AppStore中的用户反馈：一个实证研究

*摘要*——应用程序分发平台或应用程序商店（例如Google Play或Apple AppStore）允许用户以评分和评论的形式向下载的应用程序提交反馈。 在过去的几年中，这些平台已经成为应用程序开发人员和用户非常流行的平台。 但是，他们对需求工程过程的真正潜力和影响尚不清楚。 本文报道了一项探索性研究，该研究分析了来自苹果AppStore的超过100万条评论。我们调查了用户提供反馈的时间和方式，检查了反馈内容，并分析了其对用户群体的影响。我们发现，大多数反馈是在新版本发布后不久提供的，随着时间的推移频率会迅速降低。评论通常包含多个主题，例如用户体验，错误报告和功能请求。评论的质量和建设性差别很大，从有用的建议和创新的想法到侮辱性言语。反馈内容会影响下载数量：积极的消息通常会带来更好的收视率，反之亦然。负面反馈如缺点通常具有破坏性，并且会遗漏上下文细节和用户体验。我们讨论我们的研究结果及其对软件和需求工程团队的影响。

*关键词*——用户需求，用户反馈，移动需求

# 引言

Apple AppStore1，Google Play2和Windows Phone Store3等应用程序分发平台使用户只需点击几下鼠标即可查找，购买和安装软件应用程序。他们的知名度正在高速增长。截至2013年3月，AppStore和Google Play上的应用程序超过80万个，Windows Phone Store超过130,000个。下载数量是个天文数字，AppStore每个月下载大约10亿次应用程序。随着这些平台的日益普及，使应用程序的销售和部署更容易，同时他们庞大的注册用户群体对软件组织的吸引力也非常大。

这些平台（也称为应用程序商店）也提供用户反馈功能，从软件和需求工程角度来看，这是特别有用的。购买应用程序的用户可以用“星星”的数量来进行评价，并发布评论信息。 评分和评论都是公开的，其他用户和开发者都可以看到。 这样的反馈允许用户驱动的质量评估和营销。 评级较高的应用程序在“顶级列表”中排名较高，这反过来又增加了应用程序的可见性和下载数量。反馈还允许用户在应用程序上给出和获取建议，类似于酒店预订站点。

有趣的是，用户似乎也向应用程序供应商和开发人员提供一些有助于提高软件质量并识别丢失的功能[ 12 ]的反馈。通过快速浏览（例如，AppStore）显示了用户发布功能请求和其他消息以改进应用程序。产生了以下三个问题：

1. 应用商店可以作为用户和开发者的沟通渠道吗？
2. 开发者如何使用应用商店去更快更好的理解用户不断变化的要求和需求？
3. 哪些方法和工具可以支持软件组织去分析、汇总并利用这些用户反馈？

本文调查了一个实证研究，这是回答这些问题的第一个基石。 我们旨在了解应用分发平台中用户反馈的现状。 本文研究探讨了三个方面。 首先，探讨用户如何以及何时提供反馈，分析反馈元数据与其对应用程序流行度的影响之间的相关性。 其次，我们的研究确定并分类用户评论中的主题，它们的共同点，流行度和影响。 第三，本文对研究人员和工具供应商如何系统地收集和使用建设性反馈以及如何将其整合到需求和软件工程基础结构中提供了深入的见解。

我们首先介绍研究问题，方法和数据（第二节）。 然后，我们总结了三个研究问题的结果：反馈用法，内容和对用户群的影响（第三节）。 在讨论我们的发现（第四节）并提出研究的局限性（第五节）之后，我们对相关工作进行了调查（第六节），并勾画了未来的研究方向（第七节）。

1https://itunes.apple.com/us/genre/ios/id36?mt=8

2https://play.google.com/store/apps

3http://www.windowsphone.com/en-us/store

二、研究设计

A.研究问题

我们研究用户反馈功能的使用情况，反馈内容及其对用户群体的影响。

**反馈用法**描述应用程序用户如何提供反馈 为此，我们研究以下问题：

•反馈频率：用户提供反馈的频率和频率

•反馈元数据：在反馈长度，评分和实用性方面，典型的用户反馈是什么？

**反馈内容**描述反馈信息中提供的主题（即语义实体）及其频率。 我们特别调查以下问题：

•反馈类型：存在哪些类型的用户反馈？

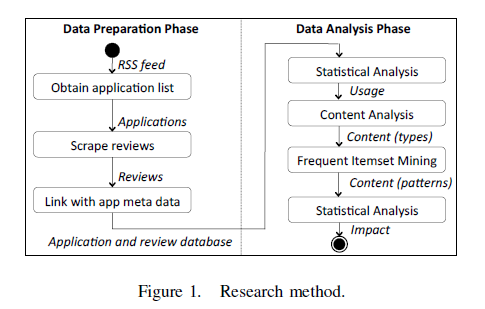
•反馈模式：在反馈类型的共同发生中是否存在模式？

**反馈影响**描述用户反馈是否影响其他用户的评分以及他们决定使用该应用程序。 我们研究以下问题：

•市场影响：具体的反馈类型或反馈模式是否会影响评级？

•社区影响：用户群体比其他人更赞赏具体的反馈意见吗？

术语反馈，我们指的是评分，评论，或两者兼而有之。回答这些问题时，我们区分免费和付费应用程序，分析定价是否对研究问题产生影响。



## B.研究方法

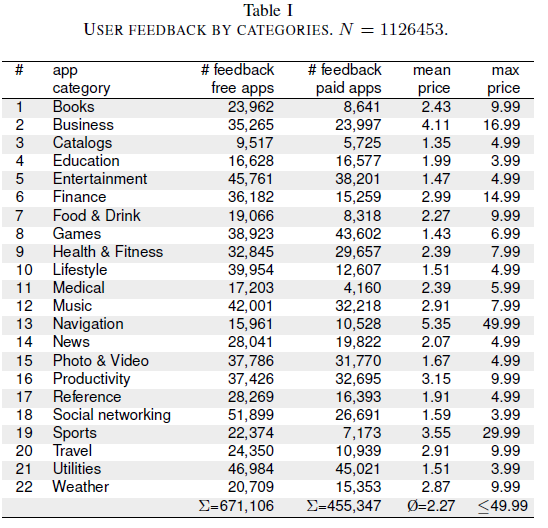
我们的研究方法包括两个阶段：数据准备和数据分析阶段，如图1所示。

1）数据准备阶段：在数据准备阶段，我们收集数据并从中创建数据库。 我们首先调查了三个最大的应用程序分发平台和可用的数据：苹果的AppStore，谷歌的Play和微软的Windows Phone商店。 这三者都包括我们感兴趣的可比数据：关于应用程序的信息和用户编写的相关评论以及评分。 我们决定使用苹果的AppStore，因为我们有技术和应用方面的经验，并且因为我们发现有可能以编程方式接收数据。 我们在不同的应用程序分发平台之间进行比较以供将来的研究。

在AppStore中，应用程序属于22个不同的类别4，以帮助用户更快地找到应用程序。此外，应用程序分为免费和付费。由于这些分类，我们决定对数据进行分层抽样，包括为每个类别提供相同数量的免费和付费应用程序。

2012年9月16日，我们使用Apple提供的RSS feed生成器查询了每个类别中前25个免费和付费应用程序的列表5。 此列表包含了下载最多的应用程序，并且每天更新。 接下来，我们解析列表以提取包含的应用程序的标识符。然后，我们使用了一个开源的抓取工具6，通过修改这个工具去遍历所有的应用标识符来删除评论列表。 在最后一步中，我们将每个应用程序的元数据（如应用程序名称，发布日期，价格等）与获得的评论列表链接起来，并将结果插入MySQL数据库。

2）数据分析阶段：数据分析阶段由三个步骤组成，分别研究使用情况，内容和影响。 为了分析反馈使用情况，我们应用了描述性统计。我们还进行了统计测试，排除危险因素并报告错误率。为了探索反馈内容并找到包含的主题，两位研究人员独立地对我们的数据集中的随机样本进行了手动内容分析[11]。之后，我们应用频繁项集挖掘[18]来识别主题中的潜在模式。最后，为了研究影响，我们将反馈元数据和内容结合在一起，并对结果数据进行统计分析。我们在相应的结果部分对每个分析步骤进行详细说明。



4excluding “Newsstand”, which is an additional tag rather than a category.

5http://itunes.apple.com/rss/

6https://github.com/oklahomaok/AppStoreReview

## C.研究数据

表I显示了我们的数据集的概述。 我们总共从1,100份申请中获得了1,126,453份评论（550份免费，550份付费）。 在AppStore中，每个版本的应用程序都会重置针对特定应用程序的评论。 因此，我们的数据集中的评论是在相应申请的最后一个版本之后完全发布的。 不到一半的评论（518,041或45.99％）指定了审查的应用程序版本。 我们无法解释清楚，但我们推测用户可以通过浏览器或通过AppStore软件输入反馈。 只有后者才能访问已安装的版本。

同样，一些评论没有指定他们的发布日期。但是，由于我们按照出现的顺序获得了反馈，并且由于反馈日期只包括当天，所以我们可以根据前任和后继评审计算所有丢失的日期。 最早的反馈是在2008年7月10日输入的。因此，我们的数据超过了4年。

大多数免费应用程序的评论都写在“社交网络”（51,889 - 7.73％）类别中，至少在“目录”（9,517 - 1.42％）类别中。 我们的数据集中的大多数付费应用程序评论属于“公用事业”（45,021 - 9.89％）类别，最低的评论发表在“医疗”（4,160 - 0.91％）类别。 平均而言，最昂贵的应用程序属于“导航”类别（平均5.35美元），而“目录”类别中的应用程序是最便宜的（平均1.35美元）。 总的来说，我们数据集中的平均付费应用程序的成本为2.27美元，而“导航”类别的最高价格为49.99美元。 我们的完整数据集（包括免费应用程序）的平均应用程序价格为0.92美元。

# 三、结果

## A.反馈用法

1）反馈频率：我们的数据包含1,126,453条评论来自918,433位不同的评论者，即每个评论者大约1.23条评论。 在这些评论中，671,106个适用于免费应用程序，而455,347个是为付费应用程序编写的。 这种差异是显着的（双样本t检验，p <0.001，CI = 0.99）。 因此，我们得出结论，总的来说，免费应用的反馈比付费应用更多。 大多数情况下，免费应用程序的用户群体较大。

在我们的数据集中，我们计算了568,599位不同的免费应用评论者和389,563位不同的付费应用评论作者。 这意味着，免费应用程序（1.46：1）的用户比例与免费应用程序（1.47：1）的查看率相匹配。

我们首先研究了个别记者的评论数量。 我们发现84,567条评论（7.51％）是由“Anonymous”发布的。 此外，我们还观察到其他几个匿名用户名，如“????”（353个评论）或“???”（330个评论）。 我们的数据中，前1000位用户中共有57位用户拥有这样的匿名用户名，共有87,282次匿名评论（7,75％）。 剩余的评论是由其他非匿名用户名撰写的，即使他们真的确定用户是否有疑问。 总的来说，每个用户发布的评论似乎遵循幂律分布。 我们的数据集中有826,874（90.03％）的评论者仅写下1条反馈。 相比之下，只有1,183（0.13％）的评论超过了5条。 这表明一小部分用户非常活跃并不断给予反馈。

平均每个应用的用户社区每天提供22.09条评论。 同样，更多的用户免费提供反馈，而不是付费应用。 我们发现免费每日评论36.87，每天只有7.18付费应用程序的评论（双样本t检验，p <0.001）。 根据中位数，不同价格类别的每日反馈数量仅略有不同，在0.38美元的应用程序中，价格为14.99美元，3.32美元为免费应用程序。

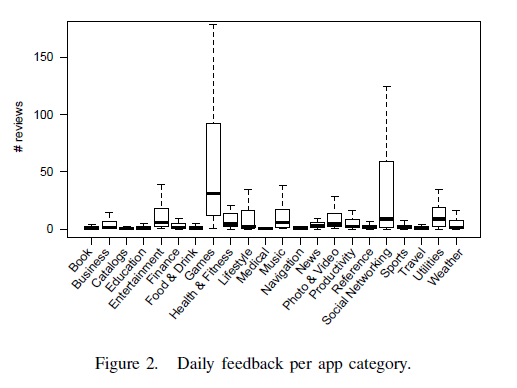
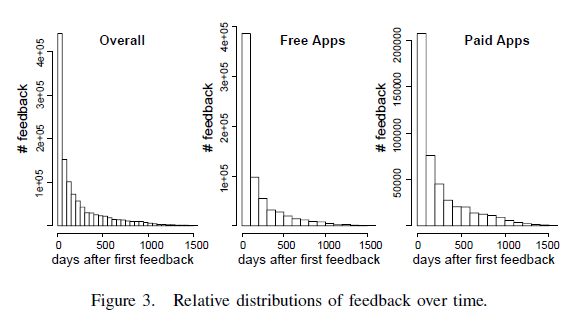


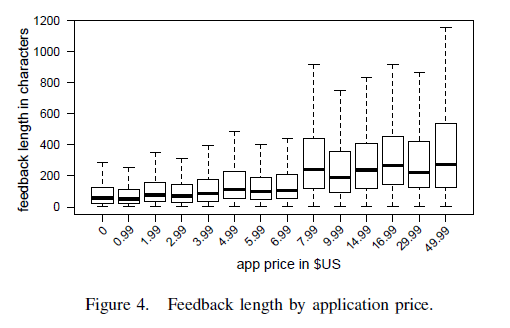
图2显示了为各种应用程序类别提交的反馈数量。由于分布是积极的倾斜，我们使用中位数来报告平均每日反馈。用户每天在游戏类别（中位数为31.24）中发布大部分反馈，其次是社交网络（中位数为8.82）和公用事业（中位数为8.75）。在目录（中位数0.30），医疗（中位数0.34）和书籍（中位数0.53）中提供了每日最少的反馈。对这个结果的一个明显的解释是流行的类别，如大用户社区的游戏获得更多的反馈，而小社区的利基类别得到较少。另一个潜在的原因是目录和书籍等类别的应用程序主要提供信息渲染功能，而游戏和实用程序则提供更复杂的功能集，从而激发更多的反馈。最后，我们可以假设游戏和社交网络等特定的社交领域鼓励用户的表达，他们花更多的时间在应用程序上，并与之建立特殊的关系。

在应用程序级别上，大多数反馈都是针对社交网络类别排名第六的免费应用程序Facebook提供的。 对于这个应用程序，用户仅仅在一天内就发布了4,275条评论。 这些评级的平均值是3.95星。 我们的数据集中最少的反馈是为应用程序Packers Radio＆Live Scores提供的，它在体育类别中排名第16，成本为0.99美元。 在这种情况下，有2个用户在303天内提供了反馈。 他们两个给了5颗星。

最后，我们调查了用户的反馈行为。 我们首先估计每个反馈和第一个反馈之间的时间间隔。 从时间跨度来看，我们计算了反馈随时间的分布。 图3显示，用户随着时间的推移可以快速减少反馈。 尽管分布不是指数性的（Kolmogorov-Smirnov检验拒绝这一假设，p <0.001），但是用户在释放后的最初几天给出了最大的倒退，导致随着时间的推移长尾。 这表明用户反馈是由新版本触发的。

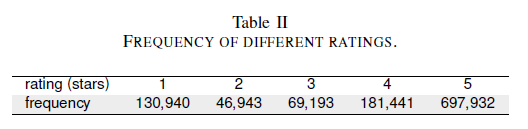
2）*反馈元数据*：我们首先研究了数据集中**反馈的长度**。 总的来说，反馈长度范围从1个字符到6000个字符。 所有应用程序的中位反馈长度为61个字符（平均值为106.09）。 2,802（0.25％）个评论仅包含一个字符，而6个应用程序评论包含6,000个字符。 这些非常长的文本都没有包含有用的信息。相反，相应的用户似乎已经写了随机字符或重复的字符和空白序列来产生视觉模式。 所有评论（863,951）的76.7％包括少于140个字符，这是Twitter消息的长度，而80.4％的评论（905,974）包含的字符少于SMS文本消息（160）。 超过99％的反馈包含少于675个字符，相当于打印页面的三分之一左右。 因此，我们得出结论：应用程序反馈主要由短消息组成，与Twitter相比，更像是类似电子邮件的其他通信工件。





从应用程序供应商的角度来看，对这一结果的一种可能的解释是，许多信息（而不是全部信息）是无用的，激发了系统的手段来过滤不相关的反馈。 从用户的角度来看，这个结果显示用户没有分配太多的时间来给予反馈。 众所周知，移动用户除了使用文本外还使用多媒体来传达复杂的内容[13]，[14]。

如图4所示，反馈时长似乎随着应用价格的上涨而增加。 尽管我们无法直接发现应用程序价格和反馈长度之间的显着线性相关性，但我们能够显示较低价格和较高价格应用程序之间反馈长度的显着增加。 为此，我们将数据分成两个不相交的集合。 较低价格套餐包括价格高达6.99美元的应用程序，而较高价格套餐则包含所有昂贵的应用程序。 双样本Wilcoxon秩和检验显示，用户写出价格较高的反馈比较低价格的申请要长得多（p <0.001）。



反馈长度似乎与星级中的评级有关。 通过使用一系列双样本Wilcoxon秩和检验，我们可以根据其等级显示反馈的中位数长度的以下关系：

m1 < m2 ∧ m2 > m3 > m4 > m5 ∧ m1 > m3 (p<0.001).

我们认为这个结果是相当有趣的。 一种解释是，用户倾向于写的越多，他们喜欢的应用程序就越少，这表明改进的空间不大。 相反，这表明书面反馈主要用于改进请求。

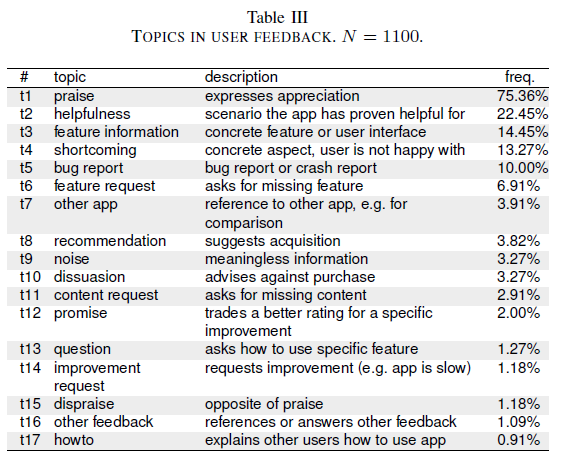
我们的数据集的**评分**总体非常乐观，导致平均评级为4.13星。 697,932（61.96％）的评价包含5星评级，879,373（78.07％）给出至少4星级。 只有130,940（11.62％）条评论评分最低（1星级）。 表二显示了基本频率分布的概况。 我们通过查询每个可用类别中最受欢迎的下载应用程序来获得我们的数据集。 尽管我们的数据样本包含大量的低收视率（177,887或15.79％的评论少于3星），但是我们不能排除应用程序的排名对整体收视率的影响（反之亦然）。 独立性的χ2检验证实了我们的数据中等级和等级不独立的假设（χ2= 2,661,333，p <0.001）。

在AppStore中，用户可以通过评估其**实用性**来评估现有反馈的质量。 在我们的数据集中，只有67,143（5.96％）的评论被其他用户评价为有帮助。 从这些，38,519（57.37％）被认为是100％有帮助。 只有19,118（28.47％）的评级评级被超过一半的评级用户评为无用。 有趣的是，16,671（24.83％）被评为完全无用。 这意味着用户群体在超过82％的案件中同意他们的意见。使用Hartigans的倾向测试[7]，我们可以发现有用性的分布是双峰的（p <0.001）。 事实证明，如果对反馈进行评估，则被用户认为是非常有帮助的或非常无用的。

## B.反馈内容

为了调查我们数据集中用户反馈的内容，我们采用了迭代内容分析技术。 我们首先画了一个分层随机样本[4]我们的数据。 分层抽样将人口分为相互排斥的群体，对应于我们数据集中的应用类别和定价模型。 我们从22个应用类别的每一个中随机选择了12个免费和付费应用的评论，得到了528个评论。 我们觉得这个门槛足够大，可以进行统计测试，而手工分析仍然可行。 接下来，两名研究人员独立“编码”随机样本，以确定反馈中包含的主题。 我们允许分配多个主题，因为大多数反馈包含的不止一个主题。

编码过程包括对单个评论的两个迭代。 我们从一组空白的话题开始。 然后，我们阅读每篇评论，并确定一个描述所包含信息的主题。 如果相应的话题在我们的列表中，我们使用它，否则我们添加一个简短的描述和一个例子。 这些描述由编码人员共享，使他们能够“解读”书面反馈。 我们没有严格地查找语法实体，而是查找语义实体。 例如，“赞”这个话题可以分配给包括“酷！”在内的反馈。 “太好了”，或“真棒”。 另一方面，有时我们必须解释讽刺或讽刺的反馈，例如“我失去了我所有的手机通讯录。 太好了谢谢！”。 这是使用手动分析而不是文本挖掘的主要动机。



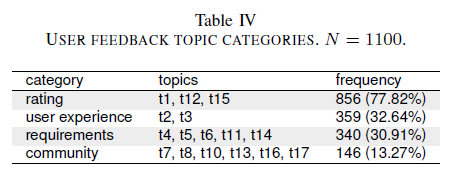
我们获得了两组主题，一组为T1有16项，T2组有21项。 通过讨论确定的主题，我们发现T1是T2的一个严格的子集，但是T2允许在某些信息实体上有更多的具体区别。 因此，我们决定以T2作为进一步编码的基础，作一个修改：我们删除了4个最不频繁的话题，因为它们只在1或2个评论中被发现。 由此产生的17个主题作为我们对其余内容分析的编码准则。

在最终的编码轮次之前，我们将随机样本加倍为1,100条评论。 这使我们能够在95％的置信度下对我们的数据集进行预测，接受3％的误差。 两位研究人员再将这个较大的样本编码，彼此独立。 最后，我们讨论了分配代码不匹配的反馈，并一起决定最终代码。

*1）反馈类型：*最终结果如表3所示。 我们发现最受欢迎的话题就是“表扬”，表示赞美这个应用。 这个主题在75％以上的分析样本中占主导地位。 第二个最流行的话题是“实用性”，它描述了一个用例或情况，其中应用程序被证明对用户有帮助。 这在20％以上的反馈中占主导地位。 进一步的话题“特征信息”以及“缺点”主要在超过13％的分析反馈样本。

在阅读样本反馈的同时，我们做了三个有趣的研究。首先，我们的随机样本的反馈质量相当丰富。一方面，高品质的应用程序评论，提出了有趣的新功能，深刻地证明了他们的建议。 另一方面，一些反馈不会增加应用程序的数字评分的任何价值。 其次，用户往往很快产生侮辱性言论。 尤其是当他们花钱的时候，他们似乎丝毫不会饶恕开发者。 像“消灭设计这个应用程序的白痴”这样的反馈远没有建设性，并且解释了开发者对用户反馈的不满。 第三，用户似乎经常抱怨删除或更改的功能。我们将此视为用户习惯使用应用程序的特定工作流程的指标。以影响用户工作流程的方式更改应用程序似乎是一个不满的源头。

为了估计反馈中的信息多样性，我们计算了每个反馈的主题数量分布。 总的来说，有6条（0.55％）评论包含5条主题，22条（2.00％）有4条评论，116条（10.55％）条评论包括3条主题，427条（38.82％）2条主题，而528条（48.00％ 只包含一个主题。 换句话说，大多数反馈（52％）包含多个主题。



为了进一步解释和比较反馈信息，我们将所得到的主题分为四个主题：

1）社区。 这些话题代表社区和社会方面。 具体来说，引用其他反馈和其他应用程序，向其他用户提出的问题，解释如何使用应用程序的howtos，以及建议和劝阻都包含在这个主题中。

2）需求。这个主题捕捉与应用程序改进相关的主题。 所有请求 - 功能，内容和改进请求 - 以及缺陷和错误报告属于这个主题。

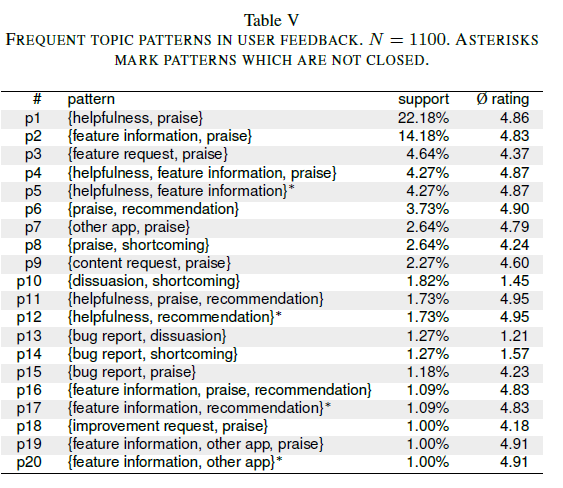
3）评分。这个主题包括与申请的判断有关的主题，即赞美和不赞成，但是也承诺，表达了用户改变其判断的意图，并作了一些改进。

4）用户体验。这个主题包含与应用程序描述相关的主题。 这些是有帮助的，它捕获应用程序证明有用的用例，以及包含应用程序功能和用户界面描述的功能信息。

表IV总结了已确定的主题和所包含的主题，并显示了它们在随机样本中的频率。 频率表示包括至少一个相关主题的反馈的数量和百分比。 在我们的随机样本中，“评级”是迄今为止最频繁的主题，频率超过77％。 这与分配本身之后的应用分发平台背后的主要意图相对应，为其他用户提供了良好的应用指标，从而保证了应用的高质量。第二个最常见的主题是“用户体验”，在所有反馈中占三分之一。我们认为这个主题的高度流行是有趣的。它表明，用户倾向于与其他用户和开发人员分享他们的经验，可能是为了证明他们对应用程序的评价，建议或劝阻。所有反馈中大约有30％是主题“需求”。 这表明，尽管评分整体相当正面，但用户往往将改进的要求外化。 最后，“共同体”的主题出现在分析样本中的13％左右。 鉴于AppStore的最初目的，我们认为这个数字相当高。这可能表明用户自然倾向于形成社区，即彼此作出反应，提出问题或发布可能有用的信息。

*2）反馈模式：*我们使用Zaki [18]提出的频繁项集挖掘算法ECLAT来识别数据样本中主题的共现。 项集挖掘[1]是一种数据挖掘方法，用于根据数据库中的同现发现不同变量之间的关系。 它将包含至少两个不同变量的数据库以及指定最小支持度σ的参数作为输入  
为了发现关系。 输出包括频繁项目集，即在数据的至少σ％中共同出现的一组值。 我们的目标是找到一系列在我们的数据集合中出现的主题比其他主题更高的主题。 因此，我们建立了一个数据库，其中包含我们随机样本中每个反馈的主题代码。 我们运行ECLAT算法，最小支持度为σ= 0.01，最小模式长度为2，这意味着结果应该只包含至少有两个主题的项目集。

表V显示了我们用这些阈值获得的20个模式。最常见的模式是{实用性，赞美}，超过所有反馈的20％。它描述了应用程序的有用性和积极的评价。我们的数据集中的一个具体例子是“非常适合在旅途中上传收据。比在计算机上协调起来要容易得多。“应用于我们随机样本的14％以上的第二最常见模式{特征信息，赞美}类似于第一种，不同之处在于它更具体地描述了一个积极的特征或功能一个应用程序。在我们的数据集中的一个具体例子是“我喜欢这个应用程序不到十秒钟，让你知道你的电池寿命！我喜欢它。“第三种模式{功能请求，赞美}占我们随机样本的近5％占优势。它说明了也包含功能请求的正面反馈。从这样的模式中，我们期望比第一种模式更低的应用评分。为了验证这个假设，我们在下面的章节中调查反馈内容和评分之间的规律性。

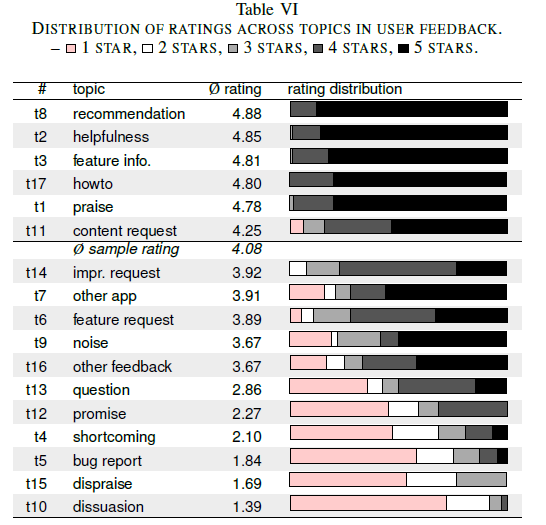


## C.反馈影响

1）市场影响：为了研究反馈的市场影响，我们研究了应用评级与反馈主题之间的关系以及反馈模式。我们首先通过一系列χ2检验测试了确定的主题的独立性和用户的最终评分。结果表明，“其他应用程序”（p = 0.90），“其他反馈”（p = 0.69）和“howto”（p = 0.56）这两个主题与评分无关，而其他的则不是（p <0.05 ）。

为了进一步研究特定反馈类型对应用评分的影响，我们为每个主题计算了相关评分的分布情况以及所有反馈的平均评分。表6说明了结果，并将其与我们随机样本中反馈的总体平均评分相关联。导致最积极评价的主题是“推荐”，其次是“实用性”和“特征信息”，而负面评价最多的主题是“劝阻”，其次是“差评”和“错误报告”。

这些结果允许两种解释。首先，我们可以根据它们对用户评分的影响来排列需求主题。内容请求（平均4.25颗星）是最不重要的要求反馈。他们的平均评级甚至高于样本的整体平均评级。改进要求（平均3.92颗星）和功能要求（平均3.89颗星）更为关键，但平均评级仍高于三颗星的理论中间。缺点（平均2.10颗星）对评级有一定的负面影响。 Bug报告（平均1.84颗星）是最关键的。其次，结果表明，用户体验主题主要包括在正面评论中。换句话说，用户不会倾向于将自己的经验包含在负面反馈的应用程序中。由于这些信息对于纠正性维护是至关重要的[20]，所以这一发现表明用户反馈意见不太可能帮助开发人员改进其应用。

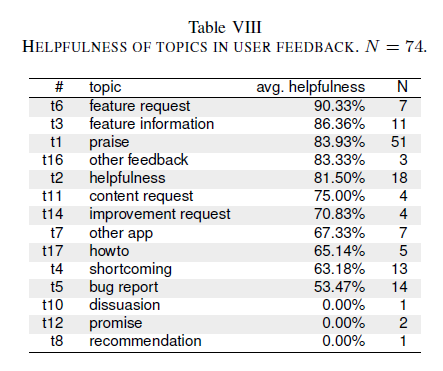


为了进一步打破主题对评级的影响，我们计算了每个评级的前五个主题。表7显示确认我们发现的结果。缺点，错误报告和功能要求对负面评价有很大的影响，对正面评价有帮助和特征信息。

接下来，我们调查了反馈模式与具体评级之间的关系。为此，我们计算了每个识别模式的平均评分。结果列于表五中。不同模式的平均评分差异很大。最积极的（封闭的）模式是{有帮助，赞美，推荐}，平均评分为4.95星。正如我们假设的那样，{功能请求，赞美}的平均评分较低（4.37星）。{错误报告，劝阻}是最低的平均评分（1.21星），这对应于报告错误和阻止其他用户购买应用程序的负面信息。

*2）社区影响：*为了研究反馈对用户群体的影响，我们研究了反馈有用性与反馈的长度，内容以及评分之间的关系。 乍看之下，我们无法观察到反馈长度与其他用户的有用性之间的任何线性或多项式相关性。 然而，χ2检验显示反馈的长度和其他用户评价的有用性在统计上是相关的（χ2= 2043547，p <0.001）。 为了进一步研究这一现象，我们根据其反馈意见将反馈分成三组。 低有用性表明，高达33％的评价反馈意见的用户认为它有帮助。 中等乐于助人指出评价反馈的用户中有33-66％认为有帮助。 高度乐于助人，包括超过66％的用户认为有帮助的反馈。

通过使用一系列双样本Wilcoxon秩和检验，我们能够根据其有用性显示以下关于中值反馈长度的关系：mlow <mmedium> mhigh∧mlow <mhigh（p <0.001）。 这个结果意味着低帮助性的反馈是最短的，而中等有帮助性的反馈是我们的数据集中最长的。 当超出描述水平时，低和高有用性反馈之间的长度差别的意义是有争议的，因为中间长度仅相差7个字符(mlow = 114,mmedium = 144,mhigh = 121)。 相反，中等乐于反馈的差异在23（19.01％）和30（26.32％）个字符之间。

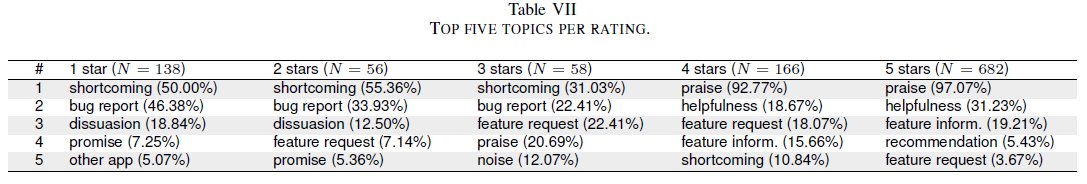


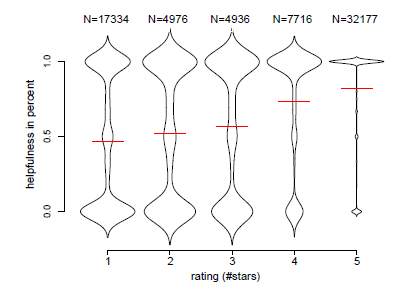
值得注意的是，所有评价有帮助的反馈都比其他反馈要长得多。共有67,143条点评数据集中位数长度为121个字符，其余1,059,310条评论中位数长度为58个字符，不到一半。这种差异是显着的（两样本Wilcoxon秩和检验，P <0.001），证实了更长的反馈更可能被其他人评价，可能是因为它包含更多的信息。

接下来，我们研究了反馈有用性与反馈内容的关系。不幸的是，随机样本中只有74条评论（6.73％）用于探索反馈话题，根据其有用性评分。这个数字不允许我们概括主题和乐于助人之间的关系。尽管如此，为了探索可能的关系，我们计算了这个集合的每个主题的平均乐于助人。表八显示了最有用的反馈中包含的主题，功能请求，功能信息和赞扬，而最有帮助的反馈则包含了推荐，承诺和劝阻。

最后，我们调查了反馈评级与乐于助人之间的关系。 χ2检验显示这些变量在我们的数据集中是统计学依赖的（χ2= 11952.97，p <0.001）。 为了进一步分析这种关系，我们计算了每个反馈评分的平均乐于助人。 图5以小提琴曲线序列的形式显示了结果，每个评分一个。 小提琴图与盒子图相似，但也说明了基础数据的概率密度。 平均乐于助人的评分标有红线。 一系列双样本t检验显示，根据其等级，平均反馈有用性具有以下显着关系：m1 <m2 <m3 <m4 <m5（p <0.001）。

这个结果意味着更好地评估应用程序的反馈被认为对其他用户更有帮助。 这个结果的一个可能的解释在于应用程序商店的主要用例，即允许用户找到好的应用程序。 我们认为用户浏览反馈意见是为了了解应用程序是否被社区认为是高质量的。 在这种情况下，他们购买猪的风险很低。 因此，用户可能会考虑更好的评论，通常将我们发现的应用程序特性和用例描述为更好的决策支持，从而评估它们更有帮助。 相比之下，很难想象用户会花时间将反馈评为有用的，从而给应用程序带来较低的评分。 在这种情况下，用户宁愿离开下载区域并浏览替代方案。





# 四、讨论

我们通过回顾引言中的主要动机问题来讨论我们的发现及其含义。

**1）应用商店可以作为用户和开发者之间的沟通渠道吗？** 我们的结果显示答案是肯定的。 应用用户每天不断发表数十条评论，解决开发人员和用户社区的问题。 对开发者的评论从赞美和感谢，例如 “伟大的应用程序，正是我所需要的”。 “自更新后，我无法打开电子邮件链接，请修复”。 “如果我低于一定的金额，我希望它会通知我”。 社区主题包括向其他用户提供的建议，关于特定功能的直接问题，这些问题的答案或关于如何使用应用程序的解释。

目前的应用程序商店缺乏双向通信功能，如回复或引用帖子（如在社交媒体中所知）。 这可以防止开发人员联系特定的用户，要求澄清问题或通知他们他们的问题已经解决。 由于大多数用户在新发布后很快发表评论，增强沟通的一种可能方式是将更改和新功能链接到用户反馈，从而影响或导致这些更改。 这将增加用户的参与，并帮助理解变化的基本原理。 我们也认为用户应该亲自参与个人账户和个人资料。 这可以防止不合格的内容，增加用户的动机，并允许开发者联系特定的用户。

**2）开发人员如何使用应用商店更好，更快地了解不断变化的用户需求？**即使大多数反馈重复自然语言的评分，用户也经常在AppStore中分享他们的需求，想法和经验，这是应用程序设计师和分析师的重要信息来源。其他研究表明，开发者将用户反馈作为重要的信息来提高软件质量和识别缺失的特征[12]。

大约三分之一的反馈包括软件要求和用户体验方面的主题，从缺点和功能需求到应用程序有用的场景以及特定功能的描述。需求分析师可以立即从这个反馈中获益，获得关于他们的应用程序如何被实际使用的见解，而不仅仅是简单的下载和销售数量，并且能够评估功能的重要性，这可能有助于发布规划。而且，在他们自己的需求推动下，一些用户开发和分享了其他人可能从中受益的扩展，解决方法和想法，并且可能为新功能提供灵感和想法[16]。最后，像特征描述或howtos这样的反馈可以用作文档的起点。

我们发现评论通常是简短的消息，类似于推文。在电影或酒店等其他领域，反馈往往会延长3-4倍[9]。用户似乎不愿花太多的时间写评论，而是专注于他们正在执行的应用程序的实际任务。但是，反馈的数量，频率和内容表明用户愿意分享他们的意见，经验和想法。这表明软件供应商应该调查最小化反馈提交工作的方法，例如，通过提供文字描述和主动鼓励用户分享他们的经验。在同样的方向，我们发现用户体验往往缺乏负面评论，如缺点和错误报告。这阻碍了开发者仅仅从这个文本反馈中改进他们的应用[20]。克服这种“僵局”的一个有希望的方法是通过对应用程序进行测试并在运行时监视用户的交互来自动收集用户体验[10]。情境数据将丰富文本反馈，并帮助开发人员更好地了解他们出现的需求和具体情况（即上下文）。

**3）哪些方法和工具应该支持软件组织分析和汇总用户反馈？** 应用商店包括有用的反馈，但也包括噪音 我们发现反馈质量差别很大，从其他用户和开发者的有用信息到侮辱性攻击。 因此，分析人员和开发人员将从系统地过滤和汇总反馈的工具和方法中获益。 特别是，我们认为研究人员应该调查预测模型，对反馈进行分类（例如错误报告和缺陷），筛选开发人员必须作出的重要“警告”，或者尽早评估项目的成功。 对采矿错误报告已经有相当多的研究，这些报告也可以在应用商店中进行测试。 检测重复或类似的反馈。

我们还发现了几个规律，可以用来建立这样的模型。 首先，我们的结果显示，大多数反馈在新版本发布后不久提供，随着时间的推移频率快速下降。 这意味着反馈与发布中的新功能和更改有关。 通过分析几个版本的反馈，可以及早发现有问题的版本或功能。 其次，一个重要的发现是，社区通常赞同反馈的有用性，这可以将其汇总。 第三，我们确定了数字评分，乐于助人和文字反馈之间的几个相关性。 我们假设如果有更多关于用户，版本和使用数据的元数据可用，那么将会有更多的相关性。 最后，我们手动分析的结果是一个注释的真实数据的语料库7，它可以用来训练一个分类器，例如 在新提交的反馈中确定可操作的数据[5]。

我们发现用户倾向于给出他们的评级的理由。这一发现的一个含义是，用户意识到评级潜力，愿意为供应商提供有用的解释。事实上，我们的研究证实，反馈内容对市场有真正的影响，因为更积极的信息通常也会导致更好的应用程序评级，反之亦然。

一般来说，正面的反馈更受用户群体的赞赏。这可以通过应用程序分发平台的主要用例来解释，即支持用户找到好的应用程序。同样，更长的反馈更可能被其他用户视为有帮助，可能是因为它包含更多的信息。从方法论的角度来看，我们确信在软件开发生命周期中集成用户反馈收集和分析是非常重要的。我们认为，分析应用商店中的反馈意见可以补充其他需求的工程活动，例如研讨会和访谈，尤其是在时间紧迫的情况下，并且计划了许多增量发布。

7The study data sets are available at <http://mining.socialse.org>

# 五、有效性结论

我们的研究既不是为所有应用程序商店推广也不代表的。 但是，我们认为结果具有高度的普遍性，特别是对于Google Play和Windows Phone Store。 虽然整个人口是公开的，但我们认为我们的结果对于AppStore中的用户反馈具有代表性。 首先，我们的数据集包括来自1,100个最下载的应用程序的所有用户反馈。 其次，我们进行统计检验，检查结果的显着性，排除危险因素。 第三，描述性统计和内容分析的结果是相似的。 然而，有三个限制，解释结果时应该考虑。

首先，我们根据最下载的应用程序清单从AppStore获取反馈，从而获得我们的数据。这可能会导致下载数量与其他变量（如收视率）之间的关系。但是由于我们的数据样本仍然包含大量的低收视率（177,887或15.79％的评论少于3星），我们对结果充满信心。具体而言，我们感兴趣的是评分如何与内容和有用性等其他变量相关联，而不是在总体平均评分中找到绝对数量。

其次，为了探索用户反馈中的主题，我们从我们的数据集中抽取了1,100条评论，因为对完整数据集的手动分析是不可行的。统计证据不如完整数据集强。我们试图通过选择一个样本规模来减轻这种威胁，这样就可以将我们的数据集推广到95％的置信水平，接受3％的误差。此外，我们选择了一个分层随机样本，这保证了所得到的样本在数据集中同样考虑了不同的类别。

第三，我们基于少数评论研究了反馈有用性和内容之间的关系。然而，其他分析，例如评级和有用性之间的关系间接证实了我们的发现。不过，我们鼓励其他研究人员复制我们的内容分析研究，有目的地根据其有用性选择反馈。

最后，我们在分析过程中做了两个简化假设，这可能部分地限制了结果的构造和内部效度。一些评论没有明确指定发布日期，所以我们需要插入这些值。由于我们按照网上订单的顺序获得了反馈，因此我们可以通过调查前后直接发布的评论日期来补偿缺失值。我们有信心，这不会影响结果数据，因为反馈日期只包括一天，数据差距不会超过1天。特别是，该算法从不改变反馈的顺序。

为了分析反馈内容，我们依靠手动分析。结果受到实验者的偏见。为了减少这种风险，两位研究人员进行了独立的对分析。我们通过改进评级标准来改进评级标准，重复这一活动。我们只报告了协议超过90％的结果。

# 六、相关工作

应用程序分发平台是最近的现象。尽管如此，已经有一些关于他们的研究。 Chen和Liu [3]对应用程序在AppStore中的普及进行了初步的研究。他们发现，排名最高的付费应用程序不一定与客户评分密切相关，这一点我们的研究证实。相反，哈曼等。[6]挖掘黑莓应用程序商店的技术，客户和业务方面的应用程序。作者发现应用程序的评级与其下载数量之间有很强的相关性，而价格和评级以及价格和下载数量之间似乎没有相关性。虽然我们发现应用程序的评级和排名在我们的数据集中不是统计独立的，但是我们无法显示相关性。相反，我们描述了价格和反馈长度之间的关系，并且显示用户为更昂贵的应用程序编写更多反馈。

周等人[19]调查了重新包装第三方Android应用程序的现象。作者发现，开发人员经常重新打包合法的应用程序从原来的商店分发他们在第三方市场。周等人显示第三方市场上托管的应用程序有5-13％被重新包装，重新包装的主要用途是替换现有的应用程序内广告或嵌入其他应用程序以“窃取”或重新路由广告收入，这说明了应用程序分发平台作为软件公司的商业模式的重要性日益增加。Yamakami [17]从商业角度分析应用程序商店，介绍潜在的商业模式和关键的成功因素。作者指出，病毒式营销取决于用户谈论他们在应用程序商店和社交网络中的经验。我们的研究证实了这一发现，并显示负面评论中通常缺少用户体验的描述。

Chandy和Gu [2]根据最近的“假”评论，旨在对AppStore中的垃圾邮件进行分类。 此类评论可能会欺骗用户下载垃圾邮件应用程序，或者忽略受到垃圾评论负面影响的应用程序。 作者提出了一个潜在的模型，能够将应用程序，开发人员，用户和评论归类为正常和恶意类别。 我们的研究证实，评论对市场和用户群体有影响。 此外，它确认了进入顶级列表的重要性，因为顶级下载的应用程序在AppStore中不一定被评为更好。 esearch正在评估应用程序和评论质量。 像Hong等人的研究人员 [8]对评论的自动分类进行工作，以便区分有用和无用的评论。

最后，Seyff等人[15]和施耐德等。 [13]建议用移动设备的反馈不断引出用户需求，包括应用上下文的信息。我们的研究证实了这些方法的重要性，并建议应用程序商店可以作为收集，交换和管理用户需求和用户体验的平台。

# 七、结论

用户反馈和用户参与对现代软件组织至关重要。用户越来越多地评估和审查应用程序分发平台中的应用程序，称为应用商店 尽管这些反馈的部分内容是肤浅的，至多对下载数量有影响，但其他的还包括有用的评论，错误报告，用户体验和功能请求。 这可以帮助开发人员了解用户的需求，将应用程序以“民主”的方式扩展到众包需求。但是，目前的平台不允许开发人员系统地过滤，汇总和分类用户反馈以获取需求，或优先考虑开发工作。 开发人员也将从使用和上下文数据丰富文本反馈中受益。

# 致谢

这项工作得到了EC（FastFix项目FP7-258109）的支持。 我们感谢Bernd Bruegge提供的宝贵意见。

# 文献

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In SIGMOD Conference on Management of Data, pages 207–216, Washington, DC, 1993. ACM.

[2] R. Chandy and H. Gu. Identifying spam in the iOS app store. In Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality - WebQuality ’12, pages 56–59, Lyon, France, 2012. ACM.

[3] M. Chen and X. Liu. Predicting Popularity of Online Distributed Applications: iTunes App Store Case Analysis. In Proceedings of the 2011 iConference, pages 661–663, Seattle, WA, USA, 2011. ACM.

[4] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting Empirical Methods for Software Engineering Research. In Guide to Advanced Empirical Software Engineering, pages 285–311. Springer- Verlag London, 2008.

[5] M. A. Ferrario, W. Simm, J. Whittle, P. Rayson, M. Terzi, and J. Binner. Understanding Actionable Knowledge in Social Media: BBC Question Time and Twitter, a Case Study. In ICWSM, pages 455–458, 2012.

[6] M. Harman, Y. Jia, and Y. Zhang. App store mining and analysis: MSR for app stores. In 9th IEEE Working Conference on Mining Software Repositories, pages 108–111, Zurich, Switzerland, 2012. IEEE.

[7] P. M. Hartigan and J. A. Hartigan. The dip test of unimodality. Annals of Statistics, 13(1):70–84, 1985.

[8] Y. Hong, J. Lu, and J. Yao. What Reviews are Satisfactory: Novel Features for Automatic Helpfulness Voting. In 35th international conference on Research and development in information retrieval, pages 495–504, Portland, Oregon, USA, 2012. ACM.

[9] N. Jakob, S. H. Weber, M. C. Müller, and I. Gurevych. Beyond the Stars: Exploiting Free-Text User Reviews to Improve the Accuracy of Movie Recommendations. In 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, 2009.

[10] W. Maalej and D. Pagano. On the Socialness of Software. In Proceedings of the International Conference on Social Computing and its Applications, Sydney, Australia, 2011. IEEE.

[11] K. A. Neuendorf. The Content Analysis Guidebook. Sage Pub., 2002.

[12] D. Pagano and B. Bruegge. User Involvement in Software Evolution Practice: A Case Study. In Proceedings of the 35th International Conference on Software Engineering, San Francisco, CA, 2013. IEEE.

[13] K. Schneider, S. Meyer, M. Peters, F. Schliephacke, J. Mörschbach, and L. Aguirre. Feedback in Context: Supporting the Evolution of ITEcosystems. In Product-Focused Software Process Improvement, LNCS 6156, pages 191–205. Springer-Verlag Berlin Heidelberg, 2010.

[14] N. Seyff, F. Graf, and N. Maiden. End-user requirements blogging with iRequire. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE ’10, page 285, New York, New York, USA, 2010. ACM Press.

[15] N. Seyff, F. Graf, and N. Maiden. Using Mobile RE Tools to Give End-Users Their Own Voice. In IEEE International Conference on Requirements Engineering, pages 37–46, Sydney, Australia, 2010. IEEE.

[16] E. von Hippel. Lead Users: A Source of Novel Product Concepts. Management Science, 32(7):791–805, July 1986.

[17] T. Yamakami. A Three-Dimension Analysis of Driving Factors for Mobile Application Stores: Implications of Open Mobile Business Engineering. In 2011 IEEE Workshop of International Conference on Advanced Information Networking and Applications, pages 885–889, Biopolis, Singapore, Mar. 2011. IEEE.

[18] M. Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3):372–390, 2000.

[19] W. Zhou, Y. Zhou, X. Jiang, and P. Ning. Detecting Repackaged Smartphone Applications in Third-Party Android Marketplaces. In 2nd conference on Data and Application Security and Privacy, pages 317– 326, San Antonio, Texas, USA, 2012. ACM.

[20] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter, and C. Weiss. What Makes a Good Bug Report? IEEE Transactions on Software Engineering, 36(5):618–643, Sept. 2010.