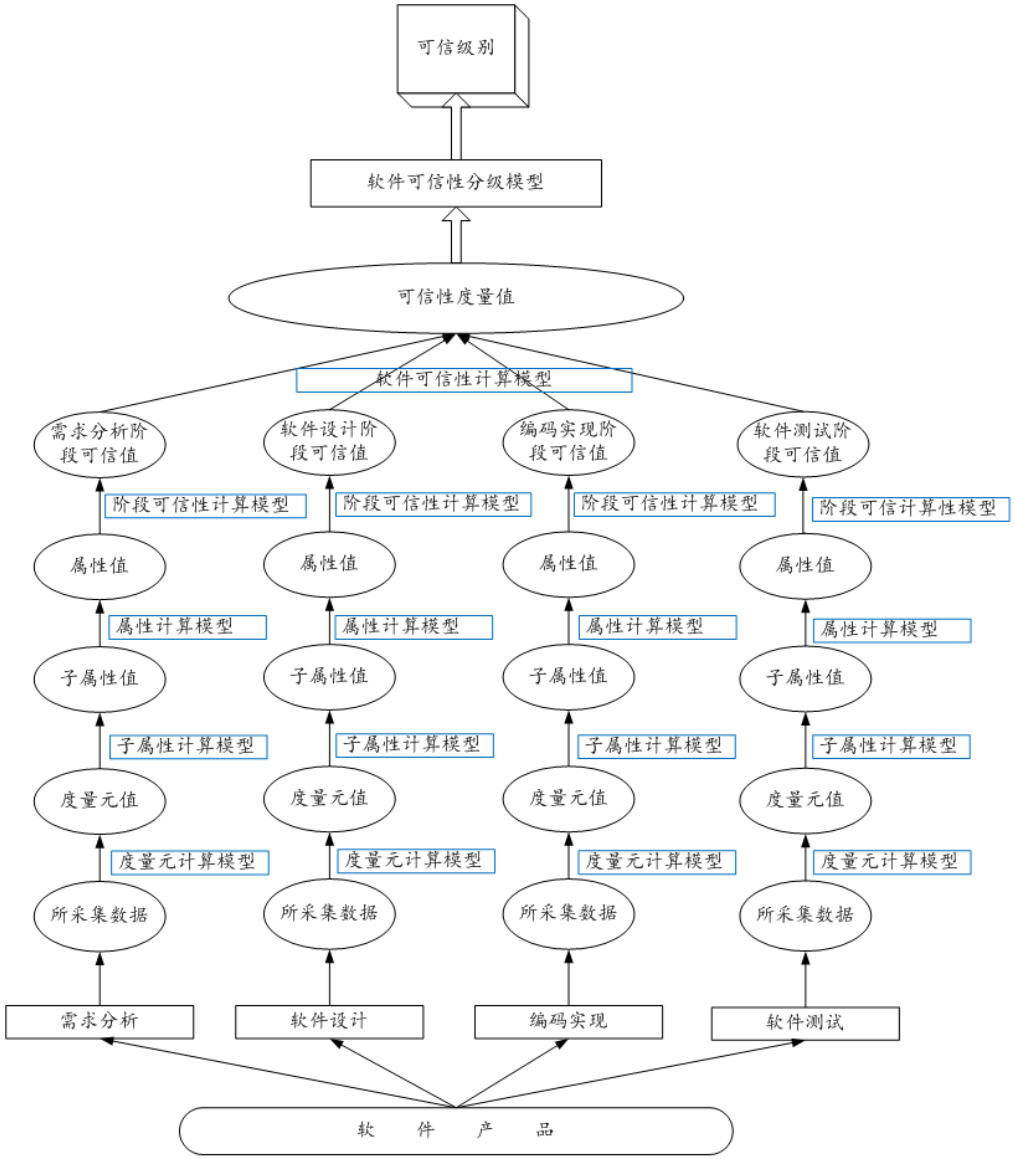

软件可信等级
分级
软件可信性度量值
计算
可信性属性度量值
计算
子属性度量值
计算
度量元值

度量元计算模型

度量元是软件可信度量与评估的最基础层，度量元计算模型依据所采集数据计算得到度量元值，根据所采集到的数据类型，其可分为定量计算模型和定性计算模型。在基于全生命周期的软件可信度量模型中两类模型均有涉及到，在基于出厂报告的软件可信度量模型中只涉及定性计算模型。

- 全生命周期度量元模型
- 出厂报告度量元模型

全生命周期度量元模型


可信级别
软件可信性分级模型
可信性度量值
软件可信性计算模型
需求分析阶段可信值
软件设计阶段可信值
编码实现阶段可信值
软件测试阶段可信值
阶段可信性计算模型
阶段可信性计算模型
阶段可信性计算模型
阶段可信计算性模型
属性值
属性值
属性值
属性值
属性计算模型
属性计算模型
属性计算模型
属性计算模型
子属性值
子属性值
子属性值
子属性值
子属性计算模型
子属性计算模型
子属性计算模型
子属性计算模型
度量元值
度量元值
度量元值
度量元值
度量元计算模型
度量元计算模型
度量元计算模型
度量元计算模型
所采集数据
所采集数据
所采集数据
所采集数据
需求分析
软件设计
编码实现
软件测试
软 件 产 品

全生命周期度量元模型

在基于全生命周期的软件可信度量中，度量元定量计算模型有两种计算公式如下：

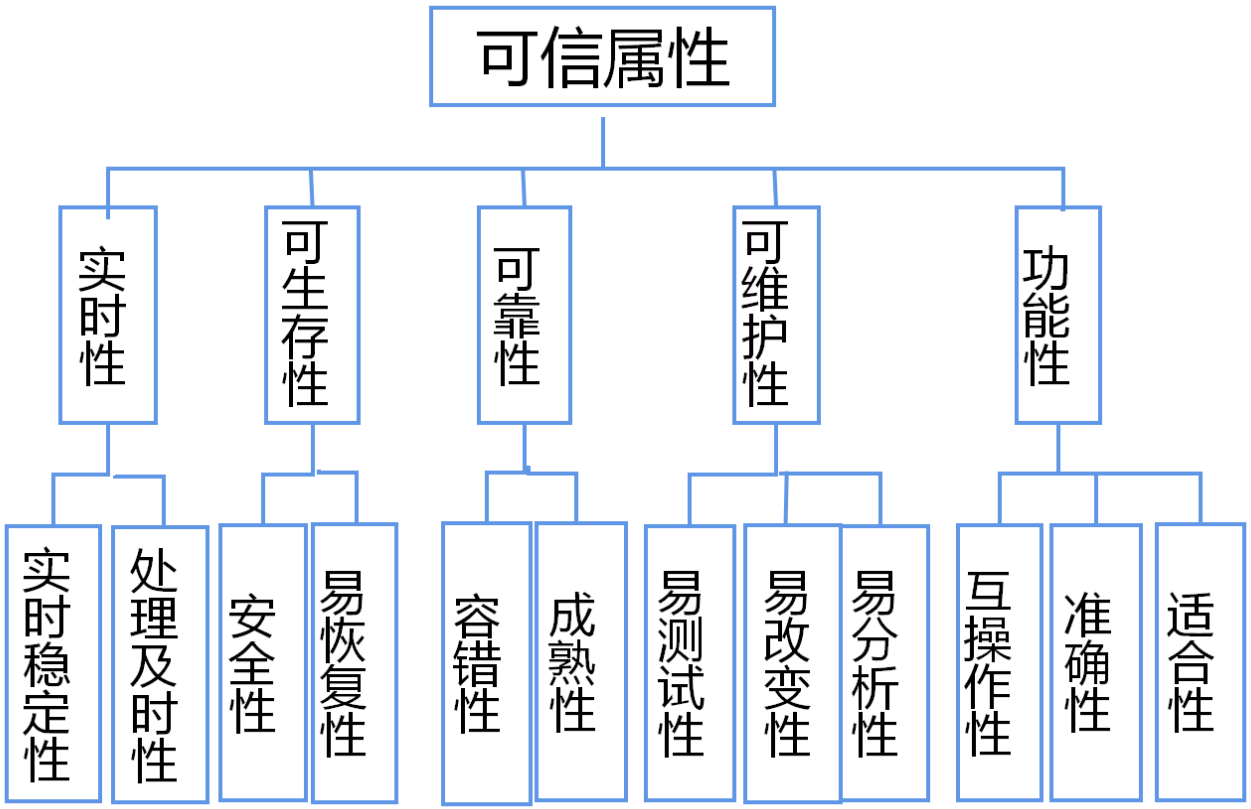
$$index = 1 - \frac{u}{v}\ (v \neq 0)$$

其中u表示未满足相应标准的不可信数据数目，v表示需满足相应标准的全部数据数目。

$$index = \frac{u}{v}\ (v \neq 0)$$

其中u表示满足相应标准的可信数据数目，v表示需满足相应标准的全部数据数目。

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

全生命周期度量元模型

在基于全生命周期的可信属性分层模型基础上中，按照属性和子属性进行编排，把子属性下的度量元按照全生命周期的四个阶段进行列表安排，构造基于全生命周期的可信度量元模型。

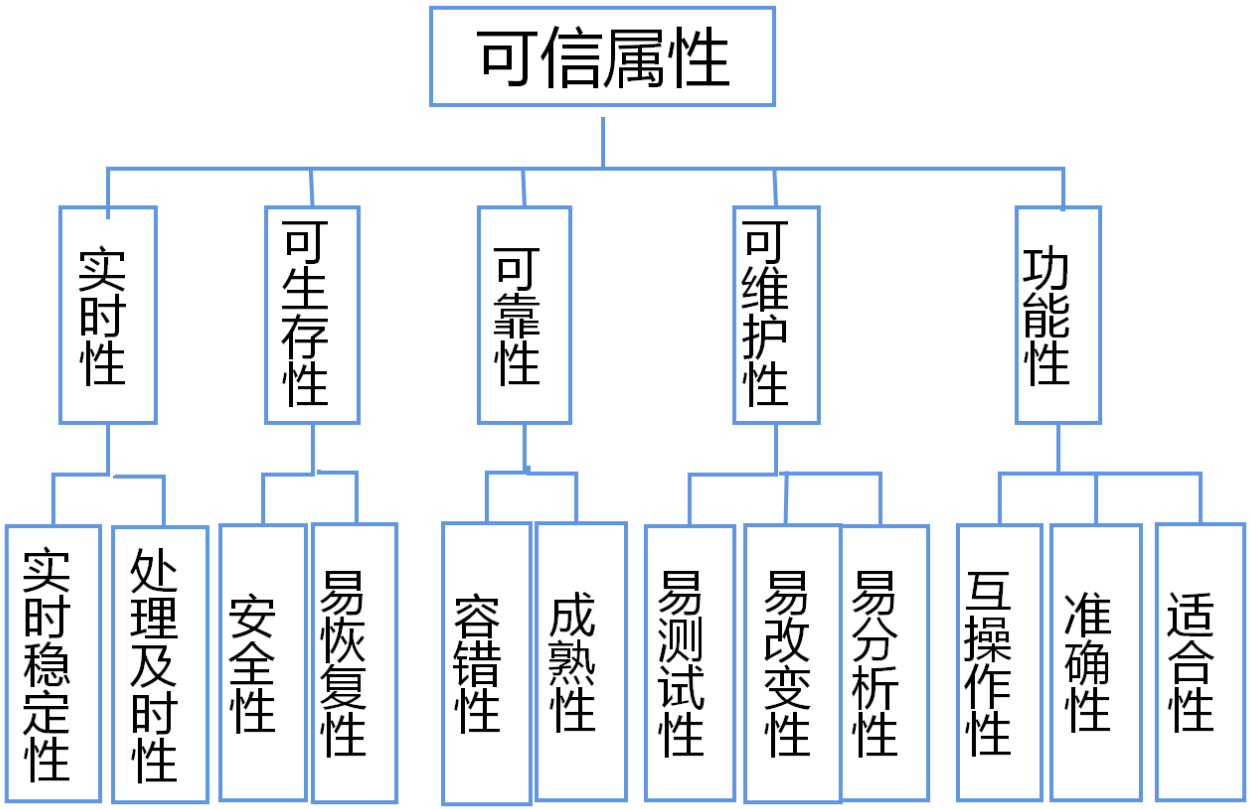
- 可信属性：功能性、可维护性、可靠性、可生存性、实时性。
- 共有111个度量元，其中定量度量元有101个、定性度量元有10个。
- 需求分析阶段有20个定量度量元、3 个定性度量元，
- 设计阶段有20 个定量度量元、4 个定性度量元，
- 编码实现阶段包含19个定量度量元、3个定性度量元，
- 测试阶段有42个定量度量元、0个定性度量元。

度量元类型

度量元类型有定量和定性之分。

- 定量度量元：
  定量是可通过采集到的数据按照计算公式获得确切数值。定量度量元的两种计算公式：$1-\frac{u}{v}$ 和$\frac{u}{v}$。前者中的u记录未满足相应标准的不可信数据数目，如，缺失定义用户需求数目或未定义精度要求的数据处理需求项数目；而后者中的u记录满足相应标准的可信数据数目，如，符合易测试性要求的数目。两者中的v都记录需满足相应标准的全部数据数目，因此在基于全生命周期的软件可信度量元模型定量度量元采用前者作为计算公式。
- 定性度量元：
  定性是不易直接获得确切的数值，对于定性度量元采用专家依据涉及到该度量元信息进行评估给出相应数值。这样每一个度量元都可以获得度量值。

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

功能性属性的可信度量元模型

功能性阐述软件产品提供满足明确和隐含要求功能的能力，拥有三个子属性：适合性、准确性和互操作性。

- 共有27个度量元，其中27个定量度量元、0个定性度量元。
- 需求分析阶段有6个，
- 软件设计阶段有6个，
- 编码实现阶段有6 个，
- 测试阶段有9个。

功能性属性的子属性适合性度量元

开发阶段	名称	类型	度量元
需求分析	功能定义充分	定量	$r=1-\frac{u}{v}$，其中: u=遗漏用户需求数目 v=全部用户需求数目
	功能定义正确性	定量	$r=1-\frac{u}{v}$，其中: u=错误用户需求数目 v=全部用户需求数目
软件设计	设计覆盖性	定量	$r=1-\frac{u}{v}$，其中: u=缺失的设计项数目 v=应有的全部用户需求数目
	设计正确性	定量	$r=1-\frac{u}{v}$，其中: u=不合理的软件设计项数目 v=全部软件设计项数目
编码实现	实现定义充分性	定量	$r=1-\frac{u}{v}$，其中: u=遗漏的实现设计项数目 v=全部设计项数目
	设计正确性	定量	$r=1-\frac{u}{v}$，其中: u=错误的实现设计项数目 v=全部软件设计项数目
软件测试	功能测试完整性	定量	$r=1-\frac{u}{v}$，其中: u=未进行测试设计（未覆盖）的用户功能数目 v=全部功能总数目
	功能测试有效性	定量	$r=1-\frac{u}{v}$，其中: u=用测试用例测试的用户功能数目 v=全部功能总数目

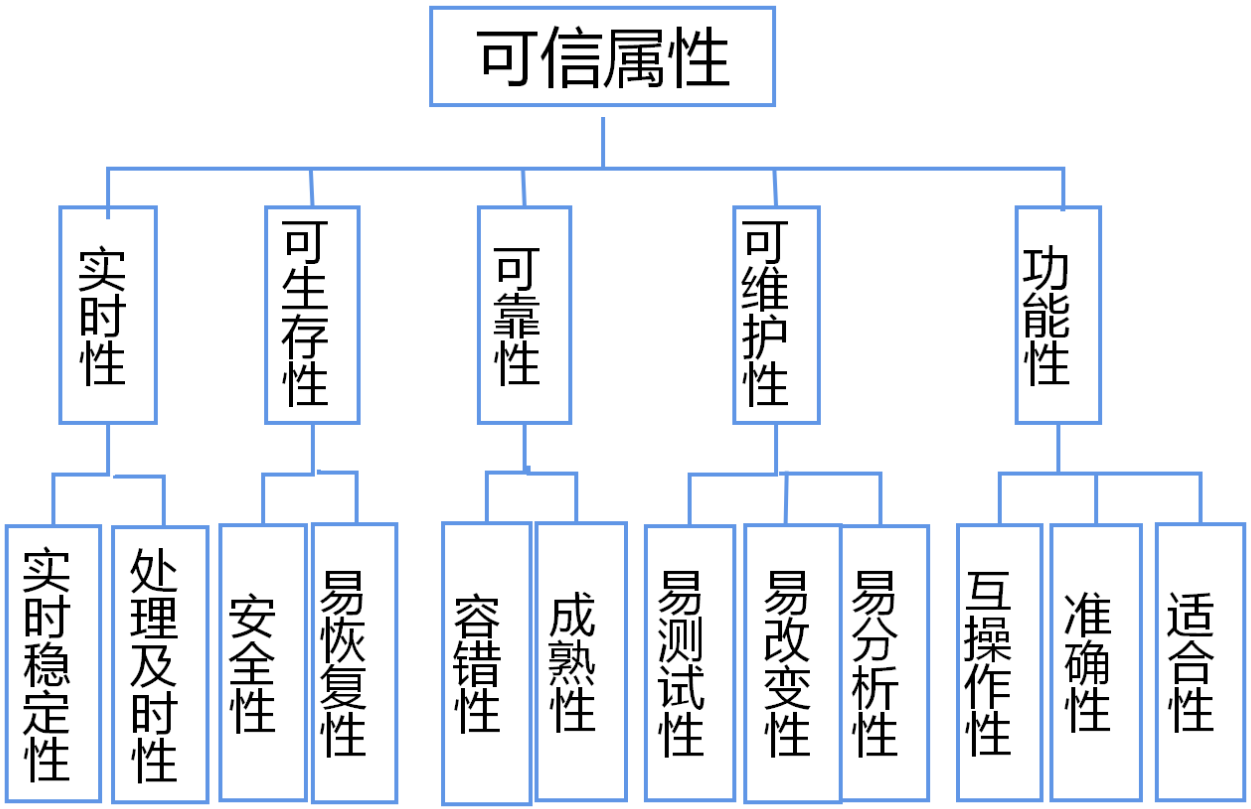
功能性的子属性准确性度量元

开发阶段	名称	类型	度量元
需求分析	数据处理精度定义	定量	$r=1-\frac{u}{v}$，其中: u=未定义精度要求的数据处理需求数目 v=全部数据处理需求数目
	计算准确性定义	定量	$r=1-\frac{u}{v}$，其中: u=未给出计算准确性定义的需求项数目 v=全部计算准确性需求数目
软件设计	数据处理精度设计	定量	$r=1-\frac{u}{v}$，其中: u=未得到相应设计的数据处理精度项数目 v=全部数据处理精度需求项数目
	计算准确性设计	定量	$r=1-\frac{u}{v}$，其中: u=未得到相应设计的计算准确性项数目 v=全部计算准确性需求项数目
编码实现	数据处理精度实现	定量	$r=1-\frac{u}{v}$，其中: u=未设计或已实现但不符合设计的数据元素/数据文件数 v=全部的数据元素/数据文件总数目
	计算准确性实现	定量	$r=1-\frac{u}{v}$，其中: u=未达到设计要求的计算准确性数目 v=全部计算准确性项总数目
软件测试	数据处理精度测试有效性	定量	$r=1-\frac{u}{v}$，其中: u=未进行有效测试的处理精度要求数目 v=全部数据处理精度要求总数目
	数据处理精度符合性	定量	$r=1-\frac{u}{v}$，其中: u=经过有效测试且符合精度要求的指标数目 v=全部数据处理指标总数目
	计算准确性测试有效性	定量	$r=1-\frac{u}{v}$，其中: u=未进行有效测试的计算准确要求数目 v=全部计算准确性要求总数目
	计算准确性符合性	定量	$r=1-\frac{u}{v}$，其中: u=经过有效测试且符合计算准确性要求的指标数目

功能性的子属性互操作性度量元

开发阶段	名称	类型	度量元
需求分析	数据可交换性（依据数据格式）	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义接口数据格式的接口数目 v=全部接口数目
	接口一致性（依据协议）	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义接口协议的接口数目 v=全部接口数据数目
软件设计	接口设计完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未充分设计的接口设计项数目 v=全部接口设计项数目
	接口设计文档化	定量	$r = 1 - \frac{u}{v}$，其中: u=未文档化的接口设计项数目 v=全部接口设计项数目
编码实现	接口实现充分性	定量	$r = 1 - \frac{u}{v}$，其中: u=未实现的接口项数目 v=全部接口项总数目
	接口实现正确性	定量	$r = 1 - \frac{u}{v}$，其中: u=与设计不符合的接口项数目 v=全部接口项总数目
软件测试	接口测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计（未覆盖）的接口数目 v=全部接口总数目
	接口测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=对接口用测试用例进行有效测试的接口数目 v=全部接口总数目
	接口符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=不符合接口定义的接口数目 v=全部接口总数目

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

可维护性属性的可信度量元模型

可维护性描述了软件产品被修改的能力，拥有三个子属性：易分析性、易改变性和易测试性。

- 共有20个度量元，其中有10个定量度量元、10个定性度量元。
- 需求分析阶段有4个，
- 软件设计阶段有5个，
- 编码实现阶段有3 个，
- 测试阶段有8 个。

可维护性的子属性易分析性度量元

开发阶段	名称	类型	度量元
需求分析	易分析性要求定义	定性	专家评分
软件设计	易分析性要求设计	定性	专家评分
	软件变化文档化	定量	$r = 1 - \frac{u}{v}$，其中: u=未文档化的软件变化项数目 v=全部软件变化项数目
编码实现	易分析性设计实现	定性	专家评分
软件测试	易分析性要求测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的易分析性要求的数目 v=易分析性要求总数目
	易分析性要求测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的易分析性要求数目 v=易分析性要求总数目
	易分析性要求符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的易分析性要求数目 v=应遵循的全部易分析性要求总数目

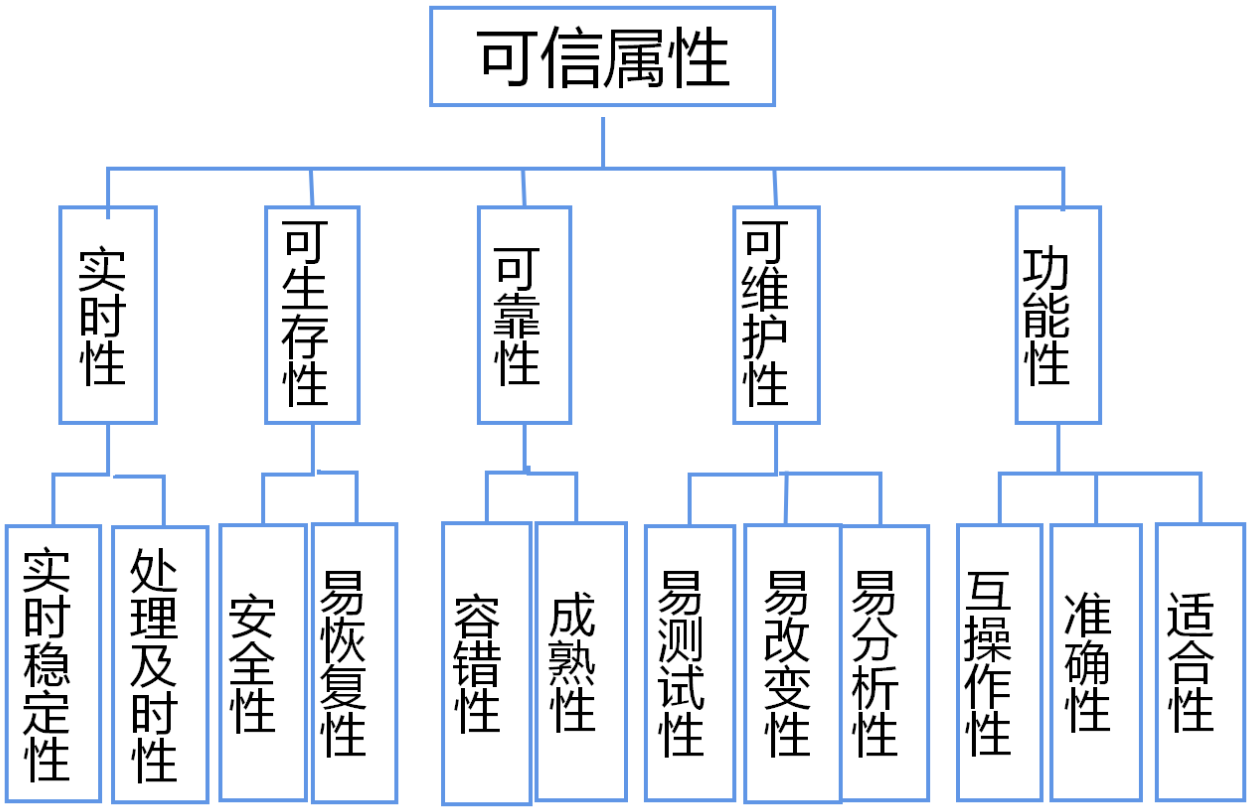
可维护性的子属性易改变性度量元

开发阶段	名称	类型	度量元
需求分析	变更方法定义	定量	$r = 1 - \frac{u}{v}$，其中: u=检查单中未通过检查项数 v=检查单中全部检查项数
	易改变性要求定义	定性	专家评分
软件设计	易改变性设计	定性	专家评分
编码实现	易改变性设计实现	定性	专家评分
软件测试	易改变性要求测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的易改变性要求数目 v=易改变性要求总是数目
	易改变性要求测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的易改变性要求数目 v=易改变性要求总数目

可维护性的子属性易测试性度量元

开发阶段	名称	类型	度量元
需求分析	易测试性要求的定义	定性	专家评分
软件设计	易测试性设计	定性	专家评分
	设计文档完整性	定性	专家评分
编码实现	易测试性设计实现	定性	专家评分
软件测试	易测试性要求测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的易测试性要求数目 v=易测试性要求总数目
	易测试性要求测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的易测试性要求数目 v=易测试性要求总数目
	易测试性要求符合有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合的易测试性要求数目 v=应遵循的全部易测试性要求总数目

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

可靠性属性的可信度量元模型

可靠性阐述了软件产品维持规定性能级别的能力，拥有两个子属性：成熟性和容错性。

- 共有15个度量元，包括15个定量度量元、0个定性度量元。
- 需求分析阶段有3个，
- 软件设计阶段有3个，
- 编码实现阶段有3 个，
- 测试阶段有6 个。

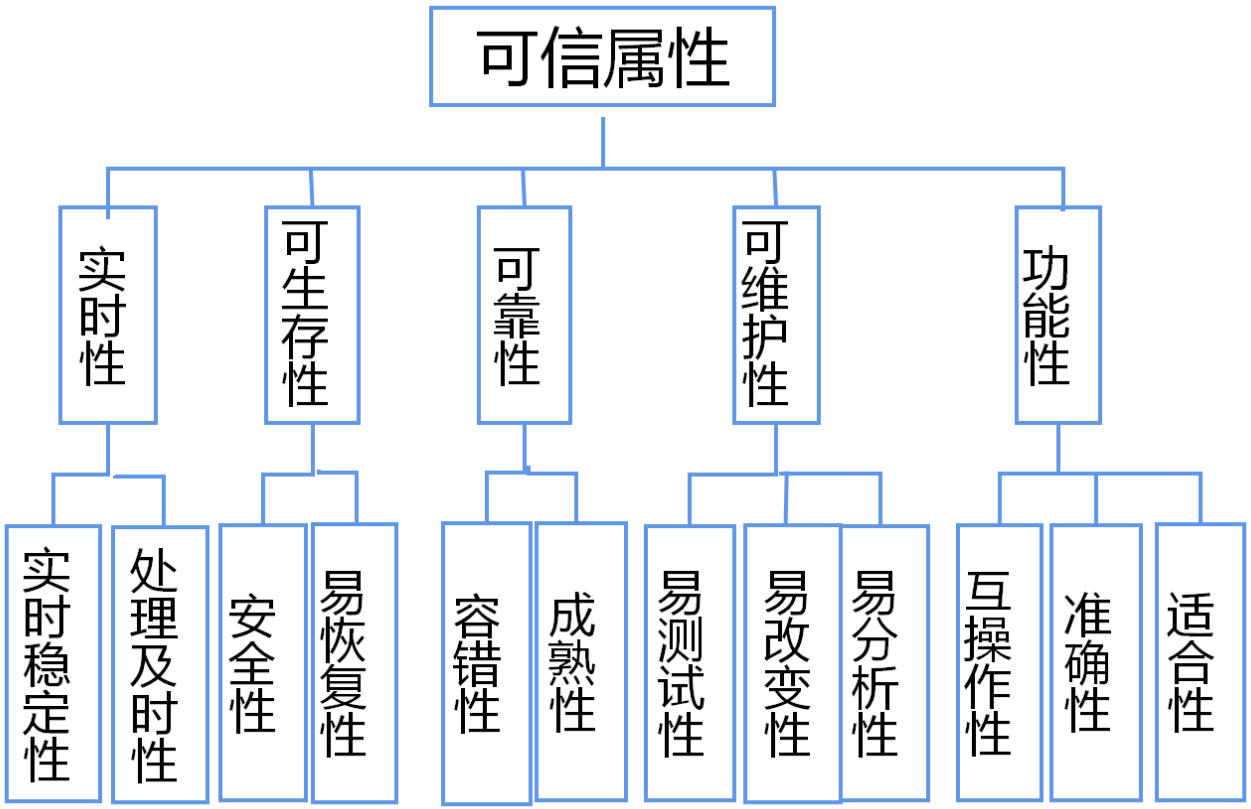
可靠性的子属性成熟性度量元模型

开发阶段	名称	类型	度量元
需求分析	成熟性需求定义的完整性	定量	$r=1-\frac{u}{v}$，其中: u=未定义或未合理定义成熟性的需求数目 v=全部应定义成熟性的需求数目。
	失效后处理措施定义的完整性	定量	$r=1-\frac{u}{v}$，其中: u=未定义或未合理定义失效后处理措施的需求数目 v=全部应定义失效后处理措施的需求数目。
软件设计	成熟性设计	定量	$r=1-\frac{u}{v}$，其中: u=未设计或未正确设计的软件成熟性需求项数目 v=全部软件成熟性需求项数目
	接口设计文档化	定量	$r=1-\frac{u}{v}$，其中: u=未设计或未正确设计的软件失效后处理措施项数目 v=失效后处理措施项总数目
编码实现	成熟性设计实现	定量	$r=1-\frac{u}{v}$，其中: u=未正确实现的软件成熟性设计项数目 v=全部软件成熟性设计项总数目
	失效后处理措施的实现	定量	$r=1-\frac{u}{v}$，其中: u=未正确实现的软件失效后处理措施设计项数目 v=失效后处理措施设计项总数目
软件测试	成熟性要求测试完整性	定量	$r=1-\frac{u}{v}$，其中: u=进行测试设计的成熟性要求数目 v=成熟性要求总数目
	成熟性要求测试有效性	定量	$r=1-\frac{u}{v}$，其中: u=未进行有效的测试设计的成熟性要求数目 v=成熟性要求总数目
	成熟性要求符合性	定量	$r=1-\frac{u}{v}$，其中: u=符合的成熟性要求数目 v=应遵循的全部成熟性要求总数目

可靠性的子属性容错性度量元模型

开发阶段	名称	类型	度量元
需求分析	错误处理规则识别	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义的错误处理需求数目 v=全部应建立的错误处理需求数目
软件设计	容错处理设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未合理设计的错误处理项数目 v=全部应容错处理项数目
编码实现	容错处理设计实现	定量	$r = 1 - \frac{u}{v}$，其中: u=未正确实现的容错处理设计项数目 v=全部容错处理设计项总数目
软件测试	错误处理规则测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的错误处理规则数目 v=所建立的全部错误处理规则总数目
	错误处理规则测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的错误处理规则数目 v=所建立的错误处理规则总数目
	错误处理规则符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合的错误处理规则数目 v=所建立的全部错误处理规则总数目

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

可生存性属性的可信度量元模型

可生存性阐述了软件产品在受到攻击或失效出现时，连续提供服务以及及时恢复所有服务的能力，拥有两个子属性：安全性和恢复性。

- 共有32个度量元，包括32个定量度量元、0个定性度量元。
- 需求分析阶段有7个，
- 软件设计阶段有7个，
- 编码实现阶段有7个，
- 测试阶段有11个。

可生存性的子属性安全性度量元模型

开发阶段	名称	类型	度量元
需求分析	权限控制定义	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理的权限控制规则数目 v=全部权限控制规则数目
	关键事务识别	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义的关键事务授权规则数目 v=全部关键事务授权规则数目
	资源安全性定义	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义的软件所占用各资源安全性要求 v=全部软件所占用各资源安全性要求数目
	关键数据识别	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义的关键数据项目 v=全部关键数据项目
软件设计	权限控制设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行正确设计的权限控制要求数目 v=权限控制要求总数目
	关键事务授权方法设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未合理设计的关键事务的授权方法数目 v=全部关键事务的授权方法总数目
	密码安全设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未合理设计的关键事务的授权方法数目 v=全部关键事务的授权方法总数目
	资源安全性设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未合理设计的软件所占用各资源安全性要求 v=全部软件所占用各资源安全性要求总数目
	关键数据完整性检查	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行完整性验证设计的关键数据数目 v=全部关键数据总数目
	敏感数据一致性检查	定量	$r = 1 - \frac{u}{v}$，其中: u=未设计或未合理设计的敏感数据一致性保护数目

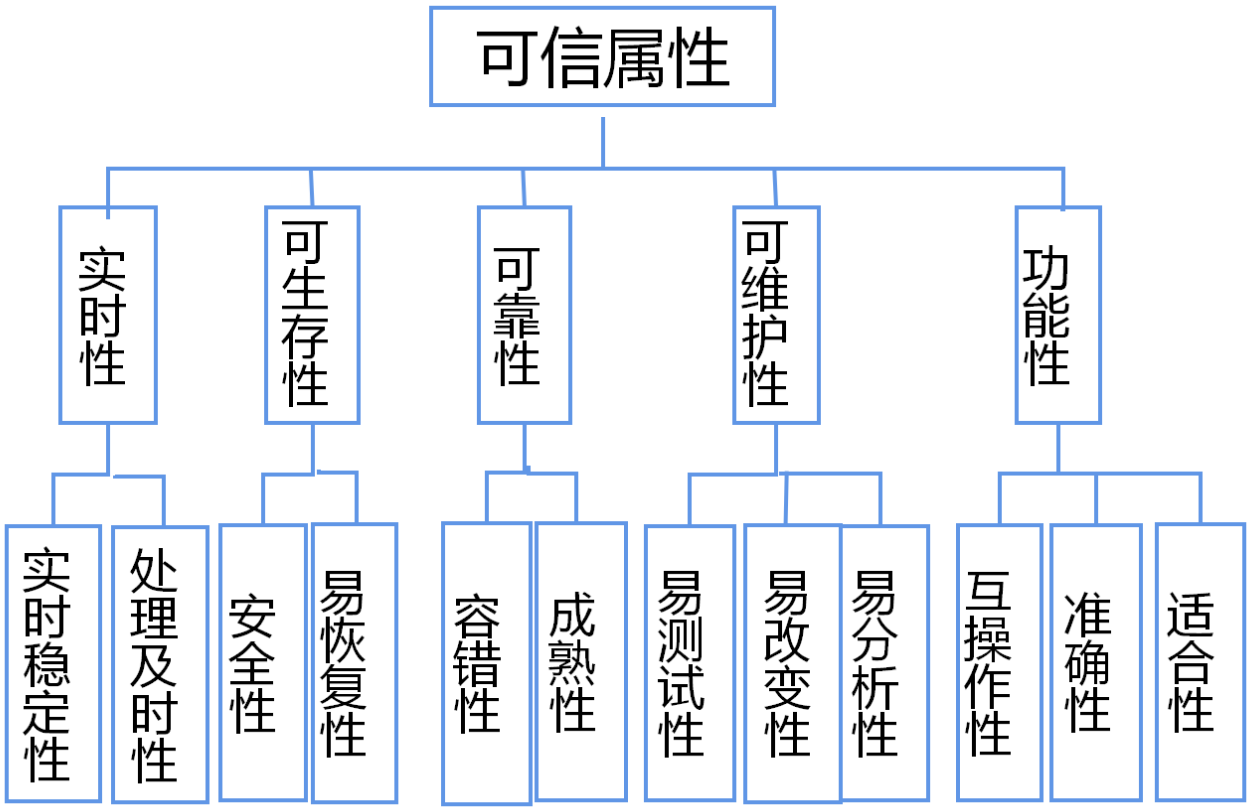
可生存性的子属性安全性度量元（续）

开发阶段	名称	类型	度量元
软件测试	安全保密性测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的安全保密性数目 v=安全保密性总数目
	安全保密性要求测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效测试设计的安全保密性数目 v=安全保密性总数目
	关键事务及资源识别符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合权限控制定义数目 v=应遵循的全部权限控制定义总数目
	关键事务及资源识别符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合关键事务及资源定义数目 v=应遵循的全部关键事务及资源定义总数目
	完整性测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效测试设计的完整性数目 v=完整性总数目
	完整性测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效测试设计的完整性数目 v=完整性总数目
	完整性符合	定量	$r = 1 - \frac{u}{v}$，其中: u=符合完整性数目 v=应遵循的全部完整性总数目

可生存性的子属性恢复性度量元

开发阶段	名称	类型	度量元
需求分析	易恢复性要求定义	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义在失效发生情况下软件重建规定性能级别并恢复受直接影响数据的能力的需求数目 v=全部应定义在失效发生情况下软件重建规定性能级别并恢复受直接影响数据的能力的需求数目
	平均恢复时间定义	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义平均恢复时间的需求数目 v=全部应定义平均恢复时间的需求数目
	最大恢复时间定义	定量	$r = 1 - \frac{u}{v}$，其中: u=未定义或未合理定义最大恢复时间的需求数目 v=全部应定义最大恢复时间的需求数目
软件设计	易恢复性要求设计	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行正确设计的易恢复性要求数目 v=易恢复性要求总数目
编码实现	易恢复性设计实现	定量	$r = 1 - \frac{u}{v}$，其中: u=未正确实现的易恢复性设计项数目 v=易恢复性设计项总数目
软件测试	易恢复性测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行测试设计的易恢复性要求数目 v=易恢复性要求总数目
	易恢复性测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u=未进行有效的测试设计的易恢复性要求数目 v=易恢复性要求总数目
	易恢复性要求符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合的易恢复性要求数目 v=应遵循的全部易恢复性要求总数目
	恢复时间符合性	定量	$r = 1 - \frac{u}{v}$，其中: u=符合的恢复时间要求数目 v=应遵循的恢复时间要求总数目

全生命周期度量元模型


可信属性
实时性
可生存性
可靠性
可维护性
功能性
实时稳定性
处理及时性
安全性
易恢复性
容错性
成熟性
易测试性
易改变性
易分析性
互操作性
准确性
适合性

实时性属性的可信度量元模型

实时性表达了软件在指定时间内完成操作或提交输出的能力，拥有及时性和稳定性两个子属性。

- 共有17个度量元，包括17个定量度量元、0个定性度量元。
- 需求分析阶段有3个，
- 软件设计阶段有3个，
- 编码实现阶段有3 个，
- 测试阶段有8 个。

实时性的子属性及时性度量元模型

开发阶段	名称	类型	度量元
需求分析	处理及时性定义	定量	$r = 1 - \frac{u}{v}$，其中: u = 未定义或未合理定义的处理时间需求项数目 v =全部处理时间需求项数目
	最坏情况下的处理时间	定量	$r = 1 - \frac{u}{v}$，其中: u =未定义或未合理定义处理时间下限的需求项数目 v =需定义最坏情况下处理时间的需求项数目
软件设计	处理及时性设计	定量	$r = 1 - \frac{u}{v}$，其中: u = 未设计或未正确设计的处理时间需求项数目 v =全部处理时间需求项数目
	最坏情况下的处理时间	定量	$r = 1 - \frac{u}{v}$，其中: u =未设计或未正确设计的处理时间下限的需求项数目 v =已定义最坏情况下处理时间的需求项数目
编码实现	处理及时性实现	定量	$r = 1 - \frac{u}{v}$，其中: u = 未实现或未正确实现的处理时间设计项数目 v =全部处理时间设计项数目
	最坏情况下处理时间的实现	定量	$r = 1 - \frac{u}{v}$，其中: u = 未实现或未正确实现的处理时间下限的设计项数目 v =已定义最坏情况下处理时间的设计项数目
软件测试	处理及时性测试完整性	定量	$r = 1 - \frac{u}{v}$，其中: u = 未进行测试的处理及时性要求的数目 v = 处理及时性要求的全部数目
	处理及时性测试有效性	定量	$r = 1 - \frac{u}{v}$，其中: u = 未进行有效测试的处理及时性要求数目 v =处理及时性要求的全部数目
	处理及时性符合性	定量	$r = \frac{u}{v}$，其中: u = 符合处理及时性要求的数目 v = 应遵循的全部处理及时性要求的数目
	最坏情况下的处理时间测试	定量	$r = \frac{u}{v}$，其中: u = 符合最坏情况下处理时间的项数目

实时性的子属性稳定性度量元

开发阶段	名称	类型	度量元
需求分析	处理时间最大抖动定义	定量	$r=1-\frac{u}{v}$，其中: u =未定义或未合理定义的处理时间抖动需求项数目 v =全部处理时间抖动需求项数目
软件设计	处理时间最大抖动设计	定量	$r=1-\frac{u}{v}$，其中: u =未设计或未正确设计的处理时间抖动需求项数目 v =全部处理时间抖动需求项数目
编码实现	处理时间最大抖动实现	定量	$r=1-\frac{u}{v}$，其中: u =未设计或未正确实现的处理时间抖动设计项数目 v = 全部处理时间抖动设计项数目
软件测试	时间抖动测试完整性	定量	$r=1-\frac{u}{v}$，其中: u =未进行测试的时间抖动指标要求的数目 v =时间抖动指标要求的全部数目
	时间抖动测试有效性	定量	$r=1-\frac{u}{v}$，其中: u =未进行有效测试的时间抖动指标要求数目 v =时间抖动指标要求的全部数目
	易恢复性要求符合性	定量	$r=1-\frac{u}{v}$，其中: u=符合的易恢复性要求数目 v=应遵循的全部易恢复性要求总数目
	时间抖动符合性	定量	$r=\frac{u}{v}$，其中: u =符合时间抖动指标要求的数目 v =应遵循的全部时间抖动指标要求的数目

定性度量元

度量元定性计算模型采用专家打分方法，评判相应度量元完成情况，分成A、B、C、D四个级别，四个级别间需区别明显，有利于专家评审。A级为最高级别，依次递减。度量元打分分数越高，表示该项完成得越好。本书采用从最低级别上升到下一个级别的值近似黄金分割的比例递减方法，表示评分级别提高难度越来越难的原则。对A、B、C、D四个级别进行量化，

- 若度量元等级评价为A，则该度量元值$index = 1$；类似地，
- 若为B，则$index = 0.9$；
- 若为C，则$index = 0.7$；
- 若为D，则$index = 0.2$。

定性度量元

若度量元是通过专家评审进行的，若度量元是通过专家评审进行的，假设有m个专家，依次获得专家评审值为：

$$index^1, \ldots, index^m$$

则度量元$index$可以通过下面计算公式求得：

$$index = \frac{index^1 + index^2 + \cdots + index^m}{m}$$

在该模型中$index$通过对各专家的度量值求平均值得到。当然也可对各专家设置权重，然后通过对各专家的度量值加权求和得到$index$。

度量元量纲

注意到，全生命周期度量元值在$[0,1]$区间，但软件可信属性以及子属性的可信度量值在区间$[1,10]$中，因而需要将全生命周期度量元从$[0,1]$映射到区间$[1,10]$，其方法有多种，比如可采用线性函数变换，即

$$index = 9 \times index + 1$$

还可以采用分段函数变换，即

$$index = \begin{cases} 1, & index \leq \frac{1}{10} \\ 10 \times index, & 其他情形 \end{cases}$$

度量元计算模型

度量元是软件可信度量与评估的最基础层，度量元计算模型依据所采集数据计算得到度量元值，根据所采集到的数据类型，其可分为定量计算模型和定性计算模型。在基于全生命周期的软件可信度量模型中两类模型均有涉及到，在基于出厂报告的软件可信度量模型中只涉及定性计算模型。

- 全生命周期度量元模型
- 出厂报告度量元模型

出厂报告度量元度量模型

软件出厂时通常会组织专家对即将出厂的软件进行评审，评审通过了才能出厂使用。评审会，一般要求专家基于软件出厂报告进行评审，为使专家在评审时能依据评审材料给出客观评价，
基于出厂报告的可信属性分层模型28 个子属性基础上，给出隶属这些子属性的度量元。
每个子属性都有4 个度量元，分别为标注为A、B、C、D，形成了4个级别，这4个级别间区别明显，有利于专家评审。
A级为最高级别，依次递减。
这样共有112个度量元，每个度量元都进行定性描述，以示度量元之间的明显差异，方便专家选择。

出厂报告度量元度量模型

在基于出厂报告的软件可信性度量中，本书仅涉及到112个度量元，而且都是定性度量元。本书也是采用从最低级别上升到下一个级别的值近似黄金分割的比例递减方法，表示评分级别提高难度越来越难的原则。对A、B、C、D四个级别进行量化，即，

- 若专家对第i个可信属性的第j个可信子属性的第k个度量元等级评价为A，则$index_{ijk} = 10$;
- 若为B，则$index_{ijk} = 9$;
- 若为C，则$index_{ijk} = 7$;
- 若为D，则$index_{ijk} = 2$。

总体策划与执行度量元模型

可信属性	子属性	度量元
总体策划与执行	开发计划与执行情况	A：软件开发计划受控、评审符合要求、软件类型清楚（I、II、III、IV类）、人员岗位明确，需求、设计编码、测试三分离，软件实际研制过程严格按照软件技术流程执行，软件实际完成情况与估算不存在偏差；
		B：软件开发计划受控、评审符合要求、软件类型清楚（I、II、III、IV类），软件实际研制过程与项目办要求的研制技术流程相比进行的调整均经过研制单位负责人（分系统副总师）审批，软件实际完成情况与估算偏差在合理范围内；
		C：软件开发计划受控、软件类型清楚（I、II、III、IV类），软件实际研制过程与项目办要求的研制技术流程相比进行的调整均经过研制单位负责人（分系统副总师）审批，针对软件实际完成情况与估算偏差不在合理范围内的情况均进行相应的分析、且处理措施合理；
		D：软件开发计划不受控，或软件类型不清楚（I、II、III、IV类），或软件实际研制过程未按照项目办要求的研制技术流程执行，或流程调整未均经过研制单位负责人（分系统副总师）审批，或针对软件实际完成情况与估算偏差不在合理范围内的情况未全部进行相应的分析、或存在偏差的情况处理措施不合理。
	功能、性能与任务书符合情况	A：软件的功能、性能、可靠性、安全性和可维护性全部满足任务书要求；
		B：软件的功能、性能、可靠性、安全性满足任务书要求；
		C：软件的可靠性、安全性满足任务书要求，功能、性能基本满足任务书要求，未满足任务书要求的功能、性能项均已办理超差申请或偏离申请、经过上级主管部门组织的评审一致通过并确认对系统无影响；
		D：软件的可靠性、安全性不满足任务书要求，或不满足任务书要求的功能、性能项未办理超差申请或偏离申请，或超差申请/偏离申请未经过上级主管部门组织的评审一致通过并确认对系统无影响。
	软件开发文档完整情况	A：软件开发及第三方评测文档齐全、受控（含变更），文档编写符合规范、内容完整、准确、一致、可追踪；
		B：软件开发及第三方评测文档齐全、受控（含变更），文档编写符合规范、一致、可追踪；
		C：软件开发及第三方评测文档齐全、受控（含变更）；
		D：软件开发及第三方评测文档不齐全或不受控（含变更）。

子属性计算模型

本小节给出子属性计算模型。
设x是可信属性y的子属性，通过该子属性的度量元值计算得到x的值。假设该子属性有n个度量元，则有n个度量元值$index_1, \ldots, index_n$。

- 度量元有权重：设每个度量元都有权重，依次为$\eta_1, \ldots, \eta_n$，则子属性x_i计算模型如下：

$$x = (index_1)^{\eta_1} \times ((index)_2)^{\eta_2} \times \cdots \times (index_n)^{\eta_n}$$

- 度量元没有权重：

$$x = \frac{index_1 + index_2 + \cdots + index_n}{n}$$

或

$$x = (index_1)^{\frac{1}{n}} \times (index_2)^{\frac{1}{n}} \times \cdots \times (index_n)^{\frac{1}{n}}$$

- 当每个度量元取值在$[1, 10]$区间内时，子属性x_{ij}也在区间$[1, 10]$内取值。

属性计算模型

到目前为止，我们已经介绍了度量元计算模型和子属性计算模型，针对所采集数据，经过上述两个模型计算后，可得到取值范围为$[1,10]$的子属性值。接下来可通过我们在第四章中所给出的可信属性度量模型进行软件属性可信性度量，可采用如下两个度量模型进行属性可信性度量。

设软件可信属性y有m个子属性，每个子属性x_i都有权重β_i，则软件可信属性y的度量值通过下面计算

- 幂积模型：

$$y = x_1^{\omega_1} \times x_2^{\omega_2} \times \cdots \times x_m^{\omega_m}$$

- 幂和模型：

$$y = \left(\sum_{i=1}^{n} \omega_i x_i^{-\rho_y}\right)^{-\frac{1}{\rho_y}}$$

- 由于每个子属性x_i在区间$[1,10]$中取值，因此属性y也在区间$[1,10]$中取值。

软件可信性计算模型

软件可信性计算模型分两类：

- 不分关键和非关键属性：幂函数乘积综合计算模型。
  设软件有N个可信属性，每个属性y_i都指导一个权重α_i，则软件可信度计算公式为：

$$T_0 = y_1^{\alpha_1} \times y_2^{\alpha_2} \times \cdots \times y_N^{\alpha_N}$$

Claim

我们这里给出的软件可信评估计算方法是幂函数乘积组合，它一方面满足软件可信性度量性质；另一方面，符合“串联规则”，体现了每个属性都是重要的；再者，更体现了“木桶原理”，为了保证软件系统的可信性，必须把每个属性都做好才行。

- 分关键和非关键属性：有m个关键属性$y_1,\ldots,y_m$，其对应权重为$\alpha_1,\ldots,\alpha_m$，和s个非关键属性$y_{m_1},\ldots,y_{m+s}$，其对应权重为$\beta_{m+1},\ldots\beta_{m+s}$；关键属性整体权重为$\alpha$非关键属性整体权重为$\beta$；$\varepsilon$和$\rho$为参数。

$$T_3=(\alpha[\min_{1\le i\le m}\{(\frac{y_i}{10})^{\epsilon}\}y_1^{\alpha_1}y_2^{\alpha_2}\cdots y_m^{\alpha_m}]^{-\rho}+\beta[y_{m+1}^{\beta_{m+1}}y_{m+2}^{\beta_{m+2}}\cdots y_{m+s}^{\beta_{m+s}}]^{-\rho})^{-\frac{1}{\rho}}$$

Claim

注意权重值和参数的取值范围。

1. ϵ：调控参数，用来调控最小关键属性对软件可信性的影响，满足$0\le\epsilon\le\min\{1-\alpha_{\min'},\frac{\ln y_0-\ln y_{\min'}}{\ln y_{\min'}-\ln 10},\rho\}$，且$\epsilon$越大，影响越大，$\alpha_{\min'}$表示最小关键属性在整个关键属性集中所占的权重；
2. ρ：与关键属性和非关键属性之间替代性相关的参数，满足$0<\rho$，且其值越大，则关键属性与非关键属性间替代性越难。

例子：出厂报告

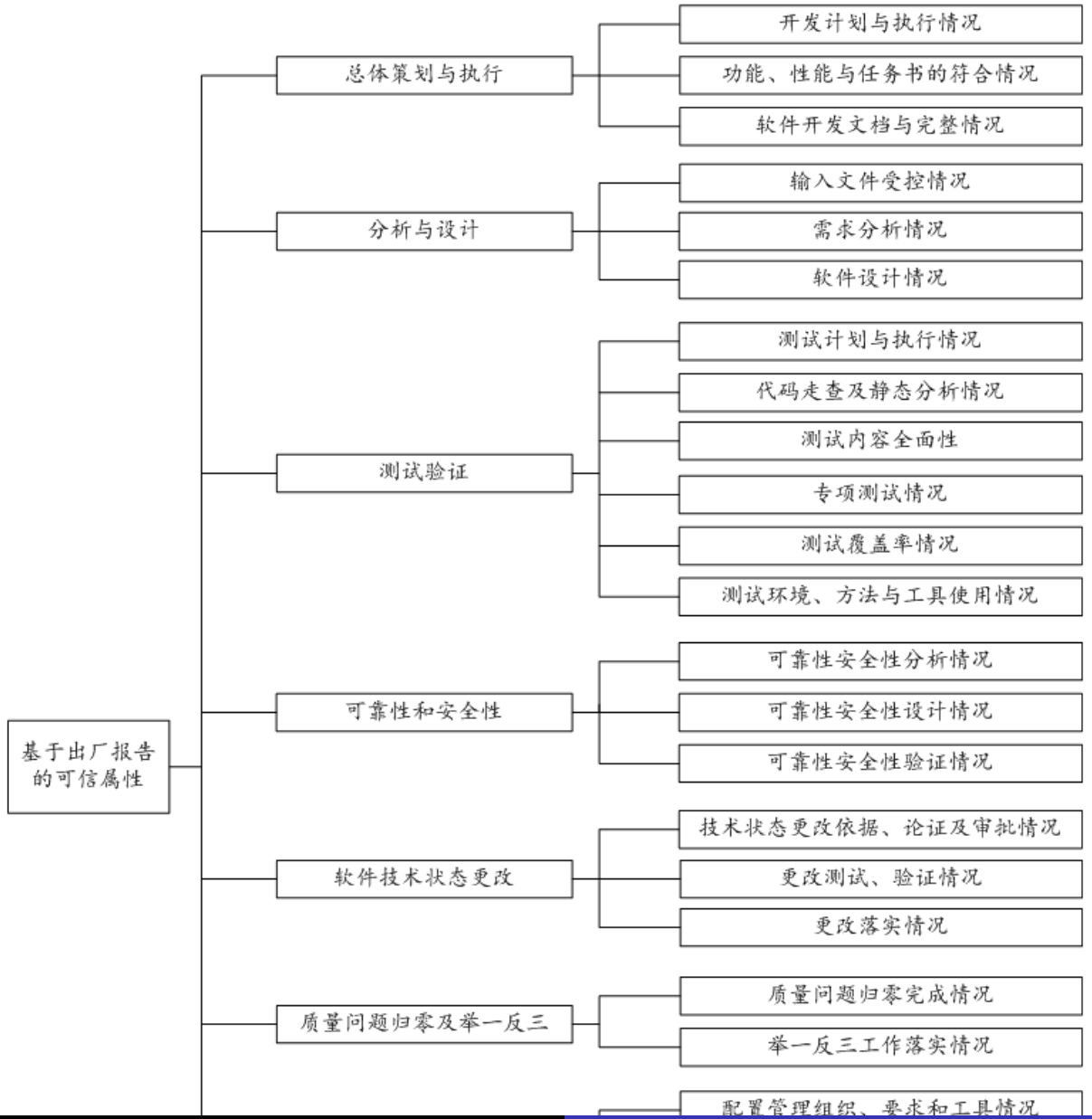
软件出厂时通常会组织评审专家基于软件出厂报告对软件进行评审，为使专家在评审时能依据评审材料给出客观评价，

- 结合出厂报告现有验收、评价方法和标准，
- 建立了基于出厂报告的软件可信性层次化模型，其由基于出厂报告的可信属性模型、基于出厂报告的可信属性分层模型和基于出厂报告的可信度量元模型构成。

出厂报告可信性属性分层模型

- 基于出厂报告的可信属性模型包含9个可信属性，即，总体策划与执行情况、分析与设计情况、测试验证情况、可靠性与安全性情况、软件技术状态更改情况、质量问题归零及举一反三情况、配置管理情况、软件开发环境情况和第三方评测情况。
- 基于出厂报告的可信属性分层模型如图所示，每个可信属性含有2-6个子属性，共计28个子属性.
- 基于出厂报告的可信度量元模型中每个子属性包含4个度量元，每个度量元依次对应A、B、C、D四个等级中的一个，共计112个度量元，

出厂报告可信性属性分层模型



出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
总体策划与执行	开发计划与执行情况	A：软件开发计划受控、评审符合要求、软件类型清楚（I、II、III、IV类）、人员岗位明确，需求、设计编码、测试三分离，软件实际研制过程严格按照软件技术流程执行，软件实际完成情况与估算不存在偏差；
		B：软件开发计划受控、评审符合要求、软件类型清楚（I、II、III、IV类），软件实际研制过程与项目办要求的研制技术流程相比进行的调整均经过研制单位负责人（分系统副总师）审批，软件实际完成情况与估算偏差在合理范围内；
		C：软件开发计划受控、软件类型清楚（I、II、III、IV类），软件实际研制过程与项目办要求的研制技术流程相比进行的调整均经过研制单位负责人（分系统副总师）审批，针对软件实际完成情况与估算偏差不在合理范围内的情况均进行相应的分析、且处理措施合理；
		D：软件开发计划不受控，或软件类型不清楚（I、II、III、IV类），或软件实际研制过程未按照项目办要求的研制技术流程执行，或流程调整未均经过研制单位负责人（分系统副总师）审批，或针对软件实际完成情况与估算偏差不在合理范围内的情况未全部进行相应的分析、或存在偏差的情况处理措施不合理。
	功能、性能与任务书符合情况	A：软件的功能、性能、可靠性、安全性和可维护性全部满足任务书要求；
		B：软件的功能、性能、可靠性、安全性满足任务书要求；
		C：软件的可靠性、安全性满足任务书要求，功能、性能基本满足任务书要求，未满足任务书要求的功能、性能项均已办理超差申请或偏离申请、经过上级主管部门组织的评审一致通过并确认对系统无影响；
		D：软件的可靠性、安全性不满足任务书要求，或不满足任务书要求的功能、性能项未办理超差申请或偏离申请，或超差申请/偏离申请未经过上级主管部门组织的评审一致通过并确认对系统无影响。
	软件开发文档完整情况	A：软件开发及第三方评测文档齐全、受控（含变更），文档编写符合规范、内容完整、准确、一致、可追踪；
		B：软件开发及第三方评测文档齐全、受控（含变更），文档编写符合规范、一致、可追踪；
		C：软件开发及第三方评测文档齐全、受控（含变更）；
		D：软件开发及第三方评测文档不齐全或不受控（含变更）。

出厂报告的软件可信属性、子属性、度量元

分析与设计	输入文件受控情况	A：输入文件齐全、受控（含变更），文档编写符合规范、内容完整、准确、一致；
		B：输入文件齐全、受控（含变更），文档编写符合规范、一致；
		C：输入文件齐全、受控（含变更）；
		D：输入文件不齐全或不受控（含变更）。
	需求分析情况	A：需求规格说明受控、经过主管部门组织的评审且项目办软件负责人及全部利益相关方均参加评审、评审问题已闭环，需求规格说明内容完整，软件运行环境、硬件接口及软件接口明确、功能划分粒度合理、描述清晰无歧义、性能指标明确、量化，需求项覆盖任务书中全部要求，所有需求项均有明确的来源且与输入文件一致，所有需求均可测试验证
		B：需求规格说明受控、经过主管部门组织的评审且项目办软件负责人及全部利益相关方均参加评审、评审问题已闭环，需求规格说明内容完整，需求规格说明中软件运行环境、硬件接口及软件接口明确、功能描述清晰无歧义、性能指标明确、量化，需求项覆盖任务书中全部要求，所有需求项均有明确的来源且与输入文件一致。
		C：需求规格说明受控，需求规格说明中硬件接口及软件接口明确、功能描述清晰无歧义、性能指标明确，需求项覆盖任务书中全部要求，所有需求项均有明确的来源且与输入文件一致。
		D：需求规格说明不受控，或需求规格说明中硬件接口、软件接口不明确，或功能描述有歧义，或性能指标不明确，或需求项未覆盖任务书中全部要求，或存在需求项无明确的来源或与输入文件不一致的情况。
	软件设计情况	A：设计报告受控，经主管部门组织评审且项目办软件负责人参加评审、评审问题已闭环，设计报告内容完整、模块划分合理（高内聚、低耦合），模块间接口明确，模块输入、输出清晰，覆盖需求规格说明中全部需求；
		B：设计报告受控，经主管部门组织评审且项目办软件负责人参加评审、评审问题已闭环，模块划分合理（高内聚、低耦合）、模块间接口明确，模块输入、输出清晰，覆盖需求规格说明中全部需求；
		C：设计报告受控，覆盖需求规格说明中全部功能需求和接口需求；
		D：设计报告不受控，或未覆盖需求规格说明中全部功能需求和接口需求。

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
测试验证	测试计划与执行情况	A：单元测试、组装测试、确认测试均有相应的测试计划，被测软件版本、测试项目、测试阶段明确，测试过程实际完成情况与测试计划一致，经过开发方（单元测试、组装测试、确认测试，及相应的回归测试）、分系统、整星（船、器）的全部测试，测试记录完整，测试时间明确，被测试功能具体、明确；
		B：单元测试、组装测试、确认测试均有相应的测试计划，被测软件版本明确，测试过程实际完成情况与测试计划一致,测试记录完整，测试时间明确；
		C：单元测试、组装测试、确认测试均有相应的测试计划，经过了开发方（单元测试、组装测试、确认测试，及相应的回归测试）、分系统测试；
		D：无单元测试计划、确认测试计划中的任一测试计划，或未经过开发方单元测试、或确认测试或分系统测试。
	代码走查及静态分析情况	A：针对待入库（受控库和产品库中）的每一个版本均进行了静态分析和项目组级代码走查，针对已入库（受控库和产品库中）的每一个版本均进行了室级或型号级走查，静态分析和代码走查报告齐全、受控，代码走查发现的问题描述清楚、处理措施明确、问题分类准确，静态分析和代码走查发现的全部问题均已闭环，集成环境工具使用验证时序正确性、程序实现正确性和控制行为正确性与分析设计阶段的一致性；
		B：针对若干关键版本和最终版本均进行了静态分析和代码走查，静态分析和代码走查报告齐全，代码走查发现的问题描述清楚、处理措施明确、问题分类准确，代码走查发现的全部问题均已闭环，静态分析发现的题如不处理，均说明原因且原因合理，集成环境工具使用验证时序正确性、程序实现正确性和控制行为正确性中的两项与分析设计阶段的一致性；
		C：进行了静态分析和代码走查，静态分析和代码走查报告齐全，代码走查发现的问题处理措施明确，代码走查发现的全部问题均已闭环，静态分析发现的题如不处理，均说明原因且原因合理，集成环境工具使用验证时序正确性、程序实现正确性和控制行为正确性中的一项与分析设计阶段的一致性；
		D：未进行静态分析或代码走查，或无静态分析或代码走查报告，或代码走查发现的问题处理措施不明确，或存在未闭环的代码走查问题，或存在不处理且未说明原因（或原因不合理）的静态分析问题，未使用集成环境工具或集成环境工具使用验证时序正确性、程序实现正确性和控制行为正确性中的任何一项与分析设计阶段的不一致。

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
测试验证	测试内容全面性情况	A：单元测试进行了软件单元的功能测试、接口测试、重要执行路径的测试、局部数据结构测试、错误处理测试及影响上述测试的边界条件测试，组装测试进行了软件单元和（或）软件部件的接口测试、软件部件和（或）软件配置项的接口测试、全局数据结构测试、运行时间测试、运行空间测试、计算精度测试（只针对有数据精度要求的软件部件）及边界条件和非法输入的性能测试，确认测试进行了功能测试、性能测试、余量测试、边界测试、⑴⑺安全性测试、可靠性测试、外部接口测试、强度测试。
		B：单元测试进行了软件单元的功能测试、接口测试、重要执行路径的测试及影响上述测试的边界条件测试，组装测试进行了软件单元和（或）软件部件的接口测试、软件部件和（或）软件配置项的接口测试，确认测试进行了功能测试、性能测试、安全性测试、可靠性测试、外部接口测试、强度测试。
		C：单元测试进行了软件单元的功能测试、边界条件测试，组装测试进行了软件单元和（或）软件部件的接口测试、软件部件和（或）软件配置项的接口测试，确认测试进行了功能测试、性能测试、安全性测试、可靠性测试、外部接口测试。
		D：单元测试未进行软件单元的功能测试或边界条件测试，组装测试未进行软件单元和（或）软件部件的接口测试或软件部件和（或）软件配置项的接口测试，确认测试未进行功能测试、性能测试、安全性测试、可靠性测试、外部接口测试中的任一测试。
	专项测试情况	A：分别进行了数据和内存访问冲突、时序、堆栈使用、指针、除零、数组溢出、条件关系、数据初始化、计算精度、变量的约束、量纲、调用、函数输入等专项复查和专项测试，复查结果无问题、测试结果正确，或复查、测试问题均已闭环；
		B：分别进行了数据和内存访问冲突、时序、堆栈使用、条件关系、数据初始化、计算精度等专项复查和专项测试，复查/测试结果无问题，或复查/测试问题均已闭环，或复查/测试问题经主管部门软件负责人确认对系统无影响；
		C：进行了数据和内存访问冲突、时序、堆栈使用、条件关系、数据初始化、计算精度等专项复查和专项测试中的一项至五项，复查/测试结果无问题，或复查/测试问题均已闭环，或复查/测试问题经主管部门软件负责人确认对系统无影响；
		D：未进行任何专项复查和专项测试。

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
测试验证	测试覆盖率情况	A：单元测试覆盖软件详细设计报告中全部软件单元、源代码和目标码的分支、语句覆盖率均达到100%、MC/DC覆盖率达到100%，组装测试覆盖软件概要设计报告中全部软件部件及软件部件接口，确认测试覆盖软件需求规格说明中全部功能、性能、可靠性、安全性、接口等需求项以及全部安全关键功能、状态/模式转换情况、源代码和目标码的分支、语句覆盖率均达到100%，系统测试覆盖系统功能、系统性能、软件配置项接口、故障模式测试，测试中发现的问题全部落实更改，回归单元、组装、确认、系统测试覆盖全部更改项且问题均已解决。
		B：单元测试覆盖软件详细设计报告中全部软件单元、源代码的分支、语句、MC/DC覆盖率均达到100%，组装测试覆盖软件概要设计报告中全部软件部件及软件部件接口，确认测试覆盖软件需求规格说明中全部功能、性能、可靠性、安全性、接口等需求项以及全部安全关键功能、状态/模式转换情况，系统测试覆盖系统功能、系统性能、软件配置项接口、故障模式测试，测试中发现的问题全部落实更改，回归单元、组装、确认测试覆盖全部更改项且问题均已解决。
		C：单元测试覆盖软件详细设计报告中全部软件单元、源代码的分支、语句覆盖率均达到100%，MC/DC覆盖率未达到100%的给出原因说明及正确性分析，组装测试覆盖软件概要设计报告中全部软件部件及软件部件接口，确认测试覆盖软件需求规格说明中全部功能、性能、可靠性、安全性、接口等需求项，系统测试覆盖系统功能、系统性能、软件配置项接口、故障模式测试，测试中发现的问题未更改的给出原因说明及正确性分析，经主管部门软件负责人确认对系统无影响。
		D：单元测试未覆盖软件详细设计报告中全部软件单元、或源代码的分支或语句覆盖率未达到100%，或MC/DC覆盖率未达到100%的未给出原因说明及正确性分析，或组装测试未覆盖软件概要设计报告中的全部软件部件及软件部件接口，或确认测试未覆盖软件需求规格说明中全部功能、性能、可靠性、安全性、接口等需求项，或测试中发现的问题未更改的未给出原因说明及正确性分析，或未经主管部门软件负责人确认对系统无影响。
	测试环境、方法与工具使用情况	A：软件测试环境与实际软件运行环境之间的差异很小，采用的测试方法对被测软件的运行情况不产生影响，能够真实的反应实际软件运行情况；
		B：软件测试环境与实际软件运行环境之间的差异较小，采用的测试方法对被测软件的运行情况的影响较小，能够较真实的反应实际软件运行情况，对软件测试环境、方法、工具与实际软件运行情况之间的差异所产生的影响、测试的有效性、测试不到的情况均进行了分析；
		C：软件测试环境与实际软件运行环境之间的差异较小，采用的测试方法对被测软件的运行情况的影响较小，能够较真实的反应实际软件运行情况；
		D：软件测试环境与实际软件运行环境之间存在较大差异，采用的测试方法对被测软件的运行

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
可靠性和安全性	可靠性安全性分析情况	A：软件可靠性安全性分析充分，根据软件研制任务书中的可靠性、安全性、可维护性要求以及通用软件可靠性安全性要求明确提出软件可靠性安全性需求，需求跟踪矩阵中体现了软件可靠性安全性需求与任务书中可靠性安全性要求的追踪关系；采用成熟的方法和技术（如SFMEA、SFTA、HSIA等）开展了软件可靠性安全性分析，并确定了安全关键软部件；
		B：软件可靠性安全性分析较充分，根据软件研制任务书中的可靠性、安全性、可维护性要求以及通用软件可靠性安全性要求明确提出软件可靠性安全性需求，需求跟踪矩阵中体现了软件可靠性安全性需求与任务书中可靠性安全性要求的追踪关系；
		C：进行了软件可靠性安全性分析，但未建立软件可靠性安全性需求与任务书中可靠性安全性要求的追踪关系；
		D：未基于软件研制任务书中的可靠性安全性要求进行软件可靠性安全性分析，未提出软件可靠性安全性需求。
	可靠性安全性设计情况	A：根据软件可靠性安全性需求进行软件可靠性安全性设计，并建立软件设计和软件可靠性安全性需求的追踪关系；针对安全关键软件进行了独立性设计、冗余设计、软件单粒子防护设计、可恢复性设计、余量设计、可复用设计、可维护性设计等专项设计工作；
		B：根据软件可靠性安全性需求进行软件可靠性安全性设计，并建立软件设计和软件可靠性安全性需求的追踪关系；
		C：进行了软件可靠性安全性设计，但未建立软件设计和软件可靠性安全性需求的追踪关系；
		D：未基于软件可靠性安全性需求进行软件可靠性安全性设计。
	可靠性安全性验证情况	A：在软件单元、部件、配置项、分系统（单机）和系统各级逐级验证了软件是否满足任务书中可靠性、安全性和可维护性要求，并建立了验证与可靠性安全性要求的追踪关系；针对安全关键软件进行了专项分析（如数据分析、中断分析、堆栈分析、单粒子效应分析、可恢复性分析等）和专项测试（强度测试、安全性测试、故障注入测试、可靠性测试、余量测试、边界测试等）；
		B：在软件单元、部件、配置项、分系统（单机）和系统各级逐级验证了软件是否满足任务书中可靠性、安全性和可维护性要求，并建立了验证与可靠性安全性要求的追踪关系；
		C：在软件单元、部件、配置项、分系统（单机）和系统各级只验证了部分软件任务书中的可靠性、安全性和可维护性要求，未建立验证与可靠性安全性要求的追踪关系；
		D：未针对任务书中的可靠性安全性要求进行相应的测试或分析验证。

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
软件技术状态更改	技术状态更改依据、论证及审批情况	A：更改依据明确、更改影响域分析全面、更改论证充分，技术状态更改经各方认可，更改审批完备、符合要求；
		B：更改依据明确、更改影响域分析全面、更改论证充分，更改审批完备、符合要求；
		C：更改有依据，进行了技术状态更改论证，更改审批完备、符合要求；
		D：更改无依据，或未进行技术状态更改论证，或更改审批不符合要求。
	更改测试、验证情况	A：针对更改影响到的全部功能、性能、接口、可靠性和安全性、故障处理情况进行了测试验证，并进行了相关的余量测试、边界测试、强度测试，测试验证均通过，更改措施有效；
		B：针对更改影响到的全部功能、性能、接口、可靠性和安全性、故障处理情况进行了测试验证，测试验证均通过，更改措施有效；
		C：针对更改影响到的全部功能、性能、接口进行了测试验证，测试验证均通过，更改措施有效；
		D：未针对更改影响到的全部功能、性能及接口进行测试验证，或测试验证存在不通过的情况。
	更改落实情况	A：更改涉及到的文件均完成更改并归档，更改文件完整、正确、一致，更改涉及到的代码完成更改并归档；
		B：更改涉及到的文件、代码均完成更改并归档；
		C：更改涉及到代码完成更改并归档；
		D：更改涉及到代码未完成更改并归档；

可信属性	子属性	度量元
质量问题归零及举一反三	质量问题归零完成情况	A：未发生质量问题或发生质量问题已进行质量问题技术归零和质量问题管理归零，质量问题技术归零满足定位准确、机理清楚、问题复现、措施有效、举一反三五条要求，质量问题管理归零满足过程清楚、责任明确、措施落实、严肃处理、完善规章五条要求；
		B：发生质量问题已进行质量问题技术归零，质量问题技术归零满足定位准确、机理清楚、问题复现、措施有效、举一反三五条要求；
		C：发生质量问题已进行质量问题分析，报告内容完整（包含故障定位过程、机理和危害度分析、采取措施有效的验证结果、不能完全归零的理由、不影响飞行试验成败的明确结论），针对风险制定了对策，经分析、评审认为不影响任务完成；
		D：发生质量问题，未全面完成技术归零或质量问题分析工作。
	举一反三工作落实情况	A：针对本型号（如果有）和其他型号质量问题的举一反三全面、彻底，举一反三记录完整；
		B：针对本型号（如果有）和其他型号质量问题进行了举一反三，举一反三有记录；
		C：针对本型号（如果有）的问题进行了举一反三；
		D：未针对本型号（如果有）的问题进行举一反三。
配置管理	配置管理组织、要求和工具情况	A：配置管理计划内容完整、配置标识明确，有针对开发库、受控库和产品库配置管理组织、要求和工具。
		B：配置管理计划中配置标识明确，有针对开发库、受控库和产品库配置管理组织、要求和工具。
		C：配置管理计划中有针对开发库、受控库和产品库配置管理组织、要求和工具。
		D：配置管理计划中针对开发库、受控库和产品库配置管理组织、要求和工具不全。
	变更控制情况	A：有三级配置库（开发库、受控库和产品库），四单（问题报告单、更改单、出库单、入库单）齐备、内容完整、审批符合要求；
		B：有三级配置库（开发库、受控库和产品库），四单（问题报告单、更改单、出库单、入库单）齐备、审批符合要求；
		C：有三级配置库（开发库、受控库和产品库），四单（问题报告单、更改单、出库单、入库单）齐备；
		D：无三级配置库（开发库、受控库和产品库），或四单（问题报告单、更改单、出库单、入库单）不齐备
	配置审计、纪实情况	A：基线审计报告、软件配置状态报告齐备，内容完整；
		B：基线审计报告、软件配置状态报告齐备；
		C：有基线审计报告和软件配置状态报告；
		D：无基线审计报告或无软件配置状态报告。

出厂报告的软件可信属性、子属性、度量元

可信属性	子属性	度量元
软件开发环境	开发配套软件	A：开发配套软件清单完整齐套（编译工具、固化软件、调试工具、测试工具）、覆盖软件全过程（需求开发、需求管理、设计、测试、代码走查）、配套软件受控（经过质量机构认可或第三方评测机构认证并颁发证书）；
		B：开发配套软件清单完整齐套、配套软件受控；
		C：开发配套软件受控；
		D：开发配套软件不受控。
	开发配套硬件	A：开发配套硬件清单齐备、校准规范完整、配套硬件使用时在有效期内、设备环境符合要求、有记录；
		B：开发配套硬件校准规范完整、配套硬件使用时在有效期内、设备环境符合要求、有记录；
		C：开发配套硬件使用时在有效期内、设备环境符合要求、有记录；
		D：开发配套硬件使用时不在有效期内、或设备环境不符合要求、或无记录；
第三方评测	评测输入和评测计划情况	A：测试输入文件齐备、输入文件受控（代码与文档版本一致）、评测计划（人员配置、时间安排、测试类型满足评测任务书要求）全面、满足要求；
		B：输入文件受控、评测计划全面、满足要求；
		C：输入文件受控、有评测计划；
		D：输入文件不受控、或无评测计划。
	评测执行情况	A：功能、性能测试覆盖率达到100%，测试环境满足测试要求，边界、余量、可靠性、强度等测试内容充分，测试数据记录准确、完整；
		B：功能、性能测试覆盖率达到100%，测试环境满足测试要求，测试数据记录准确、完整；
		C：功能、性能测试覆盖率达到100%，或未达到100%的给出原因说明且原因充分；
		D：功能、性能测试覆盖率未达到100%，且原因不充分。
	问题解决情况	A：测试问题全部修改正确，通过了回归测试，并根据问题进行了举一反三；
		B：测试问题全部修改正确，并通过了回归测试；
		C：存在未修改的问题或问题修改不全面，但对软件的功能、性能影响可接受；
		D：存在未修改的问题或问题修改不全面，并对软件的功能、性能有不良影响。

基于出厂报告的软件可信性分级模型

软件可信度量值要求	可信属性要求	可信等级
$9.5 <= T$	1. 低于9.5分的可信属性个数不超过3个 2. 没有低于8.5分的可信属性	V
$8.5 <= T < 9.5$ 或者$T > 9.5$且不能评为V级别	1. 低于8.5分的可信属性个数不超过3个 2. 没有低于7.0分的可信属性	IV
$7.0 <= T < 8.5$ 或者$T > 8.5$且不能评为IV级别及以上者	1. 低于7.0分的可信属性个数不超过3个 2. 没有低于4.5分的可信属性	III
$4.5 <= T < 7.0$ 或者$T > 7.0$且不能评为III级别及以上者	1. 低于4.5分的可信属性个数不超过3个	II
$T < 4.5$ 或者$T > 4.5$且不能评为II级别及以上者	1. 无要求	I

某类型软件可信属性和子属性权重值

属性	属性权重	子属性	子属性权重
1、总体策划与执行	0.05	1、开发计划与执行情况	0.31
		2、功能、性能与任务书符合情况	0.36
		3、软件开发文档与完整情况	0.33
2、分析与设计	0.17	4、输入文件受控情况	0.33
		5、需求分析情况	0.33
		6、软件设计情况	0.34
3、测试验证	0.20	7、测试计划与执行情况	0.16
		8、代码走查及静态分析情况	0.17
		9、测试内容全面性	0.17
		10、专项测试情况	0.17
		11、测试覆盖率情况	0.17
		12、测试环境、方法与工具使用情况	0.16
4、可靠性和安全性	0.15	13、可靠性安全性分析情况	0.33
		14、可靠性安全性设计情况	0.34
		15、可靠性安全性验证情况	0.33
5、软件技术状态更改	0.09	16、技术状态更改依据、论证及审批情况	0.34
		17、更改测试、验证情况	0.33
		18、更改落实情况	0.33
6、质量问题归零及举一反三	0.09	19、质量问题归零完成情况	0.5
		20、举一反三工作落实情况	0.5
7、配置管理	0.11	21、配置管理组织、要求和工具情况	0.33
		22、变更控制情况	0.34
		23、配置审计、纪实情况	0.33
8、软件开发环境	0.05	24、开发配套软件	0.5
		25、开发配套硬件	0.5
9、第三方测评	0.09	26、评测输入和评测计划情况	0.33
		27、评测执行情况	0.33
		28、问题解决情况	0.34

具体应用

- 2014年开展了针对某类型飞行试验器23个软件的专项审查，由10名专家组成软件专项审查专家组对软件进行评分，软件代码总规模约30万行。
- 评审方式是每个软件由2 – 3名专家采用主观与客观结合方式依据度量元表所给出的出厂报告，对28个可信子属性按照A、B、C、D四个等级进行评估.
- 最后，基于出厂报告的软件可信性分级评估方法进行可信性度量评估方法。
- 在参与审查的23 个软件中，本节选取具有代表性的11 个软件（软件编号依次为2、4、6、7、9、18、19、20、21、22、23）的评分结果进行分析和统计。

评审情况

软件编号	软件类型	1. 总体策划与执行情况			2. 分析与设计情况			3. 测试验证情况						4. 可靠性和安全性情况			5. 软件技术状态更改情况			6. 质量问题归零及举一反三情况		7. 配置管理情况			8. 软件开发环境情况		9. 第三方评测情况		
		开发计划与执行情况	功能、性能与任务书符合情况	软件开发文档完整情况	输入件控制情况	需求分析情况	软件设计情况	测试计划与执行情况	代码走查及静态分析情况	测试内容全面性	专项测试情况	测试覆盖率情况	测试环境、方法情况	可靠性安全性分析情况	可靠性安全性设计情况	可靠性安全性验证情况	技术状态更改依据、论证及审批情况	更改测试、验证情况	更改落实情况	质量问题归零完成情况	举一反三工作落实情况	配置管理组织、要求和工具情况	变更控制情况	配置审计、纪实情况	开发配套软件	开发配套硬件	评测输入和评测计划	评测执行情况	问题解决情况
1	I	B	B	B	A	B	B	/	/	/	D	/	/	B	B	B	/	/	/	A	C	/	A	B	B	B	/	B	B
2	IV	B	B	C	B	C	C	B	C	C	D	C	B	B	C	B	B	B	C	A	C	B	B	B	B	B	A	A	B
3	IV	C	B	C	B	C	C	B	B	C	D	C	B	B	B	B	/	/	/	A	C	B	B	B	B	B	A	B	B
4	III	C	B	C	B	C	B	C	D	C	D	C	B	B	C	C	C	B	B	A	C	B	B	B	B	D	B	B	A
5	**I**	**C**	**/**	**C**	**A**	**/**	**/**	**/**	**/**	**/**	**D**	**/**	**A**	**/**	**/**	**/**	**/**	**/**	**/**	**A**	**C**	**C**	**A**	**C**	**C**	**C**	**A**	**A**	**B**
6	III	C	B	C	A	B	B	C	B	C	D	C	C	C	C	C	C	B	B	A	C	C	C	C	C	C	A	A	B
7	**III**	**C**	**B**	**C**	**A**	**B**	**B**	**C**	**B**	**C**	**D**	**C**	**B**	**C**	**C**	**D**	**C**	**B**	**B**	**A**	**C**	**C**	**B**	**C**	**C**	**C**	**A**	**A**	**B**
8	II	A	/	A	/	/	/	/	/	C	D	C	A	B	/	/	B	B	A	A	C	B	A	B	C	B	A	A	C
9	III	C	B	C	A	B	C	C	B	C	C	B	B	C	C	C	C	B	B	B	C	B	B	C	C	B	A	A	B
10	III	C	B	C	/	B	B	C	A	C	D	C	A	C	C	C	C	C	A	/	C	A	A	A	B	B	C	C	/
11	I	C	B	B	B	/	/	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
12	II	C	B	B	B	/	/	A	A	C	C	A	C	B	B	B	A	B	A	A	C	A	A	A	B	B	A	A	A
13	I	C	B	B	B	/	/	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
14	I	C	B	B	B	/	/	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
15	II	C	B	B	B	/	/	A	B	C	C	B	C	B	B	B	A	B	A	A	C	A	A	A	B	B	A	A	A
16	I	C	B	B	B	/	/	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
17	I	C	B	B	B	/	/	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	B
18	IV	C	B	B	B	B	B	B	A	C	B	C	B	B	B	B	B	C	B	A	B	A	A	A	B	B	B	B	C
19	IV	C	B	B	B	B	B	B	B	B	A	B	B	C	B	B	C	B	B	A	C	A	A	B	B	B	B	B	B
20	IV	C	C	C	B	D	B	C	B	C	B	B	B	B	C	C	C	B	C	A	C	B	A	C	B	A	B	B	B
21	IV	C	B	B	A	B	B	B	C	C	A	C	A	C	C	C	B	B	B	A	C	B	A	C	B	B	A	A	C
22	IV	C	B	B	A	B	B	C	C	C	D	C	A	C	C	C	B	B	B	A	C	B	A	C	B	B	A	A	A
23	IV	B	B	B	B	B	B	B	C	B	C	B	A	B	B	B	B	A	C	A	C	A	A	A	B	B	B	B	C

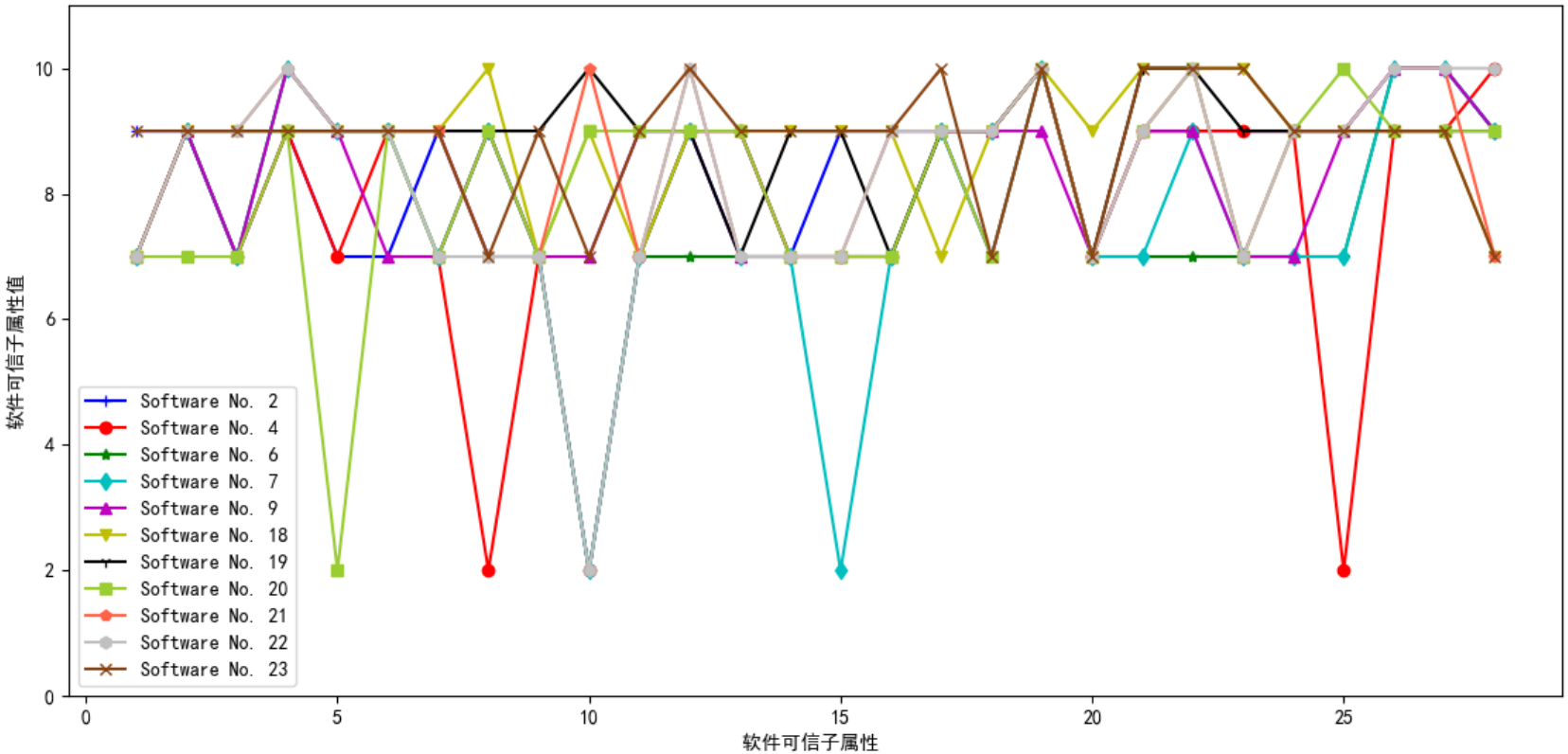
评审情况

软件编号	软件类型	1. 总体策划与执行情况 .05			2. 分析与设计情况 .17			3. 测试验证情况 .20						4. 可靠性和安全性情况 .15			5. 软件技术状态更改情况 .09			6. 质量问题归零及举一反三情况 .09		7. 配置管理情况 .11			8. 软件开发环境情况 .05		9. 第三方评测情况 .09		
		开发计划与执行情况	功能、性能与任务书符合情况	软件开发文档完整情况	输入文件受控情况	需求分析情况	软件设计情况	测试计划与执行情况	代码走查及静态分析情况	测试内容全面性	专项测试情况	测试覆盖率情况	测试环境、方法情况	可靠性安全性分析情况	可靠性安全性设计情况	可靠性安全性验证情况	技术状态更改依据、论证及审批情况	更改测试、验证情况	更改落实情况	质量问题归零完成情况	举一反三工作落实情况	配置管理组织、要求和工具情况	变更控制情况	配置审计、纪实情况	开发配套软件	开发配套硬件	评测输入和评测计划	评测执行情况	问题解决情况
子属性权重		.31	.36	.33	.33	.33	.34	.16	.17	.17	.17	.17	.16	.33	.34	.33	.34	.33	.33	.50	.50	.33	.34	.33	.50	.50	.33	.33	.34
2	IV	9	9	7	9	7	7	9	7	7	2	7	9	9	7	9	9	9	7	10	7	9	9	9	9	9	10	10	9
4	III	7	9	7	9	7	9	7	2	7	2	7	9	9	7	7	7	9	7	10	7	9	9	9	9	2	9	9	10
6	III	7	9	7	10	9	9	7	9	7	2	7	7	7	7	7	7	9	9	10	7	7	7	7	7	7	10	10	9
7	**III**	**7**	**9**	**7**	**10**	**9**	**9**	**7**	**9**	**7**	**2**	**7**	**9**	**7**	**7**	**2**	**7**	**9**	**9**	**10**	**7**	**7**	**9**	**7**	**7**	**7**	**10**	**10**	**9**
9	III	7	9	7	10	9	7	7	9	7	7	9	9	7	7	7	7	9	9	9	7	9	9	7	7	9	10	10	9
18	IV	7	9	9	9	9	9	9	10	7	9	7	9	9	9	9	9	7	9	10	9	10	10	10	9	9	9	9	7
19	IV	7	9	9	9	9	9	9	9	9	10	9	9	7	9	9	7	9	9	10	7	10	10	9	9	9	9	9	9
20	IV	7	7	7	9	2	9	7	9	7	9	9	9	9	7	7	7	9	7	10	7	9	10	7	9	10	9	9	9
21	IV	7	9	9	10	9	9	9	7	7	10	7	10	7	7	7	9	9	9	10	7	9	10	7	9	9	10	10	7
22	IV	7	9	9	10	9	9	7	7	7	2	7	10	7	7	7	9	9	9	10	7	9	10	7	9	9	10	10	10
23	IV	9	9	9	9	9	9	9	7	9	7	9	10	9	9	9	9	10	7	10	7	10	10	10	9	9	9	9	7

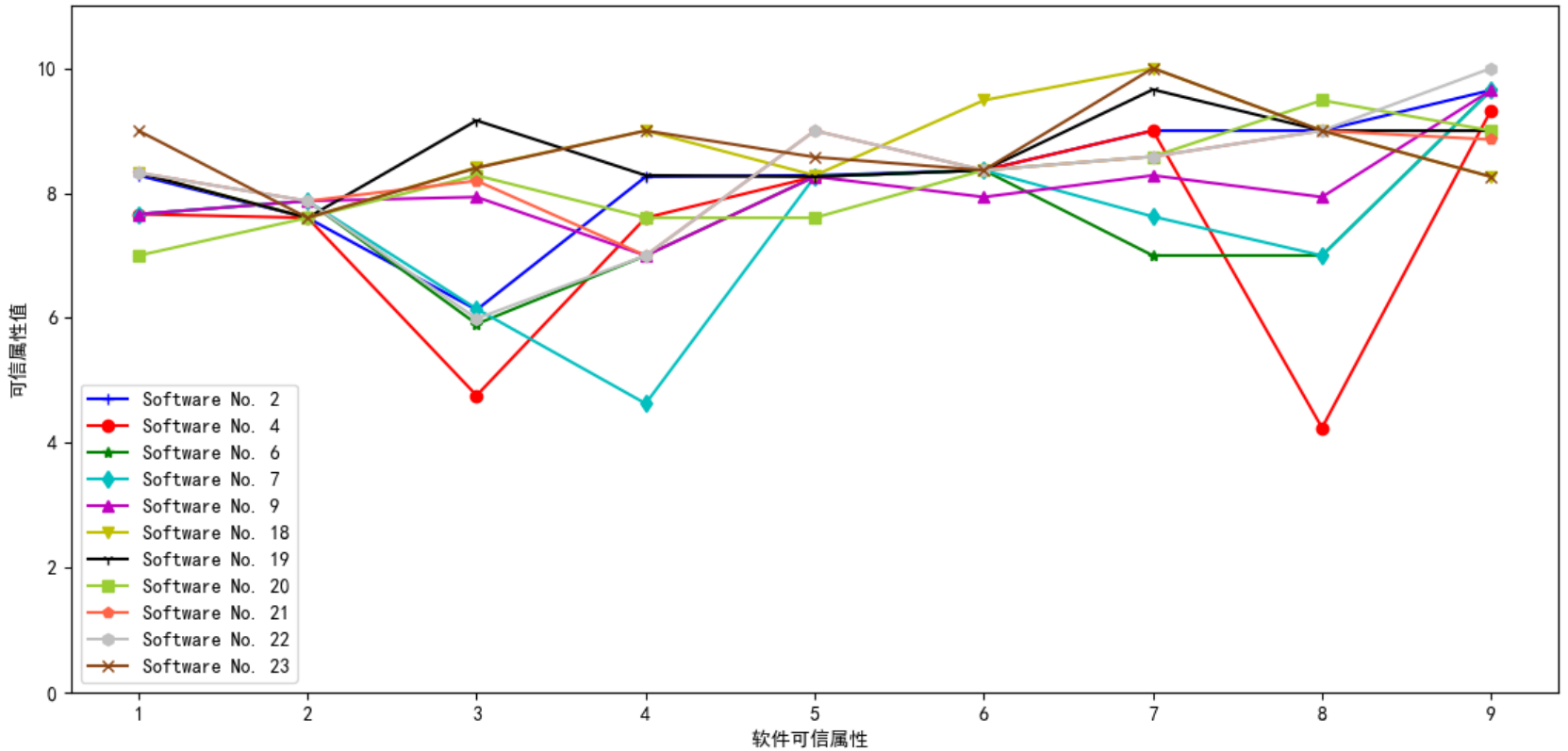
结果统计

	总体策划与执行	分析与设计	测试验证	可靠性和安全性	软件技术状态更改	质量问题归零及举一反三	配置管理	软件开发环境	第三方评测	软件可信性度量值
权重	0.05	0.17	0.20	0.15	0.09	0.09	0.11	0.05	0.09	
2	8.28	7.61	6.13	8.26	8.28	8.37	9.00	9.00	9.65	7.90
4	7.66	7.61	4.76	7.61	8.26	8.37	9.00	4.24	9.37	7.09
6	7.66	7.87	5.90	7.00	8.26	8.37	7.00	7.00	9.65	7.36
7	7.66	7.87	6.15	4.63	8.26	8.37	7.62	7.00	9.65	7.04
9	7.66	7.87	7.94	7.00	8.26	7.93	8.28	7.94	9.64	7.97
18	8.33	7.61	8.41	9.00	8.28	9.49	10.00	9.00	8.26	8.61
19	8.33	7.61	9.16	8.28	8.26	8.37	9.66	9.00	9.00	8.58
20	7.00	7.61	8.28	7.61	7.61	8.37	8.59	9.49	9.00	8.08
21	8.33	7.87	8.20	7.00	9.00	8.37	8.59	9.00	8.56	8.17
22	8.33	7.87	5.99	7.00	9.00	8.37	8.59	9.00	10.00	7.76
23	9.00	7.61	8.40	9.00	8.58	8.37	10.00	9.00	8.26	8.57

各软件（28个）可信子属性值分布情况



各软件（9个）可信属性值分布情况



9个可信属性：总体策划与执行情况、分析与设计情况、测试验证情况、可靠性与安全性情况、软件技术状态更改情况、质量问题归零及举一反三情况、配置管理情况、软件开发环境情况和第三方评测情况。

根据可信属性值计算出软件可信度及其可信等级

	总体策划与执行	分析与设计	测试验证	可靠性和安全性	软件技术状态更改	质量问题归零及举一反三	配置管理	软件开发环境	第三方评测	软件可信度量值	级别评定
权重	0.05	0.17	0.20	0.15	0.09	0.09	0.11	0.05	0.09		
2	8.28	7.61	6.13	8.26	8.28	8.37	9.00	9.00	9.65	7.90	III级
4	7.66	7.61	4.76	7.61	8.26	8.37	9.00	4.24	9.37	7.09	II级
6	7.66	7.87	5.90	7.00	8.26	8.37	7.00	7.00	9.65	7.36	III级
7	7.66	7.87	6.15	4.63	8.26	8.37	7.62	7.00	9.65	7.04	III级
9	7.66	7.87	7.94	7.00	8.26	7.93	8.28	7.94	9.64	7.97	III级
18	8.33	7.61	8.41	9.00	8.28	9.49	10.00	9.00	8.26	8.61	IV级
19	8.33	7.61	9.16	8.28	8.26	8.37	9.66	9.00	9.00	8.58	IV级
20	7.00	7.61	8.28	7.61	7.61	8.37	8.59	9.49	9.00	8.08	III级
21	8.33	7.87	8.20	7.00	9.00	8.37	8.59	9.00	8.56	8.17	III级
22	8.33	7.87	5.99	7.00	9.00	8.37	8.59	9.00	10.00	7.76	III级
23	9.00	7.61	8.40	9.00	8.58	8.37	10.00	9.00	8.26	8.57	IV级

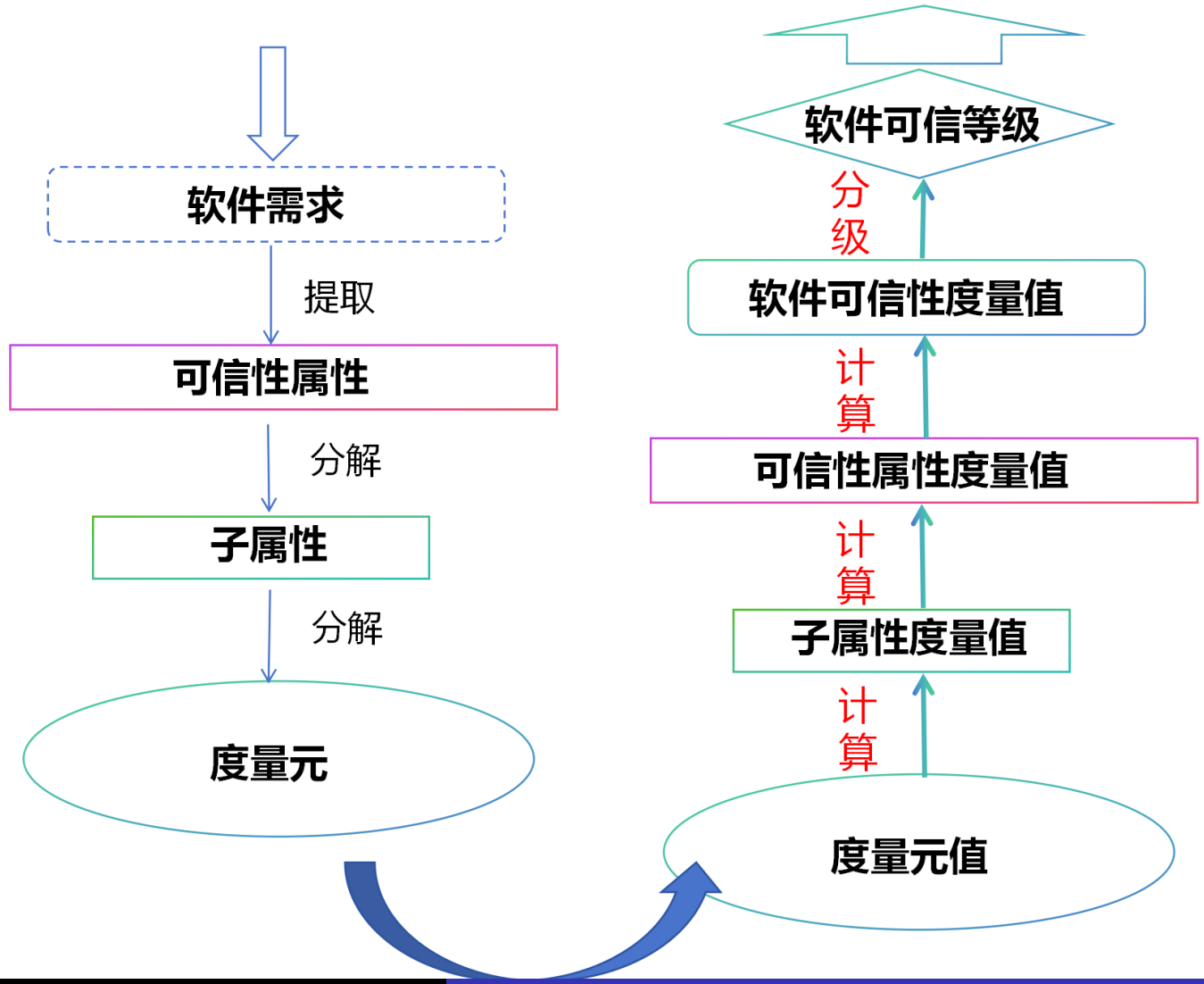
$$V: 9.5 \leq T; IV: 8.5 \leq T < 9.5; III: 7.0 \leq T < 8.5;$$
$$II: 4.5 \leq T < 7.0; I: T < 4.5$$

根据可信属性值计算出软件可信度及其可信等级

这11个软件的可信等级中有3 个软件的可信等级为IV 级，7个软件的可信等级为III 级，90%以上软件均达到较高可信等级。从表中也可看出，软件想要达到最高可信级别V 级很难，这符合可信值越大提高的难度越大，可信级别高对属性可信性值要求也高的原则。

应用情况表明，基于出厂报告的软件可信性分级评估方法适用于该类型软件可信性度量评估，以及可信属性和可信子属性设计合理、科学，模型能够有效评价软件可信性并准确发现软件产品缺陷及研制过程中的薄弱环节，对于该类型软件研制水平的提高有重要意义。在应用过程中通过可信子属性A、B、C、D的评分分布情况可以看出有些评分标准要求太高而很难达到，而有些标准要求太低很容易达到，这需要在未来应用过程中不断反馈和修正；同时，可信子属性和可信属性权重将依据评估的使用进行继续精化。

软件可信性分解与分级流程



作业

依据下表中的数据，计算编号10-17软件的可信属性度量值、软件可信度量值以及可信等级。可信属性权重和子属性权重同例子一样。从可信属性度量值计算以及软件可信度量值计算模型都使用幂积公式。

软件编号	软件类型	1. 总体策划与执行情况			2. 分析与设计情况			3. 测试验证情况						4. 可靠性和安全性情况			5. 软件技术状态更改情况			6. 质量问题归零及举一反三情况		7. 配置管理情况			8. 软件开发环境情况		9. 第三方评测情况		
		开发计划与执行情况	功能、性能与任务书符合情况	软件开发文档完整情况	输入文件受控情况	需求分析情况	软件设计情况	测试计划与执行情况	代码走查及静态分析情况	测试内容全面性	专项测试情况	测试覆盖率情况	测试环境、方法情况	可靠性安全性分析情况	可靠性安全性设计情况	可靠性安全性验证情况	技术状态更改依据、论证及审批情况	更改测试、验证情况	更改落实情况	质量问题归零完成情况	举一反三工作落实情况	配置管理组织、要求和工具情况	变更控制情况	配置审计、纪实情况	开发配套软件	开发配套硬件	评测输入和评测计划	评测执行情况	问题解决情况
10	III	C	B	C	D	B	B	C	A	C	D	C	A	C	C	C	C	C	A	D	C	A	A	A	B	B	C	C	D
11	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
12	II	C	B	B	B	D	D	A	A	C	C	A	C	B	B	B	A	B	A	A	C	A	A	A	B	B	A	A	A
13	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
14	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
15	II	C	B	B	B	D	D	A	B	C	C	B	C	B	B	B	A	B	A	A	C	A	A	A	B	B	A	A	A
16	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	A
17	I	C	B	B	B	D	D	B	C	C	C	B	C	B	C	C	A	B	A	A	C	A	A	A	B	B	A	A	B