团　　体　　标　　准

T/GDFCA 041—2019

基于区块链技术食品追溯系统的可靠性测试标准

Reliability test standard for food traceability system based on blockchain technology

（征求意见稿）

2019-××-××发布　　2019-××-××实施

广东省食品流通协会发布

目　次

前　　言.....III
1　范围.....1
2　规范性引用文件.....1
3　术语和定义.....1
4　测试范围.....2
5　测试方法.....4
6　测试过程.....4
7　测试规范.....5

前言

本标准按照《GB/T 1.1-2009 标准化工作导则 第1部分：标准的结构和编写》标准起草。

本指导性技术文件由广东省食品流通协会提出并归口。

本指导性技术文件起草单位：中山市仁达贸易发展有限公司、广州生命码科技有限公司、中科软件测评（广州）有限公司。

本指导性技术文件主要起草人：

基于区块链技术食品追溯系统的可靠性测试标准

1 范围

基于区块链技术食品追溯系统是一种基于物联网、云计算和区块链等先进技术构建的食品追溯系统，将传统追溯系统与区块链技术进行融合，实现对食品的种植、加工、仓储、运输、销售等全过程的追溯。通过区块链技术去中心化、不可篡改、开放的特性，确保食品源头可追溯、流向可跟踪、信息可查询、信息高可性，保障公众消费和食品安全。

本标准规定了基于区块链技术食品追溯系统的可靠性测试的术语和定义、测试方法和过程、系统可靠性测试规范。

本标准适用于基于区块链技术食品追溯系统的可靠性测试。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25000.10-2016 系统与软件工程 系统与软件质量要求和评价 第10部分：系统与软件质量模型；

GB/T 25000.51-2016 系统与软件工程 系统与软件质量要求和评价 第51部分：就绪可用软件产品的质量要求和测试细则；

GB／T 29832.1-2013_系统与软件可靠性 第3部分：测试方法

3 术语和定义

下列术语和定义适用于本文件。

3.1

区块链 Blockchain

一种分布式分类账，由不可更改的数字化记录的数据组成，称为数据块。然后使用加密签名将每个块“链接”到下一个块。这允许块链像分类帐一样使用，可以由具有适当权限的任何人共享和访问。

3.2

行业区块链 Consortium block chains

由某个群体内部指定多个预选的节点为记账人，每个块的生成由所有的预选节点共同决定，其他接入节点可以参与交易，但不过问记账过程，其他任何人可以通过该区块链开放的API进行限定查询。

3.3

分布式账本 Distributed ledger

是分布在多个站点，国家或机构中的一种数据库。记录一个接一个地存储在连续分类账中。分布式账本数据可以通过“许可”或“不许可”来控制谁可以查看它。

3.4

非对称加密 Asymmetric encryption

非对称加密是一种密钥的保密方法，需要两个密钥：公开密钥（简称公钥）和私有密钥（简称私钥）。公钥与私钥是一对，如果用公钥对数据进行加密，只有用对应的私钥才能解密。因为加密和解密使用的是两个不同的密钥，所以这种算法叫作非对称加密算法。

3.5

共识机制 Consensus mechanism

共识机制是通过特殊节点的投票，在很短的时间内完成对交易的验证和确认；对一笔交易，如果利益不相干的若干个节点能够达成共识，我们就可以认为全网对此也能够达成共识。

3.6

食品追溯 Food tractability

通过记录和标识，追溯食品的历史、应用情况或所处位置的活动，连接生产、检验、监管和消费的各个环节。

3.7

信息编码 Information coding

是指为方便信息的存储、检索和使用，在进行信息处理时赋予信息元素以代码的过程。即用不同的代码与各种信息中的基本单位组成建立一一对应的关系。信息编码必须标准化、系统化。

3.8

信息采集 Information collection

是指根据特定的目标和要求，将分散在不同时空域的有关信息，通过特定的手段和措施采集的过程。采集溯源单位生产企业的基本信息，产品的基本信息、产品质量安全信息。

3.9

信息交换 Information exchange

指数据在不同的信息实体之间进行交互的过程，其目标是在异构环境中实现数据的共享，从而有效的利用资源，加快数据流通，实现数据的集成和共享。

3.10

信息发布 Information release

指信息提供给企业、监管部门和消费者，不同使用者对信息要求不同，信息发布的内容、方式应满足信息使用者的需求。

3.11

系统可靠性 Software reliability

在规定条件下，在规定时间内，不引起系统失效的概率；及在规定的时间周期内，在所述条件下程序执行所要求的功能的能力。其中的概率是系统输入和系统使用的函数，也是系统中存在的故障的函数，系统输入将确定是否会遇到已存在的故障（如果故障存在的话）。

3.12

系统可靠性估计 Software reliability estimation

应用统计技术处理在系统测试和运行期间采集、观察到的失效数据，以评估该系统的可靠性。

3.13

系统可靠性测试 Software reliability test

在有使用代表性的环境中，为进行系统可靠性估计对该系统进行的功能测试。需要说明的是，“使用代表性”指的是在统计意义下该环境能反映出系统的使用环境特性。

4 测试范围

基于区块链技术食品追溯系统信息标准应包括追溯信息编码、信息采集、信息交换、信息发布四大部分，系统中的关键追溯信息记录在区块链的分布式账本节点。分布式账本节点在政府监管部门、行业协会、检测机构和利益相关企业部署。基于区块链技术食品追溯系统组成构架见图4.1，测试范围涵盖下图方框的子系统的软件部分。

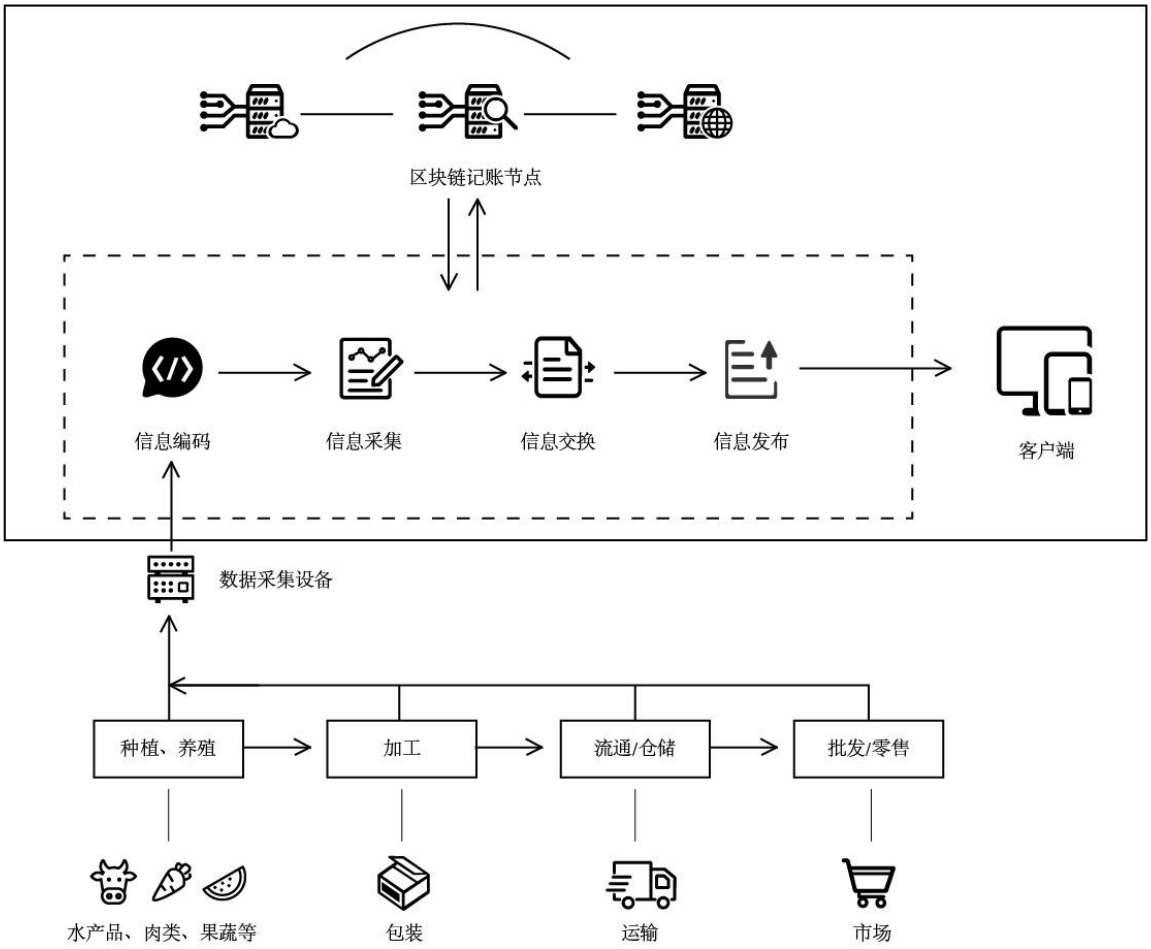

图4.1 基于区块链技术食品追溯系统组成构架图

信息编码：是为了方便信息的存储、检索和使用，在进行信息处理时赋予信息元素以代码的过程。

信息采集：是指未出版的生产在信息资源方面做准备的工作，包括对信息的收集和处理。

信息交换：是指数据在不同的信息实体之间进行交互的过程，其目标是在异构环境中实现数据的共享，从而有效地利用资源，提高整个信息系统的性能，加快信息系统之间的数据流通，实现数据的集成和共享。

信息发布：是系统面向用户终端的主要信息发送方式，是用户获取位置及相关信息的重要途径。

数据采集设备：所有的数据采集是从设备采集的。设备有多种，有些通过传感器来采集，有些设备属于智能设备，本身就是一台小型计算机，能够自己采集，不管是传感器，还是智能设备本身，采集方式一般包含两种，一种是报文方式，所谓报文就是根据你设置的采集频率进行数据传输，一般放到消息队列中。还有一种采集是以文件的方式采集，在做数据分析的时候，工业设备的数据希望是连续不断的，我们可以理解为毫秒级采集，就是设备不停的发送数据，然后形成一个文件或者多个文件。

区块链记账节点：是区块链分布式系统中的网络节点，是通过网络连接的服务器、计算机等，针对不同性质的区块链，成为节点的方式也会有所不同。

客户端：或称为用户端，是指与服务器相对应，为客户提供本地服务的程序。

5 测试方法

根据被测系统的特点，采用错误猜测方法、错误恢复、仿真模拟技术、敏感性测试、恢复性测试和稳定性测试等黑盒测试技术，通过设计系统的可靠性测试用例的方法，从成熟性、可用性、容错性、易恢复性和可靠性的依从性等方面对系统的可靠性进行质量测试，并将可靠性测试结果与系统可靠性要求比较，评价系统可靠性的符合性。

6 测试过程

基于区块链技术食品追溯系统的可靠性测试一般测试过程可包括：可靠性测试需求分析、测试用例及场景设计、可靠性测试实施、数据收集和分析等环节。

6.1 可靠性测试需求分析

可靠性需求分析是研究用户要求得到的系统可靠性的定义过程，具体来说就是了解和确认用户对系统的可靠性需求。在得出分析结果的基础上，确定可靠性测试的目标，如系统平均修复时间小于3分钟等，以便进行可靠性测试验证。

6.2 测试用例及场景设计

根据可靠性测试目标，针对复杂度高的模块设计较多的测试用例。

测试场景设计可参考如下：

1）根据相关文档进行总结分类，确定关键的失效模式；

2）确定系统在实际应用中的业务流程，根据系统需求方提供的实际业务情况确定被测系统在应用中的业务流程；

3）根据历史数据估算同时运行系统的用户数量；

6.3 可靠性测试实施

可靠性测试实施可通过测试工具本身提供的功能来实现，测试数据记录亦可通过测试工具完成。需要注意的是在测试实施的过程中，需要记录在测试场景设计中定义的持续时间以及收集生成的错误、警告和通知信息。

6.4 数据收集和分析

可靠性数据分析主要包括失效分析和可靠性分析：

1）失效分析

失效分析是根据运行结果判断系统是否失效，以及失效的程度、后果、原因等；

2）可靠性分析

可靠性分析主要是指根据失效数据，估计系统的可靠性水平，预计可能达到的水平，评价系统是否已经达到要求的可靠性水平，为管理决策提供依据。

如果系统的运行结果与需求不一致，则称系统发生失效，具体失效程度划分可参考下表 6.4.1。

失效等级	描述	说明
等级 1：关键性失效	整个系统中止或严重毁坏数据库	例如：宕机、蓝屏
等级 2：严重性失效	重要功能无法正常运行，并且没有替代的运行方式（与工作有关）	例如：程序错误
等级 3：普通失效	绝大部分功能仍然可用，次要功能受到限制或要采用替代方式（与工作有关）	例如：打印功能错误
等级 4：轻微失效	少数功能在有限的操作中受到限制	例如：界面不规范
等级 5：可忽略的失效	影响未波及到最终用户	

表6.4.1 失效等级表

7 测试规范

可靠性测试是系统可靠性保证过程中非常关键的一步，侧重点不同于一般的系统功能测试，其测试用例设计的出发点是寻找对可靠性影响较大的故障。因此，基于区块链技术食品追溯系统可靠性测试应从系统的成熟性、可用性、容错性、易恢复性、以及区块链技术和食品追溯两项技术特性的可靠性依从性等方面进行考虑。

7.1 成熟性

成熟性是指系统为避免有系统中错误而导致失效的能力，主要是指系统避免自身的错误、自身模块间的错误而导致整个系统失效，属于内部接口防范。成熟性测试一般需考虑：

1）在用户文档集陈述的限制范围内使用时，系统不应丢失数据。这种要求即使在下面的情况下也要满足：利用的容量高达规定的极限、企图利用超出规定极限的容量、由系统说明中列出的其他系统或由最终用户所造成的不正确输入、违背用户文档集中明确规定的细则。

2）系统在正常运行的情况下，连续操作（该操作不会中断系统运行）系统，查看系统是否能够连续运行24个小时（如果需求有具体要求则按需求时长测试），验证系统是否出现故障。

3）系统异常情况严重程度为微小或轻微的，或没有检测到异常。

4）在非正常运行情况下（如：有用户接口出错、应用程序自身的逻辑出错、系统或网络资源可用性引发差错），系统有继续运行的能力。

5）系统稳定，不会出现因修改系统错误而引起新的错误的情况。

7.2 可用性

可用性是对系统可使用程度的一个评价。测试时一般需考虑：

1）测试系统是否在用户需要使用时可用。可用性可以通过系统、产品或组件在总时间中处于可用状态百分比进行外部评估。因此可用性是成熟性（控制失效的频率）、容错性和易恢复性（控制每个失效发生后的宕机时间长短）的组合。

2）检测系统在使用时能够进行正常操作和访问的程度、平均无故障时间，统计测试过程中可用时间和测试过程整体时间，并计算其可用时间占比，验证是否满足需求百分比。例如，要求测试过程中可用时间/测试过程整体时间≥98%，测试过程整体时间为16小时，测试过程中可用时间为8小时，则可用时间占比为50%，不满足需求。

7.3 容错性

容错性是指在系统出现故障或违反指定接口的情况下，系统维持规定的性能级别的能力（级别指失效防护能力），这里主要指系统和外部的接口，如用户接口、硬件接口、外部系统接口等。容错性测试一般需包含以下三方面：

7.3.1 避免死机

1）在正常运行的情况下，验证系统运行过程中是否出现死机；
2）在非正常运行的情况下，验证系统运行过程中是否出现死机；
3）在硬件或网络出现问题的情况下，验证系统是否出现死机；
4）通过查看系统运行历史日志检查系统是否出现了运行情况下的死机情况。

7.3.2 避免失效

1）全面检查系统在需求规格说明或设计文档中规定的防止危险状态措施的有效性和每一个危险状态下的反应；
2）对设计中用于提高安全性的系统结构、算法、容错、冗余、中断处理等方案进行针对性测试；
3）对系统的故障模式（如数据超范围、死锁）的测试；
4）在正常运行的情况下，验证系统运行过程中是否失效。
5）在非正常运行的情况下，验证系统运行过程中是否失效。
6）在容量高达规定的极限的情况下，验证系统运行过程中是否能保证不丢失数据。
7）在超出规定极限的情况下，验证系统运行过程中是否能保证不丢失数据。
8）在其他系统或最终用户所造成的不正确输入的情况下，验证系统运行过程中是否能保证不丢失数据。
9）在违背文档集中规定细节的情况时，验证系统是否能保证不丢失数据。
10）验证系统是否能控制整个系统中止或严重毁坏数据库的故障模式，以避免关键性的失效。
11）在数据超范围或死锁的情况下，验证系统的处理情况。
12）在人为的误操作的情况下，验证系统的处理情况。
13）在多机系统出现故障需要切换时，验证系统功能和性能的连续平稳性。
14）产品描述中列出的其他程序或用户造成的错误输入时，系统不崩溃也不丢失数据。
15）系统中的设置事务的检查点、重做和还原功能，数据保存、恢复信息，避免失效措施、错误声明等应与产品说明和用户文档中的陈述一致。

7.3.3 抵御误操作和保护能力

1）在数据类型作为参数的误操作情况下，验证系统是否能抵御此误操作。
2）在输入数据序列的误操作情况下，验证系统是否能抵御此误操作。
3）在操作序列的误操作情况下，验证系统是否能抵御此误操作。
4）输入用户文档中明确规定的非法指令时，系统不崩溃也不丢失数据
5）在用户文档集陈述的限制范围之内对系统进行操作，不应丢失数据。
6）当输入违反句法条件数据时，系统应有错误或警告提示信息，并且拒绝对错误数据进行处理
7）对系统处于标准配置下其处理和保护能力的测试；
8）应进行对异常条件下系统的处理和保护能力的测试，以表明不会因可能的单个或多个输入错误而导致系统的不安全状态；
9）对重要数据的抗非法访问能力的测试；
10）必须包含边界、界外及边界结合部的测试；
11）对“0”、穿越“0”以及从两个方向趋近于“0”的输入值的测试；

12）必须包括在最坏情况配置下的最小输入和最大输入数据率的测试；

13）对安全关键操作错误测试，验证系统配置项对这些操作错误的反应；

14）验证被测试系统防止非法进入并保护系统的数据完整性能力；

15）对双工切换、多机替换的正确性和连续性的测试。

7.4 易恢复性

系统失效通常表现为死机、运行速度不匹配、计算精度不够、输出项缺损、输出项多余等，恢复性是指在失效发生后，重新建立其性能级别并恢复直接受影响数据的能力，包括原有能力恢复的程度、恢复的速度，如某路由器恢复时间、恢复期间丢包数，措施通常包括重启系统、恢复备份的数据、一键还原数据、错误操作提示、联系服务商等。易恢复性测试一般需进行：

1）采取各种人工干预的手段模拟硬件故障、网络故障、电源故障等，故意造成系统出错，并由此检查系统是否具有错误探测功能，在故障发生时能否保护正在运行的作业和系统状态；

2）在模拟生产环境下，验证系统是否能执行所有的用户操作。

3）在生产环境下，验证系统是否能执行所有的用户操作。

4）验证系统的数据保护的措施是否可用。

5）检查系统的平均恢复时间；检查系统是否具有处理程序超时或死循环故障的能力。

6）在故障发生时能否保护正在运行的作业和系统状态的测试，当可能导致系统重置的故障发生时，测试系统是否有能力保护现场。

7.4.1 可重新启动性

1）在数据库毁坏的情况下，验证系统能否重新启动。

2）在丢失多项事务的情况下，验证系统能否重新启动。

3）在丢失单项事务或临时的数据库损坏的情况下，验证系统能否重新启动。

4）系统运行失效后，应能较快重建系统。

5）探测错误功能的测试，当系统出现错误时，系统能探测错误，并给出提示信息。

6）能否切换或自动启动备用硬件的测试，从硬件角度考核其在重置发生时能否正常进行切换或重新启动。

7.4.2 易修复性

1）对系统需求规格说明或设计文档中含有恢复、重置功能或者抗毁要求的系统配置项，逐项测试在克服硬件故障或系统异常之后，系统能否正常地继续进行工作，并不对系统造成损害；

2）在测试中将系统置于极端的条件下或模拟极端条件下产生故障，然后调用恢复进程，并监测、检查和核实应用程序和数据能否得到正确的恢复。

3）在异常情况下，验证系统是否有修复能力，且修复能力是否有效。

4）验证系统是否有自身修复能力，且修改能力是否有效。

5）不会因网络异常、异常宕机、数据库实例异常停止、系统应用实例异常停止、部分交易超时异常而使系统或数据遭到破坏，且在异常情况解除后，系统可以恢复正常。

6）服务器/数据库服务器网络故障时，故障恢复后系统处理能力能否恢复正常。

7）验证系统的恢复的方法是否可用。

8）在正常运行过程中掉电，再连接上电系统和数据是否受到影响。

9）通过询问或检查系统历史日志方式检查系统因故障不能用后，每次修复所需要的时间。

10）在系统恢复后，能否从最后记录下来的无错误状态开始继续执行作业的测试，测试系统是否有能力无误的恢复现场。

11）恢复或重置功能：如重新复位、重新上电等。

7.5 可靠性的依从性

可靠性的依从性要求系统遵循与兼容性相关的标准、约定或法规以及类似规定的程度，以及需检查系统的可靠性是否遵循了所实施法规、标准和约定。