

第 1 章

设计师为什么 要懂技术

“设计师要不要学技术？”“设计师应该怎么学技术？”“设计师怎么学写代码？”相信这类问题会不时出现在设计师的脑海里。笔者将从三个角度讲述设计师为什么需要了解技术。

1.1 原因一：行业的改变和进步

1.1.1 海外懂技术的设计师好像有点多

如果读者平时关注 Google、Meta、Apple 等海外公司的设计招聘广告，不难发现有些招聘信息会提及设计师需要懂得 HTML5、CSS3、JavaScript 等技术；如果读者平时会浏览一些海外设计师的个人博客，可能会发现他们的网站采用了大量精湛的动画效果。这时候大家脑海可能会出现一个想法：为什么海外设计师好像都懂技术？

其实海外设计师懂得网站和应用开发并不是一件罕见的事情，因为海外教育更多以实践为主，近几年流行的 STEM（Science, Technology, Engineering 与 Mathematics 的缩写，指科学、技术、工程、数学四门学科）教育是一个很好的例子。在欧美很多大学的设计专业会涉及一部分技术课程和作业，例如网站开发、应用开发、新媒体技术开发等，为了实现自己的想法和拿到毕业证书，学生不得不自学掌握更多的技术来完成毕业作品。所以，海外懂技术的设计师确实比较多。

1.1.2 我们期待你是全链路设计师

2017 年中国互联网突然涌现的“全链路设计”“全栈设计”“大设计”等名词引起了设计师的热议，这些名词的诞生意味着设计师应该掌握商业分析、界面设计、交互设计、动效设计、运营设计和技术实现等能力，成为多专多能的人才。

在此之前，互联网大厂的设计师分工相当明确，交互设计师、UI 设计师、视觉设计师各司其职，突如其来的“全链路设计”让一部分设计师手足无措。

有些设计师可能认为交互设计就是画画线框图，其实不然。交互设计是产品设计和技术实现的桥梁，如果交互设计师画的线框图无法被开发人员实现，那么返稿是必需的，所以交互设计需要较强的逻辑和综合能力，这导致部分 UI 设计师转型时会遇到一定的挑战。另外，动效设计是很多设计师和开发人员的“噩梦”，因为开发人员看到一个复杂的完整动效时脑海里不知道怎么把它实现，而不懂开发实现的设计师更不知道怎么和开发人员阐述自己的想法，设计师与开发人员沟通就如鸡同鸭讲，所以如何把一个复杂的动效设计拆解成可被实现的代码同样需要设计师拥有较好的技术能力。

1.1.3 互联网泡沫、新一代设计师和人工智能同时带来的挑战

2015—2022 年，整个互联网行业饱受资本寒冬和互联网泡沫的影响，在此期间同类竞品的同质化越来越高，整个互联网行业不再需要这么多设计师，有些领导还希望两三个设计师的活可以由一个设计师完成。但是，非互联网行业的从业人员又因为互联网行业工资高一直想加入，一堆质量参差不齐的设计培训班快速培养了一群能力参差不齐的设计师，并通过各种方式将他们推荐到各个公司，导致整个互联网设计行业供大于求，就业环境更加恶劣。

除此之外，设计师同时迎来了同辈和“后浪”的冲击。美国近年来减少了 H-1B 签证的发放，加上 2020 年新型冠状病毒感染疫情的全球暴发，使得部分综合能力较强的在海外发展的设计人才和留学生回流中国。与此同时，一出生就成长在互联网时代的“95 后”也逐渐步入社会，这一代新晋设计师在校期间涉及 Web、Arduino、Processing 等课程的学习，综合能力得到较好的锻炼，所谓“长江后浪推前浪”，这一现象也加剧了互联网行业的就业竞争。

2016 年起，人工智能技术逐渐火爆全球。例如 2017 年 Adobe 推出的 Adobe Sensei 极大降低了 Photoshop、Premiere 等编辑软件的使用难度，同时推出了搜图功能；阿里巴巴推出的基于图像智能生成技术的鹿班系统改变了传统的 Banner 设计流程，输入相关的素材和想要的风格、尺寸，鹿班就能代替设计师完成素材分析、抠图、配色等耗时耗力的设计制作，迅速生成多套符合要求的设计解决方案。那时起互联网出现了一波热烈的讨论：人工智能会不会取代设计师？

由于以上原因，部分设计师将目光转向当年在微信朋友圈、抖音流行的 Python、Java 等技术培训班广告，为了建立自己的求职优势以及和其他设计师产生差异化，开始了自学技术生涯。

1.1.4 AR/VR、车联网、智能硬件等新赛道的出现

随着人工智能和新型人机交互技术逐渐成熟，AR/VR、车联网、智能硬件等领域在不断壮大。这些新领域的设计和传统互联网设计是有明显区别的。每一次技术革新必定引起新一轮设计的颠覆，所以不懂技术的设计师很难把握设计的“度”。

绝大部分的互联网产品基于一个屏幕的图形界面进行设计，用户通过鼠标、键盘和触摸屏等设备和产品交互。AR/VR 的交互界面涉及三维空间，此时基于二维界面的鼠标、键盘和触摸屏如何沿用成为研究重点。另外，基于手势和语音的多模交互成为新的交互方式，然而多模交互暂时没有明确的设计规范，因此需要设计师更多地探索和实践。

同样，座舱的设计也是基于空间的设计，除了多模交互，仪表盘、中控屏和 AR-HUD（Augmented Reality Head Up Display，增强现实抬头显示）等多个屏幕如何联动是一个全新且重要的设计问题，因为它会直接影响驾驶员在动态驾驶过程中的注意力和工作负荷。和互联网设计不一样的是，在座舱交互中人机工程和心理学将贯穿整个座舱的体验设计，其中基于驾驶员的疲劳 / 分神检测、基于驾驶任务的环境及汽车检测等技术和驾驶 / 非驾驶过程中的体验设计息息相关。以上同样没有明确的设计规范，需要设计师结合场景和技术做更多的探索和实践。

智能硬件不一定有屏幕，所以基于图形界面的设计经验和理论基础不一定适用于智能硬件设计。此外，由不同技术组合成的智能硬件有着不一样的设计限制和体验差异。以智能座舱的疲劳检测为例，为什么绝大部分的 DMS（Driver Monitoring System，疲劳驾驶预警系统）采用红外摄像头而不是普通的 RGB 摄像头？因为 RGB 摄像头在夜间或者通过隧道以及阴暗路面时识别率骤降，而红外摄像头可以有效避免该问题，并且能避免驾驶员眼镜反光或者佩戴墨镜的影响，这些都是 RGB 摄像头很难做到的。如果设计师不知道这些技术细节，无论怎样设计也无法实现良好的疲劳检测效果。

基于以上介绍，相信读者能大概了解技术对于新领域的设计影响有多大，这也导致了想拥抱新领域的互联网设计师出现水土不服的现象。如果要解决该问题，最简单、最有效的方法是多了解技术原理和细节，成为一名跨学科、一专多能的设计师。

1.1.5 政策对于设计师的影响

2018年我国第一本面向中学生的人工智能教材——《人工智能基础(高中版)》正式出版，人工智能相关课程进入了浙江、上海等地区的高中课堂；2019年北京、浙江和山东确定把Python编程基础纳入信息技术课程和高考的内容体系，山东省2019年出版的小学信息技术六年级教材也引入了Python内容；2021年全国已有数百所高校开设了人工智能专业……这意味着未来的设计师在校期间已经拥有一定的编程能力和技术基础。

2021年发布的《中华人民共和国国民经济和社会发展第十四个五年规划和2035年远景目标纲要》指出，国家将对人工智能、量子信息、脑科学等前沿领域实施一批具有前瞻性、战略性的国家重大科技项目；同时国家正在推进产学研深度融合，支持企业牵头组建创新联合体，承担国家重大科技项目。“十四五”规划意味着国家正在大力推动科技创新，未来需要越来越多的跨学科人才成为推动创新的关键角色，设计师也应该成为这样的关键角色。

1.2 原因二：技术和设计之间有着密切的关系

相信读者对图1-1并不陌生，用户体验由设计、技术和商业决定。笔者认为，优秀的用户体验最重要的是在特定场景下解决用户痛点或者满足用户需求，而设计师的工作是通过各种方式解决问题实现该目标。解决问题的方法有很多，重点在于设计师能不能识别到关键路径和设计出解决方案，尤其是在VR/AR、智能家居和机器人等处于“从0到1”的新兴领域，这时候会用到很多新型技术，例如多模交互、空间交互、传感设备、智能织物等，设计师了解技术的先进性和局限性对于设计和创新变得非常重要。

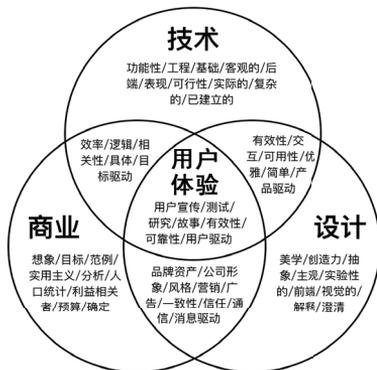


图 1-1 用户体验的组成

1.2.1 技术、设计和创新之间的关系

在 *Incremental and Radical Innovation: Design Research vs. Technology and Meaning Change* 一文中，Donald Norman 和 Roberto Verganti 提出了两种创新方式，它们分别是渐进式创新和激进式创新，如图 1-2 所示。以用户为中心的设计属于渐进式创新的一种，它就跟从 A 爬到 B 或者 C 爬到 D 一样，尽管它能有效让产品做出改进，但容易陷入局部的最大值，因为“登山者”在爬山过程中无法知道这座山的背后是否存在其他更高的山丘，例如位于 A 的登山者只知道 B 的存在，却不知道 D 的存在。激进式创新由意义变革和技术变革驱动，例如家庭电灯、汽车和飞机、广播和电视都属于激进式创新。和渐进式创新不同的是，每一个激进式创新都是在没有对个人甚至社会需求进行仔细分析的情况下完成的，但它能让“登山者”发现更高的山丘或者直接创造更高的山丘，就如同位于 B 的登山者发现了 C 和 D 一样。

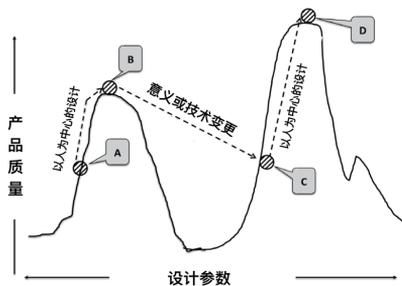


图 1-2 渐进式创新和激进式创新的区别

什么样的创新才算是激进式创新？文章中提及了 Kristina Dahlin 和 Dean Behrens 提出的三个标准：

标准 1：发明必须是新颖的，它需要和以前的发明不同。

标准 2：发明必须是独一无二的，它需要和当前的发明不同。

标准 3：发明必须被采用，它需要影响未来发明的内容。

标准 1 和标准 2 定义了创新的激进性，它们可以随时发生；标准 3 定义了创新需要是成功的，它只有在社会、市场和文化等因素协调时才会发生，简单理解就是实现天时、地利、人和。文章中提及了成功的激进式创新可能每 5 ~ 10 年发生一次，在错误的时间即使有正确的想法也会失败，例如 Apple 在 20 世纪 90 年代初期推出的 QuickTake 数码相机和 Newton 个人数字助理，它们尽管满足了标准 1 和标准 2，但因为市场上失败了，所以并不满足标准 3，导致它们不能被认定为激进式创新。

文章里提到，完全新颖的创新是不可能的，它们更多来自前人的想法和工作。有些是通过改进前人的工作，例如爱迪生没有发明灯泡但延长了灯泡寿命；还有就是通过将几个已有想法创新组合，例如基于触控和手势的 iPhone。多点触控系统在计算机和设计实验室已经有 30 多年的历史，手势识别技术也有悠久的历史，苹果公司既没有发明多点触控界面，也没有发明手势控制，但它确实改变了人类的生活习惯。所以研究前人的工作内容有助于探索新的创新方向，为设计奠定更扎实的基础。

上文提到激进式创新由意义变革和技术变革驱动，在图 1-3 中，Norman 从技术和意义两个维度定义了四种不同类型的创新，它们分别是：

技术推动式创新：创新来自技术的根本性变革，而不是基于用户的需求和痛点，所以产品的含义没有任何变化，彩色电视机的发明（在现有的黑白电视机之上）就是一个很好的例子。

技术顿悟：创新来自新技术的出现或者现有技术在全新环境中的使用。例如，Wii 视频游戏机属于技术顿悟创新，它使用的新技术和新意义从根本上改变了电子游戏的空间。

市场拉动式创新：创新来自对用户需求的分析并开发出满足用户需求的产品。文章中把以用户为中心的设计和传统的市场拉动方法都放在这里。以人为中心的设计虽然允许技术和意义的局部变化，但基本上它使产品保持在左下象限内。

意义推动式创新：创新来自对社会文化的动态理解。例如从手表作为工具到手表作为时尚配饰的转变。以 20 世纪 60 年代的迷你裙为例，它不仅仅是一条不同的裙子，还是女性自由的全新象征，标志着社会发生了根本性的变化，但没有涉及新技术。

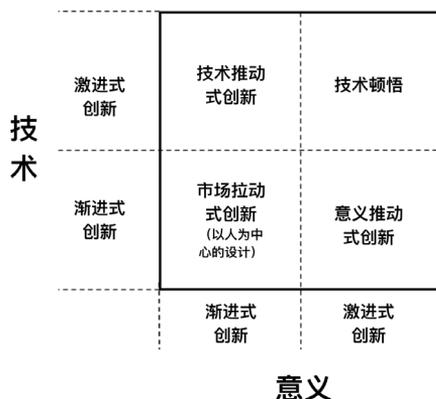


图 1-3 四种不同类型的创新

总的来说，从上述内容可以了解到技术将不断推动产品和体验的创新和进步，因此在涉及前沿项目的设计时，了解技术的先进性和局限性是设计过程中必不可少的一环，如何将技术和体验完美融合，需要更多关注用户需求和和使用场景。

1.2.2 技术、设计和用户之间的关系

在互联网公司，大部分的工作流程跟流水线没有太大差异：产品经理定义和挖掘用户场景—设计师设计解决方案—开发人员实现解决方案。这样的工作流程导致一部分从业人员认为用户场景、设计方案和技术方案是前后决定的关系。从 1.2.1 节可以了解到，新技术的诞生和成熟也可以定义新的用户场景，因此用户场景、设计方案和技术方案可以相互影响，如图 1-4 所示。

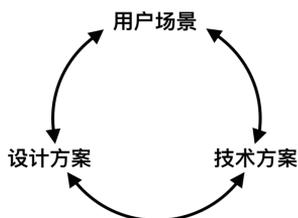


图 1-4 用户场景、设计方案和技术方案的关系

如何从技术角度推导用户场景和设计？笔者认为关键在于坚持“以用户为中心”的设计理念，同时也需要寻找技术的价值所在。以 VR 为例，为什么 VR 在 2014 年火爆之后就一直沉寂？因为这段时期的 VR 设备无法解决用户在使用过程中产生的晕眩感，这个问题主要由以下三个因素导致。

在 VR 全视角屏幕中，延迟是造成眩晕感的最大问题。假设用户的头部在 0.5 秒内向右边旋转 90 度，VR 设备也会给用户呈现右转 90 度之后的画面。然而如果 VR 设备中画面的转换需要花费 1 秒的时间，0.5 秒的时差会使用户立即产生强烈的眩晕感。经研究发现，头动和视野的延迟不能超过 20 毫秒，不然就会出现眩晕感。但是所有计算机的运算、显示处理都有一定的滞后性，延迟时间加起来不超过 20 毫秒对于 VR 设备来说是一个非常大的挑战。

眼睛看到的画面信息和前庭系统所感受到的真实位置信息不匹配，两种信息在大脑里面无法统一，也会产生晕眩感。例如坐 VR 过山车时，在视觉上玩家处于高速运动状态，但是玩家的前庭系统感觉不到自己在运动，这时会导致玩家头晕。经研究发现，通过前庭电刺激（Galvanic Vestibular Stimulation, GVS）在一定程度上可以解决该问题。前庭系统主要用于调节平衡和运动，而前庭电刺激的做法是将两个电极放在耳朵上追踪用户内耳的感知运动，并给予一定的电流刺激前庭系统，这样就能欺骗用户的大脑，让用户觉得自己的身体正在运动。

由于每个人的瞳距都不一样，对某些人来说，人眼瞳孔中心、透镜中心、画面中心三点并非一线，从而出现重影现象，看久了人也会非常容易头晕。目前大部分 VR 设备采用了滑轮调节 VR 透镜之间距离的设计，可以自由调节两个镜片之间的距离，从而避免了重影和晕眩。

从 VR 晕眩这个案例可以了解到一部分用户体验的基本需求是由人类的生理特性决定的，如果技术无法解决或满足这些基本需求，那么上层所有的设计几乎是空中楼阁，因此设计师需要提前了解当前技术是否足够成熟到解决这些问题。当技术成熟后，就要围绕技术的价值和用户需求寻找新的设计可能性。例如小米在 2020 年发布的“一指连”技术背后用的是 UWB (Ultra Wide Band, 超宽带) 技术，它可以实现厘米级定位和 $\pm 3^\circ$ 的角度测量精度，也就是说 UWB 的技术价值在于赋予手机和智能设备空间感知能力，就像室内的 GPS。从用户视角来看，大部分用户家里都有空调、风扇、电视等电器，如何远程便捷操纵它们成为用户的需求和痛点。因此小米利用 UWB 技术设计出手机对准智能设备即可操纵设备的交互

体验，省去了以往复杂的连接过程，实现了小米宣传文档里说的“智能生活，从未如此简单”。

从以上案例可以了解到，技术也可以影响用户需求，推导出设计方案。接下来将介绍在设计过程中哪些地方可以结合技术一起考虑，这样在高效产出设计方案的同时还能降低和开发人员的沟通成本。

1.2.3 自然无感交互的背后都是技术和设计的融合

基于场景的驱动，计算机为用户提供更直接、有效的服务成为行业探索的新方向，这也使得设计行业中一直倡导的“最好的设计就是没有设计”成为可能。这句话该怎么理解？智能产品会逐渐融入用户的生活和场景中，通过了解用户的行为和上下文动态改变自己的交互策略来适应用户。如何感知用户行为以及上下文都需要技术的支持，当技术无法实现百分之百准确判断时，为用户带来的是帮助还是困扰？明显后者的影响会更大。

以手势识别为例，当用户发起了一个执行手势，但是设备因为光线以及距离的原因误识别为另外一个手势，那么后续的交互流程都是错误的，因为它明显不符合用户的意图。因此，设计师需要基于技术的限制设计一套拥有容错空间的设计，此外还需要将当前场景拆解成一个个参数供技术理解，从而帮助开发人员提升整个手势识别技术的准确率。因此，基于场景驱动的设计需要设计师懂得技术和设计的融合。

除了基于场景和人工智能驱动的设计外，人机交互设计、AR/VR设计、3D渲染设计，甚至常见的动效设计都与技术有着强关联，如果是从事相关行业的读者肯定会留意到这些设计并不是那么好复制，因为它们的设计策略都是隐形在界面后，和技术原理、参数化设计绑定在一起。仍以手势识别为例，如果用户A正在用手势操纵电视，这时用户B走到电视前面也尝试用手势操纵电视，那么电视应该执行用户A的操作还是用户B的操作？如果讲先后顺序那么系统应该只执行用户A的操作而忽略用户B，但手势识别技术是无法实现该交互策略的，这时应该引入什么样的技术？这项技术对于算力是否有影响？还有什么情况下应该释放用户A的操控权？这些细节都需要设计、场景、技术和参数之间的配合，只有设计得好，设计才是自然的。

1.3 原因三：设计过程中需要考虑技术

在产品开发迭代过程中，设计师的工作职责一般包含前期调研、方案设计、设计评审和开发跟进。如果想让自己的设计稿在设计评审时一稿通过，除了理解需求，掌握以用户为中心或者以业务为中心的设计方法外，设计方案是否能被实现也是很重要的。上述每一个设计步骤都和技术息息相关，下面笔者将对此一一讲解。

1.3.1 前期调研

在互联网产品设计中，设计师在设计方案前一般都会做竞品调研和分析，但是设计师大多都是围绕着产品本身进行分析，包括功能、界面、交互设计等，也就是对竞品看得见的部分进行分析。其实除了分析竞品看得见的部分外，还要分析看不见的部分，这在智能硬件或者人工智能领域尤其重要，设计师需要关注值得关注的体验设计背后用了什么关键技术。以共享单车为例，用手机解锁美团单车背后的原理是什么？笔者尝试分析一下整个交互流程，有以下关键点。

(1) 每辆美团单车都需要有自己独一无二的 ID 信息，当用手机扫描单车上的二维码时，云端将授权信息发送给手机。

(2) 用户通过手机蓝牙将解锁指令和授权信息发送给单车的智能锁，智能锁获得授权信息后解锁，并将解锁成功的信息发送给手机。

(3) 用户的手机将解锁成功的信息发送给云端，云端开始计费。

结合解锁、锁车等交互流程大概能判断出美团单车的智能锁包括什么硬件模块，当然也能直接上网查看别人上传的拆解报告，这样在设计新的智能单车时就能判断出需要考虑哪些技术因素。除了竞品分析，前期调研还应该分析我们的设计需要做什么？还可以做什么？这时候工业界公布的信息和专利以及学术界的研究都能帮助设计师了解技术的可行性和边界在哪里。那么，设计师应该如何做相关的调研？笔者认为有以下三点。

(1) 多看发布会。例如，每年苹果的 WWDC 和 Google 的 I/O 大会，除了主论坛还有各种分论坛，从这些论坛上能了解到苹果和 Google 的最新动态和历年的关注事项。举个例子，在 WWDC 2020 的 Design for Intelligence 分论坛上苹果设计师解释了苹果怎么利用 Shortcuts、Siri Suggestion 和 Clips 打造基于意图理解的 AI 设计，这是苹果首次公布自己关于 AI 和设计融合的想法。在 WWDC 2021 上

苹果设计师讲解了空间交互的设计规范，空间交互有可能成为苹果下一阶段的重心，从 macOS 和 iPadOS 的多屏联动以及新发布的 AirTag 也能验证该可能性。

(2) 多了解每家公司申请了什么专利。通过了解每年各大科技公司和实验室在某个领域申请的专利数量，能大概了解一项技术的发展趋势是什么样的，是开始布局、技术成熟、布局完成还是技术已经被放弃。专利检索很考验设计师的搜索技巧和工具，如果只想初步了解，可以通过免费的 Google Patents 搜索想了解的专利内容，通过标题、权利要求和保护范围能粗略知道这篇专利讲了什么；如果搜索不到可以利用 Incopat 等付费工具。

(3) 多阅读某个领域会议和期刊上的论文，例如人机交互和人因研究领域的 CHI、座舱领域的 AutoUI 等。学术界不需要过多考虑工程实现和商业落地，所以能比较自由地探索新方向，但这也意味着部分教授做的事情有可能“不靠谱”，读者在阅读论文时尤其要注意这一点。

1.3.2 方案设计

设计师结合用户需求和可行性综合考虑整体方案能有效避免设计评审时为开发人员带来的技术挑战，在后期开发跟进和设计还原时也能减少不必要的麻烦。以互联网设计为例，交互设计师和 UI 设计师在设计时需要考虑的技术细节是不同的，但又彼此有关。

交互设计师在画设计稿时重点关注的事项包括交互框架的合理性和可复用性。合理性包括交互框架和交互流程是否匹配，重点在于每个控件、页面和流程的点击、跳转逻辑是否畅通。以长页面加载数据为例，当用户进入某个页面时，App 从服务器下载数十个甚至数百个资源，App 会白屏或者卡顿一段时间。为了解决这个问题，不懂技术的设计师可能会采取分页加载的设计，也就是用户滑到页面顶部点击“加载更多”按钮或者通过手势触发“加载更多”的交互事件。但如果设计师懂技术，在数据加载这个细节上可能会有更好的解决方案，那就是采用懒加载（Lazy Loading，也叫延迟加载）策略，相关细节会在本书后续章节讲解。

可复用性更看重的是设计师是否熟悉当前平台的设计规范和实现策略。例如，自己设计的控件/组件在样式和功能上是否和当前设计已有的或者平台的控件/组件设计规范保持一致，如果不一致需要开发人员重新实现一套新的控件/组件库。在这一点上如果没做好，可能每个页面的控件和组件调用的是不同开发人员写的

库，后续维护成本相当高。现在设计行业逐渐推广的原子化设计和设计系统也是为了该目的，复用程度高的设计对于开发人员来说就是调用一段甚至一行代码，所以设计师在设计相关流程时可以提前和开发人员沟通，这样有助于减少双方的工作量和后期沟通成本。

UI 设计师在画设计稿时重点关注的是界面的可复用性和美感。可复用性包括 App 界面如何适配不同平台以及不同大小的屏幕，之前设计的控件 / 组件样式是否适用于当前的需求。设计做好适配前提是了解响应式设计该怎么做，以及将这套策略和开发人员沟通清楚，包括如何通过百分比来控制控件 / 组件的大小和间距；如何通过界面布局和交互流程将一套设计自适应计算机和手机屏幕，这些工作都需要和交互设计师以及开发人员协商一套可用的交互和技术规则。有些 UI 设计师为了保持当前界面的美感会设计出和其他界面不同的控件和组件，这对于自己、组内其他设计师以及开发人员来说都不是一件好事，因为整套设计变得不好维护。在设计时 UI 设计师可能会重新调整交互设计稿中控件 / 组件的大小和摆放位置，这时候 UI 设计师应该考虑这些元素的摆放规则是什么，是居左、居中还是居右？每个元素之间的关系是什么？元素和容器之间的层级是什么？UI 设计师在设计时可能没关注这些细节，但这些细节会影响整个界面的技术实现。因此，设计师在画稿时结合技术实现考虑设计可行性，能有效提升和上下游的沟通效率。

1.3.3 设计评审和开发跟进

设计师在和产品经理、开发人员一起评审设计方案时，开发人员一般会提出各种问题来消除方案无法实现的可能性。这时候如果设计师无法解释，开发人员的回答很有可能是“做不了”，这意味着设计师要重新改稿。“做不了”其实分 3 种情况：

（1）能做但时间不允许，这时候更多是一种博弈，如果设计师能提前预估工作量，可以和开发人员协商时间问题。

（2）开发人员一时没有想清楚技术的可行性，如果设计师了解技术细节，可以帮助开发人员解决相关问题。

（3）由于时间、技术成本高，真的无法实现，这时候设计师只能重新改稿。

设计师前期做好技术调研并在设计方案时提前考虑好各种技术问题，能有效解决前面两种情况产生的问题，避免第三种情况的发生。在设计评审时，如果设

设计师能站在开发人员的角度阐述设计方案，就有助于开发人员了解设计师的想法。久而久之，当设计方案长时间没出现技术问题，设计师和开发人员之间就能逐步建立良好的信任，这对于双方的后续合作有着较好的作用。

当设计方案进入开发流程后，设计师最害怕的是开发人员突然说“这个设计做不了”，从而重新梳理整个设计方案。为了避免该问题，设计评审时双方应该达成共识并产出纪要；但有时候确实避免不了这种情况的出现，这时候设计师最应该做的就是和技术人员坐在一起排查技术问题和风险，这考验设计师的技术水平和沟通能力。笔者在以往工作中偶尔会遇到这种现象，这时候需要了解清楚是什么原因导致做不了，是技术架构、网络还是系统底层？这些问题需要双方站在全局角度一起去思考，解决的办法很多，只是看大家有没有探索出来。如果真的没有解决方案，只能结合问题和关键点重新梳理设计方案，但这种情况笔者遇到的甚少，如果前期调研、方案设计和设计评审三个流程充分思考过设计背后应该采用哪种技术方案，就能有效避免该情况的发生。

看完本章内容，相信读者应该大概了解了设计师为什么要懂点技术。在笔者的过往工作中，发现设计师多多少少会提到希望自己能学点 HTML 和 CSS 知识，其背后大部分的需求都跟上文内容有关。设计师觉得学点 HTML 和 CSS 就能掌握技术其实是一个很片面的想法，因为 HTML 和 CSS 都算不上编程语言。HTML 中文名是超文本标记语言，它通过标签将网络上的文档格式统一，使分散的互联网资源连接为一个逻辑整体；而 CSS 中文名是层叠样式表，它是一种样式表语言，用来描述 HTML 呈现。技术也不仅仅是编程和代码，在计算机相关专业里需要掌握的知识还包括网络工程、操作系统、算法、数据库等，编程和写代码只是程序员学习和应用这些知识的手段而已。在大学里系统掌握一套技术体系都需要花 4 年的时间，设计师不要觉得学点代码就能把开发人员过去 4 年学到的知识一下子都掌握了。

那么设计师应该重点掌握哪些技术知识？笔者认为，技术原理、技术架构和计算思维才是设计师应该掌握的。就像技术架构师可以不写代码，但不能不懂这些内容。技术原理能让架构师知道这门技术是怎么运用的，它的上下限分别在哪里；技术架构则是像建房子一样，将不同的技术有序组合；而怎样搭建出根基稳但又便于拓展的技术架构归根到底在于架构师的计算思维能力。笔者希望读者读完本书后，能像技术架构师一样站在技术上层思考问题，这才是对设计师来说价值最大的地方。