第八章
软件韧性可信量化模型

陈仪香

华东师范大学软件学院可信智能团队TrIG

2024年11月18日

TrIG 华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

第八章 软件韧性分析

8.1 软件韧性

8.2 软件组件

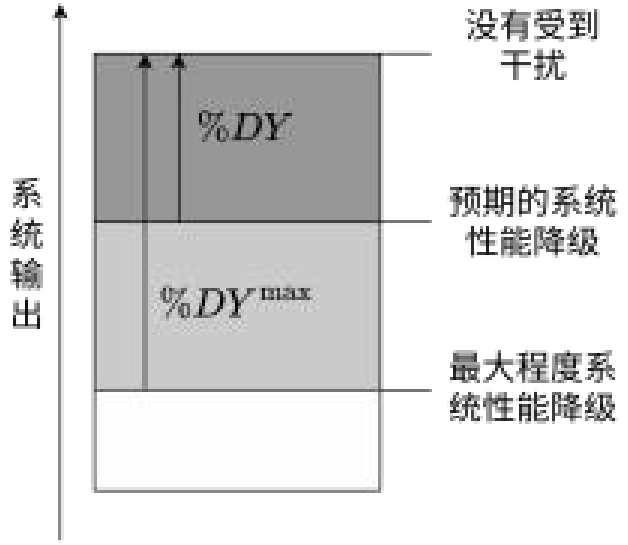
8.3 韧性可信度量

8.4 例子

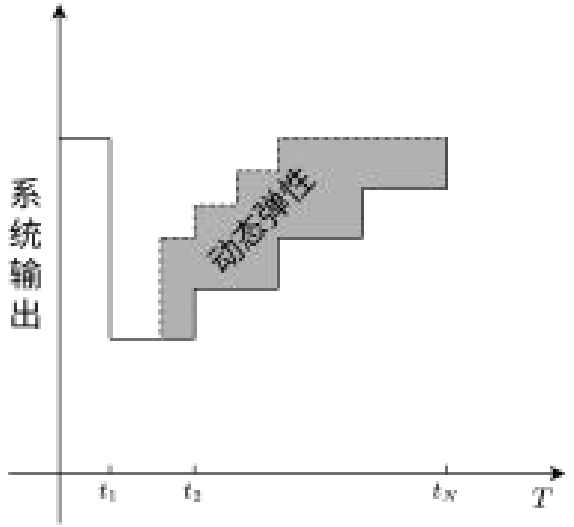
TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

8.1 软件韧性

- 软件韧性是指软件在遭受攻击或出现故障时，在不中断服务的情况下尽快恢复到正常工作状态的能力。
- 因此，通过对软件系统的韧性可 信性进行评估，使软件系统满足韧性可信性需求后再进行使用，可以减少软件系 统故障带来的经济损失和严重后果。



(a)静态模型



——：无加速条件下恢复过程；
……：加速条件下恢复过程。

(b)动态模型

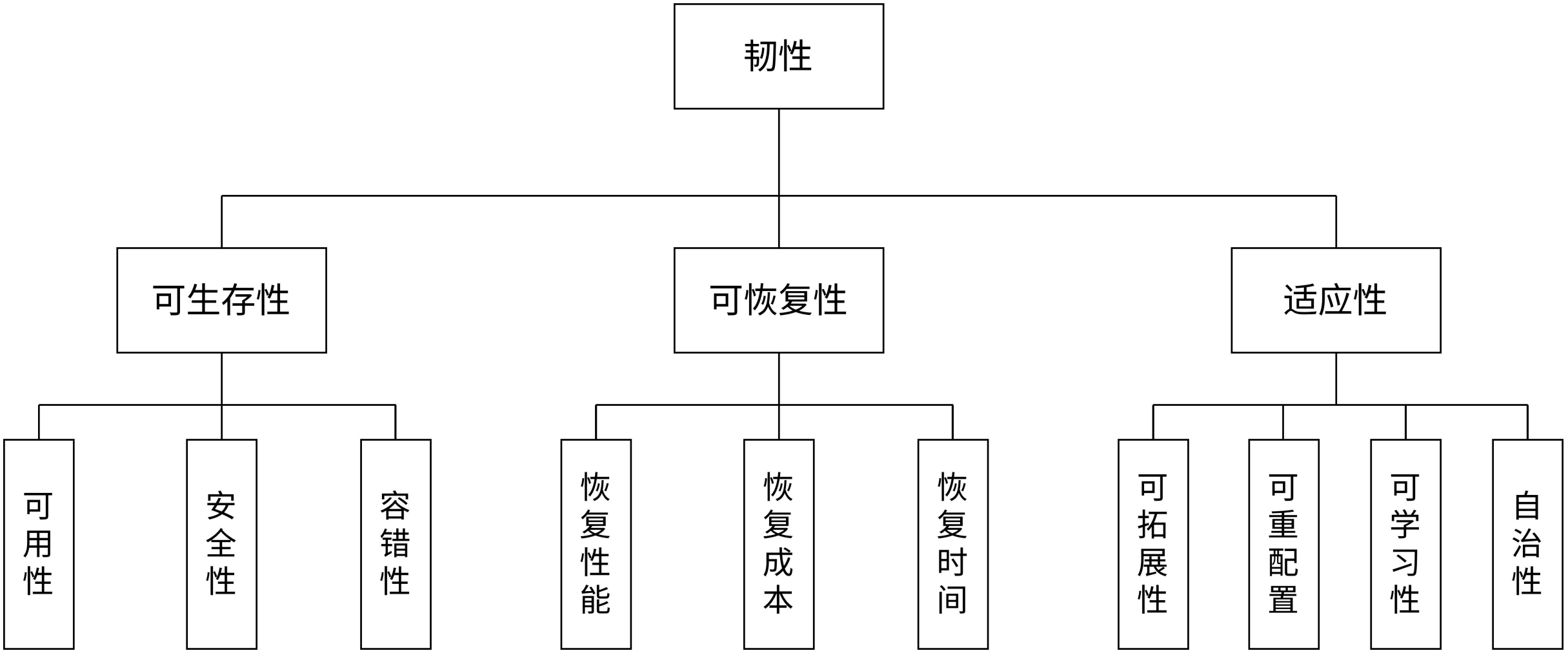
8.1 软件韧性

常见韧性定义涉及的属性

	华为	NIST	ITU	MORS	Barker	Jin-Hee	Sterbenz
可生存性	√	√	√	√	√		√
可恢复性	√	√		√	√	√	
适应性	√	√		√		√	
容错性			√			√	√
可靠性					√		
脆弱性					√		
冗余性			√				
安全性							√
可依赖性							√
性能							√

8.1 软件韧性

韧性属性分解



8.1 软件韧性

韧性属性定义

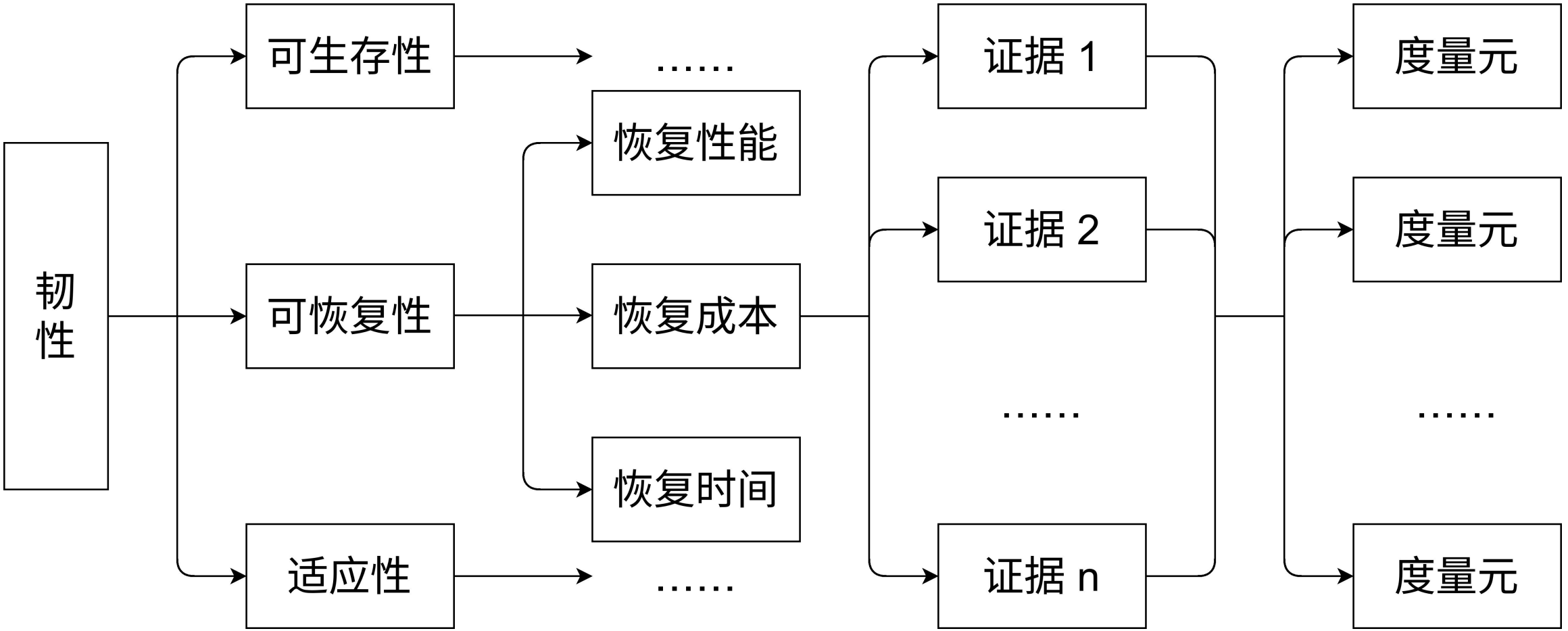
属性	定义
可生存性	系统在受到攻击、出现故障或事故的情况下，及时完成其任务的能 力。
可恢复性	系统及时恢复其服务的能力。
适应性	系统调整自身或其资源以面对改变的形势或环境来维持正常工作的 能力。

8.1 软件韧性

属性	子属性	定义
可生存性	可用性	在规定的时间段内系统能够按照要求运行的能力。
	安全性	系统保护信息和数据的能力。
	容错性	在存在包括故障、错误或攻击等威胁的情况下，系统的功能状态能 够提供适当服务的程度。
可恢复性	恢复性能	系统恢复其服务的能力。
	恢复成本	系统在恢复过程中耗费的非时间成本。
	恢复时间	系统在恢复过程中涉及到的一系列时间、延时等。
适应性	可拓展性	系统可以添加新的功能。
	可重配置	系统重新调整资源与进程关系的能力。
	可学习性	系统从动态环境中学习，做出适应性的决定来提高性能的能力。
	自治性	自主控制系统在不确定环境下长时间运行良好，并在没有外部干预 的情况下自动恢复系统故障的能力。

8.1 软件韧性

韧性可信分解架构示意图



8.1 软件韧性

韧性既有功能也有性能，因而既要对功能进行度量也要对性能进行度量

	属性	度量元
表现B	性能：实际工作时的性能， 以初始性能为基准，不同系 统或组件有不同的指标量， 例如 CPU，内存等的性能 体现为工作频率等，屏幕的 性能体现为分辨率、触摸反 馈准确率等。	性能按 0%-100% 对应到 1-10 分。
	功能：实际工作时提供的功 能，以初始功能为基准。	功能按 0%-100% 对应到 1-10 分。例外情况： 当所有关键功能无法工作时取 1分。

8.1 软件韧性

韧性既有功能也有性能，因而既要对功能进行度量也要对性能进行度量

$$表现B = \begin{cases} \mathrm{PERF}^{\alpha 1} \times \mathrm{FUNC}^{\alpha 2} \\ \alpha 1 + \alpha 2 = 1 \end{cases}$$

- PERF 为根据前表得到的性能度量值，FUNC 为根据前表得到的 功能度量值，α1 和 α2 分别为性能和功能的权重，取值范围为 [0, 1]，由专家和组 件开发人员指定。
- 表现 B 的可信度量取值区间范围为 [1, 10]。

第八章 软件韧性分析

8.1 软件韧性

8.2 软件组件

8.3 韧性可信度量

8.4 例子

TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

8.2 软件组件

- 组件可用性主要体现在功能可用性和数据可用性两个方面，不同组件具有不同的关键功能和非关键功能，数据也是类似。

- 开发人员需要在组件设计和开发 过程中确定该组件哪些功能和数据是关键的，尽可能保证关键功能和关键数据 的可用性以达到高韧性可信度。

8.2 软件组件

可用性可信证据

韧性属性	韧性子属性	韧性证据	
可生存性	可用性	组件当前关键功能与非关键功能可 用状态： 1. 组件关键功能全部可用 2. 组件关键功能部分可用 3. 组件非关键功能全部可用 4. 组件非关键功能部分可用 5. 组件全部功能不可用 （关键功能与非关键功能取决于组件 实际需求）	A: 组件满足 1 和 3 B: 组件满足 1 和 4 C: 组件满足 2 和 3 或组件满 足 2 和 4 D: 组件满足 5
可生存性	可用性	组件当前关键数据与非关键数据可 用状态： 1. 组件关键数据全部可用 2. 组件关键数据部分可用 3. 组件非关键数据全部可用 4. 组件非关键数据部分可用 5. 组件全部数据不可用 （关键数据与非关键数据取决于组件 实际需求）	A: 组件满足 1 和 3 B: 组件满足 1 和 4 C: 组件满足 2 和 3 或组件满 足 2 和 4 D: 组件满足 5

8.2 软件组件

- 组件安全性主要体现在组件能保护好数据不会泄露给未授权用户，
- 从 灾难发生前和灾难发生后两个方面考虑，以考虑到其安全措施是否周全。

8.2 软件组件

安全性可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存 性	安全性	组件数据与信息保密： 1. 组件在灾难发生前可保证数据不会被未授权用户访问得到 2. 组件在灾难发生后可保证数据不会被未授权用户访问得到 3. 组件在灾难发生前无法保证数据不会被未授权用户访问得到 4. 组件在灾难发生后无法保证数据不会被未授权用户访问得到 5. 组件在灾难发生后数据损坏，所有用户均无法访问	A: 组件满足 1 和 2 B: 组件满足 1 和 4 C: 组件满足 3 和 4 D: 组件满足 5

8.2 软件组件

- 组件容错性主要体现在组件有容错设计、容错设计合理、容错性能达标等方 面。

- 同时， 组件的故障频率（即平均故障间隔）和故障后损失的功能和性能情况 也可以反映组件的容错能力。

8.2 软件组件

容错性可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性	容错性	组件容错及其功能： 1. 对容错性有合理的测试 2. 组件具有错误透明性，能给出错误根源 3. 组件在容错阶段仍能提供服务	A: 组件当前满足其中三点 B: 组件当前满足其中两点 C: 组件当前满足其中一点 D: 组件不满足任何一点
可生存性	容错性	组件容错性能： 1. 修复前平均时间（MTTF）不高于规定值 2. 渗透阈值（Percolation threshhold）不低于规定值 3. 该组件与其他组件有合理的依赖关系	A: 组件当前满足其中三点 B: 组件当前满足其中两点 C: 组件当前满足其中一点 D: 组件不满足任何一点

8.2 软件组件

容错性可信证据（续）

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性	容错性	平均故障间隔时间（MTBF，即两次故障之间的间隔时间）	A: 平均故障间隔时间低于规定时间 B: 平均故障间隔时间在规定时间的 100%-150%之间 C: 平均故障间隔时间在规定时间的 150%-200%之间 D: 平均故障间隔时间超出规定时间的 200%
可生存性	容错性	组件损失表现：组件受到攻击前的表现 B_{origin} 与受到攻击后降低到的最低表现 $B_{disrupt}$ 之差	A: 组件损失表现不超过受到攻击前表现的 10% B: 组件损失表现占受到攻击前表现的 10%-40% C: 组件损失表现占受到攻击前表现的 40%-70% D: 组件损失表现超过受到攻击前表现的 70%

8.2 软件组件

- 组件恢复性能主要体现在其故障时对性能和功能的恢复，
- 该恢复过程又可 从两方面考虑，是否及时恢复和能恢复多少。
- 另一方面，组件的韧性可信度和可 信等级在故障时也可能发生变化，因此也需将其在韧性可信等级的恢复性能也 纳入考量。

8.2 软件组件

恢复性能可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复性能	1. 组件在规定时间内恢复了损失表现的 90% 以上 2. 组件在规定时间内恢复了损失表现的 60%-90% 3. 组件在不超过两倍规定时间内恢复了损失表现的 90% 以上 4. 组件在规定时间内恢复了损失表现的 30%-60% 5. 组件在不超过两倍规定时间内恢复了损失表现的 60%-90% 6. 组件在规定时间内恢复了损失表现的 30% 以下 具体规定时间可参考 GB/T 20988-2007 进行制定	A: 满足证据 1 B: 满足证据 2 或 3 C: 满足证据 4 或 5 D: 满足证据 6

8.2 软件组件

恢复性能可信证据（续）

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复性能	组件在系统降级后仍可恢复到降级前的状态	A: 系统降级到最低等级时组件仍能恢复到降级前状态 B: 系统降级到不低于 II 级时组件仍能恢复到降级前状态 C: 系统降级到不低于 IV 级时组件仍能恢复到降级前状态 D: 系统降级后组件无法恢复到降级前状态

8.2 软件组件

- 组件恢复成本主要体现在恢复的经济和时间成本，
- 但为避免证据及子属性 的重叠，时间成本和其他时间相关证据在恢复时间中进行考量。
- 此处仅考虑组件在恢复过程中耗费的经济成本。

8.2 软件组件

恢复成本可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复成本	组件当前恢复所耗费的开销	A: 当前耗费了原组件成本的 30%以下用于恢复 B: 当前耗费了原组件成本的 30%-60% 用于恢复 C: 当前耗费了原组件成本的 60%-90% 用于恢复 D: 当前耗费了原组件成本的 90%以上用于恢复

8.2 软件组件

- 组件恢复时间主要从三个方面，即从故障到开始恢复的延迟时间、用于恢复的时间和恢复总时间体现，
- 若这些时间低于预期设计，则说明组件在恢复时间方面可达到韧性要求。

8.2 软件组件

恢复时间可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复时间	1. 平均完全恢复时间（MTTFR）低于规定时间 2. 平均恢复时间（MTTR）低于规定时间 3. 恢复延迟低于规定时间 (MTTFR 指从组件受到攻击到完全恢复耗时，MTTR 指从发生降级到完全恢复耗时，恢复延迟指从表现降到最低到开始恢复工作耗时)	A: 组件当前满足其中三点 B: 组件当前满足其中两点 C: 组件当前满足其中一点 D: 组件不满足任何一点

8.2 软件组件

组件可拓展性则体现在组件能否通过更新添加新的功能，
从关键功能和非关键功能两方面考虑。

可拓展性可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可拓展性	组件可以添加新的功能以适应环境： 1. 组件可以添加新的关键功能 2. 组件可以添加新的非关键功能 3. 组件无法添加新的关键功能 4. 组件无法添加新的非关键功能	A: 组件满足 1 和 2 B: 组件满足 1 和 4 C: 组件满足 2 和 3 D: 组件满足 3 和 4

8.2 软件组件

组件可重配置则主要指组件能在不同情况下进行不同地配置，以更好地提 供服务等。

可重配置可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可重配置	1. 组件在新环境中很快就能改变配 置并正常提供服务 2. 组件有很多套已有配置可选择以 面对各种环境或攻击 3. 组件可以根据需求调整自己使用的资源数量	A: 组件当前满足其中三点 B: 组件当前满足其中两点 C: 组件当前满足其中一点 D: 组件不满足任何一点

8.2 软件组件

组件自治性主要体现在组件能在没有人为干预的情况下做出正确决策并高 效地提供服务。

自治性可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	自治性	组件可以自动应对变化做出决策	A: 组件可以根据自己的观测和规定以最高效率完成目标， 且无需人为干预 B: 组件可以根据自己的观测和规定完成目标，在人为干预的情况下可以使效率最大化 C: 组件可以根据自己的观测和规定完成目标，但即使在人为干预的情况下也无法使效率最大化 D: 组件无法独立完成目标， 必须人为干预

8.2 软件组件

组件可学习性主要体现在组件能否根据已有的工作信息进行学习，以在未 来更好地提供服务。

可学习性可信证据

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可学习性	1. 组件可以学习到攻击者的行为以应对未来的攻击 2. 组件可以学习以往恢复工作的开销-恢复比，以更好地开展恢复工作 3. 组件可以自动分析未知的系统漏洞及缺陷，以在未来进行修复	A: 组件当前满足其中三点 B: 组件当前满足其中两点 C: 组件当前满足其中一点 D: 组件不满足任何一点

8.2 软件组件

权重

韧性可信属性权重计算结果

属性	可生存性	可恢复性	适应性
权重	0.322	0.410	0.268

可生存性子属性权重计算结果

子属性	可用性	安全性	容错性
权重	0.353	0.324	0.323

8.2 软件组件

权重

可恢复性子属性权重计算结果

子属性	恢复性能	恢复成本	恢复时间
权重	0.369	0.294	0.337

适应性子属性权重计算结果

子属性	可拓展性	可重配置	可学习性	自治性
权重	0.251	0.237	0.261	0.251

8.2 软件组件

权重

可恢复性子属性证据权重取值表

子属性	恢复性能		恢复成本	恢复时间
证据	证据 1	证据 2	证据 1	证据 1
权重	0.6	0.4	1	1

可生存性子属性证据权重取值表

子属性	可用性			安全性	容错性			
证据	证据 1	证据 2	证据 3	证据 1	证据 1	证据 2	证据 3	证据 4
权重	0.4	0.4	0.2	1	0.3	0.2	0.2	0.3

第八章 软件韧性分析

8.1 软件韧性

8.2 软件组件

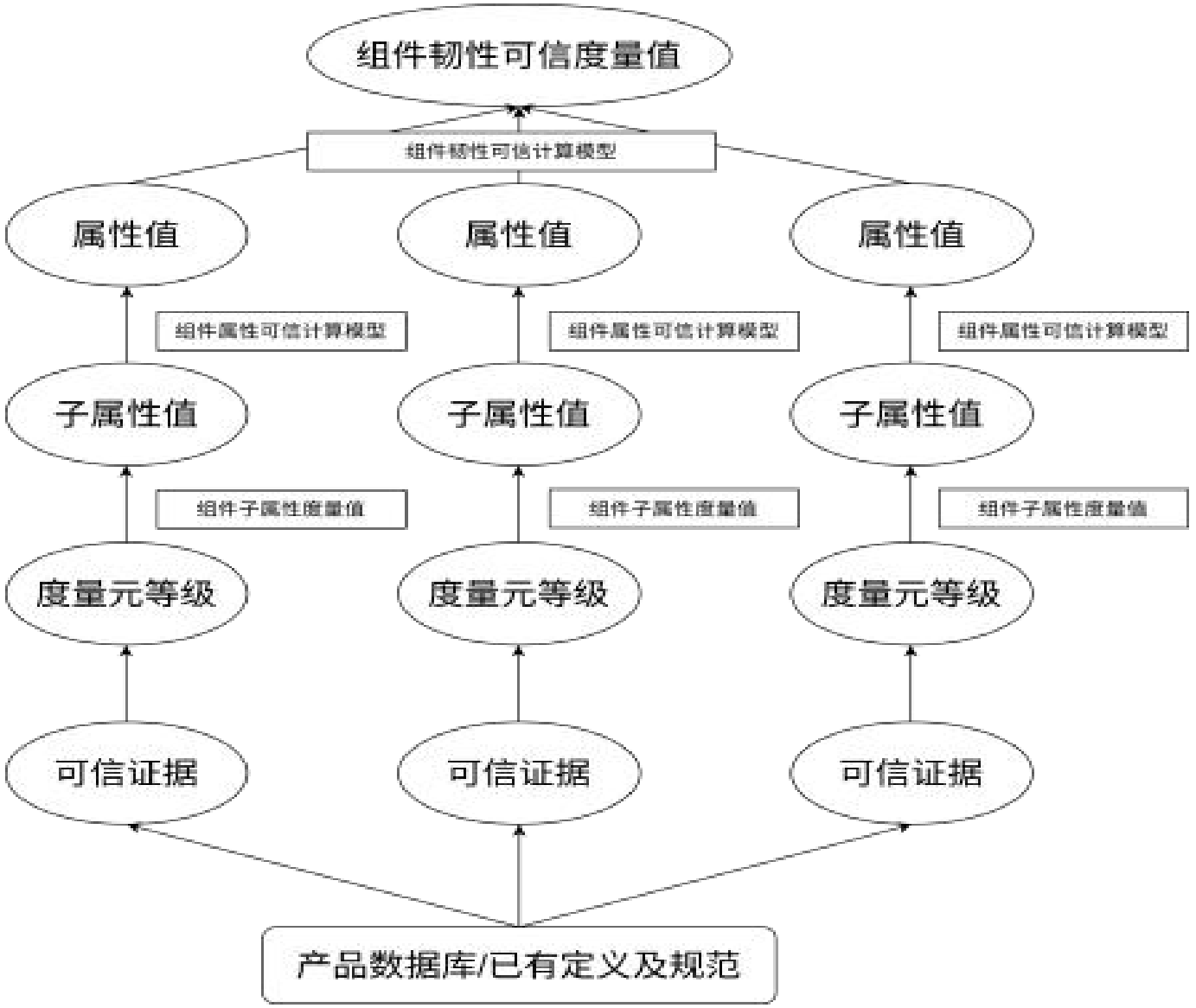
8.3 韧性可信度量

8.4 例子

TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

8.3 韧性可信度量模型

组件韧性可信性计算模型



- 属性的可信值由子属性值计算可得，
- 组件子属性的可信值由可信度量元获得，
- 可信度量元等级的选定从可信证 据中获得，
- 可信证据从产品数据库中获得。

8.3 韧性可信度量模型

度量元都是A,B,C,D形式

因而直接定义ABCD的取值就可以了：

A:10

B:7

C:4

D:1

8.3 韧性可信度量模型

子属性可信值计算：

$$子属性j可信值=\begin{cases}\prod_{i=1}^{m}\mathrm{Metric}_i^{\gamma_i}\\ \sum_{i=1}^{m}\gamma_i=1\end{cases}$$

其中 Metric_i是子属性j的第i个证据对应的度量值，而γ_i是第i个证据的权重

8.3 韧性可信度量模型

属性可信值计算：

$$属性i可信值=\begin{cases}\prod_{j=1}^{n} 子属性_j^{\beta j} \\ \sum_{j=1}^{n}\beta_j=1\end{cases}$$

其中 子属性$_j$是属性i对应的第j个子属性的度量值，而β_j是第j个子属性的权重

8.3 韧性可信度量模型

韧性可信值计算：

$$\text{韧性可信值} = \text{可生存性}^{0.322} \times \text{可恢复性}^{0.410} \times \text{适应性}^{0.268}$$

第八章 软件韧性分析

8.1 软件韧性

8.2 软件组件

8.3 韧性可信度量

8.4 例子

TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

8.4 例子

例子：选取 GitHub 上的一个游戏服务器系统为例，其包括一个网关组件， 网关组件运行在服务器上，

- 其关键功能是：(i) 负责与客户端进行网络互联，(ii) 向 客户端提供游戏数据；
- 非关键功能是：帮助用户选择网络质量最佳的服务器，确保用户以较低的网络延迟连接服务器。
- 关键数据是游戏内部数据，
- 非关键数据 是网络调试信息。

8.4 例子

例子：选取 GitHub 上的一个游戏服务器系统为例，其包括一个网关组件， 网关组件运行在服务器上，

由于网络服务器需要对全球各地的玩家提供服务，因此其可用性及可拓展性有一定额外需求，可用性及可拓展性中增加了多条证据，这些证据在该组件使用过程中或受到攻击时会随时间发生相应改变。其添加的证据如下表

8.4 例子

网关组件添加的韧性证据--可用性

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性	可用性	组件当前对各地区提供的服务质量：延迟不高于平均延迟、网速不低于均网速、丢包率不高于平均丢包率即达到质量要求	A: 组件对各地区均能提供达到质量要求的服务 B: 组件对超过 50% 地区提供 的服务能达到质量要求，少于50% 地区无法达到要求 C: 组件对少于50% 地区提供 的服务能达到质量要求，超过50% 地区无法达到要求 D: 组件对所有地区提供的服 务均无法达到质量要求

8.4 例子

网关组件添加的韧性证据-可拓展性

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可拓展性	组件在添加新功能时对已有的影响情况	A: 组件可以在用户无感知地情况下添加功能 B: 组件添加功能时不影响正运行的功能，但用户下次使用时需更新 C: 组件添加功能时不影响部分功能，但需中断提供其他功能，正在使用被中断功的用户会受到影响。所有用户在下次使用组件时均要更新 D: 组件添加功能时需中断提供所有功能，用户下次使用 时需更新

8.4 例子

- 作为对比，后台管理组件也是该游戏系统的一部分，其主要功能是：

  (i) 使 管理员可以管理游戏用户，对其进行封禁等，

  (ii) 使管理员可以实时改变一部分 游戏数据；

- 非关键功能是自动识别使用外挂的用户并提醒管理员。其关键数据是

  (i) 游戏信息、数值配置数据，

  (ii) 用户账户信息数据；

- 非关键数据是日志数据。其添加的一部分证据如下表 所示：

8.4 例子

后台管理组件添加的韧性证据--可学习性

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可学习性	组件可以学习并识别外挂玩家的行为	A: 组件识别外挂玩家准确率高于 90% B: 组件识别外挂玩家准确率在60%-90%之间 C: 组件识别外挂玩家准确率在30%-60%之间 D: 组件识别外挂玩家准确率低于 30%

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（1）	可用性（1）	组件当前关键功能与非关键功能可 用状态： 1. 组件关键功能全部可用 3. 组件非关键功能全部可用 该组件关键功能是：(i) 负责与客户 端进行网络互联，(ii) 向客户端提供 游戏数据；非关键功能是：帮助用户 选择网络质量最佳的服务器，确保 用户以较低的网络延迟连接服务器。	A: 组件满足 1 和 3

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（2）	可用性（2）	组件当前关键数据与非关键数据可 用状态： 1. 组件关键数据全部可用 3. 组件非关键数据全部可用 该组件关键数据是游戏内部数据， 非关键数据是网络调试信息。	A: 组件满足1 和 3

8.3 韧性可信度量

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（3）	可用性（3）	组件当前对各地区提供的服务质量：延迟不高于平均延迟、网速不低于平均网速、丢包率不高于平均丢包率即达到质量要求	A: 组件对各地区均能提供达 到质量要求的服务

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（4）	安全性	组件数据与信息保密： 1. 组件在灾难发生前可保证数据不 会被未授权用户访问得到 2. 组件在灾难发生后可保证数据不 会被未授权用户访问得到	A: 组件满足 1 和 2

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（5）	容错性（1）	组件容错及其功能： 1. 对容错性有合理的测试 2. 组件具有错误透明性，能给出错误根源 3. 组件在容错阶段仍能提供服务	B: 组件当前满足其中两点 （满足 1 和 3）

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（6）	容错性（2）	组件容错性能： 1. 修复前平均时间（MTTF）不高于规定值 2. 渗透阈值（Percolation threshhold）不低于规定值 3. 该组件与其他组件有合理的依赖 关系	A: 组件当前满足其中三点

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（7）	容错性（3）	平均故障间隔时间（MTBF）（MTBF 即两次故障之间的间隔时间）	A: 平均故障间隔时间低于规 定时间

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（8）	容错性（4）	组件损失表现：组件受到攻击前的表现 B_{origin} 与受到攻击后降低到的最低表现 $B_{disrupt}$ 之差	A: 组件损失表现不超过受到 攻击前表现的 10%

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性（1）	恢复性能（1）	1. 组件在规定时间内恢复了损失表 现的 90% 以上 2. 组件在规定时间内恢复了损失表 现的 60%-90%	A: 满足证据1

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性（2）	恢复性能（2）	组件在系统降级后仍可恢复到降级前的状态	A: 系统降级到最低等级时组件仍能恢复到降级前状态

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性（3）	恢复成 本	组件当前恢复所耗费的开销	A: 当前耗费了原组件成本的30% 以下用于恢复

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性（4）	恢复时间	1. 平均完全恢复时间（MTTFR）低于规定时间 2. 平均恢复时间（MTTR）低于规定时间 3. 恢复延迟低于规定时间 (MTTFR 指从组件受到攻击到完全 恢复耗时，MTTR 指从发生降级到 完全恢复耗时，恢复延迟指从表现 降到最低到开始恢复工作耗时)	A: 组件当前满足其中三点

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性（1）	可拓展性（1）	组件可以添加新的功能以适应环境： 1. 组件可以添加新的关键功能 2. 组件可以添加新的非关键功能 3. 组件无法添加新的关键功能 4. 组件无法添加新的非关键功能	A: 组件满足 1 和 2

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性（2）	可拓展性（2）	组件在添加新功能时对已有的影响情况	D: 组件添加功能时需中断提供所有功能，用户下次使用时需更新

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性（3）	可重配 置	1. 组件在新环境中很快就能改变配置并正常提供服务 2. 组件有很多套已有配置可选择以面对各种环境或攻击 3. 组件可以根据需求调整自己使用的资源数量	B: 组件当前满足其中两点（满足 2 和 3）

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性（4）	可学习性	1. 组件可以学习到攻击者的行为以应对未来的攻击 2. 组件可以学习以往恢复工作的开销-恢复比，以更好地开展恢复工作 3. 组件可以自动分析未知的系统漏洞及缺陷，以在未来进行修复	C: 组件当前满足其中一点（满足 1）

8.4 例子

游戏系统网关组件的全部韧性证据参见附录，根据其实际情况可知网关组 件初始满足度量元如表

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性（5）	自治性	组件可以自动应对变化做出决策	B: 组件可以根据自己的观测 和规定完成目标，在人为干 预的情况下可以使效率最大 化

8.4 例子

可生存性子属性证据权重取值表

子属性	可用性			安全性	容错性			
证据	证据1	证据2	证据3	证据1	证据1	证据2	证据3	证据4
权重	0.4	0.4	0.2	1	0.3	0.2	0.2	0.3

$$
\begin{aligned}
SURV &= (10^{0.4} \times 10^{0.4} \times 10^{0.2})^{0.353} \times 10^{0.324} \\
&\times (7^{0.3} \times 10^{0.2} \times 10^{0.2} \times 10^{0.3})^{0.323} \\
&= 10^{0.353} \times 10^{0.324} \times 8.985^{0.323} \\
&= 9.660
\end{aligned}
$$

8.4 例子

可恢复性子属性证据权重取值表

子属性	恢复性能		恢复成本	恢复时间
证据	证据 1	证据 2	证据 1	证据 1
权重	0.6	0.4	1	1

$$REC = (10^{0.6} \times 10^{0.4})^{0.369} \times 10^{0.294} \times 10^{0.337} = 10$$

8.4 例子

适应性子属性证据权重取值表

子属性	可拓展性		可重配置	可学习性	自治性
证据	证据1	证据2	证据 1	证据 1	证据 1
权重	0.6	0.4	1	1	1

$$\mathrm{ADAPT} = (10^{0.6} \times 1^{0.4})^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251}$$

$$= 3.981^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251}$$

$$= 5.250$$

8.4 例子

游戏系统网关组件的韧性可信性度量值

$$CP_{gateway} = 9.660^{0.322} \times 10^{0.410} \times 5.250^{0.268} = 8.321$$

8.4 例子

该游戏由于主要在美国运营，因此在美国部署服务器较多，后期在中国宣传 较多，在一次版本更新后，中国玩家数量增加，中国地区服务器压力剧增，无法 提供与美国地区同等质量的服务，玩家将明显感到延迟增加，部分玩家甚至无法 连上服务器，对应部分度量元改变如表

8.4 例子

网关组件故障时满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（1）	可用性（1）	组件当前关键功能与非关键功能可用状态： 1. 组件关键功能全部可用 2. 组件关键功能部分可用 3. 组件非关键功能全部可用 4. 组件非关键功能部分可用 5. 组件全部功能不可用 该组件关键功能是：(i) 负责与客户 端进行网络互联，(ii) 向客户端提供游戏数据；非关键功能是：帮助用户选择网络质量最佳的服务器，确保用户以较低的网络延迟连接服务器。	C: 组件满足 2 和 3 或组件满 足 2 和 4（版本更新前为 A，此时网关组件无法向部分用户提供与客户端进行网络互联功能，无法帮部分用户选择网络质量最佳的服务器）

8.4 例子

网关组件故障时满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（2）	可用性（2）	组件当前对各地区提供的服务质量： 延迟不高于平均延迟、网速不低于 平均网速、丢包率不高于平均丢包 率即达到质量要求	B: 组件对超过 50%地区提供 的服务能达到质量要求，少 于50% 地区无法达到要求（版本更新前为 A，此时网关 组件在中国以外地区仍能提 供达到质量要求的服务）

8.4 例子

网关组件故障时满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性（3）	容错性	组件损失表现：组件受到攻击前的 表现 B_{origin} 与受到攻击后降低到的 最低表现 $B_{disrupt}$ 之差	B: 组件损失表现占受到攻击前表现的 10%-40%（版本更 新前为 A，此时受到影响用 户约占总用户 30%，且部分用户受到影响较为轻微，损 失表现约为 25%）

8.4 例子

网关组件故障时满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复性 能	1. 组件在规定时间内恢复了损失表 现的 90% 以上 2. 组件在规定时间内恢复了损失表 现的 60%-90% 3. 组件在不超过两倍规定时间内恢 复了损失表现的 90% 以上 4. 组件在规定时间内恢复了损失表 现的 30%-60% 5. 组件在不超过两倍规定时间内恢 复了损失表现的 60%-90% 6. 组件在规定时间内恢复了损失表现的30% 以下	D: 满足证据 6（版本更新前为 A）

8.4 例子

第二次计算出网关组件初始可生存性（SURV）、可恢复性（REC）、适 应性（ADAPT）及韧性可信度量值分别为

$$\begin{aligned}
\mathrm{SURV} &= (4^{0.4} \times 10^{0.4} \times 7^{0.2})^{0.353} \times 10^{0.324} \\
&\times (7^{0.3} \times 10^{0.2} \times 10^{0.2} \times 7^{0.3})^{0.323} \\
&= 6.454^{0.353} \times 10^{0.324} \times 8.073^{0.323} \\
&= 7.996
\end{aligned}$$

$$\mathrm{REC} = (1^{0.6} \times 10^{0.4})^{0.369} \times 10^{0.294} \times 10^{0.337} = 6.006$$

$$\begin{aligned}
\mathrm{ADAPT} &= (10^{0.6} \times 1^{0.4})^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251} \\
&= 3.981^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251} \\
&= 5.250
\end{aligned}$$

8.4 例子

第二次计算出网关组件初始可生存性（SURV）、可恢复性（REC）、适 应性（ADAPT）及韧性可信度量值分别为

$$CP_{gateway} = 7.996^{0.322} \times 6.006^{0.410} \times 5.250^{0.268} = 6.353$$

前一次可信性值计算

$$CP_{gateway} = 9.660^{0.322} \times 10^{0.410} \times 5.250^{0.268} = 8.321$$

8.4 例子

- 游戏开发人员及时增加了服务器以解决了该问题，
- 同时调整了更新策略，
  - 让 一部分正在游玩的玩家继续游玩，未游玩的玩家才需进行更新，以免导致服务器 突发压力过大。
- 由于开发成本属于正常运营成本的一部分，
- 因此并未计算到恢复 成本中，对应度量元改变如表

8.4 例子

网关组件修复后满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存性	可用性（1）	组件当前关键功能与非关键功能可 用状态： 1. 组件关键功能全部可用 2. 组件关键功能部分可用 3. 组件非关键功能全部可用 4. 组件非关键功能部分可用 5. 组件全部功能不可用 该组件关键功能是：(i) 负责与客户端进行网络互联，(ii) 向客户端提供游戏数据；非关键功能是：帮助用户选择网络质量最佳的服务器，确保用户以较低的网络延迟连接服务器。	A: 组件满足 1 和 3（修复前为 C）

8.4 例子

网关组件修复后满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存 性	可用性（2）	组件当前对各地区提供的服务质量：延迟不高于平均延迟、网速不低于平均网速、丢包率不高于平均丢包率即达到质量要求	A: 组件对各地区均能提供达 到质量要求的服务（修复前为 B）

8.4 例子

网关组件修复后满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可生存 性	容错性	组件损失表现：组件受到攻击前的表现 B_{origin} 与受到攻击后降低到的最低表现 $B_{disrupt}$ 之差	A: 组件损失表现不超过受到攻击前表现的40%（修复前为 B）

8.4 例子

网关组件修复后满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
可恢复性	恢复性能	1. 组件在规定时间内恢复了损失表现的 90% 以上 2. 组件在规定时间内恢复了损失表现的 60%-90% 3. 组件在不超过两倍规定时间内恢复了损失表现的 90% 以上 4. 组件在规定时间内恢复了损失表现的 30%-60% 5. 组件在不超过两倍规定时间内恢复了损失表现的 60%-90% 6. 组件在规定时间内恢复了损失表现的 30% 以下	A: 满足证据 1（修复前为 D）

8.4 例子

网关组件修复后满足的度量元

韧性属性	韧性子属性	韧性证据	韧性度量元
适应性	可拓展 性	组件在添加新功能时对已有的影响情况	B: 组件添加功能时不影响正运行的功能，但用户下次使用时需更新（修复前为 D）

8.4 例子

第三次计算出网关组件初始可生存性（SURV）、可恢复性（REC）、适 应性（ADAPT）及韧性可信度量值分别为

$$\begin{aligned}
\mathrm{SURV} &= (10^{0.4} \times 10^{0.4} \times 10^{0.2})^{0.353} \times 10^{0.324} \\
&\times (7^{0.3} \times 10^{0.2} \times 10^{0.2} \times 10^{0.3})^{0.323} \\
&= 10^{0.353} \times 10^{0.324} \times 8.985^{0.323} \\
&= 9.660
\end{aligned}$$

$$\mathrm{REC} = (10^{0.6} \times 10^{0.4})^{0.369} \times 10^{0.294} \times 10^{0.337} = 10$$

$$\begin{aligned}
\mathrm{ADAPT} &= (10^{0.6} \times 7^{0.4})^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251} \\
&= 8.670^{0.251} \times 7^{0.237} \times 4^{0.261} \times 7^{0.251} \\
&= 6.383
\end{aligned}$$

8.4 例子

第三次计算韧性可信度量值（修复）

$$CP_{gateway} = 9.660^{0.322} \times 10^{0.410} \times 6.383^{0.268} = 8.768$$

第二次可信性值计算（多布置）

$$CP_{gateway} = 7.996^{0.322} \times 6.006^{0.410} \times 5.250^{0.268} = 6.353$$

第一次可信性值计算（初始）

$$CP_{gateway} = 9.660^{0.322} \times 10^{0.410} \times 5.250^{0.268} = 8.321$$

8.4 例子

- 组件正常工作时韧性可信度量值为 8.321，
- 压力较大发生故障后，韧性可信 度量值降低到 6.353，
- 程序员进行修复和升级后，韧性可信度量值恢复并提升到 了 8.768。
- 从该案例的计算结果可以看出，在灾难后及时对系统或组件修复可保 证韧性可信值不低于灾难前地韧性可信值，
- 同时，通过分析灾难形成的原因，对 系统或组件进行提升，可进一步提高韧性可信值。

作业

- 软件韧性的定义是什么，包含了哪三个可信属性和十个子属性？其定义是什么？
- 韧性既有功能也有性能，因而既要对功能进行度量也要对性能进行度量，一般情况下，如何对其性能进行分级度量的？并以可生存性的容错性度量元定义加以说明
- 例子中的游戏系统网关组件，在客户大量增加后，性能发生了哪些变化？运行商是如何维护和解决的？解决效果如何？