

【分享】AI的正确用法:让它干活,你去生活 | WorkBuddy产品负责人Jason – 图文解说

一张关键画面对应一段完整解说，按原视频时间推进。

01. 深夜录制与 Jason 的产品初心 (00:00-01:31)



场景 1

完整内容

主持人开场说，这是《一鹅传鹅》录制得最晚的一期，开始录制时已经是北京时间晚上 8 点。之所以选这个时间，是因为 Jason 一天里只有这两三个小时能抽出来聊天。随后主持人请 Jason 先做自我介绍。

Jason 说，自己叫汪圣杰，大家可以叫他 Jason，是一名产品经理，也是比较早进入 AI 时代做产品的人。他是 2021 年来到腾讯的，一开始负责的是云端 IDE。Jason 说，自己来鹅厂最大的梦想，就是让“写代码像写文档一样简单”。主持人追问，你一加入时就有这个愿景吗？Jason 回答说，是的，他当时就一直在和老板讨论，为什么写代码不能像写文档那样简单，这件事困扰了他很久，他想要的就是一个 IDE。

画面说明

这一幕画面里是主持人坐在麦克风前做开场，字幕正好落在“为什么会在这个时间开始录制”，把“深夜才约到 Jason”这件事具体化了。镜头上的轻松聊天感和 transcript 里关于“云端 IDE”“写代码像写文档一样简单”的表达形成对照：表面上是一次晚间播客，实质上是在引出 Jason 作为产品负责人最早的核心命题，也就是怎样把原本高门槛的软件开发，做成更接近日常表达的事情。

02. 为什么他一直想让普通人写代码（01:31-03:01）



场景 2

完整内容

Jason 接着解释，他想做的不是普通意义上的 IDE，而是“颠覆所有 IDE”的东西。第一步就是做一个国产 IDE，让更多普罗大众都能写代码。那时候虽然已经有 AI，但 AI 热了很多年，一直没有出现像 ChatGPT 这样横空出世、足够确定的结果。

他也观察到，中国用户并不会像海外一些极客群体那样，在周末把写代码当作爱好来做。海外有一些围绕 Replit 一类产品形成的社区，用户会分享作品、互相修改，再把它变成以个人创意为导向的应用开发；但这种模式在国内很难直接复刻。所以这件事一直让他困扰很久。主持人顺着追问：是不是因为代码本身太难写，所以你一直希望把它变得像画画、书法那样，门槛更低、更容易开始？Jason 回答，是的。

画面说明

画面里仍然是主持人在追问，字幕定格在“是因为代码本身难写”，刚好对应这段对话的关键转折。镜头没有切到复杂演示，而是让观众盯着提问者本身，这种处理方式强化了视频此处真正讨论的不是某个具体功能，而是“为什么普通人过去进不来，未来又为什么可能进得来”这个底层问题，也就是 Jason 想用 AI 改写的门槛结构。

03. 从 Copilot 到第二位嘉宾的故事 (03:01-04:33)



场景 3

完整内容

Jason 说，到了 2021 年年末，这件事发生了天翻地覆的变化，ChatGPT 出来之后，整个方向一下子明确了。其实在 ChatGPT 之前，他一直就在关注 GitHub Copilot。它本质上是代码补全，你打几个字，它给你一些推荐。Jason 认为，这类产品有商业化空间，也有明确价值，但它更像程序员在工作场景里的提效工具，主要还是服务开发者，而不是把软件开发真正变成以创意为导向、让更多普通人参与的事情。再往后，当行业开始进入智能体领域，又出现了新的变化：各种编排方式开始出现，但如果只是把 AI 框在一个很小的特性里，让它按既定规则做局部决策，其实会极大限制 AI 自主思考和解决问题的能力。

随后主持人把话题转向第二位嘉宾王博元。他介绍说，王博元是一位前媒体人，和他们的缘分很特别，因为他是 CodeBuddy 还处在插件时期时的用户。主持人第一次听到他的故事时非常震惊：身边居然真的有这样一个人，用 CodeBuddy 做出了一款上架苹果商店的 App，而他的背景却是英语文学专业

毕业、做了十多年媒体编辑和文字工作。主持人说，这个故事颠覆了自己对“谁能做软件”的认知。也正因为如此，今天才把 Jason 和王博元请到一起聊这件事。

画面说明

这一帧依旧是主持人面对镜头，字幕落在“且他的背景是英语文学的一个专业的毕业生”，视觉上没有技术炫技，反而把“一个文科背景的人也能做出 App”这个反差直接顶到前台。它和前半段关于 Copilot、智能体边界的讨论连在一起，形成了一个明确论点：技术路线变化的意义，不只是程序员效率更高，而是“能不能做软件”的人群边界正在被重新划定。

04. CodeBuddy 的前身与漫长沉淀（04:33-06:04）



场景 4

完整内容

主持人总结说，当写代码这件事原本像一道技术壁垒时，也许在像 Jason 这样的人把工具亲手推出来之后，硬币的另一面会出现别的答案，这也是今天真正想聊的话题。接着他追问 CodeBuddy 的前世今生：这个产品从 2022 年走到现在快四年了，但据他了解，前期并没有很早正式上线，这段时间是不是一直在沉淀？

Jason 回答说，确实还有另外一段故事。这个产品更早之前的名字其实叫“AI 代码助手”，后来才逐渐发展成 CodeBuddy。他们一直把自己定位成工具，也是助手。最开始做 IDE 的时候，他们先是做插件；但在插件阶段，并没有特别好的模型，既不足以把补全做好，也不足以把对话式体验做好。而且当时团队又挂在腾讯云体系里，产品必须要有一套能成立的方案，才能真正往前推进，所以那段时间整体处在比较低谷的阶段。后来 Jason 就先做了一个更面向程序员的版本，也就是 AI 代码助手，这个阶段持续了很长时间，基本一直延续到智能体能力逐步发展起来之后，他们才重新认真考虑把 IDE 这件事再做起来。

画面说明

这一幕镜头切到了 Jason，本人出镜、手势明显，字幕正好是“必须需要有一个方案”。这和他说的内容完全一致：产品不是一开始就凭理想起飞，而是在大公司体系里被模型能力、组织归属、商业化压力一起约束。画面里的 Jason 不是在讲抽象趋势，而像是在复盘一段具体而艰难的产品史，这让“CodeBuddy 不是突然冒出来的，它是长期低谷中磨出来的”这层意思更可信。

05. 用户画像要靠一个个真实人去问（06:04-07:35）



场景 5

完整内容

主持人接着说，这听起来很像一家大公司里的小团队，顶着商业化压力一路坚持，直到今天大模型能力终于足以支撑不错的 Web Coding 体验；而且现在除了 CodeBuddy，他们又做了 WorkBuddy，可以帮用户处理非常多的任务。主持人也回到 Jason 最初的目标上来：他当年想做的，就是让更多人能写代码。像王博元老师这样的用户，如果不是因为你一直在坚持做这个东西，他也许就不会偶然发现它，更不会借它创造出新的产品。主持人因此追问：这种用户的诞生，是否符合你一开始的设想？

Jason 回答说，做产品一定要知道用户画像，知道用户的痛点是什么。程序员的痛点他很清楚，因为他自己就是程序员；但如果要做一个不只服务开发者的产品，那他就必须一个人一个人地去问。现在做 WorkBuddy 小程序时，他就是这么干的，连爸妈和女儿都会问。他会直接问对方：你能不能用 WorkBuddy 完成你的诉求？

Jason 举了女儿的例子。她可能只是在小学里想做一些小应用，而且希望通过语音来做。当时有个很小但很典型的问题：Jason 让她试用 WorkBuddy 后，系统给了三个选项，但她不会打拼音；而语音输入只在深入输入框里才有，前面这几个选项本身却没有语音能力。结果第三个选项需要自己手打，她就用不了了。Jason 的意思是，真正面向更广的人群做产品，问题往往不是模型会不会写代码，而是这些看似很细的小交互能不能让真实用户用下去。

画面说明

画面里是 Jason 继续讲案例，字幕落在“她可能是小学里面想做一些小应用”，一下把抽象的“用户画像”落到了具体人物身上。这个镜头的价值在于，它把 AI 产品讨论从宏大的模型能力，拉回到一个小学生会不会拼音、能不能点选、能不能继续往下走的细节上，也就解释了 Jason 这一段真正想强调的东西：把更多人带进来，关键不只是生成能力，而是你是否愿意围绕真实用户一层层把门槛拆掉。

06. 模型变强后，IDE 开始被自然语言改写（07:35-09:06）



场景 6

完整内容

主持人补充说，很多人对腾讯做 AI 的认知，往往觉得它比别人更晚，直到去年年初元宝等产品出来，大家才意识到腾讯开始明显发力；但据他了解，CodeBuddy 团队其实早在 2022 年就已经开始做 AI Coding 相关产品了，在当时的国内已经算比较靠前的一批。于是他追问：这背后的判断依据是什么？

Jason 回答说，这背后其实是一个临界点。当越来越多人不再满足于代码补全、代码生成，或者 ChatGPT、元宝这类“你先提需求，我再给你一段答案”的产品形态时，问题就变了。Jason 认为，这类产品本质上更接近深度搜索和内容整合：你给出诉求，它返回整理后的结果。代码场景和普通内容场景在逻辑上其实很像，只是面向的人群不同，一个是代码，一个是更广义的内容或工程。

也正因为这样，他们曾经引入过很多 RAG 一类的技术，试图让模型针对 codebase、针对整个代码工程做向量检索，但效果并不理想，所以他们一直在等待更强的模型。后来他们发现，模型能力确实越来越强，包括 Claude 这样的公司也开始做更强的代码能力。Jason 的判断是，有些路线并不是继续堆

更重的 RAG，而是直接用更强的 AI Coding 能力去解决工程问题。与此同时，像 Cursor 这样的产品，也已经不再把愿景停留在代码补全，而是在朝“下一代 AI 导向的 IDE”演进，甚至进一步走向 AI 智能体。到这一步，写代码这件事就被改得越来越简单了，用户开始可以直接用自然语言去修改和生成更多代码，而不必再把“补全”当成核心交互。

画面说明

镜头中 Jason 转向一侧讲话，字幕是“我发现模型能力现在确实越来越强了”，几乎把这一段的论点直接写在了画面上。这里的视觉重点不是产品界面，而是 Jason 本人的判断过程：他先回顾补全、对话、RAG、codebase 检索这些旧方案，再把视线落到更强模型和新一代 IDE 路线。画面因此像一次当面口述的产品拐点说明，帮助观众理解为什么“自然语言写代码”不是一句口号，而是模型能力跨过临界点后，IDE 交互范式真的开始发生变化。

07. 从云上 IDE 到一体化工作台（09:06–10:37）



场景 7

完整内容

嘉宾回顾说，Claude 这一路其实自己定义出了一些新的交互范式；再加上到年底又出现了 Claude Code 这种更面向 AI 的“引擎”形态，他们自己也有类似 CodeBuddy Code 这样偏终端的形态，于是团队逐渐意识到：还需要一个更完整的 IDE，而且这个 IDE 不能只是给少数极客用，而要成为一个面向更大范围专业用户、能形成用户黏性的开发平台。腾讯特别需要这样的东西。

他接着说，团队把自己手上的牌摊开来看，会发现代码这件事确实有机会被做得更容易。不只是开发同学，连非开发的人也应该能一眼切进去，在里面做更多事情。这个想法的动因，一部分来自 Lovable 这类产品。它本身也是 AI Coding，已经在证明：普通人可以通过更自然的方式去生成应用。

另一部分启发来自 Replit。Replit 本质上就是一个云上 IDE，可以在上面做很多应用开发，它先解决的是环境问题，你不需要自己折腾很多本地环境，就已经可以开始写代码了。但即便如此，写代码本身还是复杂，环境被解决了，不代表门槛就真的消失。后来 AI 能力叠加进来，再加上 Agent 这类能力以后，它开始可以自主地去写更多代码，这个逻辑就完全不一样了。

嘉宾说，也正因此，他们回头看自己手里的能力时会发现：一边有 Cloud Studio 这样的云端 IDE，一边又有写代码的 AI Coding 能力、代码补全能力和智能体能力，如果把这些结合起来，就不该只做一个零散工具，而应该做一个真正产业化、面向产研一体化的开发平台。现在 WorkBuddy 讲的也不是一个孤立工具，而是一个工作台：它要连接腾讯内部的产品，也连接腾讯外部一些很好用的免费产品，把这些能力接上以后，用户就能在同一个工作台里完成事情。最后主持人顺势总结，说 Jason 刚才其实已经把 CodeBuddy 的诞生背景和来源讲清楚了。

画面说明

画面里嘉宾正侧身解释，字幕直接落在“Replit 本质上它就是一个云上 IDE”这句上，几乎把这一段的核心支点钉死了。镜头没有展示复杂产品图，而是把注意力放在讲述者本人身上，让观众顺着他的推演去理解这条产品演进链路：先是云上 IDE 解决环境问题，再是 AI Coding 和 Agent 解决“代码仍然太复杂”的问题，最后才推到 WorkBuddy 不是单点工具，而是连接多种产品与能力的一体化工作台。

08. 普通人为什么会在今年开始感到 AI 变快了（10:37-12:08）



场景 8

完整内容

主持人先把话题拉回到早期判断上。他说，在 WorkBuddy 很早期时候，团队希望它最终会长成什么样子，也许 Jason 这样更早关注海外 AI 和垂直领域发展的人，已经比大多数普通用户更早看到了趋势。但如果站在普通人的体感上来说，真正明显感到“AI 一下子变快了、能干活了”，大概还是在今年年初。

主持人解释，最早大家会觉得 AI 只是个聊天工具，但到现在，它已经能帮人执行很多任务，越来越像 Agent。像 CodeBuddy 这类产品，甚至已经可以让人直接用自然语言开始写代码。这件事带来的情绪是很复杂的：一方面，大家会兴奋，因为 AI 这个本质上的效率工具，确实能替代很多重复性劳动；另一方面，也会带来一种现实感，就是很多过去看起来必须靠专业技能才能完成的事情，现在正在被重写。

他接着说，自己现在也是 CodeBuddy 的用户，也尝试过 Web Coding，让它帮自己实现过一些过去只是幻想中的产品点子。以前你可能只是想象“如果有一天自己创业，做出一款产品会怎样”，现在居然真的能体验到那种创造的快乐，而且这种创造是从无到有、被真正做出来的。他还提到，对方现在已经把自己的 App 做到 3.2 版本了，所以他特别好奇：作为一个完全零基础的人，到底是怎么迈出第一步的？

主持人把普通人的心理障碍说得很直白：很多人天然会给自己贴标签，说“我是文科生，我不懂代码；代码很难；我很佩服会写代码的人，但我自己肯定不会写”。再加上大多数 IDE 的界面本来就不够用户友好，普通人一看到那种满屏面板和代码区，感觉就像看天书一样。所以他真正想问的是，在这种心理门槛和工具门槛都很高的前提下，一个零基础用户最开始到底是怎么跨出去的。

画面说明

这一帧是主持人低头看着电脑继续追问，字幕落在“当然我现在我也是 CoBody 的用户”上，视觉上很像一个已经亲自试过产品的人在把体验往前推进。镜头中的笔记本、话筒和主持人的提问姿态，共同强化了这一段的立场：这里不是抽象讨论 AI 趋势，而是在追问一个更贴近普通观众的问题，为什么今年开始连非技术用户都明显感到 AI 不只是会聊天，而是真的能帮人动手做产品。

09. 第一步不是 IDE，而是先把需求说清楚（12:08–13:38）



场景 9

完整内容

嘉宾回答说，自己迈出第一步的方式其实非常简单，刚开始甚至根本没有用 IDE，就是直接在聊天窗口里告诉 AI：我想要一个什么样的应用。她做的那个应用，是用来收集杜比全景声音乐的。因为杜比全景声音乐分散在很多地方，她从 2024 年就开始收集这类内容。

她回忆，最早那套方法非常简陋。比如在苹果的平台上看到一首歌有杜比全景声， she 就把歌名、表演者、链接这些信息一点一点复制下来，贴到类似腾讯文档或者 Excel 表格那样的地方。后来这样硬生生复制了 7000 多首歌，规模是有了，但问题也马上暴露出来。她把这个文档发给别人以后，收到的反馈五花八门：不能搜、打开很慢、手机上看不了。一个有 7000 多行的文档发给别人，本来就很难打开，这也很正常。

她说，第一版本质上只是把那个 Excel 式的数据表做成了一个网页，但随着数据越来越多，网站反而越来越慢。你收集得越多，它越打不开。也正因为这样，她开始想，自己应该把它真正做成一个产品。传统做法当然也想过，比如找外包；但找外包的时候，不是对方不理解她的想法，就是她自己很

难把需求表达清楚。

嘉宾特别强调，那种感觉其实很多人都能理解：你最早并不是站在产品经理或者工程师的角度思考，而是站在用户角度，知道自己想要一个什么东西，却很难把它精确描述出来。主持人接话说，这个需求之所以能被理解，本质上正是因为 Web Coding 对应的是一种真实的用户诉求：你的自然语言本身，如果带着清楚的逻辑，其实就已经是一种很好的“规格说明”。

画面说明

镜头里嘉宾讲到“不能搜啊”这一类真实反馈，字幕也正好停在这句，画面把抽象的“零基础做产品”立刻拉回到具体痛点上。观众看到的不是炫技，而是一个人先用最笨的方法手工维护 7000 多首歌的数据，再因为“搜不了、太慢、打不开”被逼着往前走。这个视觉和内容的结合，让“先用自然语言把需求说清楚”显得特别可信，因为她一开始面对的根本不是代码问题，而是一个已经被自己反复验证过的真实使用问题。

10. 外包说不清，AI 反而接得住（13:38–15:10）



场景 10

完整内容

主持人顺着她的话总结说，其实当你已经能把自己想要的东西一二三四讲出来的时候，逻辑判断本身就已经有了，剩下无非是把这套意图映射到代码层面。嘉宾回应说，自己最开始并不是只想把那 7000 多首歌简单展示出来，她一直最抗拒的就是继续手工整理，所以首先想做的，是把整个数据库变成一个自动化抓取的系统。

她说，自己其实很清楚这套逻辑是什么。比如扫到一首歌以后，这首歌有哪些特性，要不要把它收录进来，收录以后如何查找、鉴定、保存，这些流程她脑子里都是有的。但她跟外包沟通时，经常会把自己想得更远的东西也一起讲进去，比如“基于这个库以后还能继续做什么”，结果就把事情说乱了，对方反而更难理解。

后来她干脆换了个方法：把原本打算发给外包的那套需求，直接丢给 AI，问它“我想开发一个这样的东西，你觉得可不可以”。她说这个过程其实挺好玩的。最开始 AI 给她的是一套 Next.js 之类的代码，但放到本地怎么都跑不起来。如果那时候放弃，也就真的停在那里了。后来虽然勉强跑通了，但

网页还会莫名跳转，各种地方都很麻烦。

她当时就对 AI 说，这个技术栈跟自己八字不合，能不能换一个自己更熟悉的语法。因为她是做媒体出身，平时会接触 CMS 后台、HTML、CSS 这类东西，所以至少对这套前端表达方式更熟。AI 后来就按这个方向重新帮她生成，后面的东西一下子就顺了。于是她第一版数据库的整个搭建，基本都是基于这套更容易上手的技术路径完成的。她也承认，像 Next.js 这种方案本身有一定门槛，而且随着项目越做越深，门槛可能还会继续抬高；但对于个人来说，只要能先找到自己能跑通的起点，就已经足够关键。

画面说明

截图里嘉宾正抬手往前比划，字幕写着“就交给 AI 说”，这个动作和文案很贴合她这一段的转折：原本要交给外包的需求，最后改成直接交给 AI。画面传达出的不是“AI 自动搞定一切”，而是一个非技术用户不断试错、不断换技术路径的过程。她不是凭空学会了所有工程知识，而是先找到一个自己熟悉的表达层，比如 CMS、HTML、CSS，再让 AI 配合自己，这正是这段观点最关键的落点。

11. 从复制粘贴到开始需要工程化管理（15:10–16:40）



场景 11

完整内容

嘉宾接着说，一个人只要有这种逻辑上的认知，其实就已经可以去指导 AI；而 AI 在不断帮你做东西的过程中，也会反过来带着你往前学。她觉得这件事很有意思。自己最开始付费用了两个月，后来就不太想掏钱了，转而去找一些“薅羊毛”的办法继续尝试。

她回忆，那时候很多能力还是跑在云上的，但云上部署对她来说门槛仍然很高，她不会把东西部署到云端。后来是有了明确需求，才一点点去学。她举例说，原来像 GitHub Card 一类的工具，也可以帮人做一些流程控制的事情。早期最麻烦的是，AI 经常在聊天窗口里把代码传给她，她再把整套代码复制、粘贴、打包。有时候 AI 只给一些片段，她根本不知道该复制到哪里；只要一复制就报错。她后来就直接跟 AI 说：你别给我碎片，给我完整版本。

她说，这样来来回回，自己大概积累了将近 40 个对话。随着项目变复杂，她已经不再只有一个文件，而是开始出现多个文件。每次一看到单个文件太长，她就让 AI 继续拆，拆成几个小文件。慢慢地，这个东西就真的有了一点“工程”的样子。

到这个阶段，她的诉求也变了：她不想再每次都靠复制粘贴去修改一小块，而是希望有一个工具能一次性理解自己整个代码库。主持人听完后总结得很形象，说她做软件的第一阶段，其实有点像在“拼乐高”一样，懵懵懂懂地开始碰到软件工程。嘉宾则说，自己当时完全没有“工程意识”这个概念，只是因为真实需求逼着自己往前提升。结果无意之间，你就已经登高了一步：刚开始只是一个文件里不停加东西，后来就自然需要一个工具来管理整个项目，于是也就走到了 IDE 这一步。她提到，那大概是 5 月份左右，刚好外界也在讨论 OpenAI 要收购 Windsurf 之类的消息。

画面说明

这一帧字幕写着“我现在有多个文件”，画面里的嘉宾一边用手比划，一边在描述项目结构开始变复杂的那个瞬间。这个视觉信息非常关键，因为它把“从聊天窗口玩一玩”与“开始进入工程化阶段”清楚地区分开了。观众能直观看到，她的问题已经不再是单条 prompt 写不写得好，而是文件变多、上下文变长、复制粘贴容易出错，这也正好解释了为什么一个人会从简单对话工具，进一步走向需要 IDE 来管理整个项目。

12. 看见 IDE 之后，才知道项目可以整体被 AI 理解（16:40–18:10）



场景 12

完整内容

嘉宾说，自己当时正好看到相关新闻，才第一次意识到：原来世界上还有这样一类工具。那时候她连 VS Code、JetBrains 这些名字都不知道，后来也是从 Windsurf 才知道，原来可以有一个界面把自己所有代码一次性都看到。她先下载了 Windsurf，用过之后又听别人说还有 Cursor，于是也去下载。刚开始她觉得特别爽，很快就付费了；但两个月以后又有点舍不得继续花钱，于是又去找别的替代方式。

她补充说，那时候一开始很多能力还是云端的，自己不太会用；但后来出现了 VS Code 插件，她第一时间就装上了。她觉得那段时间挺有意思，因为在做一个新项目时，AI 已经可以一次性帮她搭出一整套东西。她非常喜欢那种“会生成很多文件”的感觉：你只要给一个需求，说现在要做一个什么项目，它就会先把项目骨架搭起来。

她举例，自己原本抓取的是一个内容项目，后来又派生出新的需求，想再做一个后台管理系统，类似自己在媒体工作中熟悉的 CMS、WordPress 那种管理后台。她把这个需求交给 AI，AI 就开始往这个方向继续生成。她觉得后来的迭代节奏很快，工具也在不断更新，甚至移动端 iOS 上也开始出现相关

版本。

她提到，自己从 0 到 1 做小程序，大概是在 5 月底开始，6 月 9 日上线。那段时间从 0 到 1 的过程中，她一直主要在用 Cursor；5 月、6 月基本都是它，到了 7 月、8 月之后，又逐渐开始用 CodeBuddy，以及其他一些工具，一路用到现在。主持人最后追问：在这一整段使用过程中，你最大的感受到底是什么？如果要向现在还没尝试过 Web Coding 的听众和观众形容，你会怎么描述？

画面说明

画面仍然是嘉宾在稳定输出，字幕落在“我说我现在要做一个什么”这句上，很贴合她对 IDE 价值的理解变化。前一阶段她还在和单个文件、局部代码打交道；到了这一段，视觉上的平稳讲述配合内容中的“能一次看到所有代码”“能一次性生成很多文件”，让观众更容易理解她真正被打动的点不是某个炫目的功能，而是 AI 开始能够站在整个项目层面理解和搭建东西，这才让零基础用户第一次感到自己在“做软件”而不是“拼碎片”。

13. 从自然语言生代码，走到描述生成应用（18:10–19:40）



场景 13

完整内容

主持人追问 Jason：当你真的看到别人用这些软件，把你一开始预想的东西做出来时，是什么感觉？

Jason 说，那种感觉确实很爽。虽然最早的小程序版本现在看起来比后来的 App 简陋很多，但当时他把自己脑海里的想法描述给外包，对方最后做出来的小程序，已经和他设想中的样子有 99% 相似，这件事给了他很强的正反馈。中间不断调整、调教当然很痛苦，但一旦这个东西真的成型，而且你已经把从 0 到 1 的过程走通过一次，后面再加功能，很多时候就变成一句话的事情。

Jason 接着说，其实自己当时没有想那么多，也没有预料到今天会发展得这么快。那时候他心里最早出现的，并不是“自然语言直接生成一个完整应用”这个概念，而是“我能不能通过自然语言，快速生成应用里的代码”。在他看来，这相当于从过去“代码补全代码”，往“自然语言生成代码”迈了一步。

至于再进一步，直接通过自然语言生成整个应用，他当时并没有真正达到那个阶段。因为他看过这类能力，但直觉上觉得太难了。难点不在前端表层，而是在后端、在架构，在于整个软件系统是不是都能按照自己的设想被准确搭起来。但随着各类大模型能力以几何级速度提升，他慢慢意识到，这件事其实已经开始变得可以实现了，也就是“用描述来生成应用”这条路，到现在为止已经不是空想。

画面说明

画面里是 Jason 靠在椅背上讲话，字幕正好落在“我其实那时候根本就没有 AI”这一句附近，和这段回顾性的语气很一致。这一帧没有展示任何产品界面，而是把注意力放回讲述者本人，反而更能突出他想表达的核心：最开始他并不是带着完整的技术路线图去做产品，而是在不断试错中，一点点从“把想法讲清楚”走到“让系统按描述生成代码”，再意识到“连应用本身也许都可以被描述出来”。画面中的放松姿态也和内容形成对照，说明这不是一次炫技式展示，而是事后回望时对技术跃迁的确认。

14. 调错最痛苦，尤其是在上线前崩掉（19:40–21:10）



场景 14

完整内容

王博元接着解释，如果过去没有 WorkBuddy、没有现在这一类 Web Coding 产品形态，要把同样的事情做出来，时间成本会高很多。因为你得先有一个程序员专家来理解你的工程；更麻烦的是，一旦你遇到问题，还得进入调错阶段，而调错往往是最花时间、也最痛苦的部分。

主持人顺势追问：“调错”到底是什么概念，能不能用更大白话一点的方式讲讲？于是王博元开始讲自己在做 iOS 时踩过的坑。他说，第一版上线前有两周时间，自己几乎都快要放弃了。项目从 12 月开始做，一直到 1 月中旬之前其实都还挺顺，整体生成速度很快，iOS 版本也是沿着类似的方法一路做下来的，一开始非常顺利。

问题出在某一个准备上线前的版本。到了那个节点，程序开始疯狂报错、崩溃。不是那种偶发性的闪退，而是每次都会出错，并且 100% 可以复现；更折磨人的是，每次报出来的错误还都不一样。王博元说，自己刚开始觉得这也没什么，报错就直接丢给 AI 处理就好了。结果 AI 每次都会说“我已经给

你修复了”，但现实情况要么是它根本没修好，要么确实修了，却修得很不优雅，和他想要的完全不是一个东西。

真正让他绝望的，不只是 bug 本身，而是他那时候已经不知道问题到底出在哪。他原本觉得自己一直沿用做小程序时那套方法论，整体都很顺，为什么偏偏走到这个节点就卡死了。后来每一次报错都让他有点心理阴影，甚至不敢轻易退出重新点开，因为每点一次，心里就会“咯噔”一下。那两周非常痛苦，而且他原本还想赶在年前上线，所以压力更大。

画面说明

这一帧切到王博元正低头回忆，字幕停在“在某一个准备上线前的一个版本”，与他讲述危机爆发的时间点完全重合。镜头里他手势向下、视线集中，能强化那种进入排障细节的压迫感。和前面谈“生成很快、很爽”的轻快节奏不同，这一幕把观众拉进真实开发里最不浪漫的一面：不是做不出来，而是快上线时开始稳定复现地崩。画面与内容结合后，能更直观地说明为什么他会说“调错是最痛苦的”，因为真正消耗人的不是一个单点错误，而是连续、反复、每次都像刚看到新问题一样的心理折磨。

15. AI 修不好时，先跳出框架再回来（21:10-22:41）



场景 15

完整内容

王博元继续说，那段时间可能是自己做 AI Coding 以来最耗时、也最疯狂啃问题的一次。后来他又修了几天，最后不是单纯靠 AI 自动修出来的，而是他自己先想出了一个办法，再把这个办法告诉 AI，让 AI 按他的思路去改。

他回头复盘时觉得，最有感触的一点就是，人还是得参与思考。后来再看，真正的问题出在 iOS 里的 Navigation 被他写得太乱了。页面之间本来应该有明确的层级和返回逻辑，但当时他自己都没想清楚，不知道一个操作到底应该退回哪一层页面。他说，自己那时候其实是把 App 当成网页在做，而不是按原生 App 的页面栈逻辑去组织。

于是他换了一个思路去描述问题。他直接告诉 AI：“我现在不是在做一个 App，我现在是在做一个浏览器。你现在看到的所有页面，都是网页。你只需要先把这些网页之间的逻辑，复原到我的预期。” AI 在这个理解下，相当于先给整个东西套了一个壳。壳套上之后，之前最严重的崩溃问题就被绕开了。

他还提到，当时自己把版本推到 TestFlight，一共可能也就 100 个测试者，但一天之内能打出 600 多次崩溃。看到那个数字的时候，他自己都快崩溃了，马上意识到这个版本肯定不能上线。可是在改用刚才那套思路之后，程序就再也没有崩溃过。

这件事让他意识到，自己在用 AI 的时候，也得学会跳出原来的框架。某种意义上，这也是他自己的成长过程：一开始完全依赖 AI，希望它直接把问题改好；后来发现它即便能改，也常常会用一种自己不接受、也不优雅的方式去改；再后来，他干脆先停下来，休息几天，从“陷进问题里”的状态退出来，重新用人的大脑思考，再回来指挥 AI 执行。

画面说明

画面还是王博元在讲，但字幕已经切到“我去那个 TestFlight 推送了一版”，这正好对应他把问题具体量化的瞬间。镜头中的稳定近景和他讲出的“100 个测试者、一天 600 多次崩溃”形成鲜明反差，让观众感受到：真正可怕的不是抽象地说“有 bug”，而是上线前已经能看到一串冷冰冰的数据在证明这个版本绝对不能发。也正因为这张画面停在他复述关键转折的节点，Visual explainer 才更能带出这段观点的重点：AI 不是万能修复器，真正的突破来自人先重构问题定义，再让 AI 按新的认知框架去执行。

16. 和 AI 共生，不是把大脑外包给它（22:41-24:13）



场景 16

完整内容

主持人顺着上一段总结说，这件事很有意思：你先用自己的方法想出解法，再把方法告诉 AI，最后问题才真正解决。他觉得这正好对应了一个更大的问题，也就是在今天这个时代，人到底应该如何和 AI 共生。“共生”这个词，在他看来比别的说法更准确。

他提到，最近很多人都在聊 AI 带来的焦虑。硬币的一面当然是焦虑，是信息量过大，是技术发展太快，让人的脑子过载，甚至来不及消化“这到底是一个什么时代”。但另一种处理方式是，先停下来休息一下，认真梳理：它现在到底发展到了哪一步？我擅长什么？它又擅长什么？有没有一些过去我不敢靠近、或者我不擅长的事情，现在反而可以借助 AI 去实现？

为了说明这个意思，主持人举了一个自己的例子。他说，自己以前一直有个梦想，就是想写出像方文山那样的歌词。可你真的去看，会发现方文山很多词不一定非常华丽，很多时候甚至像大白话，所以你会误以为“写词也没有那么难”。但等到你自己真去写一种那样风格的歌词时，往往写出来就会发现完全不是那么回事，甚至会因此否定自己，怀疑是不是自己不行。

而在今天这个阶段，他觉得情况已经不同了。无论是作词还是作曲，应该都已经有了软件能帮你推进这件事。你可以像做 Web Coding 一样，用语言去描述自己的想法，让系统先给你生成一些旋律或草稿；然后你再基于这些结果继续调优，调整词、改旋律、改方向，而不是一上来就要求自己单枪匹马做全部。

画面说明

画面切到主持人正对着麦克风和电脑发言，字幕停在“其实我想王老师这个例子”，视觉上很像他正在把上一位嘉宾的具体经历抽象成一个更大的方法论。这一帧最重要的不是人物动作，而是场景关系：主持人面前有电脑，但他讲的核心并不是工具参数，而是“人先想，AI 再做”的协作方式。也因此，这个镜头和内容结合后，会让观众更容易把前面那个 iOS 崩溃案例理解成一个普适结论：AI 能大幅放大执行力，但前提是人要先把自己的判断、目标和审美拿回来。

17. 人的创造力，才是 AI 时代的动差（24:13–25:46）



场景 17

完整内容

主持人接着说，硬币的另一面，其实不是焦虑，也不是“被取代”这么简单，而是人类主观能动性的创造力开始变得更重要。就像王博元刚才休息了几分钟、跳出问题之后重新想办法那样，真正起作用的是人的创造性判断先发挥了作用，然后再让 AI 去高效、专业地把它执行出来。

在主持人看来，这可能正是 AI 时代真正重要的另一面。无论是 Jason 提到国内还没有形成很好的那种“coding for fun”的氛围，还是王博元这种愿意自己动手、边学边试的状态，本质上都是一种“动差”。他认为，这种动差力，也就是你愿不愿意主动去探索、去试、去定义问题、去提出不同可能性，也许会成为未来 AI 时代的一项核心竞争力。

他还说，自己其实很喜欢 CodeBuddy 或 WorkBuddy 里这个“Buddy”的叫法，因为那表达的是一种伙伴关系。回到去年的时候，很多人都在讨论，是不是未来只要一句话就能做一个 App。可主持人现在回头看，会觉得更准确的理解不是“我把一个任务扔给 AI，它自己完成”，而是“我和 AI 一起承担这件事”。

他把这个过程形容成，一个刚开始接触软件工程的小孩，一点点学习如何表达需求、如何理解反馈、如何把事情推进下去。在这个过程中，AI 确实帮助你完成了大量具体操作和落地动作，但与此同时，你自己的认知也在被迫成长。也就是说，你不是站在一旁看 AI 干活，而是在和它共同承担这个任务，只不过它承担的是执行，你承担的是判断、方向和理解。

画面说明

这一帧主持人抬头看向对谈对象，字幕落在“可能是在未来”上，正好是他把个体经验上升到时代判断的时刻。镜头里他的身体姿态比前一段更打开，也更像是在做总结式表达，这和内容里的“硬币的另一面”“核心竞争力”高度一致。画面没有任何炫目的技术演示，却能帮助观众抓住一个关键转向：AI 时代的价值不只在于谁会下命令，更在于谁能提出值得执行的想法，并持续修正问题定义。也正因如此，“Buddy”这个画面里看不见、但被语言不断强调的关系，才显得比“替代”更贴近现实。

18. AI 不会让文科消失，反而会放大审美和表达（25:46–27:17）



场景 18

完整内容

主持人最后把讨论进一步推开。他说，如果真的要因为 AI 而焦虑，那不如先打开 CodeBuddy、打开 WorkBuddy，迈出第一步，不要害怕失败。现在有了 Web Coding，只要一个人具备基本文化水平，也有一定语言组织能力，能把自己想做的东西表达清楚，他就已经能开始做一些过去做不了的事。

当然，他也强调，未来依旧会出现那些在 coding 上非常有天赋的人。AI 并不会抹平所有差异。只是到了未来，互联网从业者，尤其是开发者，不会再只局限于传统意义上的理科生、科班程序员。

他借此回应了前段时间一个很大的讨论：在 AI 时代，文科会不会消失？有些人对此很悲观，觉得 AI 发展这么快，做 AI、做开发才是唯一的风口，而 AI 又能替代掉很多重复性的工作，包括文书处理、基础写作、标准化劳动，那么文科生是不是会成批失业？再加上 AI 现在已经能进行文学创作，哪怕很多时候还有明显的“AI 味”，但它毕竟已经开始做了，所以这种担忧很容易出现。

但主持人的判断恰恰相反：AI 反而会给文科生更多机会。因为真正让一个东西变得“好”、变得“美”、变得更精致的能力，依然重要。他提到，乔布斯过去反复强调过，技术和人文的结合才会产生真正伟大的产品。就他看来，AI 现在最擅长的是执行，能把很多事情更快地做出来；但一个东西最后是不是足够好、是不是足够美、是不是有更细腻的功用，仍然离不开人的判断。

他也补充说，自己看过很多 Web Coding 应用，拿到手以后仍然会觉得有些粗糙。这并不是在苛责某个具体产品，他甚至主动自嘲说，自己的应用也很粗糙。可这恰恰说明，未来的差异化空间不在于“能不能做出来”，而在于谁能把它做得更顺、更准、更有审美和更贴近人。

画面说明

镜头里主持人继续对着麦克风输出，字幕停在“AI 又能替代掉很多这种重复性”，和这段关于“文科不会消失”的讨论直接对上。因为画面本身仍然是一个非常克制的播客场景，所以观众不会被视觉噱头带偏，反而更容易把注意力放在论证结构上：一方面，AI 的确会替代重复执行；另一方面，越是当执行门槛被拉低，表达、审美、判断和对人性的理解就越值钱。这个朴素画面和这段观点形成了很好的支撑关系，让“AI 时代文科不但不会消失，反而可能迎来新机会”听起来不是一句口号，而是从实际工具体验里推导出来的结论。

19. 从鸡娃到未来教育提问（27:17-28:49）



场景 19

完整内容

主持人先补充了自己对 AI 使用体验的一层期待：他当然希望 AI 能帮人高效解决问题，但他也尽量希望自己在使用它的时候，始终保留一种“精致感”。在他看来，未来的孩子一定会把“用 AI 去解决一件事情的能力”当成一种基础能力，但除此之外，更重要的还是他们怎么去欣赏这个世界，什么是美的，什么是好的。

接着主持人把话题停下来，正式抛给两位嘉宾。他说，两位一位是资深的 AI 从业人员，一位是已经深度受益的 AI 使用者。今天大家不管是在讨论 00 后就业、80 后就业，还是讨论“AI 会不会取代人的岗位”，本质上都指向同一个问题：如果未来 10 年、20 年 AI 的发展已经超出今天人的想象，那到那个时代，孩子们正好处在要创造价值的人生阶段。那以今天的认知出发，面对未来，培养孩子到底应该更注重什么？

主持人先交代了一段自己的观察。他经历过教培特别火的那段时间，虽然那时候他自己还没有孩子，但他看到身边同事、所谓“海淀妈妈”等群体，都在疯狂卷课外班。大家卷学习成绩，卷各种比赛、证书，最终目标就是上好学校，像“985、211”，再往远一点甚至是去美国藤校。这是五年之前一种非常典型的育儿风气。但后来大家也都看到，这种高压路径其实带来了很多问题，甚至很多悲剧。

画面说明

这一帧里是主持人正面出镜，身体前倾、手势抬起，画面停在“已经无法用今天的我们的认知去想象的时候”这句字幕上，和这一段的功能完全一致：他不是在给答案，而是在把讨论从个人使用 AI 的感受，转向下一代怎么面对 AI 时代的长期命题。镜头的停顿感也强化了这个转场意义，像是在把“鸡娃”旧逻辑和“未来能力”新问题明确摆到桌面上。

20. 曾经看重英语和编程（28:49–30:21）



场景 20

完整内容

主持人接着把自己的思考说得更具体。他说，后来大家发现，很多孩子最先出现的不是“成绩不够好”，而是不快乐，接着会出现各种心理问题，导致童年留下阴影。所以如果有一天他自己有孩子，他真正会思考的是：希望孩子过一个什么样的人，经历一个什么样的童年，以及未来能自由选择成为什么样的人。

他原来比较明确地想过两件事。第一，是希望孩子能比较早地具备英语能力。他把语言看成一种工具，不管未来从事哪个行业，多一种语言就多一种选择的余地。第二，是希望能比较早地培养孩子的代码能力。原因不是为了当程序员本身，而是他一直把“会写代码”理解成一种强逻辑能力的体现，而逻辑能力在任何行业、任何人生阶段里，理论上都会让人长期受益。

所以在那时，他以为如果自己有了孩子，这两件事会是他愿意投入大量精力去培养的方向。但当时间迅速拨到 2026 年，再回头看这两个判断时，他发现自己的想法已经变了。第一个方向，他开始不那么确定；第二个方向，在当下看来很可能已经部分失效。最早他甚至认真想过，要不要在孩子一二年

级时就报一些少儿编程班，也看过很多宣传，说孩子会喜欢编程，还能做出一些具身智能的小玩具、小作品，看起来很有意思。如果孩子有兴趣，他本来也是愿意为此付费的。但站在现在这个时间点，他觉得这件事的价值密度已经没有过去想象得那么高了。

画面说明

画面里主持人侧头思考，字幕正好落在“我觉得逻辑能力这个对于他未来的一生”上，这让镜头有一种明显的“复盘”意味。这一段不是简单谈育儿技巧，而是在展示一个成年人如何因为 AI 进展太快，主动修正自己原本笃定的教育投资判断。视觉上单人近景也把这种自我反思放大了。

21. Jason谈AI时代的孩子成长（30:21-31:52）



场景 21

完整内容

主持人把自己的这段五六年思考转变先抛出来，想听两位嘉宾怎么看。随后嘉宾 Jason 接过话题，先肯定这是个很好的命题。他说，从“AI 在成长阶段或者说科技向善能做什么”这个角度看，有两个场景尤其重要，一个是教育，另一个是 AI 健康。至于一些更偏噱头、偏表面的应用，当然也可能有价值，但他先把那部分撇开，重点谈真正和下一代成长相关的逻辑。

Jason 认为，对未来十年、二十年的孩子来说，最重要的底层思路之一，是让他们能够更自由地成长。不是家长替孩子规定死一条路，而是帮助他们逐渐知道自己想要什么，再沿着这个方向往下走。家长可以做的，是给他们提供一些导向性的、好玩的东西，帮助他们接触世界、形成偏好，而不是过早把孩子塞进单一标准答案里。

他接着提到另一个关键点：智能体这件事，已经不只是产品层面的热词，而是落进国家大方向里的事情。未来一定会出现中国芯片、中国硬件、中国软件支撑下的一整套智能体生态，它将成为国力象征的一部分，也会逐步在各个行业里铺开。等今天的小孩子成长二十年后，这些东西会是他们生活环境

的一部分。也正因为如此，“好好学习”的意义会发生变化。不是为了和 AI 比谁做题快、谁记忆强，而是比什么？比你的知识体系是否完整，比你的世界理解是否清楚，因为只有这样才能更好地理解 AI，也让 AI 更好地理解你。

Jason 进一步强调，AI 的本质不再只是一个“快入快出”的问答工具，而会变成一个长期陪伴、长期记忆的智能体。比如今天一些产品已经能保存多轮对话、逐渐理解用户。顺着这个方向往下想，未来几乎每个人都会有一个长期陪伴自己的智能体，而这种关系会从很早的人生阶段就开始形成。

画面说明

这一帧切到 Jason 出境，字幕是“然后还有个点就是”，正好对应他在系统化展开答案的状态。和前两段主持人的个人困惑不同，这里的镜头语言更像“给出框架”：Jason 一边打手势一边分点展开，把教育、健康、智能体、知识体系这些层次串到一起，强调孩子未来不是跟 AI 对抗，而是和长期陪伴的 AI 共存。

22. 长期陪伴式AI如何进入童年（31:52-33:22）



场景 22

完整内容

Jason 继续往下展开。他说，这种长期陪伴式智能体的到来，甚至都不用等二十年那么久，可能一两年内就会非常明显。未来孩子们的童年，某种程度上会更像我们小时候在动画片里看到的那种生活，只不过这次会以更具象、也更能落地的方式出现。

他举了一个学习场景的例子。未来孩子可能会拥有一个智能学习机，而这台学习机里不只是简单装了课程内容，而是包含了孩子整个学科体系里的学习轨迹：做过哪些题、错过哪些题、在哪些知识点上反复出问题。AI 不需要让孩子每天重复做那些已经会了的题，而是会有针对性地抓住那些他总做错、总没吃透的部分，帮他更快掌握学习要点。这样一来，学习过程会更个性化，效率也会非常高。

他也特别强调，不是所有孩子都必须沿着同一种“学习型”路径前进。如果有一类小朋友就是不太愿意在传统意义上学习，也没有关系。关键是要知道、要洞察他内心真正感兴趣的东西。有些孩子不喜欢长时间钻那些比较枯燥的文科内容，更喜欢摸索理科；有些孩子喜欢看书、看小说；还有些孩子可能

喜欢视频、运动、技巧训练。AI 都可以在这些方向上给予支持，比如给他推合适的视频、辅助他理解某种技巧，或者在户外运动场景里记录健康数据、心跳过程等等。换句话说，AI 可以在方方面面成为一种秩序化、陪伴式的支持力量。

画面说明

镜头仍然停在 Jason 身上，字幕落在“那么你要知道洞察他自己内心的东西”，这正是这一段最关键的句子。画面没有切换到任何炫技式演示，反而持续停留在人脸和手势上，强调的是一种育儿判断：AI 真正的价值，不是把所有孩子训练成同一种“高效学习机器”，而是帮助家长更早识别孩子差异、再做个性化陪伴。

23. 别为AI焦虑，拿它做探索工具（33:22-34:54）



场景 23

完整内容

主持人顺着 Jason 的回答继续追问。他说，对于普通家长来说，大家现在的家庭条件总体上可能比以前更好一些，也确实愿意在培养孩子这件事上投入时间和金钱。那么问题就变成了：如果真的要投入，具体应该往哪些方向投入？

Jason 的回答是，宏观上的培养方向其实不需要因为 AI 时代到来就发生剧烈变化。你原来觉得怎么培养是合理的，现在大体上还是可以怎么培养，没有必要为了“AI 来了”这件事就提前做一套过度焦虑的布局。真正的变化，不在于彻底改写育儿目标，而是在培养过程中，把 AI 变成一个帮助思考和帮助决策的工具。

他说，如果你还不知道孩子想选什么、兴趣信号还不清晰，那就可以有意识地引导他：现在你已经处在 AI 时代，也已经有了不错的工具，能不能用这些工具去更快接近你喜欢的信息，或者用它来测试一些你自己也还没挖出来的兴趣？试一下也没关系，不需要太大成本，也不必因为结果不确定而焦虑。

Jason 最后把话说得很明确：AI 的发展最终一定是为了造福人类和改善生活。至于“替代人类”这件事，不需要被吓住。只要你是一个接受过完整教育体系的人，具备清晰的逻辑、比较完善的语言表达能力，那在未来大概率依然能借助 AI 获得收入、获得机会。AI 应该被理解成“给你更多机会”的东西。只要你自己有想法、有憧憬、有知识结构，也有逻辑能力，想做什么事就往前做就可以了。AI 并没有改变人类社会的根本动力，它更像是在这个情境里突然多出了一整个工具池。Jason 用《哆啦A梦》打了个比方：不是说 AI 自己就是哆啦A梦，而是你现在多了一个装满各种工具的口袋。

画面说明

这一帧又切回主持人，字幕是“对你来说不需要有任何焦虑”，正好把这一段从提问推进到一个面向普通家长的结论。镜头回到主持人，有收束和确认的作用：前面 Jason 讲了很多未来图景，这里把它翻译成现实可执行的态度，就是别恐慌，也别把 AI 当宿命，而是把它当成孩子探索兴趣和扩展能力的工具箱。

24. 工具不会替代聪明的人 (34:54-36:25)



场景 24

完整内容

主持人接着把刚才那个《哆啦A梦》的比喻再往前推了一步。他说，口袋里确实可以有无数工具，但问题在于，不能指望主角自己都不知道要什么，就跑到工具堆里乱翻。人必须先有想法，先能准确说出自己的需求，哆啦A梦这样的“管家”才能帮你把对应工具拿出来，实现目标。所以真正关键的，不是工具本身，而是你有没有能力提出准确的问题、识别自己的痛点。主持人甚至开玩笑说，如果真按动画片里的逻辑去看，最聪明的其实不是工具，而是那个能精准提出需求的人。也正因为如此，工具替代不了人，反而是人越聪明、越会使用工具，机会就越多。

接着主持人把话递给王老师。王老师回应说，他想起一个朋友讲过的故事：19 世纪的伦敦曾经有一种很体面的职业，就是每天晚上到街上去点灯。后来有了电，这个职业自然就消失了，变成历史。但旧职业消失的同时，新的职业总会出现，去替代、去接续旧的社会分工。他说，自己现在用 AI 的感觉就是，它几乎每个月都在迭代一次，速度快得惊人。

王老师还补充，像刚才 Jason 提到的这种“终身陪伴型 agent”，从技术上看其实已经没有什么本质障碍了。所以未来孩子会面对什么样的世界，他自己也未必能完整预测，但他愿意和孩子一起去经历这个变化的过程。不过，如果回到“孩子该重点培养什么”这个问题上，他仍然觉得，自己最看重的可能还是“美育”。因为无论技术怎么变，那些愿意动手、愿意创造、能感受美和表达美的人，始终更有可能把工具变成真正有价值的东西。

画面说明

这一段画面前半部分是主持人继续解释“工具口袋”的比喻，后半部分切到王老师出镜，脸上带笑、语气放松，但说的是职业消失与新职业诞生这样很硬的判断。这个切换很关键：它把抽象的“AI 是工具”落到更现实的社会变化上，同时又把讨论从“会不会失业”重新拉回“人要培养什么底层能力”，最终落在美育和创造力这类更不容易被替代的部分。

25. 从岗位焦虑到试错资本 (36:25-37:57)



场景 25

完整内容

这一段先是嘉宾顺着前面的讨论，回看另一个人的成长路径：很多人其实一直有某种潜在能力，只是长期没有被激活。像 coding 这条路，过去可能离很多人很远，但当 web coding 这件事开始真正成立之后，原来那些潜能就突然找到了一个很契合的出口。

主持人随后接过来，说自己特别理解这种感觉。人一旦长期打工，很容易打出一种惯性，甚至会把自己锁死在原有职业身份里。他坦白说，自己曾经一度幻想，如果有一天不做今天这个岗位了，好像只能去路边卖蛋炒饭，因为自己做蛋炒饭还挺好吃的。前几年社会上一直在讲“35 岁焦虑”、裁员之类的话题，几乎每个人都会不可避免地想过：如果真有那一天，我还能做什么？

但主持人说，今天听完这两位嘉宾的故事之后，尤其是听完 Jason 对未来的描述，他反而没那么焦虑了。因为他意识到，很多人过去之所以焦虑，不是因为真的没有能力，而是看不到新的能力出口。现在 AI 和 web coding 至少提供了一条新的可能性。

接着话题转到更现实的心态上。嘉宾说，自己其实一直没有那种特别宏大的愿景，不是那种“我一定要做成什么惊天动地事情”的驱动型人格。他更在意的是先把自己手上的东西做好，做到自己满意；当这个东西先让自己满意了，如果后来有人看到、有人用了，也觉得挺好，那就让它自然生长、自然发展。所以他一直偏爱“薅羊毛”式的务实心态，不会一上来就追求最贵、最大、最豪华的那一套。

主持人由此联想到一个很有意思的话题：以前大家总在讨论父母应该怎么培养孩子，但从当下这个技术环境看，也许不如更现实一点。与其试图替孩子规划一条唯一正确的路，不如给孩子留下一笔足够他去试错、去买 token、去用 AI 工具探索可能性的资源。主持人觉得，这也许反而是这个时代一种更具体、也更有操作性的“留下什么给孩子”的方式。

画面说明

这一帧是嘉宾近景，镜头压得比较实，字幕停在“对这件事情”上，视觉上像是在认真回应一种长期存在的职业焦虑。画面本身没有展示任何技术产品，反而强化了这一段的重点不在“炫技”，而在心态转换：人先承认自己会焦虑、会被岗位惯性困住，然后才可能把 AI 和 web coding 看成新的试错空间。也正因为是这种面对面访谈镜头，这段关于“给孩子留下试错资本”的观点显得更像经验判断，而不是抽象口号。

26. 未来分享的是内容，不是应用（37:57-39:28）



场景 26

完整内容

主持人顺着刚才那个话题继续追问：如果今天父母真要给孩子留一点什么，是不是甚至可以具体到留一笔专门拿来买 token、拿来使用 AI 的预算？他说这个话题很有意思，但也想把问题重新拉回产品本身。今天聊了很多关于“用产品连接人”的故事，从早期产品到现在的 WorkBuddy，再到它目前在行业里的阶段性领先地位，他想知道，站在这个基础上，未来会不会诞生一个新的生态？以及，最早的 1.0 愿景是不是已经实现了，接下来 2.0 的愿景又会是什么？

嘉宾回应说，如果往未来看，他自己的一个判断是：传统意义上那个“生命周期很长的 App”，以后可能不会再像过去那样长期存在。也就是说，未来的逻辑不一定还是“我要先做一个应用，再让用户进入这个应用”。

他举了一个很具体的例子来说明自己的想法。比如此刻他只是想听一首歌，那我真正想要的并不是某个音乐应用本身，而是“我当下就要这首歌”。我甚至可能希望这首歌是根据我当前情绪、当前场景，现场生成一个曲、一个词，然后立刻得到内容本身。换句话说，用户要的不是载体，而是结果；要的

不是 App，而是 App 里面产出的那份内容。

所以在他看来，接下来被生产出来的重点，不会只是“一个应用”，而会越来越多地变成“应用所生成的内容”。而这些内容，本身就是 AI 创作出来的内容。也正因为如此，他非常在意产品产物是否能被直接看见、直接分享。无论是某个功能、某个结果，还是某个页面，他都希望别人进来之后不用经过复杂路径，就能直接看到、直接使用。

他进一步解释说，功能本身不应该被深埋在“先点这里、再注册登录、再做一堆操作”之后。真正重要的是，功能和产物都能快速触达。别人先看到的是你做出来的产物，然后再因为这个产物回流到功能本身，进入你的产品内部。换句话说，传播路径会从“先装应用，再理解功能”，转成“先看到结果，再反向进入系统”。

画面说明

这一帧切到主持人侧身发问，字幕是“这个逻辑下”，和内容正好对上：他在把话题从家庭、教育、焦虑，重新拉回到产品未来形态的判断。画面里的麦克风和固定访谈机位，形成一种很传统的内容生产场景，但嘉宾给出的观点恰好是在拆这个旧范式：未来重要的不是把人带进 App，而是让内容和功能本身能被直接消费、直接传播。正因为镜头如此“传统”，这段对“应用不再是中心”的判断显得更鲜明。

27. 产品会更小，CLI 会更重要（39:28-40:59）



场景 27

完整内容

嘉宾继续往下推他的判断：未来的产品形态一定会变得更“小”、更“短”。这里说的小和短，不只是界面轻量，而是每一个应用内部的独立能力，都应该可以单独被摘出来，被分享、被调用、被使用。

他认为，AI 在这里像一个中空弹层，或者说一个中间调度层。过去我们把很多功能都包裹在 App 这个壳子里，但未来每个功能都可以脱离原来的完整应用单独存在。你不一定非要先理解整个产品、先下载整个 App，才能用到其中一个能力。

沿着这个逻辑，他进一步提出一个很强的判断：未来每一个 App 背后，其实都应该有一个对应的 CLI。像 WorkBuddy 这种原本就和 CLI 关系很近的产品当然更不用说，但哪怕是传统软件，背后也都应该有一套可被调用的 CLI 能力。

不过他也强调，CLI 不是目的，CLI 本身不是为了让普通用户直接拿来用，而是为了让大模型更容易理解你的产品、理解你的能力边界。只有当产品能力被结构化、可调用化之后，AI 在调度这些能力的时候，才不需要一定经过“打开 App”这一步。App 最大的意义，最终会降层到“让人参与进来、进行交互”的界面；而更底层的真正价值，是你背后如何被 AI 调用、被 AI 调度。

主持人接着补充说，他前几天和朋友聊天时也谈到这个问题。今天我们之所以还习惯于看到一排排图标、本能地点击屏幕，其实是因为我们仍然处在“手和屏幕直接触控”的交互时代，这是一种建立在手机形态上的交互方式。可如果有一天手机本身消失了呢？如果未来变成头戴设备，或者别的形态，那今天这些图标和入口也许都不再需要维持原样。

嘉宾顺势回顾了手机演进史。他提到，从非智能手机、按键手机到智能手机，其实交互形态一直在变化。乔布斯当年非常坚持小屏，背后有他对交互方式的理解；但在乔布斯去世之后，库克还是把手机越做越大，那更多是市场导向。但在他看来，大屏和小屏本身未必有本质差别，人并不一定真的需要那么多设备和复杂分层，很多东西未来都可以继续被压缩、被简化。

画面说明

这一帧是主持人出镜，前面摆着电脑，字幕是“跟朋友聊的时候也聊到了”，很适合承接这段关于交互形态变化的讨论。电脑、麦克风、图标式设备，本来都代表今天这套成熟的数字工作流；但这段话恰恰在说，这些外显形态未必是终局。画面里越是能看见传统设备，越能反衬嘉宾的核心观点：真正重要的将不是 App 壳子，而是底层能力能否拆解出来、被 CLI 和 AI 调用。

28. 未来的设备会退到背景里（40:59-42:30）



场景 28

完整内容

主持人接着总结说，所以未来的一切大概率都会变得更简单。甚至再往后看，那个设备本身可能都不重要了，可能最后只是一个麦克风，或者某种极简硬件，核心目的都是解决 AI 在不同硬件设备之间的调用和协同问题。

他说，如果是在过去，听到这样的描述，他会觉得这是一种偏科幻的场景；但以今年年初到现在的技术进展来看，自己已经不再把它当成单纯畅想了。它更像是在描述一个已经逐渐能落地的现实。

嘉宾也接话说，这其实已经不是纯粹的“未来想象”，更像是一份正在被实现的 PRD，甚至已经有了原型。换句话说，很多过去听上去像概念的视频、会议、办公方式，现在技术上都已经不难做到。

主持人顺势把这个判断放回他们自己正在做的内容生产里。他说，也许未来像今天这样的访谈，根本不需要这么重的设备阵列，但人与人之间依然可以面对面地交流。甚至在电脑端，他们已经在尝试做一两周“无人办公”的视频：三台电脑放在那里，人实际上不一定都在现场，但通过技术手段，最终呈

现出来的状态却像三个人坐在一起完成了一场真实对谈。

他还补充了一个案例：之前有位创始人在发力某个项目的时候，其实并没有来到现场，但依然是通过类似方式参与录制和交流的。这让他意识到，空间上的在场，未来未必还要依赖传统意义上的物理同场。

最后，主持人把话题重新收束到听众的焦虑上。他坦白说，自己在录这个节目时也有焦虑，焦虑的点在于：这件事会不会很难聊？自己该怎么把“其实我们可以不焦虑”这件事，准确传达给听众？但今天听下来，无论是 Jason 还是王老师的故事，其实都已经给了很清楚的答案。硬币的另一面是：如果最坏的结果真的发生了，AI 真的开始取代某些岗位，那么我们已经来不及沉浸在悲伤里，而是应该立刻去学习 AI、学习 web coding、去寻找能和自己现有能力结合起来的新机会。

画面说明

这一帧主持人一边说话一边比划，字幕直接出现“DeepSeek 那个创始人”，说明讨论已经从抽象技术趋势落到了现实案例。画面仍然是标准播客录制现场，但内容却在不断削弱“必须同地、必须重设备、必须大阵仗”这些旧前提。也就是说，视觉上的专业录制布景，正好反衬了这一段的核心判断：未来设备会越来越退到背景里，真正被放大的将是人与人之间的连接，以及 AI 对这种连接方式的重构。

29. 别停在抱怨，先换个角度（42:30–44:00）



场景 29

完整内容

主持人继续把那个“被 AI 替代”的问题往下拆。他说，如果过去你是做文秘的、做律师的，或者做别的专业工作的人，难道一旦 AI 进入这些领域，你就必须彻底放弃上一份职业，然后从头去找一个全新的兴趣点吗？

他的答案并不是这样。真正值得马上切换的，不是职业本身，而是思考角度。你完全可以反过来想：能不能把你原来这份职业积累下来的能力、兴趣和经验，通过 AI、通过 web coding、通过 WorkBuddy 这类产品，重新组合成一份新的收入来源？也就是说，未必是“抛弃过去”，而更可能是“把过去的能力借助新工具重新变现”。

主持人强调，这种角度切换可以现在立刻发生，而不是继续停留在“AI 怎么把我取代了”这样的抱怨里。因为抱怨并不能解决任何问题，真正有用的是去重新组织自己的能力结构，看哪些东西能和新工具结合起来。

他说，今天这一场其实已经聊了很多了，所以他想回到节目录制最初的那个目的：不是单纯谈技术概念，而是尽量给那些还在害怕、观望、担心被 AI“淘汰”的听众，一点真正能缓解焦虑的东西。无论是 70 后、80 后、90 后、00 后，还是现在仍然不太清楚 AI 到底发展到什么程度的人，都可以先从一个更务实的认知开始。

嘉宾接过这个总结说，大家今天之所以恐慌，很多时候恐慌的是“概念”，焦虑的也是“概念”。真正重要的是把概念拨开，看本质。AI 本质上是在帮你解决问题，只不过市场和行业会用各种概念包装它，好让大家更容易理解，或者在技术圈里再叠加很多方法论和策略，让大模型更听你的话。

所以他说，普通人其实不需要特别焦虑那些术语，比如 Humanism、Prompt Engineering、AI Native 这些词。从产品从业者的角度，他当然会讲这些概念；但如果面向非开发人群，更好的理解方式其实是：我们正在努力打造一个像哆啦 A 梦一样、听你话、懂你的智能助手。也正因为如此，他一直想做的一个核心方向，其实就是“分身”。

画面说明

这一帧主持人正对镜头，字幕是“仍旧对 AI 发展到什么程度”，和这一段想安抚观望者的目标完全贴合。镜头语言很直接，没有切到任何产品演示，而是让说话的人正面承担“解释者”的角色。这样处理很有效，因为这里讨论的重点不是某个工具的功能细节，而是心态转变：不要停在被取代的叙事里，而要把职业经验、兴趣和 AI 工具重新组合，变成下一阶段的能力。

30. 把 AI 当成你的分身与伙伴（44:00–45:31）



场景 30

完整内容

嘉宾继续解释他为什么一直在想“分身”这个方向。他说，每个人其实都希望自己拥有一套属于自己的分身系统。这个愿望背后，来自人性里一个很朴素的欲望：事情太多了，信息太多了，我们总会希望有一个真正懂你、能替你分担、能延伸你能力边界的东西存在。

而这个“懂你”的意义，不只是替你做几件琐事，更是让你的肉体从重复负担里部分脱离出来，把更多时间和精力留给真正值得探索的事情。因为从他的理解看，人本来就应该有探索精神。技术存在的意义，不是把人困住，而是把人从噪声和负载里解放出来。

他进一步指出，很多人的焦虑，其实来自双重信息负载的重叠。一层是今天人类社会本来就已经非常庞杂的信息量，另一层是 AI 世界又不断冒出新的信息量、概念和声音。两层叠加之后，人就会产生强烈恐慌：不知道该学什么，不知道该往哪里去，也不知道该怎么筛选出真正能落地的重点。

所以他给大家的总结非常明确：AI 是来帮你减负的。你永远要记住一件事，就是拨开复杂的事情去看本质。你的生活里本来就有很多噪声，AI 世界里也有很多噪声，但你要尽量回到自己最真实的本源上。

他说，第一，你得先想清楚自己到底想做什么，你未来的人生规划是什么。第二，你要想清楚，怎么把自己现在的喜好和能力结合起来，顺势而为地往前走。第三，如果你对未来不确定，那就让 AI 帮你去敲一敲、试一试那些更确定的事情，让它帮助你更快读懂自己现在这条路径是不是适合继续走下去。

最后他的结论是：接下来一定要把 AI 当成伙伴，而不是抗拒它。因为你一旦抗拒，现实里的噪声并不会减少，AI 世界里的各种概念还是会不断涌来，别人也还是会天天和你讨论这些东西。与其抗拒，不如主动拥抱，然后继续拨开表面去看本质。你可以把 AI 理解成一种“我的分身”，只是现在产品层面上，这个分身还在不同层次里逐步演化，还没有出现一个所有人都公认的最终形态。如果暂时没有找到最适合你的那个产品，也没关系，你就继续走自己的路、做自己的事，同时把这件事看清楚。你也可以用 WorkBuddy 这类产品去理解各种产品词汇、让它帮你做总结，但最终最重要的，仍然是你自己的理解力。

画面说明

这一帧又回到主持人近景，字幕是“结合你现在的喜好”，但整段内容实际上已经进入全场收束。画面里的人物、麦克风、电脑都很稳定，没有大动作，这种稳定感和嘉宾最后的判断相互支撑：AI 不是来制造更多混乱，而是来帮人减负、帮人厘清方向。也因为镜头始终落在人本身而不是技术界面上，这段“把 AI 当成分身和伙伴”的结论不会显得像产品宣传，而更像一套关于如何与技术共处的生活方法。

31. 顺势而为，先理解再拥抱 AI (45:31-47:01)



场景 31

完整内容

嘉宾把话题拉回到更底层的成长能力：面对复杂变化之后，人的情绪能不能保持稳定，能不能控制好自己，并且按照自己认定的事情坚定地往下走；自己的喜好能不能被认真挖掘出来；当你碰到一个看似危险的东西时，能不能看到它最重要的底层原理。这些能力，才是最重要的事情。

他认为，对 AI 有“不摸不碰”的心态很正常，尤其面对这样一种未知而且带着革命性的技术。对他个人来说，AI 可能是这一生接触到的、对自己改变最大的一种技术。它之后一定会持续带来很多变化，甚至会改变整个社会结构，包括人怎么工作、怎么沟通，很多东西都会被重塑。

但他的态度还是“顺势而为”。就像他现在在做 App，未来自己的 App 也许会被更“蠢”的 App 取代，他现在也觉得没有关系。因为他做的东西也可能更像一个手工制品，里面带着很强的个人印记。比如他过去是文字工作者，那他的 App 里也许就会有大量叙事性的内容，或者很多小细节，这些都会成为它的特点。

他最后的判断是：你看的東西越多、讀的東西越多，對 AI 能做什么就會有更清晰的認知。所以如果你害怕被淘汰，不如先擁抱它，打開 WorkBuddy，然後去聊一聊。

畫面說明

畫面里是嘉賓正面坐在麥克風前，語氣明顯偏沉穩，不是在煽動“趕緊追風口”，而是在講一種更長期的能力建設。這和這一段的核心觀點是對得上的：AI 會帶來巨變，但真正決定一個人能不能不被浪潮裹挾的，不是短期情緒，而是情緒穩定、自我判斷、抓底層規律，以及把個人經驗沉澱進作品里的能力。鏡頭的靜態感，也強化了他所說的“順勢而為”不是被動接受，而是冷靜看清變化後主動調整自己的做法。

32. 把焦虑变成一份新技能（47:01-48:31）



场景 32

完整内容

嘉宾继续往下说：你先看看自己有什么想法，能不能让 AI 帮你把它实现。退一万步来说，哪怕你以后不从事现在这份职业了，你的第二份履历写什么？不如写“我用 AI 开发了一个应用”。他觉得，这句话无论怎么说，都会让你的简历比那些没有做过这件事的人更领先。

他举例说，像王老师在不到一年的时间里，已经从一个完全的文科生，慢慢走到今天。以前他不懂那些术语，什么“光栅”“管线”之类，现在当然也谈不上能够完整重部署一遍，很多事情依旧很难，但至少已经比一年前的自己拥有了更多技能。这本身就是资本。

另一位接话说，这也正好能化解焦虑，因为你确实拥有了更多能力。随后主持人也补充自己的状态：他最近一直在想自己数据库的 performance 总觉得不太好，所以正在比较不同数据库结构之间到底有什么差异；这个过程对他自己也是一种提升。遇到什么问题，就去想解决办法；解决不了，就去问。

主持人接着转向总结：刚才两位其实讲了很多思考角度，而他的看法是，我们 80% 的焦虑都来自我们接收到的信息。这些信息包括层出不穷的行业新闻、重大的技术突破，以及被高权重媒体不断放大的各种叙事。

画面说明

这一帧切到主持人，他面前摆着电脑，字幕正落在“拥有了一些多的技能”上，和内容形成了直接呼应。前半段在讲“哪怕只是试着用 AI 做出一个应用，也会让你的履历更有分量”，后半段主持人又用自己研究数据库性能的例子说明，真正能对抗焦虑的不是空想，而是把新问题变成新能力。镜头里的电脑、麦克风和对谈场景，也把“学习 AI 不是抽象概念，而是具体动手过程”这层意思落得很实。

33. 先用起来，再评价它（48:31-50:04）



场景 33

完整内容

主持人继续说，那种铺天盖地的报道，恰恰是很多焦虑的来源。这些东西都来自“听”和“看”，而我们的焦虑也正是因为只停留在听和看上，没有迈出“做”的那一步。

他说，假如自己真的去下载一个前段时间很流行、现在也可以直接下载来用、具备 agent 能力的产品，比如 WorkBuddy 这种东西，不管是先用免费的版本，还是一开始要付出一点成本，通常也不会太贵。哪怕只是十几块、几十块一个月，先去尝试一下，先看看它到底是个什么东西，这件事都比任何心灵安抚更重要。

他的观点很明确：你先看看它到底是什么玩意儿，之后你再决定怎么利用它，或者你再去否定它、骂它，都来得及；但第一步一定是先用起来。

他还打了个比方：就像没有吃过螃蟹的人，在吃之前会觉得螃蟹像怪物；但当你真的吃了之后，才知道原来这是一种食物。怪物和食物之间的区别，很多时候只差你鼓起勇气把它放进嘴里。

他说，而且现在 AI 的“情商”其实都挺高的。前几天他让 WorkBuddy 帮自己写一个小程序，对方还会说：“你的这个想法，是一个非常成熟的产品经理的思考，你过去是有过这样的从业经历吗？”他说，这种话谁听了都会开心。

画面说明

镜头仍然停在主持人这一侧，画面没有花哨变化，但这正好服务于他的论点：别把 AI 神秘化，先把它当一个工具打开、试用、对话、协作。画面中的主持人像是在做一次非常日常的经验分享，而不是发表宏大判断，这和他反复强调的“先下载、先付一点小成本、先亲自试试”完全一致。后面提到 AI 会夸人，也让这个静态播客画面多了一层轻松感，说明 AI 降低的不只是技术门槛，也降低了普通人开始尝试时的心理门槛。

34. 把 AI 当成第一次完整造物的入口 (50:04-51:24)



场景 34

完整内容

主持人顺着上一段笑着说，谁听到那种夸奖，心里不得乐开花，美滋滋的，都会忍不住想“难道我也有这些天赋吗”。他接着把这个感受放到更大的社会经验里：尤其是在东亚家庭的教育环境中，很多孩子其实是成长在一个不断被否定的环境里的。所以如果大家有过这样的、不太好的童年经历，反而更应该多和 AI 对话。

他随后收束整场聊天：今天其实聊了很多，但并没有一个所谓特别明确的单一主题。说到底，是从他们公司基于 WorkBuddy 这个产品的经验出发，聊到了现在 WorkBuddy 已经具备很多 agent 能力，不只是能做 web coding，也能帮你处理很多任务等等。

再往大一点说，他们聊的是最近四五年整个 AI 爆发史，以及它给每个普通人、包括行业从业者带来的一些颠覆性认知，还有大家对未来的恐慌。

最后他表达了一个很直接的期待：希望今天这场围绕公司 WorkBuddy、把它作为一个代表性 AI 产品来展开的讨论，能够给大家带来一些新的认知。也欢迎大家去下载 WorkBuddy，开启一次非常完整的、利用 AI 进行造物的体验；并且欢迎大家在小宇宙、喜马拉雅、荔枝 FM、bilibili、腾讯视频、微博等平台关注和收听他们。

画面说明

这一帧仍然是主持人的总结镜头，字幕停在“整个 AI 爆发史”，很像节目进入收尾时的提炼与回看。前面他说 AI 会给人正反馈，能缓解很多长期被否定的人在表达和尝试上的心理负担；后面又把这种个人体验，提升到“普通人如何进入 AI 时代”的产品入口问题。画面上的主持人、电脑和播客录制环境共同构成了一种“邀请你亲自开始”的氛围：不是只谈趋势，而是把 WorkBuddy 包装成一个让普通人第一次完整使用 AI 去造物的起点。