一个人怎样带动一群 AI：Atrium 的三个目标

同时开七八个 AI 编码会话，你会发现自己成了最慢的那一环。多开会话解决不了这个问题，要的是让 AI 组织起来，并且学会你的判断。这篇讲 Atrium 想做到的三件事、它现在怎么做，以及哪些还没做到。

2026-09-29 · 5 分钟阅读



写代码这件事，AI 已经做得够好了。这两周我真正的瓶颈是自己：同时开着七八个 AI 编码会话，每个隔几分钟就停下来问一句——要不要删掉旧接口？测试挂了三个怎么办？PR 开好了，帮我看一下。我在窗口之间来回切，每个 AI 都在等我。

再多开几个也没用。每个会话从开始到上线，都有几处离不开人：说清要什么，回答它中途的问题，核实它说的「测试通过了」是不是真的，合入代码，上线后看一眼有没有坏。假设一个会话每 6 分钟找你一次，每次切过去要一两分钟，开到四五个，你就忙不过来了。

Atrium 想解决的就是这件事：让一群 AI 像一个团队那样分工，只在需要你判断的时候来找你。它自己的代码就是这样写出来的，最忙的一天合入了 105 个改动，由我一个人带着做。



要做到哪三件事

要让一群 AI 替你干活、你又不用盯着，光会分工还不够。它得做到三件事，缺一件你都放不下心。

像你一样判断。你不在场时，它做出你本来会做的决定：这个接口该不该删，这个方案值不值得做。它判断得越像你，你要纠正的就越少，你也就越敢把决定交给它。

自己找事。你给方向和底线，它自己发现该做的事，提出来让你选，而不是等你一件件派。

越做越好。干活的 AI 做完就走，但做事时学到的东西要留下来：哪种做法有效，哪里踩过坑，你纠正过什么。同一个错误不该犯第二次。

这三件事是连着的：判断像你，你才敢让它自己找事；它自己做的事多了，才有东西可学；学到的又让它的判断更像你。

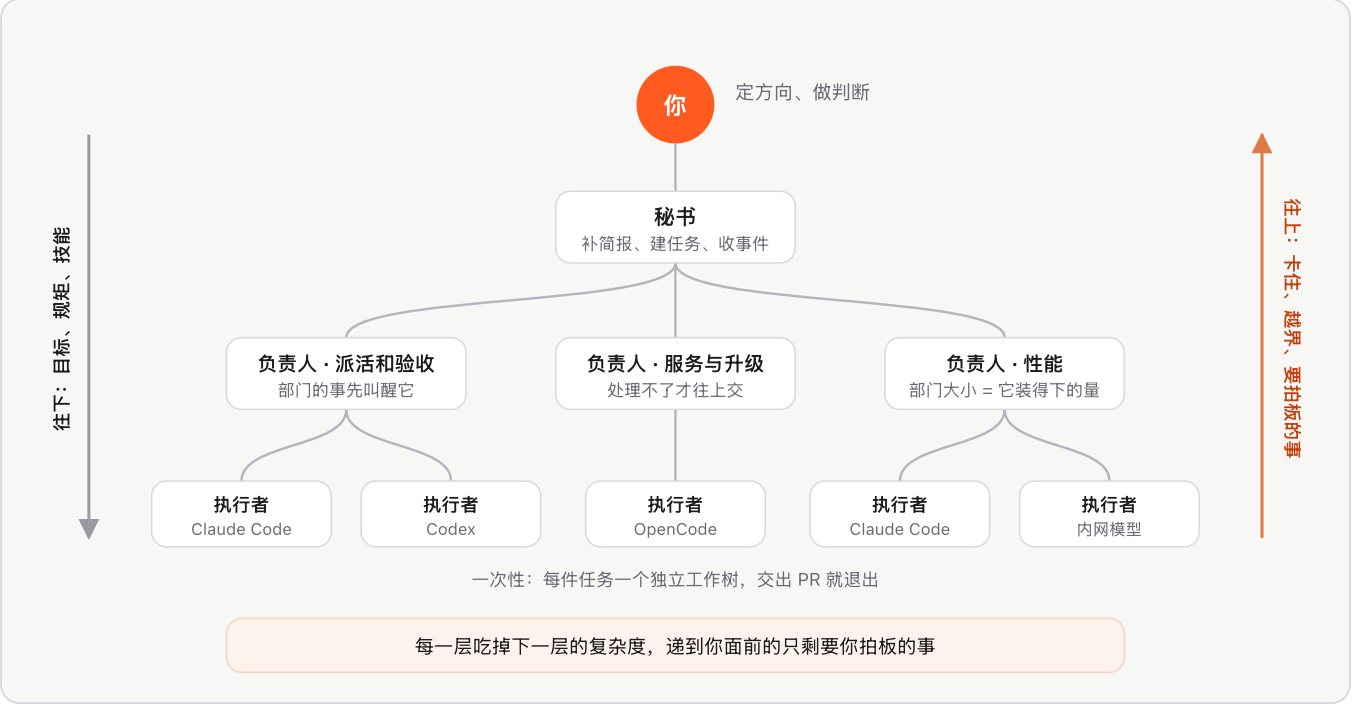
下面用今天中午的一件事，说明 Atrium 现在是怎么做的。

一次发版事故

Atrium 每次合入代码都会自动发一个新版本，服务自己下载、重启。今天中午 12:20，一次自动升级失败了：新版本刚发布的那一秒，GitHub 上的文件还没传完，服务只下载到一半就报错，停在了旧版本上。

从出事到修好上线用了 10 分钟，这期间我在做别的事。这 10 分钟正好把 Atrium 的几个做法都走了一遍。

事情在下面消化



你只对秘书说话；负责人各管一个部门；执行者做完一件事就退出。往上走的只有卡住的、越界的和要你拍板的事。

Atrium 把 AI 组织成一棵树。我只跟最上面的秘书说话，它是一个 Claude Code 会话。往下是按职责分的部门，每个部门有一个负责人。最下面是真正写代码的执行者：每件任务临时起一个编码工具，比如 Claude Code 或 Codex，在一份单独的代码副本里干活，交出代码就退出。

升级失败时，服务把这件事报给了秘书。秘书看了日志，先手动把服务升上去，让卡住的活接着走；然后把问题写成一个修复任务派了出去。一分钟后，一个执行者接下任务开工。

结果由系统自己查

两分钟后，执行者说「修好了，PR 开好了」。Atrium 不采信这句话，它自己去 GitHub 查：PR 在不在，检查过没过。过了才放进合入队列；合入前再把代码更新到最新、跑一遍检查，保证合进去的正是检查过的那一版。合入后自动发版，服务升级完先跑一遍只读的检查，通过了才记「已上线」。12:30，修复上线。

要是检查没过，任务会退回给同一个执行者重做，最多两次；再不过，就交给部门负责人处理。



从一句话到上线，你只出现在开头和结尾；中间每一步是否完成，都由系统自己查。

你的判断跟着每件活走

这个执行者之前从没接触过 Atrium，却知道该往哪个方向改。原因是派活时，Atrium 把这个部门和它上级部门的规矩逐条附在了任务说明里，其中一条是「随时能升级重启，不等空闲」，另一条是「先找最简单的做法，能删的就删」。

这些规矩都是我以前定下的，或者纠正它时说过的话。同一个判断只记一处：对所有事都成立的写进全局原则，某一块的规矩写进那个部门，做法和口味写成技能。之后每次派活都自动带上，我不用一遍遍重复。

只在需要你判断时找你

这 10 分钟里没有人来问我，因为这件事用不着我判断：问题清楚，改法也清楚。

真需要我决定的事，比如先做 A 还是先做 B，会整理成几个选项，每个写清收益和代价，我挑一个就行。平时组织能自己转，靠的是一条规则：每件没做完的事都有人负责、有时限，到时间没有进展，就叫醒负责的人，还不行就往上交，不需要谁一直盯着。我也随时可以用一条命令让所有自动的动作停下来。



Atrium 的网页只用来看。这一天没有等我处理的事，上线了 21 件。

三件事做到了哪一步

三件事的进度差别很大：

目标	现在做到的	还缺的
像你一样判断	我的判断写在全局原则、部门规矩和技能里，派活时自动带上	还没统计我纠正了它多少次，说不出它离我有多近
越做越好	我纠正一次，秘书就把它写进规矩或技能	都靠秘书手动；干活的 AI 不会从结果里自己学，同类问题会不会再犯也没有记
自己找事	定期派活、整理选项让我挑，这些做法已经有了	还没用起来，它现在不会自己去发现新的事

所以下一步是先把「我纠正了几次」「同类问题有没有再犯」记下来。前两件事能看出进展了，再放手让它自己找事。

还有哪些代价

- **额度会用完**。AI 用的是各家订阅的额度，用完就得换一个。今天就有一件活跑到一半撞上了限额，只能换一个 AI 重做。
- **秘书还是单点**。秘书的会话关掉以后，要它处理的事会排队等着；下面的部门照常转。
- **出错就停**。现在还在开发阶段，出了问题不自动恢复，要有人去看。今天这次事故就是这样发现、修掉的。
- **还没接公司内网**。最终要在公司内网里替一整个团队干活，现在只接了 GitHub。

代码在 github.com/liu-zhengdong/atrium。

原文 liuzhengdong.babelgo.cn/atrium/ · 演示版 liuzhengdong.babelgo.cn/atrium/talk/