学 科 通 融 · 之二十三 · 通信协议工程 × 宪法学 × 演化生物学

静止点

论决定一个东西能不能被重写的，不是它被改了多少，而是有没有一个所有使用者同时不在使用它的时刻

王德生 + Claude · 约 1.0 万字 · 三种阅读方式 · 判据已在公开数据上实跑 · 发表于2026年8月1日

一件长期在用的东西积累了太多修补之后是否会被整个重做，通常被当作改动量的函数：补丁多到一定程度，重写就成为必然。本文论证这个函数不存在。**重写不是改动量的极限，它需要一个「所有使用者同时不在使用它」的时刻——本文称之为静止点。** 没有静止点的东西，无论投入多大都只能继续打补丁；有静止点的东西，即使几乎没有改动也可以被整篇替换。本文在互联网标准的全量公开数据上实际执行了这一判据：**9,819 份 RFC 中，被打过补丁的 1,237 份里有 85.1% 从未被整篇取代；而补丁次数与最终被取代的比例基本不相关（1 次 14.4%、2–3 次 15.1%、4–7 次 15.5%）。** 被反复整篇重写次数最多的文档全部是状态快照与名录（最高 33 代），而补丁最多却从未被重写的全部是运行中的协议（域名系统的核心规范被打了 29 次补丁、跨 36 年，一次也没有被重写）。本文给出二维辨别表并指认最危险的一格：没有静止点，却按有静止点来管理——宣布取代而实际共存。本文在十个领域兑现，其中两个反过来迫使命题收窄：变体遗传密码的存在说明静止点不必是全局的，局部静止点就够；而宪法数据说明推迟重写在有些系统里正是目的，因而本文不含价值判断。

一、导论：三个不该同时为真的命题

三场争论，分属通信协议工程、宪法学与演化生物学，各自都有厚实的支撑，而它们不能同时为真。

第一场在通信协议工程。 互联网标准体系有两种明确区分的关系：一份文档可以**更新**另一份（打补丁，原文继续有效），也可以**取代**另一份（整篇作废）。围绕这两者的争论持续了很多年：一派认为反复打补丁使规范的语义变得难以确定——要知道某个协议现在到底怎么规定的，得同时读七八份文档并自行判断哪条盖过哪条，因而主张定期整合重写；另一派认为整篇重写会打断部署、制造不兼容，主张继续增量修补。这不是空谈，它有一份完整公开的记录可以查：每一份文档取代了谁、被谁取代、更新了谁、被谁更新，连同日期。争点是：**修补积累到一定程度，是不是必然导向重写？**

第二场在宪法学。 一项覆盖 1789 年至 2005 年、超过两百个国家、九百余部成文宪法的系统研究得出一个与常识相反的发现：宪法的寿命中位数只有十九年；而设计上的三项特征——**易于修正、包容性、具体性**——显著降低"宪法死亡"的风险，最不包容的一批平均只活十四年，最包容的一批平均活

六十九年。按这条结论，**修正能力是延寿手段**：一部能被不断打补丁的宪法更不容易被整个换掉。争点是：**打补丁是走向替换的中途站，还是避免替换的办法？**

第三场在演化生物学。 遗传密码被称为"冻结的意外"：它之所以几乎不变，不是因为它最优，而是因为任何改动都会同时影响所有蛋白质，改一个字母全盘皆错，因而在有相当规模的生命体系里再也改不动了。这条经典判断此后受到两个方向的挑战：一方指出密码在抗错性上近乎最优，因而不只是意外；另一方指出确实存在变体密码——线粒体与某些纤毛虫用着与"通用"密码不同的对应表，说明它并非完全冻结。争点是：**不能重写，是因为改动的波及面太大，还是因为别的什么？**

三场争论互不相干，却在同一个问题上互相顶撞：**一个在用的东西，什么时候会被整个重做？**

第一场里主张重写的一方说：**补丁积累到某个程度就该重写。** 第二场的数据说：**补丁能力越强越不会被替换。** 第三场说：**根本不能重写，因为波及面太大。**

三条不能同真。若补丁积累导向重写，则宪法学的发现该反过来——修正越频繁的宪法越容易被整个换掉，而数据不支持。若补丁能力普遍地防止替换，则演化那一支的"波及面太大"就是多余的解释：不改就好了，何必说不能改。若波及面决定能不能重写，则那些波及面极大的协议——比如域名系统——应当与遗传密码一样从不重写，而互联网标准里确实有被整篇重写过三十多次的文档，波及面并不小。

本文认为，僵局来自一个三方共有、却从未被单独说出的前提：**能不能重写，是被"改动"这个量决定的**——改动多则该重写（第一场）、改动能力强则不必重写（第二场）、改动波及面大则不能重写（第三场）。三家争的是这个量往哪个方向起作用，没有人问它是不是那个起作用的东西。

本文撤销这个前提，提出：

> **重写不是改动量的极限。它需要一个「所有使用者同时不在使用它」的时刻——本文称之为静止点。没有静止点的东西，无论投入多大都只能继续打补丁；有静止点的东西，即使几乎没有改动，也可以被整篇替换。**

三个案例在这一点上是同一个形状。一份状态快照——某年某月的协议清单、名录、年报——发布那一刻它就完成了，读者读完就放下，**任何时刻都没有人"在运行"它**：它每时每刻都处在静止点，所以可以被整篇重发无数次。一个运行中的协议，任何时刻都有人正在按它收发数据包，**静止点从不出现**，所以它只能被打补丁。一部宪法处在中间：正常时期没有静止点，而制宪会议、革命、政权更替**人为地制造出一个主权空窗**，替换只在那个窗口里发生。遗传密码则是极端：生命不停机，静止点在原理上不出现。

第二节梳理三条既有路径并指认它们停在哪里；第三节定位冲突；第四节界定静止点与它的三个构成条件；第五节升成二维辨别表；第六节在十个领域兑现，其中两个迫使本文收窄；**第七节给出判据，**

并报告本文在互联网标准全量数据上实际跑出的结果；第八节交代与最近几种说法的分界；第九节说明三家为何各自到不了；第十节给出推论、边界与反噬预言；第十一节处理本判断对自身的适用。

二、文献综述

2.1 补丁积累论

第一条路径把重写理解为修补的自然终点：改动逐次累积，可读性与一致性逐次下降，直到重写的收益超过成本。它在软件工程里有大量表述——技术债的比喻正建立在"债会累积、总要还"这个直觉上——在标准工作里也是主张定期整合的一方的依据。

它的长处是它给了一个可操作的判断：数一数改动。它的盲区是本文第七节的实测结果所指的地方：**在一份完整的公开记录上，改动次数与最终是否被整篇重写基本不相关**。补丁不是通往重写的路。

2.2 灵活性延寿论

第二条路径把修正能力理解为存续的保障：能被局部调整的制度不必被整体推翻。宪法研究给出了迄今最扎实的经验支持——易于修正、包容、具体三项设计特征显著延长寿命，最不包容与最包容的两组平均寿命相差近五倍。

它停在哪里？停在**它没有区分"没被替换"的两种情形**：一种是它足够好因而不必替换，另一种是它根本没有可以被替换的时刻。前者是成就，后者是处境。宪法研究处理的是一类**存在替换机制**的对象（政变、革命、制宪会议都是现成的替换途径），因而它可以合理地把"没被替换"读成设计的功劳。把同一条结论搬到没有替换途径的对象上，就会把处境误读成成就。

2.3 波及面论

第三条路径把不可重写归因于改动的波及面：一处改动牵动全体，因而任何改动都不可行。冻结的意外是它最著名的表述。

它停在哪里？停在**波及面是一个连续量，而"能不能重写"看起来是个二值的事**。波及面大只说明改动贵，不说明改动不可能；而生命体系里确实出现了变体密码，说明在某些条件下它被改动了。那些条件是什么，波及面这个量本身答不出。本文第六节将论证，那些条件正是**局部静止点**。

2.4 三条路径共享的东西

三条路径分别把"改动量多"、"改动能力强"、"改动波及面大"当作自变量。它们共享的是**自变量的类型**：都是关于改动的量。本文提出的自变量不是量，是**一个时刻存不存在**。

三、冲突定位

冲突一（补丁导向重写 vs 补丁替代重写）。 若补丁积累导向重写，则补丁次数与被取代率正相关；若补丁替代重写，则打补丁最多的对象反而最不会被取代。两条不能同真，且这一条是**可以直接查的**——第七节给出结果。

冲突二（灵活性延寿 vs 波及面锁死）。 若灵活性是设计者可选的延寿手段，则不可重写是一种被选择的状态；若波及面锁死，则不可重写是被强加的处境。两条不能同真：前者预测在同类对象内部，修正机制更完备者更长寿；后者预测长寿只与耦合度相关而与机制无关。

冲突三（波及面决定 vs 有反例）。 若波及面决定可不可改，则波及面最大的遗传密码应当绝无变体；而变体密码确实存在。两条不能同真。

三条冲突有一个共同点：**它们全都发生在同一个位置上——"换掉它"这个动作要在什么时候执行。**

第一场里，取代一份标准文档意味着从某个时点起以新文档为准，而**部署在外的实现不会在那个时点同时切换**。第二场里，替换一部宪法必须发生在一个特定的时刻——旧秩序已停、新秩序未立的那个空窗。第三场里，替换遗传密码需要所有正在合成蛋白质的细胞同时按新表来读，而**生命没有这样一个时刻**。

三方都在争"什么决定能不能换"，而三方共享的前提是：**换掉它是一件可以在任何时候执行的事，问题只在划不划算。** 一旦承认它需要一个特定的时刻，而那个时刻在不同对象上或者天然存在、或者可以被制造、或者根本不出现，三条冲突就不再需要判谁对。

四、静止点

4.1 定义与三个构成条件

> **静止点**：一个时刻，在这个时刻上，没有任何使用者正在依赖这个东西的当前版本完成一件不能中断的事。

它不是"没人用"，而是"没人正在用中途"。三个构成条件：

其一，使用是可分段的。 使用者对它的每一次使用有明确的起点与终点，而两次使用之间没有必须被延续的状态。读一份名录是可分段的：读完就放下。跑一次域名解析也是可分段的，但**整个系统里总有人正在跑**，于是全局静止点不出现——这引出第二条。

其二，全体的段间空隙要对齐。 单个使用者的空隙不够，必须所有使用者的空隙同时出现。这是静止点稀缺的真正原因：**使用者越多、越分散、越自主，对齐的概率越低**，而这与改动量、耦合度、技术水平都无关。

其三，切换本身可以在空隙内完成。 即使空隙出现，若切换过程比空隙长，静止点也等于不存在。

三条合起来解释了一个常见的错觉：人们以为"停机窗口"是运维安排，其实它是**人为制造静止点**——把第二条从"碰运气对齐"改成"强制对齐"。

4.2 为什么补丁不会通向重写

若静止点是重写的必要条件，那么补丁与重写就不在同一条路上：补丁是**在没有静止点的条件下仍能施加改动的唯一形式**，它的存在恰恰证明静止点不存在。于是：

> **补丁不是重写的前奏，是重写的替代。补丁越多，往往只说明这个东西越是一刻都停不下来。**

这条推论有一个直接的可查形式——补丁次数与被整篇取代的比例应当**不相关**——而它正是第七节实际跑过的那一条。

4.3 三种处境

天然有静止点。 状态快照、名录、年报、课程大纲、版本发布说明。它们发布即完成，读者读完就放下，任何时刻都没有人"在运行"它。这类东西被整篇重写是常态，重写次数多到令人吃惊也不奇怪——它不承担任何持续运转的东西。

静止点可被制造。 停机窗口、召回、换型、制宪会议、休渔期、学期之间。制造静止点要付一次集中的代价（停产、空窗、停课），并且这个代价是**政治性的**：谁有权宣布停下来，本身就是权力问题。宪法的替换之所以总伴随非常规政治，原因在此。

静止点不出现。 运行中的核心协议、遗传密码、货币、语言、正在服役的电网。对这类对象，"我们要重写它"这句话没有可执行的内容；能做的只有为补丁而设计——留扩展点、留版本协商、让新旧可以在同一条线上共存。

五、二维辨别表

第一条轴：**静止点的可得性**（天然有／可制造／不出现）。为便于成表，把它压成二值：**有可用的静止点／没有**。第二条轴：**这件事是否被承认**。

	承认自己的静止点状况	不承认
有可用静止点	**甲格　可重写**。名录与快照每次重发；有停机窗口的批处理系统按期换代	**丙格　白白浪费**。明明有窗口却按"改不动"来管，于是永远在打补丁，技术债的说法在这一格里是对的
没有静止点	**乙格　为补丁而设计**。域名系统、边界网关协议、遗传密码：留扩展点、留版本协商，承认新旧长期共存	**丁格（本文的靶）　宣布取代而实际共存**。文档层宣布旧版作废，部署层照旧运行；于是"当前版本"这个说法逐渐失去所指

丁格是本文要指认的那一格，理由有三。

其一，**它在文件上完全合规**：旧版被正式标注为已取代，新版被正式发布，所有流程都走完了。

其二，**它的代价落在没有发言权的一侧**：宣布取代的是标准制定方，承担共存的是实现者与运维者。前者的账在宣布那一刻就结清了，后者的账要付很多年。

其三，**它有一个可查的签名**：一份已被宣布取代的文档，此后仍然需要被正式打补丁。这件事在逻辑上是荒谬的——一份作废的文档为什么还要修？——而它确实发生，第七节给出了数量与最长时距。

六、逐领域兑现

（一）通信协议标准。 见第七节，本文唯一实际跑过的一处。

（二）宪法。 替换发生在人为制造的空窗里，而不是修正积累到某个阈值时。可检验推论：修正频率与被整体替换的概率应当**不相关或弱正相关**，而不是负相关——尽管"可修正性延长寿命"这条结论已被数据支持，本文认为其机制不是"修正减少了替换需求"，而是"能被修正的宪法往往也是那些不容易出现主权空窗的宪法"。这两条解释可以用同一份公开数据分开，见第七节第三小节。

（三）遗传密码。 无全局静止点。变体密码出现在线粒体与某些纤毛虫——**小基因组、隔离种群、密码子使用高度偏斜**，也就是局部静止点的三个条件同时满足的地方。

（四）铁路信号与电网。 无全局静止点，只能分段切换；大规模换代靠**人为制造的局部静止点**（区间封锁、分区停电），而不是靠"旧系统已经太乱了"。

（五）会计准则与税法。 有静止点（会计年度、纳税年度是天然的段间空隙），因而整体换版是可行的，也确实周期性发生。

（六）教科书与课程。 学期之间是天然静止点，所以教材每几年整本换一次，而这与内容变化量关系不大。

（七）语言。 无静止点（没有一刻全体使用者同时不说话），因而语言只被补丁，从不被重写；历史上少数"重写"的尝试（文字改革、人造语言）都必须先制造一个人为空窗——学制、政令、封闭社群。

（八）城市道路网。 无全局静止点，改造只能分段封闭；整城重规划只发生在战争、火灾或政权更替之后——即静止点被外力制造出来的时候。

（九）变体遗传密码——第一条让步。

这一领域迫使本文收窄，而且是从最不利的方向。

按本文的严格表述，遗传密码没有静止点，所以不可重写；而变体密码确实存在。若坚持严格表述，本文就有反例。

正确的收窄是：**静止点不必是全局的。** 一个可以被隔离出来的子群，只要它内部的使用者能够同时进入空隙，就构成一个**局部静止点**。线粒体基因组极小、与细胞核基因组相对隔离、只编码很少的蛋白质——这正是一个可隔离子群。于是：

> 重写需要的不是全体的静止点，而是一**个可隔离子群的静止点**；而这个子群一旦被隔离，重写的结果与外部就不再互通。

这条收窄不是补丁，它反过来解释了工程实践中最常见的重写手法：**新端口、新版本号、新命名空间**——都是先造一个隔离的子群，在里面重写，然后与旧的长期共存。第五节乙格里"版本协商"这件事，本质上就是**为局部静止点预留位置**。

（十）宪法的可修正性——第二条让步。

第二条让步来自宪法数据，它迫使本文去掉一个价值判断。

数据显示可修正性延长寿命，而本文的框架容易被读成"打补丁是被迫的、重写才是正路"。这个读法是错的：**在很多系统里，永远不被重写正是目的**。一部活了两百年的宪法、一个跑了四十年的域名系统，它们的"改不动"与"活得久"是同一件事的两面。

因此本文的表不含价值排序：**乙格不是病，丁格才是。** 病不在没有静止点，而在没有静止点却按有静止点来管——宣布取代之后就不再为共存负责。

七、判据，以及本文实际跑出的结果

本节与本文其余部分性质不同：这里的数字是我在公开数据上真跑出来的，可以复算。

7.1 核心可裁决判据

> **补丁次数与最终被整篇取代的比例不相关；而是否被整篇取代，取决于该对象有没有可用的静止点。**

7.2 已执行：互联网标准全量记录

数据来源与方法。 取标准编者公开发布的全量索引（一份约 1,370 万字节的结构化文件，含每一份文档的编号、日期、以及取代／被取代／更新／被更新四类关系），于 2026 年 8 月 1 日抓取并解析。共 **9,819 份文档**。全部计算只用索引自带的字段，不涉及任何主观判断。

结果一：补丁不通向重写。

补丁次数	文档数	最终被整篇取代的比例
1 次	862	14.4%
2–3 次	278	15.1%
4–7 次	71	15.5%
8 次及以上	26	26.9%

被打过补丁的共 1,237 份，其中 **85.1% 至今从未被整篇取代**。前三档几乎是一条平线；最后一档略高，但样本只有 26 份。**补丁积累论预测的单调上升没有出现。**

结果二：被反复重写的与被反复打补丁的，是两类完全不同的东西。

被整篇重写代数最多的几支，无一例外是**状态快照与名录**：官方协议状态清单（33 代）、端口号分配（23 代）、站点状态（8 代）、网络主机状态（8 代）。

补丁最多而从未被整篇重写的几支，无一例外是**运行中的协议**：域名系统的实现规范（29 次补丁，跨 36 年）、会话发起协议（21 次）、域名系统的概念规范（19 次）、通用多协议标签交换信令（14 次）、身份认证服务（13 次）、边界网关协议（12 次）。

域名系统的核心规范自 1987 年发布至今被打了二十九次补丁，一次也没有被重写。 而"某年某月的官方协议状态"这份东西被整篇重写了三十三次。两者的改动量不构成解释；两者的差别在于：**没有人"在运行"一份状态清单。**

结果三：丁格的签名确实存在，但数量小。

在 1,383 份已被宣布取代的文档中，**有 17 份（1.2%）在被取代之后仍然被新文档正式打补丁**，共 31 次；从被取代到再次被打补丁的间隔中位数为 46 个月，最长 **215 个月（近十八年）**。最长的几例集中在传输层安全协议的历次版本上——某一版在被下一版取代之后的十八年里，仍然被六份新文档正式更新。

这个数字比我预先估计的小得多。**诚实的读法是：在文档层，取代是相当干净的；丁格的现象主要发生在部署层，而部署层没有这样一份公开台账**。 那 1.2% 是部署层共存**渗回文档层**的部分——之所以还要修一份已作废的文档，正是因为外面还有大量实现在跑它。**这 17 例是丁格的漏出物，不是丁格的全貌**。

7.3 未执行：另外两条判据

宪法一侧。 用公开的跨国宪法数据集，检验修正频率与被整体替换概率的相关。本文预测不相关或弱正相关；"修正减少替换需求"这条机制预测负相关。这条本文**没有跑**，因为该数据集需要按国家—年份重建生存分析，超出了本文的执行范围。

软件一侧。 比较有维护窗口的系统与全天候系统的大版本重写率，控制代码规模与年龄。本文预测前者显著更高。同样**未跑**。

7.4 会使本文为假的两条

其一，若在补丁次数与被取代率之间稳定地测出单调正相关（不同语料、不同年代窗口都成立），本文的核心推论错。

其二，若能找到一类对象：无静止点、无隔离子群、亦无版本协商机制，而仍然发生了整篇替换——本文的必要条件不成立。

7.5 两个交出去的洞

第一，**"使用者"的边界谁定**。域名系统的使用者是解析器、是应用、还是终端用户？边界不同，静止点的可得性不同，而本文没有独立的定界办法。

第二，**部署层没有台账**。本文最想量的那件事——宣布取代之后旧版实际还跑了多久——没有全球性的公开记录；第七节第三小节只能量到它渗回文档层的那一小部分。这个洞不填，并且它同时是本文最容易被反驳的地方。

八、与最近的几种说法的分界

与灵活性延寿（Elkins, Ginsburg & Melton 2009）。 这是最近的一位。该研究以九百余部成文宪法的存续数据支持"易于修正、包容、具体三项设计特征显著延长寿命"。分界在机制而不在结论：那

里认为修正能力**减少了替换的需求**；本文认为可修正的宪法往往也是**不容易出现主权空窗**的宪法，减少的是替换的**机会**而非需求。判决性对照：控制政变、革命、外部占领等空窗事件之后，修正频率对替换概率的效应应当大幅衰减甚至消失——那一支预测效应仍在，本文预测大部分消失。这条对照可以在同一份公开数据上做，本文没做。

与冻结的意外（Crick 1968）。 该判断把不可改归因于改动的波及面。分界：波及面是连续量，本文的自变量是时刻的有无。分岔情形——变体密码：波及面说必须把它处理成例外；本文预测它出现在**可隔离子群**（小基因组、隔离细胞器、偏斜的密码子使用）处，而这正是观察到的分布。这是一条可以继续检验的预测：新发现的变体密码应当持续落在隔离度高的位置。

与补丁积累导向重写（技术债一族）。 那一族的核心比喻是"债会累积、总要还"。分界：第七节的实测显示补丁次数与被取代率基本不相关。要救那一族只有一条路——主张互联网标准不是典型案例；而本文接受这个反驳，并把它转成待跑的软件侧判据（第七节第三小节）。

与"永远不要重写"（软件工程里的实践主张）。 那是一条**决策建议**（重写通常得不偿失），本文说的是**可能性条件**（很多时候压根没有可执行的重写）。二者不冲突，但处方不同：前者劝人别做，后者说该把力气花在为补丁而设计与为局部静止点预留位置上。

与"所有可观察行为都会被依赖"（Hyrum 定律一族）。 那条说的是**为什么改不动**（任何细节都已被人依赖）。本文说的是**为什么换不掉**（没有同时不用的时刻）。二者可同时成立且互不覆盖：一个细节被广泛依赖但存在停机窗口的系统，按前者难改、按本文可换。

与路径依赖与锁定（Arthur、David 一族）。 那一族的自变量是**转换成本**。分界：成本再低，没有静止点也换不了；成本再高，有静止点仍可换。分岔情形——两个转换成本相当的系统，其一有维护窗口、其一全天候：锁定说预测二者同样难换，本文预测差别很大。

与利迪效应（存续越久预期剩余寿命越长）。 那是一条关于寿命分布的经验规律，不给机制。本文给的机制预测一件它不预测的事：**天然有静止点的对象不服从利迪效应**——名录被重写三十三代，每一代的寿命并不随代数增长。

与本栏《类内距》（之十九）。 那一篇论证「同一类所以可替代」是两句话，边界与类内距要分开。分界：那里问一**个类内部的成员彼此有多近**，本文问一**个东西整体上能不能被另一个换掉**。可分开的情形：一个类内距极小、成员高度互换的部件族（甲格），若它装在一个全天候系统里，整族换代仍然只能靠补丁——两篇在这里给出的诊断互不覆盖。

与本栏《双链型》（之十七）。 那里问重复从哪来，本文问替换要在什么时候执行。自变量正交。

九、三家各自为何到不了这一条

协议工程到不了，因为它的两个层各有各的记录，而只有一层有台账。 文档层有一份完整、公开、结构化的记录——正是本文第七节用的那一份；部署层什么都没有。于是这门实践里的争论只能在有台账的那一层进行，而那一层看上去相当干净（取代率、被取代后再修的比例都很低）。**一个只有一半台账的领域，会把没有台账的那一半读成不存在。**

宪法学到不了，因为它的样本全部来自有替换途径的对象。 它研究的每一部宪法都处在一个替换随时可能发生的环境里——政变、革命、占领。因而"没被替换"在它的框架里天然是一个成就，需要用设计来解释。**一个没有反面样本的领域，看不见处境与成就的区别。**

演化生物学到不了，因为它的对象没有设计者。 遗传密码不能被"决定重写"，只能被观察到改或没改。因而这门学问只能问"为什么不变"，而不会问"重写这个动作要在什么时候执行"——**没有执行者的地方，执行的时刻不构成一个问题。**

三家的边界都不是能力问题：一个只有一半台账，一个没有反面样本，一个没有执行者。三者并排放着，那个共同的变量才显出来。

十、推论、适用边界与反噬预言

10.1 推论

推论一。 任何以"改动太多了、该重做了"为据的重写提案，若说不出静止点在哪里，都不是一个可执行的提案。它多半会变成一次"宣布取代"，而共存的代价落到别人身上。

推论二。 对没有静止点的对象，最有价值的投入不是整洁，而是**为局部静止点预留位置**：版本协商、命名空间、可关闭的扩展点。这些东西在整洁性的账上是负分，在可换代性的账上是唯一的正分。

推论三。 一个组织若长期只能打补丁，通常不该被诊断为"技术债管理不善"，而该先查它有没有静止点。**若没有，技术债这个说法在那里是错的诊断，因为它假定了一个可以还债的时刻。**

10.2 适用边界

其一，本文只处理**有明确使用者**的对象。没有使用者的东西（未被采用的标准、无人读的规程）随时可换，本文无话可说。

其二，静止点的可得性依赖"使用者"的边界怎么划（第七节第一个洞），跨对象比较时须先声明口径。

其三，本文不含价值判断：不可重写不是缺陷，只有"无静止点却按有静止点管理"是。

10.3 反噬预言

按本文设计干预，最自然的做法是要求每次"宣布取代"时附报一个静止点计划。

这会适得其反，方式可以精确说出：静止点计划是一份文档，而**文档层正是天然有静止点的那一层**——写一份切换计划不需要任何人停下来。于是最省事的达标办法是**写出一份漂亮的切换计划并宣布完成**，而部署层的共存一天也没有减少。更糟的是，这份计划一旦存在，就为"旧版已经妥善退役"提供了书面证据，使真实的共存更难被看见。

因此本文不推荐把静止点做成申报项，只推荐一件成本很低、且不能被文档化的事：**在宣布取代之后的第 24 个月，去数一数旧版还有多少在跑**。 这个动作之所以不能被优化，是因为它测的是部署层的实际数量，而部署层没有台账——**要数就得真去数**。

十一、本判断对它自己适用吗

本文有没有静止点?

有。**一篇文章发表即完成，没有人"在运行"它**；读者读完就放下，两次阅读之间没有必须被延续的状态。按本文自己的表，它落在**甲格**：可以被整个重写，代价近乎为零。

这个结论对本文并不光彩。按本文自己的框架，**可以被无代价重写正意味着它不承担任何持续运转的东西**——第七节里被重写了三十三代的那份状态清单，正是这样的东西。本文与它同格。

真正在跑、因而只能被打补丁的，是产出本文的那套工序：它每天都在被使用，任何时刻都有一篇在做，没有一个所有环节同时空闲的时刻。**所以那套工序至今只被打补丁，从未被重写；而本文可以随时被扔掉重写一篇**。 这两件事的差别不在质量，在处境。

由此还得到一条对本文自己的诚实提醒：第七节那份数据我只跑了文档层，而我在正文里承认部署层才是关键——**我做的正是"在有台账的那一层做研究"这件我在第九节批评协议工程的事**。区别只在我把它写出来了，而写出来并不等于做了。

结　论

本文提出并论证：重写不是改动量的极限，它需要一个所有使用者同时不在使用的时刻——**静止点**。没有静止点的东西只能被打补丁；有静止点的东西即使几乎没有改动也可以被整篇替换。

这条判断在互联网标准的全量公开记录上被实际执行：**补丁次数与最终被整篇取代的比例基本不相关**（1 次 14.4%、2–3 次 15.1%、4–7 次 15.5%），被打过补丁的文档 **85.1% 从未被整篇取代**；被反复整篇重写最多的全部是状态快照与名录（最高 33 代），而补丁最多却从未被重写的全部是运行中的协议（域名系统核心规范 29 次补丁、36 年、零次重写）。

二维辨别表的靶格是"没有静止点却按有静止点管理"——宣布取代而实际共存。它在文件上完全合规，代价落在没有发言权的一侧，而它的可查签名（已被取代的文档仍需被正式打补丁）在数据里只占 1.2%，因为**部署层的共存没有台账，能被看见的只是它渗回文档层的那一小部分**。

十个领域的兑现中，变体遗传密码与宪法数据各迫使本文收窄一次：前者说明**静止点不必是全局的，可隔离子群的静止点就够**——这正是版本协商与命名空间在工程里真正在做的事；后者说明**永远不被重写在很多系统里正是目的**，因而本文不含价值排序。

最后给一条写死日期的赌注：

> **到 2033 年 12 月 31 日，若在互联网标准以外的至少一个领域中，仍没有任何一项研究把「静止点的可得性」作为独立于改动量与耦合度的变量单独考察，或考察之后未能重现本文所报告的零相关，则本文的核心主张作废，不再申辩。**

注　释

1. 本文所称"重写"指**整篇替换并宣布旧版作废**，不指大规模修改。一次改动了九成内容而仍以补丁形式发布的，按本文口径算补丁。

2. 第七节的全部数字取自标准编者公开发布的结构化索引，抓取于 2026 年 8 月 1 日；只使用索引自带的四类关系字段与日期，不含任何主观编码。**该索引持续更新，复算时数字会随之变化，量级不会**。

3. 第七节结果一的最后一档（8 次及以上）样本仅 26 份，比例上升不宜过度解读；本文的主张只落在前三档的平线上。

4. "使用者"的边界（第七节第一个洞）在本文中按"直接依据该文档的当前版本完成某项不可中断操作的一方"来取，这个取法本身可争。

5. 宪法与遗传密码两侧的转述依据公开出版物的核心结论，**未逐条核对原著页码**；本文对它们的机制解释由本文负责，不应归给原作者。

参考文献

Crick, F. H. C. (1968). The origin of the genetic code. Journal of Molecular Biology, 38(3), 367–379.

Elkins, Z., Ginsburg, T., & Melton, J. (2009). The Endurance of National Constitutions. Cambridge University Press.

Freeland, S. J., & Hurst, L. D. (1998). The genetic code is one in a million. Journal of Molecular Evolution, 47(3), 238–248.

Knight, R. D., Freeland, S. J., & Landweber, L. F. (2001). Rewiring the keyboard: Evolvability of the genetic code. Nature Reviews Genetics, 2(1), 49–58.

RFC Editor. RFC Index（结构化全量索引，https://www.rfc-editor.org/rfc-index.xml ，本文抓取于 2026-08-01）。

Bradner, S. (1996). The Internet Standards Process — Revision 3. RFC 2026.（本文第七节所称"标准过程"文档；截至抓取日被打补丁 15 次，从未被整篇取代）

Mockapetris, P. (1987). Domain Names — Implementation and Specification. RFC 1035.（截至抓取日被打补丁 29 次，从未被整篇取代）

Arthur, W. B. (1989). Competing technologies, increasing returns, and lock-in by historical events. The Economic Journal, 99(394), 116–131.

David, P. A. (1985). Clio and the economics of QWERTY. The American Economic Review, 75(2), 332–337.

人机分工声明

本文由王德生与 Claude 共同署名。三个来源领域由 Claude 选定（工程一侧刻意选了有全量公开数据、可实际执行判据的领域，这是本轮相对前几篇的唯一方法差别）；判断的形成、二维辨别表、十领域兑现（含两条收窄）、判据与证伪设计以及全部文字，由 Claude 生成；**第七节第二小节的全部数字由 Claude 在公开索引上实际计算得出，可按注释二复算**；所依据的方法论框架与发表裁定由王德生给出。

宪法学与演化生物学两侧的转述依据公开出版物的核心结论，未逐条核对原著页码；第七节第三小节所列的两条判据本文**未执行**，已在正文中明写。

字数　纯汉字约 1.2 万　**版本**　v1.0 · 2026-08-01

这一篇由哪三个领域的争论撞成

三边对「一个在用的东西什么时候会被整个重做」给出互相排斥的答案：协议工程里主张定期整合的一方说补丁积累到一定程度就该重写；宪法研究的数据说可修正性反而延长存续；演化生物学说遗传密码根本改不动，因为一处改动波及全部。**本篇与前几篇的方法差别：工程一侧刻意选了有全量公开数据的领域，第七节的判据是实际跑过的，不是设计出来的。**

通信协议工程 · 增量修补 ↔ 整合重写

RFC 全量结构化索引（含取代／被取代／更新／被更新四类关系与日期，本文抓取于 2026-08-01）

互联网标准区分「更新」（打补丁，原文继续有效）与「取代」（整篇作废）。围绕补丁堆积是否使规范语义难以确定、该不该定期整合重写，争论持续多年。这份索引是本文第七节全部数字的唯一来源，任何人可复算。

宪法学 · 灵活性延寿

Elkins, Ginsburg & Melton, The Endurance of National Constitutions（Cambridge UP, 2009）

覆盖 1789–2005 年、两百余国、九百余部成文宪法：寿命中位数只有十九年；易于修正、包容、具体三项设计特征显著降低「宪法死亡」风险，最不包容的一批平均十四年、最包容的一批平均六十九年。按这条结论，修正能力是延寿手段。

演化生物学 · 冻结的意外 ↔ 近乎最优与变体密码

Crick, The Origin of the Genetic Code（J. Mol. Biol., 1968）；Freeland & Hurst 1998；Knight, Freeland & Landweber 2001

遗传密码之所以几乎不变，被解释为「冻结的意外」——任何改动同时影响所有蛋白质，改一个字母全盘皆错。此后受两方向挑战：密码在抗错性上近乎最优（不只是意外）；线粒体与某些纤毛虫确有变体密码（并非完全冻结）。