|  |
| --- |
| 论迭代式的产品开发方法 |
| |  |  | | --- | --- | |  | **对基础构思的完善和原型化**  　　一款游戏从创意到开发，抽象来看可以分为两大阶段：基础构思的阶段，和迭代开发的阶段。任何游戏在最早的时候都只是一个或者一组零散而不确定的构想，策划人员将这组构想加以整理，抽取其中相互联系的规则组成核心规则集，这就是产品最初的框架。譬如说俄罗斯方块最初的规则可能包括：方块连成一行就消除并加分；头顶随机掉落新的方块；方块可旋转，等。  　　一般来说，在这个阶段，游戏开发者会寻求利用这组核心规则建立一个简单的DEMO，用来验证游戏本身的可玩性。这个DEMO往往是缺乏美术效果和友好的UI的，但是其遵循的游戏主循环一般来说与后来的商业版本并没有太大的不同。 　 　　譬如说如果你在做弹弹堂，你可能会先搞一个只有一种炮弹、一个怪一张地图的演示版，虽然内容简单，但是回合规则与弹道公式是与后来的版本基本上一致的。 　 　　对于一个前期的产品构思来说，究竟要花多少时间和精力来做DEMO？又要花费多久来测试这个DEMO？不同的公司和团队，对这个问题的回答往往大相径庭。 　 　　在这里就出现了一个很有争议性的问题：缜密的构思加上完善的策划文档，是否能够替代对于DEMO的开发和评估？答案是否定的。 　 　　为什么要做原型呢？既然原型的代码基本上不可能被用在商业化的成品里，既然这只是给一小部分人看的演示，跳过去有何不可？ 　　 　　核心原因在于，作为人，我们的能力和经验从本质上说，是有限且不完备的。而游戏又是一种体验性的产品，一款游戏的可玩性，无法通过逻辑和数理的方式来验证，而必须通过一部分人员通过实际的游戏过程，主观的去感受和评判。而这也就是为什么说游戏设计是一门艺术的理由所在。 　 　　很多游戏策划人员骨子里对市场和运营是持反感态度的，他们说，游戏是一种艺术，游戏性是我追求的灵魂，这比庸俗的充值要重要。在原型的设计和评估阶段，他们是对的。 　 　　对于原型的评判，一般来说，是要看游戏的核心规则是否清晰容易掌握，同时根据用户的操作又能够得到各种不同的选择结果，再就是技术角度的一些基础性验证。譬如说，一个游戏可能规则复杂变幻叵测，但是需要一个月的时间才能上手；又或者一个游戏3分钟即可掌握，但是玩来玩去每一次的流程都差不多。这些，都是需要在原型化过程中去分析和判断的问题。 　 　　如果一个原型做下来，大家不觉得这个东西好玩，接下来该怎么办？ 　 　　很简单，放弃。扔掉一切，重新开始设计。这里有一个很大的误区，一方面把游戏性不佳归结于DEMO的内容量不足，指望着内容量增加之后可玩性变好；另一方面以DEMO早晚会放弃不应当投入太高成本为借口，认为“这么小的DEMO能达到这样的水准，如果……的话，游戏肯定不错”实际上这都是自己给自己挖坑跳的思想和行为。 　 　　一款游戏，小到俄罗斯方块泡泡龙，大到魔兽世界天龙八部，本质上都是有自己的核心玩法的，大型产品的核心玩法构成可能更复杂，甚至是由一组相互关联的子玩法相互配合所组成，但是无论大游戏小作品，都是由一个个独立的玩法模块搭建起来的，一个大型的MMO，可能其中很多玩法并不特别出色也能获得成功；但是一个玩法模块很多、但是每一个都不算出色的产品，是不可能仅仅凭借功能比别人多来赢得玩家的。 　 　　就是说，成功的游戏不见得每一个玩法都精彩，但是没有至少一个比较精彩的玩法的游戏，一定会失败，无论多久、无论成本多高、无论程序美术策划多努力。实际上这是1和1后面的0的关系，没有1，则再多的0加起来也是0。 　 　　我讲过一个简单的理论：游戏的核心玩法是一款游戏的纵轴，而内容的增加是一款游戏的横轴，一个好的纵轴，可以支撑很广阔的横轴，就像弹弹堂或者疯狂的小鸟，基于自己的核心玩法，可以不断设计推出新的地图；而如果纵轴不够强大，横向的扩展越多，产品死亡的速度就会越快。这就像盖房子没有把房梁搭好就往上放砖，建的越快就塌陷越快。 　 　　所以，在游戏的初期，对于核心玩法和DEMO的反复修改不断锤炼，是决定了这个产品能走多远的基础和地基，打个比方你能设计出疯狂的小鸟或者植物大战僵尸的核心玩法规则，那接下来要做的无非是找一堆美术和关卡外包干活罢了。这在网页游戏和社交产品领域同样是成立的，就像傲视天地的推图和战斗系统，都应该是在早期就开始勾勒并且贯穿了整个产品开发始终的东西。 　 **迭代式开发的核心思想与理念** 　 　　好了，接下来，把艺术的感性收起来，我们要进入迭代式开发的阶段了。 何为迭代？盛大以前有一句很形象的形容，叫做小步快跑。 　 　　以下文本来自百度：迭代式开发。在软件开发的早期阶段就想完全、准确的捕获用户的需求几乎是不可能的。实际上，我们经常遇到的问题是需求在整个软件开发工程中经常会改变。迭代式开发允许在每次迭代过程中需求可能有变化，通过不断细化来加深对问题的理解。迭代式开发不仅可以降低项目的风险，而且每个迭代过程以可以执行版本结束，可以鼓舞开发人员。  　　其实中国还有一句老话，叫做走一步看一步。本质上，迭代式开发承认开发者当前对于用户需求的了解和把握是欠完备的，开发者并不追求对产品需求进行一次性、全局的理解和把握，而是针对每一个产品细节，收集用户行为和反馈，提出可能的解决方案，加以实现并验证是否解决了问题，然后再迈向下一步的一种循环。 　 　　可以这么说，迭代式开发的起点，是从一个版本的发布开始：版本发布，通过数据统计和分析，以及直接的用户调查和问询，得到用户行为的直接反馈；基于反馈分析原因并提出可能的解决方案及验证方法，开发完成这一解决方案，观察用户的行为是否有所改善。如果没有得到改善，那就去尝试另一个解决方案，重复这个循环；如果经过证实得以改善，那就继续接下来的开发流程。 　 　　一个典型的迭代式开发过程中，有三个关键原则：尽可能短的迭代周期、明确的效果验证方法、低成本的修正方案。 　 　　在不对用户和运营产生过大困扰的情况下，迭代周期越短越好。这就要求把一个比较大的版本规划切分成若干个小版本，分别针对某个特定的问题。本质上每一次的迭代循环，相当于开发人员与用户/市场之间的一个对话周期，而对话进行的越频繁，对市场的了解和把握就越深入。一对每年只通话一次的朋友，关系一定比不上每个礼拜都一起吃饭的朋友亲密，而开发者和市场之间关系越疏离，距离成功的道路就越遥远而不可见。 　 　　明确的效果验证方法。这是绝大多数产品开发过程中会犯的错，就是有意无意的省略了对效果的验证。发现产品中的一个问题，譬如某个高流失率环节，讨论，提出了某个改善方案，制作完成上线。然后……没了。 　 　　实际上我一直在强调，迭代式开发的潜台词就是承认我们对用户和市场的无知。只有经过了验证的方法，才能称为一种“经验”，而这种经验的积累，代表了游戏团队水平和竞争力的提升。仅仅满足于提出或者完成某种改善方案，这本身没有任何特别的意义。因为你压根不知道，这个改善方案到底是对的，还是错的？ 　 　　我们往往以成本和工期为理由，用自己或者领导的想当然，来替代繁杂但却可靠的运营数据和用户行为分析，并且振振有词说这就是我的水平，其实这只导致了一个结果，就是曾经犯下的错误，必然在未来某个时间点重复再犯一次。工期越长、时间越久，这种差距就越来越大。就像同样是成立5年的公司，zynga真正比我们强悍的地方，其实在于每过一年，他们做一个新产品的时候所需要犯的错误就少一些，而我们5年前犯的错和今天做一个产品犯的错有可能是一样多的。 　 　　低成本的解决方案设计。这意味着当我们发现某个环节有待改善时，应当首先评估那些实施成本最低的方案。因为每一次方案的提出和实施都是一次风险投资，在目标明确的前提下，投入越少，性价比越高。低成本的方案往往也意味着更迅速的开发时间和更短的迭代周期。有趣的是，虽然理智上我们很容易支持这一点，但是许多时候这样的要求，和作为开发者本身的人性是相违背的。 　 　　作为产品的创造者，我们经常会在发现产品的种种不足之处时，提出一个“全新的、更好的、一揽子的”解决方案，并且认为那就是完美的答案。这可以称之为一种“创新冲动”，但也可以叫做“创新性的陷阱”。实践证明，这种看起来很完美的方案，经常只是因为其细节没有被考虑充分罢了。大多数时候，在现有的方案上稍加调整，就可以解决问题的90%，这时候优先选择的一定是成本更低的解决方式。 　 　　有人会问为什么不投入更大的精力，追求产品的完美？这经常是一种很有冲击性的提问方式，就好像我们是在裹着裹脚布走路的小老太太，被五四青年当街质问一般。 　 　　其实答案很简单，因为你我这一刻所认为的完美方案，往往一点都不完美。人经常陷入一个思维误区，就是对某个特定方式的非理性推崇和崇拜。进而认为所有不同意这个方式的人都是品味或者能力有问题。然而一两个礼拜之后，就发现其实全然不是如此这般。之前的完美方案之所以看起来如此诱人，只是因为我们只是一厢情愿的看到了其优点，而不愿意面对其不足。 　 　　当一个方案真的在各方面都远胜从前的时候，我们自然应当勇往直前将其付诸实施，据我的经验，这种情况一般来说只占十分之一的比例。一段时间的沉淀、替代方案的讨论以及广泛听取周围人的意见，有助于我们去判断哪些东西是真金，哪些东西是空包弹。 　 　　一句话：为解决问题而提出、并且经过了事实的验证被证明有用的创新，才是有效性的创新，而一系列有效性的创新的累加，就是一款成功的商业化产品。 　 　　所以总体上，当我们面对一堆版本反馈和数据的时候：1、首先要做的，是提取其中对产品的进一步改善有显著标示性的片段；2、针对这些片段，提出关于其原因的假设和潜在的修改方案；3、通过逻辑性和既往经验，去掉部分可能性和可实施性较低的方案；4、然后，针对剩下来的方案，设计对其调整的有效性的验证方法；5、在成本允许的情况下，尽可能尝试所有的方案，寻找效果最好的一种，固定下来。 　 　　相对于拍脑袋，这是个辛苦得多的过程。而这个辛苦的过程所带来的，就是我们和真正一流的研发公司之间的巨大鸿沟。 　 　　另外一个很现实的问题是，当一款产品还没有推出到市场上的时候，应当如何实现有效的需求迭代？目前来看，有几个方法可以参考：首先，尽量压短前期开发时长，让产品可以更早的面向至少一部分的用户群；其次，在有限范围内寻找参与者，典型的譬如公司内的自发性组织和测试，也包括受控制的小范围封闭测试（这几乎是最常见的做法之一）；最后，记住一件事：对于开发团队来说，真正的开发周期是从产品上线那天开始算起。这可说是一种超越了具体方法本身的价值观。 我认为，丰富的原始创意/敏锐的市场嗅觉、对既往成功产品（包括市场上的流行产品）成功做法的分析和总结/模板化、迭代化的开发思想/强大而迅速的执行能力，这若干项的合理组合，就是一家成功的游戏开发公司内在的基因所在。诚然，对今天的我们，这些要求很遥远也很难达到，但，这起码告诉了我们未来的方向。 　 **一些哲学层面的提升** 　 　　如果我们仔细研究人类的思维方式，你会发现，任何创新的点子在被酝酿出的一瞬间，都是基于一个特定的市场假设模型的。这个模型存在于我们的脑海里，且相互之间不可复制。譬如说我脑子里可能出现一个点子：加了芥末的酸辣黄瓜会大卖。这里的大卖，就是我基于我脑子里对顾客的理解和认知，模拟出来的一个餐饮市场的模型，我把自己的主意放到这个模型里，然后发现计算结果是“十分乐观”，接下来该怎么做？马上改行当厨子去？ 　 　　且慢。首先我必须问自己一个问题，我了解餐饮业么？作为一个从来不做饭的人，我是否可以仅仅凭借自己脑子里的臆测，就认定一种产品会成功与否呢？ 　 　　如果你仔细阅读了之前的文字，这里你就会明白，实际上我头脑中的市场模型并不完备。顺带你也可以懂得，其实没有任何人脑子里的模型，是与市场完全等价的。本质上，我们对于现实世界的了解，永远是局部、片面并且带有主观倾向的，这是一种常态。当然，经常去了解和分析市场的人，其模型的偏差程度，会比我这种外行要小，所以他们的判断相对更可靠一些；而我们基于对市场的不完备理解，所设计出来的解决方案，本质上也必然是有欠完美的。这是一种非常哲学化的思辨，但在产品开发过程里，这几乎可以当成警句来使用。 　 　　我们对用户的需求不够了解 >> 我们设计出来的方案充满缺陷 >> 缺陷的方案提交给不够了解的用户，必然会产生意料不到的偏差。 　 　　这几乎是一种宿命的悲观论调。如果我们推演下去，不完备的方案进而会影响和改变用户原有的需求，那这基本上是乔治索罗斯著名的“反身性原理”的游戏开发版。 　 　　可是另一方面来说，正因为缺乏完美的解决方案，才导致了市场上各种游戏设计思路都有其机会，没有完美答案所以每个人都可以提出自己的方案，然后在市场中竞争并测试其有效性。生产钢材的完美方案只有一种或者几种，所以新的创业者基本上无法去开个炼钢厂。做出好游戏的思路千变万化，所以我们每个人就都有了自己的机会。 | |