

在软件工程理论中,软件开发的第一步通常是需求分析。从很多软件工程的相关书籍中都包含了标准化的需求分析方法与工具,包括用户画像、用户场景、产品用例等。然而本书并不打算采用这些高度理论化、标准化、工程化的方法来分析用户的需求,而是回到软件作为“为人服务的产品”这一本源,来探讨为什么要做产品、为谁服务、用户到底需要什么,以及软件开发者能为用户做点什么。

3.1 重视项目问题的质量

大学课程中接触的项目通常在解决两种问题:①教师给定的问题,例如使用 C 语言实现队列和栈,并模拟电梯系统的运行过程;②学生自选的问题,但问题的解决必须符合教师给定的流程,例如使用软件工程方法开发一个应用,要求必须实现标准化的需求分析、系统设计、系统实现等过程。

与第一种问题相比,第二种问题看起来似乎更偏重实践。然而,这两种问题的内核却通常是相同的:它们都是为了强化对某些理论的学习,例如队列和栈,又如“基于用例的需求分析方法”。

学习理论固然重要,但这种“轻实践,重理论”的理念却让理论学习成为了最终目的,而项目问题的质量却常常被忽视了。

这种忽视往往会带来一些不理想的结果。

同学 A 试图向教师 B 介绍自己的项目解决了什么问题。

同学 A: 我的项目来自于我平时生活的切身感受。我是一个很喜欢听歌的人。但是受到版权的限制,有些歌只有网易云音乐上有,有些歌只有 QQ 音乐上有。当我想要找一首歌时,我就需要分别在两个平台上搜索,非常麻烦。

教师 B: 好的,这个观察非常棒。

同学 A: 于是我的项目试图解决这样一个问题:如何在一个应用里,让用户能够分别搜索网易云音乐和 QQ 音乐。

教师 B: 我好像没太听懂你的意思。

同学 A：是这样的。在我的应用里有一个 Entry 和两个 Button。用户只需在 Entry 里输入关键字,点击第一个 Button 就会搜索网易云音乐,点击第二个 Button 就会搜索 QQ 音乐。

教师 B：所以你觉得“不能在一个应用里分别搜索两个音乐平台”是你作为用户遇到的问题。

同学 A：是的。

教师 B：但我认为,“没有一个应用能够帮我同时搜索两个音乐平台,并告诉我到底哪个平台上有我想找的歌”有可能是用户遇到的问题。

同学 A：我没有考虑这一步。

从上面的例子可以看到,同学 A 给出了一个很不错的观察：在多个音乐平台上搜索歌曲是一件很麻烦的事情。然而,同学 A 提出的问题是“如何能够在在一个应用里分别搜索不同的音乐平台”。一部分用户认为这是他们遇到的问题,但“如何同时搜索不同的音乐平台”也许对用户更有价值。

在学校中,忽视项目问题可能只会影响一点成绩。然而,如果在步入社会后依然不能提出高质量的项目问题,那结果可能会严重得多。

这段对话来自创业纪录片《燃点》^①。一位创业者试图向投资人介绍自己的项目。

投资人：现在在做什么项目？

创业者：现在做美食应用。

投资人：具体内容是什么呢？

创业者：我们拍摄美食短视频,帮助商家做美食推广,并联合这些商家推出一些活动。

投资人：你们帮他们拍短视频的时候收费吗？

创业者：收费的。一部短视频收费 5000~8000 元。

投资人：商家为什么要通过你的应用来推广呢？

创业者：我们只选择一些精选的商家,不是所有的商家都能达到我们的要求的。

投资人：这个项目做了多久了？

创业者：8 个月。

投资人：项目的数据怎么样？

创业者：目前拓展了 17 户商家,拍摄了 20 多部视频。

投资人：有收入吗？

创业者：有收入,收入来源主要有过拍摄短视频和推出联名卡。我们的一张联名卡 99 元。用户使用联名卡可以在一个月内到我们拓展的商家享受总计 4 次美食。

投资人：好吧……你有什么问题吗？

创业者：不知道您怎么看这种“餐厅共享”的概念？

^① 纪录电影《燃点》见 <http://www.ahavideos.com.cn/a/71/607.html>。为了增加可读性,对话内容按照本书作者的理解进行了修改。

投资人：我不理解为什么这可以被称为“共享”。你帮商家拍摄了一部短视频，商家给了你钱，不代表你能够让很多商家成为你的客户。如果没有商家的数量作为基础，那么对于商家来讲，你就只是一个短视频拍摄服务提供商。拍摄结束时，你和商家之间的合作也就结束了。我不认为你提出的你与商家之间的互动与推广逻辑是成立的。

一周后，该项目正式结束。

上面两个例子说明了忽视项目问题的质量可能带来的后果：这么做可能会影响开发者的成绩，也可能会毁了一个项目甚至公司。那么，如何才能提出高质量的项目问题呢？

本书给出的建议是回到产品的本源。产品永远是为人服务的。深入地了解“人”，认真地了解人遇到的问题，才有可能提出真正有价值的问题。

3.2 “认真地”观察用户：使用 5W 方法

要提出有价值的问题，很重要的一步是找到遇到问题的人，并认真地观察他们究竟遇到了什么问题。这里强调要“认真地”进行观察：找到他们，去到他们身边，和他们一起工作或生活，看看他们究竟在什么情境遇到了什么问题。

下面这个案例来自加州大学圣地亚哥分校的慕课“以人为本的设计：概述”。高通（Qualcomm）公司曾经为卡车司机设计过一款设备。可惜的是，卡车司机们不怎么喜欢这款设备。

这款设备上有很多很小的按钮。当开发者们观察卡车司机怎么使用这款设备时，他们惊讶地发现多数卡车司机的手都很大。并且，由于需要搬运沉重的货物，他们通常都戴着手套。这时，卡车司机完全无法使用设备上的小按钮。

基于这一观察，开发者们修改了设计，如图 3-1 所示。他们首先为设备安装了一块



图 3-1 高通公司为卡车司机设计的设备^①

^① https://customer.omnitracs.com/training/docs/mcp200/doc_cust_mcp200_installation_d.pdf。

以当时的标准来看巨大的触摸屏。其次,每次设备要求司机录入信息时,最常用的回答会直接显示在屏幕上,这一设计显著地降低了卡车司机录入信息的数量。最后,开发者们还提供了触控笔支持,以便在极端情况下,卡车司机还能使用触控笔来进行输入。

事实上,观察用户的过程也存在着一定的技巧。一种常见的技巧是观察“5个W”: Who、When、Where、What 和 Why,即“谁,在何时、何地,遇到了什么问题”,以及“为什么会遇到这个问题”。

接下来遵循这5个W来观察一位遇到问题的用户。这个案例将会贯穿本书。我们将一步一步地了解这位用户遇到的问题,提出解决方案,并最终解决他的问题。

这个案例的用户是一位年轻的奶爸F,他最近正在帮助自己的宝宝B背诵古诗。宝宝B很喜欢背诗。事实上,宝宝B将背诗视为一种游戏。无论是在上学的路上,还是在爷爷奶奶的家里,几乎在任何时间、任何地点,宝宝B都可能会缠着奶爸F背诗。

宝宝B要背诵的古诗记录在一本诗词本上。当宝宝B和奶爸F待在家里时,奶爸F可以拿着诗词本陪宝宝B背诗。但是,当他们不在家里时,由于奶爸F不可能总是带着诗词本,宝宝B就不能随时随地开心背诵古诗了。宝宝B很沮丧,奶爸F也很沮丧。因此,奶爸F的问题来了:如何才能随时随地陪宝宝B背诗呢?

针对奶爸F遇到的问题,可以做出如下5个W的观察。

谁(Who): 奶爸F,他有一个喜欢背诗的宝宝B。

在何时(When): 奶爸F与宝宝B在一起的几乎任何时候都有可能。

在何地(Where): 除了奶爸F与宝宝B的家里之外的任何地方,因为诗词本通常都放在家里。

遇到了什么问题(What): 当奶爸F与宝宝B不在家里时,由于奶爸F不能总是随身带着诗词本,因此宝宝B就不能背诗了。宝宝B与奶爸F都很不开心。

为什么会遇到这个问题(Why): 导致问题的原因是多方面的。

(1) 奶爸F不是个记忆超人,因此他不能分毫不差地复述出诗词本的内容。

(2) 诗词本有点大,也有点重,不能方便地放在口袋里。

(3) 奶爸F没有随身带着手提包的习惯,可能多数男士都没有这种习惯。

(4) 奶爸F认为当宝宝B想要背诗时,满足宝宝B的要求是一件很重要的事情,奶爸F不想伤害宝宝B想要学习的积极性。

如何帮助奶爸F解决他遇到的问题呢?可以要求奶爸F把诗词本上所有的诗词都背下来,并且还要能够按顺序复述出来。把所有的诗词都背下来不难,但是按顺序复述出来却很难。可以要求奶爸F换一本轻巧的诗词本,然后在每次出门前像带手机和钥匙一样带上诗词本。不过这也不太可能,因为奶爸F可能还需要再带上生字本、单词本等十几个小本子。我们也可以要求奶爸F随身带着手提包,或者干脆忽略宝宝B提出的背诗请求,但这些都不是能解决客户问题的方案。

开发者应该做的,是进一步了解用户到底想要什么,以及采用怎样的解决方案对用户更加有利。

3.3 进一步了解用户：面对面访谈

认真地观察用户是一个很好的开始,但观察方法本身也存在着很多不足。首先,有很多事实是观察不到的。例如,开发者通常不可能 24 小时跟在用户的身边。其次,有些事明明就摆在面前,我们却完全没有注意到它们。例如,很多 C 语言编程新手会忘记写分号,导致编译错误,却怎么也找不出错误在哪里。最后,观察用户时可能会产生一些问题。直接向用户询问问题的答案可能比继续观察用户并自己寻找答案更为高效。

意识到观察方法存在的这些(以及更多其他)问题时,就可以准备好进一步了解用户了。进一步了解用户有很多种方法。这里主要关注一种颇为直接地了解用户的方法——准备一次面对面的访谈。

访谈是一种有目的的聊天。访谈用户是为了进一步了解用户,而达到这一目的的手段则是向用户提问。因此,在访谈用户之前,首先要准备一份问题单,列出打算提问的问题。

而决定访谈成败的关键就是,向用户提出什么样的问题?

Sharan B. Merriam 在 *Qualitative Research: A Guide to Design and Implementation* 一书中指出,好的访谈问题应该是开放式的(open-ended),并且应该带来描述性的(descriptive)回答(甚至可以是一段故事)。用户的回答越具体越好,描述性越强越好。一些好的、开放式的访谈问题包括如下。

- 给我讲讲上一次当你……
- 能不能给我一个关于……的例子?
- 能否进一步介绍一下……
- 当你……时,你会怎么做/怎么想?

在准备问题时,还应该避免一些典型的设计不良的问题。

Sharan 还指出了一些应该避免的问题。

避免一次提出多个问题:“你觉得你的教师怎么样?作业多不多?课程进度合理吗?”同时提出多个问题会分散用户的注意力,导致用户不能很好地回答每一个问题。

避免提出带有导向性的问题:“将开发过程标准化,是否是为了方便替换开发人员,从而压低开发人员的薪水?”带有导向性的问题通常是为了得到有利于提问者,符合提问者预期的回答。提出这类问题,要么是提问者没能抵御住“偷懒”的诱惑,要么是有意为之。标准化的开发过程能够提升沟通与开发的效率,提高生产力,创造更多的价值,从而提高全行业的收入。标准化的开发过程确实降低了替换开发人员的成本,但“标准化”的目标和主要效果从来不是“方便替换开发人员”。

避免非黑即白的问题:“学习累不累?”除了“是”和“否”,我们通常很难从这类问题中得到更多的信息,更何况现实生活中的问题通常不是非黑即白的。有趣的学习可能是累但却快乐的,这个时候累不累可能并不太重要;无趣的学习可能一点都不累,但却太过于无聊导致内心非常疲劳。

与准备问题密切相关的一种技能是“追问”。在访谈过程中,如果在用户的回答中发现了新的问题,就一定要及时追问。追问相当于在现场随机应变地准备问题。

Sharan 指出追问可以有多种形式。

- 给用户一段沉默的时间,可能用户自己就会开始做出进一步的解释。
- 发出一些声音,例如“呃……”。
- 提出一个关键词,例如“诗词本是指……”。
- 提出一个完整的问题。

但是无论采用何种方法,都不要过度地逼迫用户(pushing too hard),或者过快地提问(pushing too fast)。记得给用户充足的思考时间。

准备好一系列问题后,可以尝试按照 U 形理论组织问题与访谈的进度:从简单、轻松的问题开始,逐步提出复杂、敏感的问题,再以简单、轻松的问题收尾,如图 3-2 所示。这将给用户及我们自己带来一段更加舒适的经历。

最后,在正式开始访谈之前,还需要找两位用户测试一下访谈问题,根据测试用户的反映调整问题,直到用户能够给出有价值的回答。

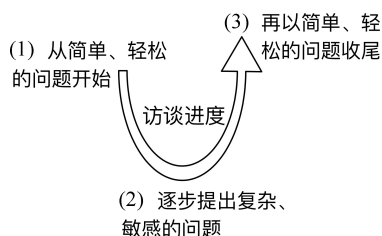


图 3-2 访谈问题与进度的 U 形组织

3.4 同理心

在很多时候,观察与了解用户考验的并非专业技能,而是人际交往技能。贯穿观察与了解用户方法的一项重要技能是同理心。同理心,或者说是换位思考能力,指的是我们应该考虑如果我们自己是那个用户,那么我们会怎么想。

因此,在观察用户时,要真的与用户一起工作/生活,一起面对他们的问题。这不仅为了能够亲身体验用户面临的问题,还是在建立用户的信任。换位思考用户肯定不太愿意相信一个只会夸夸其谈,但却从来没有与其站在一起的开发者能够真正地解决问题。

所以,在准备访谈问题时,要选择开放性的题目,并且避免提出多个/具有导向性/非黑即白的问题,以便用户能够充分地进行表达。因为用户从提问者的问题中感受到心不在焉的懈怠或是先入为主的傲慢时,必然不想给出发自内心的回答。要让访谈轻松地开始,轻松地结束,并避免过分地逼迫用户。因为没有人喜欢充满压迫感、盛气凌人的对话。

如果没有同理心,“为人服务”就只是一句虚假的口号,其内核也只是为自己服务。我们要做的,则是培养自己的职业技能,并且让自己不要变成没有同理心的人。

3.5 奶爸 F 的观察与访谈总结

下面给出奶爸 F 的观察与访谈总结。这份总结文档将指导本书后续的开发工作。

很多软件工程书籍都介绍了各式各样的标准化文档格式。但本书并没有采用标准化

的文档格式来撰写各类文档,而是采用一种自然、易懂的撰写方法。然而,通常可以很容易地将这些文档转换为标准化的格式,如用户故事等。

奶爸 F 与宝宝 B 在最近一次背诗时,都做了哪些事情?

奶爸 F 拿出诗词本,随意念出一首诗词的题目,并让宝宝 B 背诵诗词的内容。在宝宝 B 背诵完之后,奶爸 F 会再次随意念出一首诗词的题目。这个过程会重复若干次。

之后,奶爸 F 带着宝宝 B 一起朗读诗词本的最后一首诗。诗词本的最后一首诗是一首新诗。奶爸 F 会在几天的时间里,带着宝宝 B 反复朗读这首诗,直到宝宝 B 记住这首诗为止。

新诗是从哪里来的?

新诗没有固定的来源。它可能来自宝宝的图画书,或是电视上的一句话,或是公园里的一处景色。

新诗是如何被添加到诗词本的?

奶爸 F 会首先用搜索引擎搜索新诗的题目、作者、内容片段等信息,从而找到诗词的全文。接下来,奶爸 F 会将诗词抄写到诗词本上。

如果我们开发一款软件来帮助奶爸 F 和宝宝 B 背诗,那么这款软件应该运行在什么平台上?

奶爸 F 是半个极客,他拥有所有平台的设备:装有 Windows 10 系统的计算机、iPad 平板电脑以及 iPhone/Android 手机。奶爸 F 会随手拿起一台设备使用,因此软件需要运行在所有平台上,并且能够在不同设备之间同步数据。

3.6 动手做

图 3-2 所示的 U 形理论不仅可以用于组织问题,还可以用于组织语言。当想指出其他人工作上的不足时,便可以使用 U 形理论组织语言。我们可以首先肯定他人的工作:“我非常喜欢你做的……,其中……简直太棒了!”接下来,提出我们的见解:“我还发现里面有一个小小的问题,我觉得如果改进一下有可能会让整个项目更加出色。”最后,再次肯定他人的工作:“即便如此,我还是觉得……已经非常棒了!”试着在下一次向他人提出意见或建议时,采用 U 形理论,并将感受分享给其他人。

3.7 给 PBL 教师的建议

从本章开始,使用本书的 PBL 教师就可以跟同学们一起准备 PBL 项目了。可以让同学们试着从自己的团队、亲友或其他来源寻找遇到问题的人,观察他们,并开展访谈。

如果没有特殊的要求,观察记录、访谈问题,以及访谈总结等文档不必设置模板,只需要列出一些你认为重要的要点,从而让学生充分发挥想象力,培养学生的多样性思维。你肯定会遇到敷衍了事的学生,但你同样也会被同学们的创造力感染。

教师需要与学生一起评估选择的问题。可能需要评估的方面包括如下。

- 团队的目标是什么？是想做个好产品，还是想拿个好成绩，或者只是想通过考试？团队成员的目标统一吗？
- 解决这个问题可能用到什么技术？是否满足课程学习目标的要求？
- 问题的复杂度如何？在课程的时间范围内，教师和同学们能在多大程度上解决这个问题？能否交付一个成型的产品？
- 团队的成员与资源情况如何？每个人每周可以贡献多少工作时间？有没有人需要兼顾其他课程的成绩，或者游戏排名？

有些团队可能需要经历几次反复才能选定问题。请给同学们足够的时间，例如几天甚至一两周^①，同时经常与他们坐在一起，聆听他们的讨论，并只在他们遇到无法解决的问题时才给出自己的意见。请充分地展现同理心，成为团队的一员而不是监管者，让学生成为推动项目进展的核心力量。

^① 直到第10章开始之前，都可以让学生把时间花在选定问题上。这么做的前提是，每一次讨论都要有实质性的进展，包括否定一个不可行的问题并从中吸取经验。