

# 互联网模式-软件开发

互联网提供了低成本的万物互联的物质基础，当然这其中最重要的是人与人的互联。这种低成本的信息交流方式，可以有效解决传统社会中由于信息不对称、信息交流成本高等造成的问题。比如现在的电商、打车软件等，通过直接连接生产商和消费者，降低了交易成本，获得了许多人的认同。

互联网的发展经过了两个阶段，在第一个阶段，人们主要使用PC个人电脑访问互联网；第二个阶段，以苹果公司2007年发布第一代iPhone为起点，移动互联网迅速发展，人们除了使用电脑访问互联网外，使用手机访问互联网比例迅速增大，同时人们对互联网的使用越来越偏向于个人目的。

互联网改变了世界上的许多行业。而软件行业作为与互联网行业联系最深的行业之一，互联网的发展对软件开发提成了什么新的要求？有没有对软件开发方法产生影响？现在的软件开发方式和互联网未出现前有无不同？

## 一、互联网时代软件产业的不同

互联网普及要求更多的软件应用，这些软件和之前的软件有所不同。在互联网大规模普及前，软件主要用于科学计算、企业业务需求、办公自动化等领域，主要针对企业用户，一般直接针对消费者的软件较少，很多软件是单机版或运行在一个局域网中。软件的主要用户是企业用户。互联网普及后，尤其是移动互联网普及后，社会对软件的需求出现了新的趋势，现在越来越多的软件开发是主要以个人消费者为直接用户的，这使得对软件产生了新的不同要求。

1、对某一款软件而言，个人消费者用户往往数量很多，他们的需求多变并难以把握。个人消费者的需求往往不容易通过传统企业软件开发的方式，比如访谈、问卷调查等获取。他们的需求不会和一定工作流程或技术有关，更多的是个人的需求，很多并不明确，不容易获取。比如在社交应用大规模流行前，很多人无法想象，在吃饭前对食物进行拍照并分享给朋友的需求如此广泛。

2、消费者用户往往不喜欢复杂的应用，不愿意接受复杂的培训（企业员工也不愿意，但往往企业会提出要求强制进行培训），从而对软件用户体验的要求非常高。

3、消费者用户往往不像企业，他们没有足够的动机和预算（也不倾向于）来购买软件，免费软件逐渐流行（商业模式是另外一个话题）。用户对于免费软件和购买了许可证的软件的心态有所不同。付费购买了软件商业许可证的用户会认为“顾客是上帝”，软件产品不能有bug，而且因为已经付费，为了避免前期投入浪费，即使不满意，他们也不会很快的迁移到其它类似产品。而免费软件的用户，他们对软件产品的bug会有更高一些的容忍度；但他们会用卸载来表示自己的不满，并且迁移到别的竞争产品的速度会很快（除非软件产品本身有非常好的用户粘度，比如社交关系网络）。

面向企业的软件市场仍然存在，传统软件开发方法论对于这类软件开发的经验仍然适用，但对于互联网时代新的软件形态，新的一些软件开发方法已经出现。

互联网时代软件的分发渠道和用户反馈方式发生了巨大的变化。在互联网发展的早期阶段，人们在开源软件的网站上可以下载到最新的软件；现在人们习惯于从各种应用商店下载软件（App Store, Google Play等）。总之，软件开发者可以使用互联网使得自己的软件在非常短的时间内、低成本的将自己的软件分发到目标用户，为软件的高频率发布提供了可能。通过互联网，普通用户可以方便的对软件提出自己的反馈意见，可以是反馈软件的bug，也可以是自己对软件新需求。这些信息可以通过网站论坛、Email、应用商店的评论等等迅速反馈给软件开发。在没有互联网这种媒介前，人类历史上没有任何的通讯工具可以支持进行多对多的如此频繁的信息交流。这些，为新的软件开发模式提供了基础。

## 二、 开源软件开发模式

伴随着互联网在1990年左右在全球的迅速发展，最先利用互联网进行软件开发的是自由软件（freeware），最具代表性的是Linux（现在我们一般使用开源软件open-source software来描述Linux，但开源软件这个名词1998年才出现。自由软件和开源软件两个词的确切定义和相互关系比较复杂，请参考<http://opensource.org/osd>）。

Eric S. Raymond（简称ESR）在1997年发表了《大教堂与市集》（The Cathedral and the Bazaar）一文（简称CatB），阐述了Linux的开发模式和他自己对软件开发的一些看法，产生了重大影响，1998年Netscape公司受该文影响开源了其浏览器的源代码。

在CatB一文中，ESR解释了Linux的成功，它背后的软件工程方法学原理，包括黑客的文化等等。其中，ESR提到：

“Linux是第一个有意识并成功的把全世界当成它的头脑库的项目。我不认为Linux的孕育和WWW的诞生相重合是一个巧合，而且1993-1994年Linux发展的早期正是互联网服务提供商（ISP）快速发展和互联网获得社会主流认可爆发式增长的时期。Linus是第一个学会在互联网普及的情况下，利用新游戏规则的人。”

ESR提出了描述了Linus开发的几点经验：

“将用户作为协作开发者是快速改进代码和有效调试的最有效方式”。

“早发布。常发布。倾听用户。”

“如果有足够多的beta测试人员和协作开发人员基础，几乎所有的问题都可以被快速定位并被某些人解决。或者，不正式的说，如果有足够多的眼睛，所有的bug都是浅显的。我称这为Linus定律。”

在Linux的开发过程中，参与者可以基本分为两类，一类是开发者；一类是使用者。开发者开发和改进了软件后，迅速发布在一个可以公开访问的网站上，其他开发者和用户可以访问到可运行软件和源代码。其他用户在使用该软件的过程中会发现软件的bug，并在这个网站上进行反馈。而开发者可以看到这些反馈并根据这些反馈修复bug；同时，用户和开发者也可以使用互联网交流对软件功能的需求，从而确定软件的下一步开发目标，迅速演进软件。

这样的开发方法和实践在Linux和其它很多开源软件的开发过程中已经被证实是非常有效的软件开发方法。

### 三、 敏捷软件开发

软件过程是软件工程领域的一个重要组成成分，对于软件开发的进度和质量都有着重要的影响。我们可以看到，在互联网大规模开始应用以来，在同一时间维度上产生的软件过程主要是各种敏捷软件开发方法。而且，当前有大量案例和公开资料显示敏捷软件开发方法已经成为一种主流软件开发方法。

敏捷软件开发主要由有经验的软件工程师和咨询师提出。在1997年左右，Kent Beck在1996年成为克莱斯勒C3项目（克莱斯勒综合报酬系统）的负责人后，提出了极限编程的开发方法。2001年，来自业界的十七位软件工程实践者在美国犹他州探讨软件开发如何取得成功，研讨会期间，他们形成了一些关于软件开发的共同观点，即敏捷宣言：“个体和互动胜过流程和工具；可以工作的软件胜过详尽的文档；客户合作胜过合同谈判；响应变化胜过遵循计划”。该宣言定义了敏捷软件开发的核心价值观和原则，而这些原则具体是由敏捷实践以及敏捷实践的组合----敏捷方法来实现的。

随着敏捷软件开发的发展，敏捷方法的数量逐渐增多，大约有20种左右的方法，当前重要的敏捷方法主要有Scrum、XP、Kanban.

伴随着敏捷软件开发的发展，关于敏捷软件开发和传统基于计划的软件方法的争议也越来越多。Nerur给出了两者的区别：

表1 传统软件开发与敏捷软件开发对比

传统软件开发	敏捷软件开发
基础假定	可以完整描述系统需求，系统是可预测的，通过周密、完整的计划来开发软件。
	基于需求变更和快速反馈，通过小团队持续设计、测试、改进完成高质量的软件开发。
管理风格	命令和控制 领导力和合作
知识管理	显式 非正式
交流	正式 非正式
开发模型	生命周期模型（瀑布、螺旋或其它变种） 演进-交付模型
组织结构	机械的（正式的）大型组织 灵活的小型组织
质量控制	严格的计划和控制，后期大量测试 持续演进需求、设计、方案。持续测试。

Version One公司每年都会进行一次全球敏捷开发的调查，已经进行了9次。虽然这不是严格意义上的学术研究，但在工业界和学术界的影响力都非常巨大，是持续时间最长，参与人数最多的敏捷调查。2015年发布的第9届敏捷状态调查报告指出94%受调查者所属企业使用敏捷方法，而且受调查者中有35%的企业规模超过5000人；同时45%受调查者所属企业中多数团队使用敏捷方法。报告宣称的敏捷带来的益处排在前3位的是：（1）87%认为敏捷帮助他们获得了管理变更优先级的能力；（2）84%认为提高了团队生产率；（3）82%认为提高了项目可视性。

敏捷宣言的起草者之一Martin fowler，在他的文章“新方法论”中描述到敏捷软件开发和传统方法的区别主要在于敏捷方法是自适应的，而传统方法是可预测的；敏捷方法以人为中心，而传统方法以过程为中心。

## 四、 互联网软件开发模式

基于互联网时代软件用户的特征、需求、软件分发模式、用户可用的意见反馈模式，合理的开发模式应当是：

1、 将用户看作协作开发者和**beta**测试人员。

2、 在团队内部使用敏捷软件开发方式，引入敏捷的思想和实践。

### 4.1将用户看作协作开发者和beta测试人员

尊重用户，将用户视为软件开发成败的最关键因素之一，如果没有这种一致的认识是很难顺利进行互联网模式软件开发的（在营销中有粉丝经济的说法，本文仅讨论软件开发）。

#### 4.1.1 提供合理、通畅的途径获取用户反馈

可以开设专门的论坛，比如小米公司和魅族公司的论坛都非常活跃，当中有大量的用户反馈系统bug，同时提出新功能的要求和建议；也可以使用应用市场中的反馈，但这种方式双向交流不方便；或者采用其它交流手段。可以设计一些模版帮助用户方便的提交bug和新功能，这样也可以方便团队成员利用用户反馈知识。

#### 4.1.2 团队中的开发人员必须直接获知用户反馈

应当要求团队中的每一位开发人员都直接参与论坛或其它途径和用户的交流，这会使得团队的所有成员直接的获取用户反馈，演进软件。团队的成员会在这种交流中感到自己工作的价值，并且直接从用户处感受到压力；用户会在和团队成员的直接交流中感到自己被尊重，也有利于提高用户满意度和口碑。

#### 4.1.3 提高反馈频率

在互联网软件的开发中用户反馈是下一步开发的基础和依据。反馈的频率十分重要，开源软件中提出“早发布、常发布”的理念，这对互联网软件的开发也同样重要。MIUI的发布频率为一周，早期Linux甚至每天发布一个内核的新版本以获取用户的反馈。

### 4.2敏捷软件开发

#### 4.2.1 为什么？

大部分的互联网软件都存在需求不够明确和变更速度快的问题，从当前可见的学术论文、专注、经验软件工程研究结果来看，敏捷软件开发相对于传统软件工程方法而言更加适应互联网软件需求模糊、快速变更的特点。所有的敏捷软件开发都是迭代式软件开发，而且当前的趋势是迭代的周期越来越短，在每一次迭代的开始都可以成本很低的引入新的功能（Kanban等方法甚至支持在更短的时间间隔内变更需求）。敏捷软件开发中有很多实践可以有效的支持迭代式软件开发。比如Scrum方法的过程框架，有助于保持良好的开发节奏；比如eXtreme Programming方法中的测试驱动开发、持续集成、简单设计、重构等实践帮助开发团队进行演进式设计。

#### 4.2.2 建议的敏捷软件开发方法

敏捷软件开发有多种具体方法，具体采用哪种方法每个团队都有自己的考虑。根据Version One的市场调查，当前采用率最高的方法是Scrum（56%），Scrum/XP（10%）。因为Scrum中主要是过程框架和管理实践，建议使用Scrum/XP混合

方法。

在使用敏捷软件开发方法时，应当注意必须同时引入敏捷的价值观和实践。价值观是知识和理解的一个抽象层次，价值观是在某种处境中我们喜欢或不喜欢某事情的根源。比如XP的价值观包括沟通,简单,反馈,勇气,尊重。没有价值观，实践很快会变成生搬硬套，缺乏目的或方向。如果只有价值观和没有实践，因为价值观往往是非常抽象的内容，我们将可以以价值观的名义进行任何的活动。实践是清晰明了，可以让价值观清晰可见。每个人都知道我是否参加了早上的站立式会议，但我是否重视沟通并不是那么明确和具体的。

敏捷软件开发的引入和具体实践开发有大量的文献讨论，本文不做过多涉及。初学者建议参考以下网站：

<http://agileclub.org/%E6%95%8F%E6%8D%B7abc/>

## 五、 不适合使用互联网开发方法的情况

任何一种软件工程方法学都不可能适配所有的软件开发场景和团队。

如果你开发的软件是面向企业的软件，你的软件需求是明确的并且变更较少，传统计划驱动软件开发方法或许更加合适。

另外一个需要考虑的问题是，不建议在一个互联网项目启动初期引入大量用户的介入。大量用户的作用更好的发挥场景是功能的演进和寻找bug。在早期，团队应当向用户展示一个有价值有新意的项目，吸引用户的参与，用户可以基于这个初始的思想演进出更完善的功能，而最初思想的提出仍然应该由开发团队完成。如果寄希望大量用户讨论就能得到一个有新意的项目，可能的结果只是混乱的讨论而已。

## 六、 一些思考

大家都在中学政治课中学过“经济基础决定上层建筑”的说法，可以认为经济基础指生产力，上层建筑泛指一切政治、法律、道德、社会制度等（可能不严谨）。我们可以将互联网工具看成是生产力，而在互联网时代的开发模式则是一种制度是上层建筑。如果从这个角度来看，由于互联网提供了廉价、即时、广泛的信息传播方式，互联网开发模式的产生是一个必然的结果，所以Raymond说Linux开发模式的诞生和互联网早期的蓬勃发展出现在同一个时期并不是一种偶然。敏捷软件开发的兴起在2000年前后，距离互联网的诞生有一定的滞后，但在互联网改变人类的这20年中，也获得了长足的发展。而在此期间，传统计划驱动软件工程方法的进展偏少。这是否也表明，敏捷方法更好的适应了互联网时代的软件开发需求？

## 七、 引用

[S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, Communications of the ACM (May) (2005) 72– 78. ][VersionOne. 2015. 9th Annual State of Agile Development Survey. ]