

我最近编程上瘾了，但瘾的不是代码

原创 约翰老师 五宝爸约翰聊英语 2026年6月18日 10:01 新加坡

最近一个月，我几乎每天都在写东西——一个PTE模考网站，一个单词听力诊断工具，还有一个正在搭骨架的英语学习智能体平台。朋友问我，你又不是程序员，图什么？

我想了很久，发现自己其实答不上来"图什么"，因为这个问题问错了方向。它假设我在追求一个**结果**——一个能用的网站、一份漂亮的成绩单、一个能上线的产品。但回头看这一个月，真正让我上瘾的，从来不是某个项目"做成了"的那一刻，而是中间那个说不清道不明的过程：一个东西从一句模糊的念头，慢慢长成一个能跑起来的、有自己脾气的系统。

这件事最近被我用一套理论照亮了——这套理论叫SDE本体论，核心命题只有一句话：

世界不是被发现的，而是发生的。

听起来有点玄。但当我用它回看自己这一个月在干什么，突然所有的"上瘾感"都有了解释。

一、为什么"做成"不是重点，"发生"才是
先说一个最朴素的对比。

如果你问我"PTE模考网站做出来了吗"，答案是做出来了，能用。但如果你问我"这个网站是怎么来的"，答案完全是另一个故事——它不是从一份完整的需求文档里"提取"出来的，它是一路撞出来的。

最开始我以为自己要做的是一个**官网风格的展示页**，介绍考试、展示题型。做完之后我才意识到，这玩意儿连"模考"都算不上——它更像驾校的宣传册，不是模拟驾驶舱。这个认知不是我提前规划好的，是做完第一版、盯着它发愣的那一刻冒出来的。

然后才有了第二版：真正能计时、能判分、能模拟阅读和写作的系统。再往后，又冒出"做一套题会被记住答案"的问题，于是有了三套题轮换、随机抽题的机制。

每一步新功能，都不是上一步计划的延伸，而是上一步**暴露出的缺口**逼出来的。

SDE本体论把这种现象叫作"发生"，区别于"发现"。发现范式假设事情的样子是先定好的，你只是在按图施工；发生范式认为，事情的样子是在做的过程里，被一连串"撞上去的差异"逼出来的——你不知道下一步该长成什么样，直到你撞上那个让你别无选择的瞬间。

我现在回头看，能上瘾的从来不是"完成"，是这种**被撞出来的生长感**。

二、三个项目，三种“裂缝”

SDE本体论里有个说法我特别喜欢：每一次真正的创造，都是从一道“裂缝”开始的——旧的做法在哪里失效了，旧的语言在哪里突然说不下去了。我这一个月做的三件事，刚好对应三种不同的裂缝。

裂缝一：“听得懂”和“听得出”不是一回事

第二个项目是个词汇诊断小工具，最初的版本很简单：给一个单词、一个例句，做四选一的“这个词是什么意思”。

跑了几轮之后，一个问题冒出来：有些词学习者明明选对了意思，但放到真实听力材料里，他完全反应不过来。这道裂缝逼着我重新想——“认识一个词”原来不是一件事，是**三件事**：认识它的意思、能从语流里听出它、能听懂它在句子里干了什么。

这恰好是SDE说的“S是连续谱，不是二元判断”——一个词的掌握程度从来不是“记住了/没记住”两档，而是从“见过”到“认识”到“听得出”到“能用”的一整条光谱。把它压缩成对/错两档，诊断必然失真。

所以工具后来变成了三关递进：形义关过了才进辨音关，辨音关过了才进理解关。任何一关卡住，立刻停在那里，标记清楚是卡在哪一层——而不是笼统地说“这个词没掌握”。

这道裂缝带来的不只是一个功能升级，是一次诊断逻辑的重新发生：从“对不对”换成了“卡在哪一层”。

裂缝二：词汇学习真正缺的不是重复，是“差异”

这套三关诊断逻辑跑通之后，下一步是想把它嫁接到一个更大的系统里——一个完整的英语学习智能体：看视频、听音频、过单词、跟读、朗读、复述，一整条流程，还要做后台素材管理和前端学员系统。

这是一次明显的升维，从“一个晚上能写完的小工具”跳到“需要真正后端和数据库的双端平台”。架构问题一下子全冒出来了：素材怎么存、流程怎么推进、多用户怎么隔离数据。这些问题不是规划出来的，是上一个项目长大之后**自己暴露**出来的。

但更深的裂缝在这里：几乎所有主流单词记忆法——艾宾浩斯曲线、间隔重复——共享一个隐藏假设：单词的意义是先存在的，学习就是把它重复“压”进记忆里。

这套假设解释不了一个常见现象：为什么有些词背了五遍还是转头就忘，但某次在剧里听到的一句台词，却一辈子记得清清楚楚？

SDE给出的诊断是：真正驱动记忆的不是重复的次数，是**差异的质量**。每一次重新遇到一个词，如果是在新的语境、新的情绪、新的对比下撞上它，才是真正的“习得动作”在发生；单纯的同质重复，反而会让这个词加速沦为听不见的背景音。

这就是为什么这个学习平台不打算只做"刷词卡"——它要做的是让每个词被"看见、听见、用出来"的场景反复切换语境，而不是反复曝光同一个语境。重复的是词，变化的是它出现的世界。

裂缝三：工具本身也要回头审视自己的根基

第三个项目其实不是一个软件，是一个思维操作台——围绕SDE本体论本身搭的一套互动工具，帮你把任何一个领域的旧思想拆开、找裂缝、重新锻造成新概念。

这个项目暴露的裂缝最深，也最有意思：**它逼着我用SDE审视SDE自己**。比如我曾经问，这套理论能不能用在孩子的听读训练上——孩子看过视频、听过音频，内容很熟，但还不识字，文本对他来说是一堆没被激活的形状。这其实是同一个结构：内容的"纠缠"（E）已经很厚了，缺的是"结构显露"（S）的最后一步。

这道裂缝让我意识到，前两个项目表面上是"做产品"，但骨子里都在回答同一个问题：**怎么把一个人已经具备的、还没被命名的东西，推到他自己看得见的那个临界点上。**

三、为什么这个过程让人上瘾

回到最初的问题：图什么？

我现在的答案是——图的不是某个项目的完工，是反复经历"被裂缝逼着重新想"这件事本身。

每一次代码跑起来又跑不通，每一次发现"哦原来我想错了不是这样"，都是一次微小的"发生"现场。这跟刷刷、打游戏的爽感不一样——那种爽感来自于**被设计好的反馈**，而编程上瘾，更接近于你亲手撞开了一道原本没注意到的裂缝，然后看见一个之前不存在的结构，在你手底下长出来。

SDE本体论书里有句话我很认同：永恒的不是某个最终状态，永恒的是发生本身。

这话用在编程上瘾这件事上，意外地贴切。不是某个网站、某个工具值得上瘾，是那个"原来可以这样长出来"的瞬间，值得反复追。

小猪佩奇单词学习智能体即将开启试用，扫码入群了解：

