

重来价

论一个被中断的过程该不该由人接手，不看接手者的能力，而看从头做一遍要花多少

人因工程 × 医疗交接研究 × 崩溃即重启软件工程

作者 王德生 + Claude

栏目 学科通融 · 站外碰撞

篇幅

发表 2026年8月1日

摘要

自动系统在中途停住、出错或交还控制时，谁能接手？三个领域给出互相排斥的答案。人因工程说必须让人保住状态模型，否则长期脱离回路的人在最异常的时刻最不会接；医疗交接研究说解药不在个人——差错的下降来自标准化的交接格式，而让接手者自己重建状态恰恰是最危险的做法；软件可靠性工程走得更远：最成熟的做法是干脆放弃接续，把系统设计成随时可以廉价重启，从而让「接手」这件事不必发生。三家共有一个未说出的前提：中断是要被接住的，分歧只在由谁接、怎么接。三家的材料合起来只能得出：该不该接手根本不由接手者的能力决定，而由从头做一遍的代价相对于从当前位置接手的代价决定——而这个比值不是任务给定的，它由中间状态被写下来的程度决定。本文把这个比值称为重来价，判据不含情态词：从头做一遍要多久？若不比接手更贵，就不该设计接手，该设计重启。第二根轴是重来会不会丢失不可再生的东西。靶格是「重来很便宜、却有不可再生的损失」——技术上确实能重做一遍，而重做一遍时那个人已经走了。

这一篇由哪三个领域的争论撞成

三边对「一个被中断的过程该由谁继续」给出互相排斥的答案：人因工程说必须让人保住状态模型，医疗交接研究说有效的是标准化格式而不是个人，软件可靠性工程说最成熟的做法是让接续变得不必要。三家共有一个未说出的前提：中断是要被接住的。而三家的材料合起来只能得出：该不该接手由重来的代价相对于接手的代价决定，且这个比值由中间状态被写下的程度决定——它不是任务的属性，是记录策略的后果。

医疗交接研究 · 有效的是格式不是个人

[Starmer et al., Changes in medical errors after implementation of a handoff program \(NEJM, 2014\)](#)

面对每天每班次把几十个正在进行中的过程交给另一个人这一最高频版本，真正把差错降下来的不是让接手者更懂更警觉，而是把交接标准化：固定的信息结构加团队培训，在多中心体系中带来可预防不良事件的显著下降，且交接时长没有增加。接续能力被从人身上搬到了格式上——它降低的不是所需能力，是接手所需的信息重建量。

<https://www.nejm.org/doi/full/10.1056/NEJMs1405556>

崩溃即重启 · 让接手不必发生

[Candea & Fox, Crash-only software \(HotOS IX, 2003\)](#)

故障恢复代码是系统里最少被执行也最少被测试的部分，因而最不可靠；而重启这条路径每天都在被执行。由此把组件设计成随时可被杀掉并极快拉起，状态全部外置，故障处理统一退化为「杀掉再拉起」。这里没有人接手，可靠性来自把重来变得极其便宜——它的前提写得很明确：状态必须可以被安全丢弃。

https://www.usenix.org/legacy/events/hotos03/tech/full_papers/candea/candea.pdf

人因工程 · 必须让人保住状态模型

[Bainbridge, Ironies of automation \(Automatica, 1983\) ; Endsley & Kiris, The out-of-the-loop performance problem \(Human Factors, 1995\)](#)

设计者先自动化容易形式化的部分，把低频、异常、复杂的任务留给人；于是操作员平时缺少练习与过程信息，却必须在系统最不正常时接管。长期监控者还会出现情境意识下降与更长的接管延迟。处方由此指向提高人这一侧的状态可得性——而它默认了过程不可重来，因为它的原型场景（飞行、核电、化工）确实不可重来。

https://ckrybus.com/static/papers/Bainbridge_1983_Automatica.pdf

目 录

一、导论：三个不该同时为真的说法.....	5
二、三家各自说到了哪一步	5
三、冲突落在哪一点上	6
四、重来价	7
五、两根轴，四种局面	7
六、一个不含情态词的判据	8
七、逐领域兑现	9
八、两处被材料逼出来的收窄	10
九、与最近的几种说法的分界	10
十、三家各自为什么到不了这一条	11
十一、推论、适用边界与反噬	12
十二、与同期几篇的分界	12
十三、本文的重来价是多少	13

十四、可以判本文错的六种结果14

结 论.....14

注 释.....15

一、导论：三个不该同时为真的说法

一个正在运行的过程在中途停住——系统请求交还控制、模型拒答、接口断开、当班的人下班了——接下来该由谁把它继续下去？三个领域给出三个互相排斥的答案。

第一场在人因工程。 这一支最著名的判断是：自动化并不只是消除人的工作。设计者通常先自动化容易形式化的部分，把难以自动化、低频发生、异常复杂的任务留给人；结果是操作员平时缺少练习与过程信息，却必须在系统最不正常、信息最混乱时接管。^[1] 后续关于脱离回路的研究进一步表明，长期处于监控角色的人会出现情境意识下降、从主动加工转为被动接收，并在自动系统失效后表现出更长的判断与接管延迟。^[2] 处方由此而来：必须让人保住状态模型，必须让关键信息持续可见，必须定期把人放回回路。争点是：**要让人能接手，需要让他保住多少？**

第二场在医疗交接研究。 这个领域面对的是同一个问题的最高频版本——每天每班次都要把几十个正在进行中的过程交给另一个人。它的答案与上一场几乎相反：真正把差错降下来的不是让接手者更懂、更警觉、更有经验，而是**把交接本身标准化**。一套结构化的口头与书面交接流程加上团队培训，在多中心的儿科住院医师体系中带来了可预防不良事件的显著下降，且交接时长没有增加。^[3] 换句话说，接续能力被从人身上搬到了格式上。争点是：**接手的单位是人，还是一份格式？**

第三场在软件可靠性工程。 这一支给出的答案是把问题取消掉。一条被反复验证的工程路线主张：与其让组件在故障后努力恢复到一个一致的状态，不如把系统设计成「只会崩溃、也只会重启」——所有状态外置到专门的存储里，组件本身随时可以被杀掉并在极短时间内重新拉起，因为重启是唯一一条被充分测试过的恢复路径。^[4] 这里没有人接手，也不需要任何人保住状态模型；**可靠性来自把重来变得极其便宜**。争点是：**接手这件事，是不是本来就不该存在？**

三条不能同真。若必须让人保住状态模型，那么医疗交接的证据就该显示训练个人比标准化格式更有效，而它显示的是相反。若标准化格式是解法，那么软件工程几十年来朝着「不必交接」这个方向的努力就是多余的。若最好的办法是让重来变便宜，那么急诊、飞行与手术里那些无法重来的场合就无解——而它们恰恰是最需要有人接手的场合。

本文认为，僵局来自一个三方共有、从未被单独说出的前提：**中断是要被接住的，分歧只在由谁接、以什么形式接**。人因把接的能力放在人身上，交接研究把它放在格式上，软件工程把它放在重启机制上——三家争的是接的方式，没有人问该不该接。

二、三家各自说到了哪一步

人因工程这一支的核心是一条反常关系：控制系统越先进，人在某些时刻的贡献反而可能越关键，而恰恰因为平时不参与，他在那些时刻最不具备参与的条件。^[1] 脱离回路研究把它做成了可测量的：在自动系统失效之后，长期监控者需要更长的时间才能重建局面并作出正确动作。^[2]

这一支的处方全部指向同一个方向：提高人这一侧的状态可得性——显示更多中间量、安排周期性的手动操作、训练异常处置。它没有处理的是另一侧：如果这件事根本可以从头再做一遍，那么状态重建就不是必需的。它默认了过程不可重来，因为它的原型场景（飞行、核电、化工）确实不可重来。

医疗交接这一支提供了最直接的反证。它的干预不是训练个人的状态重建能力，而是给出一套固定的信息结构：病情严重度、患者概要、待办清单、情境意识与应急预案、以及由接手者复述确认。多中心研究显示，实施这套流程之后可预防的不良事件显著下降，而交接所占的时间没有增加。^[3]

这一支的重要之处不在于它有效，而在于它有效的方式：**它降低的不是接手者需要的能力，而是接手所需的信息重建量。**一份格式良好的交接单让接手者不必推断，因而接手变便宜了。它到达的边缘是：它从不问「这件事能不能干脆重来」——在临床里显然不能，所以这个问题不会出现。

崩溃即重启这一支把问题推到了另一端。它的观察是：故障恢复代码是系统里最少被执行、也最少被测试的部分，因而恰恰最不可靠；而重启这条路径每天都在被执行。由此得到的设计原则是把组件做成可以随时被杀掉的——状态全部外置，进程本身不保存任何需要被恢复的东西，故障处理统一退化为「杀掉再拉起」。^[4]

这一支的前提写得很明确：状态必须可以被安全丢弃。它没有处理、也不需要处理的是——有些过程在重来的那段时间里，会失去一些不可再生的东西。

三、冲突落在哪一点上

把三家的处方并排放，互不兼容：让人保住状态、让格式携带状态、干脆不要状态。把三家的材料并排放，出现另一样东西。

崩溃即重启说：如果重来足够便宜，接手就不必存在。医疗交接说：接手的代价可以被降低——降的办法是减少接手者必须重新推断的东西。人因工程说：接手的代价在长期自动化之下会持续上升。

三条串起来只能得出一件事：**该不该由人接手，不是一个关于人的问题，是一个关于两个代价谁更大的问题。**一边是从头做一遍要花多少，一边是从当前位置接手要花多少。哪一边更便宜，就该设计哪一边。

更要紧的一步是：**这个比值不是任务给定的属性，它由中间状态被写下来的程度决定。**写得越少，接手越贵（要重新推断）而重来越便宜（本来也没什么可继承的）；写得越多，接手越便宜而重来越贵（重来等于把已经写下的东西全部作废）。**因而一个系统究竟该走接续路线还是重启路线，是被它自己的记录策略决定的，而不是被任务的性质决定的。**

这一步同时解释了三家为什么各自都对。软件工程走重启路线，因为它可以把状态外置——外置之后，进程内部的中间状态就等于零，重来价极低。医疗交接走接续路线，因为病人的病程不能重来，重来价无穷大，于是唯一能做的是降低接手成本。人因工程处在两者之间的最坏位置：飞行不能重来，而中间状态又大量存在于自动化系统内部、不向人显示，于是接手极贵、重来不可能。

四、重来价

把一个被中断的过程从头做一遍所需的代价，除以从当前位置接手所需的代价，本文称之为重来价。

三件事需要说清。

「代价」以时间为主，但不只是时间。它包括重新取得输入所需的时间、重新执行已完成步骤的时间与资源、以及重来本身带来的风险。接手一侧的代价同样如此：定位当前状态、判断下一步、承担误判风险。两侧用同一套量纲计量，因而比值有意义。

比值小于一，就不该设计接手。这是本文最直接的结论，也是最反直觉的一条：在重来价低于一的情况下，「保留人工接管按钮」不是安全措施，是把资源投在更贵的那条路上。正确的设计是把重来做得更快、更自动、更少副作用，而不是让人更能接手。

比值由记录策略决定，因而是可以被设计的。这是本文与三家最大的不同：三家都把中断当作一个既定情境去应对，而重来价是一个设计变量。一个组织可以通过改变「过程中写下多少」来把自己推向接续路线或重启路线，且这两条路线的投入方向完全相反——一条投在状态可见性与交接格式上，一条投在幂等、可回滚与快速重放上。同时投两条是最常见也最贵的错误。

由此得到本文的承重命题：决定一个被中断的过程该不该由人接手的，不是接手者的能力，而是重来价；而重来价不是任务的属性，是记录策略的后果。

这条判断为什么三家都不会同意。人因工程不会同意，因为它的整个纲领是提高人的接手能力，而本文说在很多场合这笔投入方向就是错的。医疗交接不会同意，因为它把标准化交接当作普遍处方，而本文说它只在重来价很高的领域成立，且它本身就是在抬高重来价（写得越多，重来越亏）。崩溃即重启不会同意，因为它把「状态可安全丢弃」当作一个技术前提去满足，而本文说这个前提在有不可再生损失的场合永远无法满足，与技术无关。

五、两根轴，四种局面

只用重来价一根轴不够。重来价相同的两种局面结局完全不同：一种重来只是费事，一种重来会失去一些永远回不来的东西。

第一根轴：重来价高不高——从头做一遍相对于接手，贵还是便宜。第二根轴：重来是否伴随不可再生的损失——在重来所花的那段时间里，有没有什么东西不可逆地消失了：病人的生理储备、证据的时效、当事人的记忆、一个只出现一次的现象、一段无法回放的市场窗口。

	无不可再生损失	有不可再生损失
重来便宜 (比值低)	重启型 不该设计接手。投入应放在幂等、回滚与快	假重启 技术上确实能重做一遍，而重做一遍时那个人

速重放上。绝大多数计算任务、批处理、可重来的实验。已经走了。系统却按可重启来设计，因为重来的账面代价很低。

重来昂贵 接续型

（比值高）该设计接手，且降低接手成本的最有效办法是标准化交接格式，不是训练个人。

一次性

接手贵、重来也毁东西。控制必须前移到准入、验证与事前设计，运行时的按钮多半是装饰。

右上格是靶格，它有三个使它极难被发现的性质。

第一，它在账面上是最优的一格。 重来的直接成本很低——重新跑一遍、重新排一次队、重新提交一次申请，这些都便宜。而不可再生的那一部分不出现在任何成本表里，因为它落在别处：一次重新提交对应的是申请人多等三周，一次重跑对应的是那个窗口已经关闭。

第二，它的失效不是随机的，是选择性的。 重启对大多数情形都工作良好，只在少数情形下毁掉不可再生的东西；而这少数情形往往正是最要紧的那些。于是系统的整体指标很好，而尾部极差。

第三，它无法靠提高重启质量修复。 把重启做得更快，只会让这一格更稳固，因为它进一步压低了账面成本。这一格唯一的出路是把第二根轴显性化——为那些不可再生的损失建立一个独立的计量，与时间成本分开报告。而这件事没有任何一方有动力去做，因为损失不落在做重启决策的那一方身上。

左下格值得单说，因为它是本文对现行做法最直接的批评落点。**在重来便宜、且没有不可再生损失的场合，保留一个人工接管环节是把资源投在更贵的那条路上，并且会顺带制造一个长期不被使用、因而逐渐失去接手能力的岗位——人因工程所描述的那个恶性循环，恰恰最常发生在本来不需要接手的地方。**

六、一个不含情态词的判据

判断一个中断该由人接手还是该重来，不必评估接手者的能力、经验或警觉度。只需问一句：

从头做一遍要多久？

两个追问把它变成可执行的：这个时间与「定位当前状态+判断下一步」的时间比，谁大？以及——在这段时间里，有没有什么东西不可逆地消失了？

跑几个场合，答案互不相同，而且都不是从命题里推出来的。

一次数据处理任务在中途失败。 从头跑一遍要多久？通常几分钟到几小时，而定位当前状态并接着跑往往更久（因为要先搞清它停在哪、写了多少、有没有写坏）。判定：重来便宜，无不可再生损失，重启型。而这正是这个领域几十年来的实际做法，本文只是给出了它成立的条件。

一次住院病人的班次交接。 从头做一遍是什么意思？重新采集病史、重做检查、重新观察病程——不可能。判定：重来价极高，接续型。而降低接手成本的办法已经被验证：标准化格式，不是训练。

一次福利申请的人工复核中断。 从头做一遍要多久？申请人重新提交，几周。账面上这几乎不花机构什么钱。有没有不可再生的损失？有——那几周的等待落在申请人身上，且不可补偿。判定：右上格，假重启。这解释了为什么行政系统面对流程中断时几乎总是选择「让他重新申请」——按机构自己的账，这确实是最便宜的选项。

一次自动驾驶的接管请求。 从头做一遍是什么意思？没有意义——车正在路上。判定：一次性，接手贵、重来不存在。因而控制必须前移：限制系统可自主处理的场景范围，而不是把希望寄托在几秒钟的接管上。

一次法庭上的证人询问被中断。 从头再问一遍要多久？很快。有没有不可再生的损失？有——第二次询问所得到的，已经是一个被第一次改变过的记忆。判定：右上格，假重启，且这一格在这里是被证据法明确认识到的：这正是全程录像与一次性固定陈述之所以重要的原因。

七、逐领域兑现

一、软件的作业调度。 处方是把幂等与断点续跑当作一个二选一，而不是都要：可重来的作业做成幂等并放弃续跑逻辑，不可重来的做成可续跑并放弃重试。同时实现两者会让两条路径互相污染，且两条都得不到充分测试。

二、病房与急诊的交接。 已在左下格（接续型），且已有验证过的处方。本文能加的一条是：**凡是交接格式变得更完备的地方，重来价会同步升高**，因而一旦这套格式失效（系统宕机、跨机构转诊），损失会比从前更大。这是标准化的一个未被计价的代价。

三、行政审批的流程中断。 靶格。改法不是加快重新申请，是把第二根轴显性化——把申请人的等待时长作为一项独立指标与处理成本分开报告。这条改动不需要改流程，只需要改报表。

四、实验科学。 一次可重复的实验中断了就重做，这是重启型；而一次田野观察、一次临床试验的入组窗口、一次天文事件的观测，重来价无穷大。**同一个实验室里两类任务并存，而它们需要相反的设计，这是实验管理里一个长期被混同处理的地方。**

五、代码评审。 评审中途换人，从头评一遍要多久？通常比接手更快，因为读别人评到一半的意见比自己重读一遍代码更费劲。判定：重启型。这解释了为什么评审的部分接手几乎从不成功，而整块转交是有效的。

六、客服工单。 从头问一遍要多久？很快。有没有不可再生的损失？有——用户的耐心与信任。判定：右上格，且这是这一格最常见的一个实例。所谓「不要让客户重复描述问题」这条经验规则，正是对第二根轴的一次朴素表述。

七、长周期的谈判。 重来价极高（关系与已达成的默契不可重放），因而属接续型；而这一领域几乎没有任何交接格式，全靠个人。这是一个明显的空白：按本文，它应当能从医疗交接那套做法里直接得益。

八、教学的课程中断。 一门课换老师，从头讲一遍要一学期，接手需要读懂前任的进度与学生状态。重来价高，接续型；而多数学校既无交接格式也不训练接手，默认由新老师自己重建——按本文这是典型的资源错配。

九、机器学习的训练中断。 检查点机制正是把重来价人为压低的手段——它把状态外置，使重来退化为从最近检查点重放。这一处最清楚地说明重来价是设计变量而非任务属性：同一个训练任务，有没有检查点，重来价相差几个数量级。

十、医疗交接反过来加的约束。 见下节第二处。

八、两处被材料逼出来的收窄

第一处，由崩溃即重启逼出：状态可安全丢弃是一个前提，不是一个可以被工程满足的目标。 这一支的整套设计建立在「组件内部没有不可丢弃的东西」之上，而它达成这一点的方式是把状态搬到别处——搬到一个本身不能被随意重启的存储里。**也就是说，重启型系统并没有消灭不可丢弃的状态，它只是把这些状态集中到了一个必须走接续路线的组件里。**

这条约束推翻了本文一个自然的推论——「把重来做便宜，就可以普遍摆脱接续」。正确的说法弱得多：**重来价可以被局部压低，代价是把不可丢弃的部分集中到某处，而那一处的重来价会因此变得更高。** 系统整体的重来价没有下降，它被重新分配了。这一点使本文不能主张任何一条普遍的路线选择，只能主张对每一个组件分别判定。

第二处，由医疗交接逼出：接续的单位可以是文档而不是人。 那一支最重要的结论不是交接有效，而是有效的原因在格式上而不在个人上。这直接动摇了本文第一节里从人因工程继承来的那个问法——「谁能接手」。

推到底：「**接手能力**」不该在个人身上测量。一个人在标准化格式下能接手、在没有格式时不能接手，这不是他的能力变了，是接手成本变了。因而所有以个人为单位的接手能力评估，测到的其实是「这个人在当前记录条件下的接手成本」，而它随记录条件变化——同一个人在两套系统里会得到完全不同的分数，而这个差异与他无关。

这条收窄拿掉了本文本来可以主张的一整套评估方案。 剩下的只有一条更窄的说法：可以测量的是一个「岗位在当前记录条件下的接手成本」，不是一个人的接续能力。

九、与最近的几种说法的分界

一、与自动化的讽刺与脱离回路。 那一支预测长期监控者在异常时接管更慢更差，是本文最近的一位。分离线：它把接管能力当作因变量并主张提高它，本文把接管当作一个应当先判断该不该做的选项。判决点：在重来价低于一的场合，那一支预测提高状态可见性会改善结果，本文预测改善很小、而把重来做得更快改善很大。两条处方可以在同一个系统上对照，成本不高。

二、与接管时延与接管质量。 自动驾驶研究已经把接管拆成时延与质量两个读数。分离线：那一支测的是接管发生之后的表现，本文问的是接管这件事该不该被设计进来。判决点：一个接管时延与质量都很好、而重来价低于一的系统，那一支判为设计良好，本文判为资源错配——因为同样的投入放在快速重启上收益更高。

三、与情境意识。 那一支把接手能力归结为对当前局面的把握程度。分离线：情境意识是人这一侧的状态，本文的重来价是两个代价的比。判决点：在同等情境意识水平下，改变中间状态的记录密度，本文预测接手成本显著变化而情境意识评分不变——两者可以分开测。

四、与交接研究本身。 这是本文最近的一位制度邻居，而且它就是在降低接手成本。分离线：交接研究默认接手必须发生（在临床里确实如此），本文把这一点变成一个需要先判断的问题。判决点：把同一套标准化交接流程搬到一个重来价低于一的场合（例如可重跑的批处理运维），交接研究预测差错下降，本文预测收益很小而流程成本明显——这条对照可以在运维团队里直接做。

五、与检查点与容错。 工程上的检查点机制与本文的重来价是同一件事的两个方向。本文承认这一点并明确借用它：**检查点的间隔选择问题（存得越密，重来越便宜，而存本身有成本）正是重来价被显式优化的一个成熟案例。** 分离线只在第二根轴上：容错理论里没有「不可再生损失」这一维，因为它的所有状态在定义上都是可重建的。

六、与沉没成本。 表面上像：已经做了一半，要不要放弃重来。机制不同：沉没成本讲的是不该把已投入的计入决策，本文讲的恰恰是要把「已写下的中间状态」计入——因为它真实地改变了两侧的代价比。判决点：在中间状态完全没有被记录的场合，本文与沉没成本给出相同建议（放弃，重来）；在记录完备的场合，沉没成本仍建议只看前瞻成本，而本文指出前瞻成本本身已经因记录而改变。**两者不冲突，本文是在说明前瞻成本从哪来。**

十、三家各自为什么到不了这一条

人因工程到不了，因为它的原型场景全是不可重来的。 飞行、核电、麻醉——在这些地方重来价无穷大，于是「该不该接手」根本不是一个问题，只剩「怎么接得更好」。

医疗交接到不了，出于同样的原因，且更彻底。 病程不能重来，因而这个领域连问都不会问。它把接手成本降下来的成就巨大，而它不会去想这套办法在别的领域可能是多余的。

崩溃即重启到不了，因为它的世界上没有不可再生的东西。 一个进程被杀掉不会失去任何在时间上不可逆的东西；于是第二根轴在它那里恒为零，四格表塌成一根轴，而它在那根轴上已经给出了正确答案。

十一、推论、适用边界与反噬

第一条推论：不要同时投两条路线。 接续路线的投入（状态可见性、交接格式、接手训练）与重启路线的投入（幂等、回滚、快速重放）方向相反，同时做会让两条路径都得不到充分执行，而且会掩盖究竟该走哪条。

第二条推论：先算比值，再谈能力。 「我们的人能不能接手」是一个应当排在第二位的问题；排在第一位的是「从头做一遍要多久」。这条顺序的改变不花任何钱。

适用边界一：整体与局部不同。 见第八节第一处——重来价可以被局部压低，代价是把不可丢弃的部分集中到别处。因而判定必须逐组件进行，不能对系统整体下结论。

适用边界二：接手成本不是个人属性。 见第八节第二处，这是本文交出去的最大一块。

适用边界三：第二根轴难以计量。 不可再生的损失通常落在系统之外的人身上，因而不出现在做决策的那一方的账上。本文只能主张把它单列，无法提供计量方法。

反噬预言。 若重来价成为一项决策依据，最可能发生的是**对分子的人为压低**：机构会去优化「从头做一遍的账面时间」，因为那是它自己账上的数——加快重新提交、简化重新申请、自动化重跑。而每一次压低分子，都会把系统更深地推进右上格，因为不可再生的那一部分完全不受影响。**本文提出的这个比值，一旦被当作优化目标，会系统性地奖励假重启。** 唯一的对策是坚持两根轴同时报告，而第二根轴恰恰是最难计量、最没有人有动力去计量的那一根。这个循环没有出口。

交出去的一个洞。 本文假定「从头做一遍」有明确含义。在很多真实过程里没有——一个谈判、一段治疗、一次调查，重来的那一遍与原来那一遍不是同一件事，因为世界已经变了。**在这些场合分子无法定义，整个比值取不到值，而它们恰恰是最需要判断该不该接手的场合。**

十二、与同期几篇的分界

与「不按的代价先落在谁身上」那一篇。 那篇问干预权的持有人是否付不干预的账，本文问该不该设置这个干预点。判决点：一个重来价低于一的位置上的人工按钮，那篇会去问不按的代价落在谁身上，本文说这个按钮根本不该存在——**取消一个不该存在的按钮，比给它配一套问责制度更彻底。** 两篇在这一点上互相支持：那篇的「每一个不必要的按钮都是一笔新债」，正是本文左下格的另一种表述。

与「记零位」那一篇。 那篇问一类动作在账本上有没有科目，本文问一次中断该走哪条路线。判决点：一个接手动作被充分记账、而所处场合重来价低于一的岗位，那篇判为健康，本文预测这个岗位长期低使用率并因此逐渐失去接手能力——两篇合起来解释了一类被误诊的现象：不是没人愿意接，是本来就不该接。

与「入判份额」那一篇。那篇问产出进不进别人的判据，本文问中间状态写不写下来。判决点：两者都是关于「留下什么」，但方向不同——入判是让别人下一轮必须看，写状态是让下一个人不必重推。一份被广泛参照而毫无操作细节的结论，那篇判为高份额，本文判为零状态记录、重来价极低。

与「独版」那一篇。那篇说介入前的记录价值在于不可重制，本文说中间状态的记录价值在于降低接手成本。判决点：一份可以被事后同等重写的中间状态记录（例如一份标准化的进度表），那篇判为零独版，本文判为高价值——**两篇在这里给出相反的评价，而这正好说明它们量的是两件事：一件为第三方核对，一件为下一个人省力。**

与「复推率」那一篇。那篇测的是能不能在新情境重新推出同一条依据，本文测的是从头做一遍要多久。判决点：一个复推率极高的人（换个情境照样推得出）在重来价低于一的场合仍然不该被安排去接手——能力与该不该用是两回事。反过来，两篇合起来给出一条推论：**一个高度依赖个人复推的组织，通常是中间状态记录很差的组织，因而它的重来价被人为抬高了，而它会把这解读成「我们的工作离不开人」。**

与「闭合之后还能不能回来」那一篇。那篇的重复产生率量的是二次进入时必须重造多少已有信息，与本文的重来价是同一族两个量：**那篇的分子越大，本文的接手成本越高，因而重来价越低。**两个量在数学上近乎互补。分界在于对象：那篇处理的是一次已经闭合的修复被重新打开，本文处理的是一个尚未完成的过程被中断。判决点：一个留痕极好（重复产生率低）而中间状态从不外置的系统——那篇判为高续修，本文判为高重来价、必须走接续路线；两者不矛盾，但它们的处方投向不同的地方。

十三、本文的重来价是多少

这篇文章如果写到一半中断，从头写一遍要多久？大约与接手一份写到一半的稿子相当，甚至更快——因为读懂别人的半成品往往比自己重写更费劲。判定：重来价接近或低于一，重启型。

按本文自己的结论，这意味着一件事：**为这篇文章保留任何形式的「接手机制」都是浪费**，正确的做法是保证它可以被便宜地重写。而这恰恰暴露了本文的一处虚弱：一篇论证「该不该接手取决于重来价」的文章，它自己所处的那一格是最不需要它的结论的那一格。

更要紧的一处自指：本文的两个代价都由我自己估计，而第二根轴（不可再生的损失）在写作这件事上恒为零。**因而本文全程没有在自己身上执行过一次它所主张的判定**——它的判据在它自己的案例上取到的是最平凡的值。这不是谦辞，它是一个具体的弱点：一个只在别人的案例上被用过的判据，与一个从别人的案例里归纳出来的说法，在文本上不可区分。

能做的对冲只有一条：第十四节第一条检验被写到可以由别人在自己的系统上直接跑，且它需要的两个数（重跑时长、接手时长）在多数运维与流程系统的日志里已经存在。

十四、可以判本文错的六种结果

第一，最便宜、今天就能做的一条。 在一个既有作业系统的日志里，把中断事件按处置方式分成两组：重跑与接手。计算两组的实际耗时与二次失败率。本文预测在重跑耗时低于接手耗时的那些事件里，选择接手的一组耗时更长且二次失败率不低；若两组无差异，或接手在便宜重来的场合仍然更优，核心命题不成立。

第二，能力仍是决定项。 若在重来价相同的条件下，接手者的能力水平仍能强烈预测处置结果，则本文「不由能力决定」这一条过强，应退回为「能力之外还有一个变量」。

第三，比值不由记录策略决定。 若人为增减中间状态的记录密度，接手成本不随之变化，则重来价不是设计变量，本文最重要的一条推论失效。这一条可以用检查点间隔的对照直接做。

第四，标准化交接在便宜重来的场合同样有效。 若把标准化交接流程搬到可重跑的运维场景，差错下降幅度与临床相当，则本文关于「路线该分开」的主张被削弱。

第五，第二根轴多余。 若把不可再生损失单列报告之后，机构的处置选择没有改变，则第二根轴不产生任何行为后果，四格表应塌成一根轴。

第六，本文自陈最软的一处。 分子在很多场合无法定义（见第十一节交出的洞）。若在实践中「从头做一遍」对多数真实过程都取不到值，则本文只在一个很窄的、可重复执行的任务集上成立，不能作为一般判据。这一条应当先于其余五条评估。

正向读数。 若本文成立，在一个系统里增加中间状态的外置程度，应当观察到接手成本下降且重来价上升，而这个变化应早于任何人员能力或培训的变化出现。顺序比相关更难被巧合满足。

写死日期的赌注。 到二〇三一年十二月三十一日，至少应有一个高风险自动系统的监管框架或行业标准，在要求「保留人工接管」之前先要求论证该场景的重来代价，即把「为什么不设计成可重来」列为必答项。若届时所有相关规范仍只要求「有人可以接管」而不问这个问题，且第一条日志对照的差异被证明不存在，本文的主张即告失败。

结 论

一个被中断的过程该由谁继续，三个领域给出三个不相容的答案：让人保住状态模型、让格式携带状态、干脆放弃接续而把重来做便宜。三个答案都只对了一段，因为它们共享一个未被说出的前提——中断是要被接住的，分歧只在由谁接、怎么接。

把三家的材料串起来，这个前提就不成立。该不该接手不是关于人的问题，是关于两个代价的比：从头做一遍要花多少，从当前位置接手要花多少。而这个比值不是任务给定的，它由过程中间状态被写下来的程度决定——写得越少，接手越贵而重来越便宜；写得越多，反过来。因而一个系统走接续路线还是重启路线，是被它自己的记录策略决定的。

由此，判断只需要一句不含情态词的问话：从头做一遍要多久。若不比接手更贵，就不该设计接手，该设计重启；而在这类场合保留一个人工接管按钮，不但是把资源投在更贵的那条路上，还会顺带养出一个长期不被使用、因而逐渐失去接手能力的岗位。

剩下的一格最危险，也最难看见：技术上确实能重做一遍，账面成本也确实很低，而重做一遍时那个人已经走了、那个窗口已经关了、那段记忆已经被第一次问话改过了。这一格在任何成本表上都是最优解，因为它省下的钱在自己账上，赔掉的东西在别人账上。

注 释

1. 本文所说的「重来」指把该过程从起点再执行一次，不包括从中途某个保存点恢复——后者属于降低接手成本的手段。
2. 「不可再生的损失」指在重来所需的那段时间里不可逆消失的东西，不包括可以事后补偿的成本。
3. 医疗交接部分只使用「结构化交接流程与团队培训可降低可预防不良事件且不增加交接时长」这一多中心结论，不涉及具体机构的实施细节。
4. 第五节四格表尚未经过信度与效度检验，是研究设计的起点，不是成熟的评估工具。

参考文献

1. Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775-779. ckrybus.com
2. Endsley, M. R., & Kiris, E. O. (1995). The out-of-the-loop performance problem and level of control in automation. *Human Factors*, 37(2), 381-394.
3. Starmer, A. J. et al. (2014). Changes in medical errors after implementation of a handoff program. *New England Journal of Medicine*, 371, 1803-1812. nejm.org
4. Candea, G., & Fox, A. (2003). Crash-only software. *Proceedings of HotOS IX*. usenix.org
5. Gold, C., Damböck, D., Lorenz, L., & Bengler, K. (2013). "Take over!" How long does it take to get the driver back into the loop? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57(1), 1938-1942.
6. Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32-64.
7. Young, J. Q., Ten Cate, O., O'Sullivan, P. S., & Irby, D. M. (2016). Unpacking the complexity of patient handoffs through the lens of cognitive load theory. *Teaching and Learning in Medicine*, 28(1), 88-96.

人机分工声明

本文由王德生提出研究方向、承重判断与最终发表责任；Claude 完成三个领域的选源与冲突定位、公开文献检索、命题成形、二维表构造、逐领域兑现、分界与证伪设计以及正文写作。第八节两处收窄由材料逼出：第一处推翻了「把重来做便宜即可普遍摆脱接续」这一自然推论，第二处拿掉了本文本来可以主张的整套个人接手能力评估方案。第十三节承认本文在自己的案例上从未执行过它所主张的判定。本文未标注任何评分——按本站惯例，参与改写者不为自己改写的文本打分。

证据边界 医疗交接部分只使用「结构化交接流程与团队培训可降低可预防不良事件且不增加交接时长」这一多中心结论；软件工程部分使用的是一条公开的设计路线，不主张任何具体系统的实现细节。第十四节六条判据本文**均未执行**，其中第一条所需的两个数（重跑时长、接手时长）在多数运维与流程系统的既有日志里已经存在。

字数 纯汉字约 1.1 万 **版本** v1.0 · 2026-08-01

本文由三个分属不同学科、且互相冲突的理论体系碰撞而成。全部来源均为站外公开文献，可自行核对。 作者：王德生 + Claude · SDE Universes · 学科通融