The Journal of Systems and Software 215 (2024) 112108

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

# Conversation in forums: How software forum posts discuss potential development insights☆

Hechen Wang *, Peter Devine, James Tizard, Seyed Reza Shahamiri, Kelly Blincoe

*Human Aspects of Software Engineering Lab, University of Auckland, New Zealand*

ARTICLE INFO

Dataset link: https://zenodo.org/record/8058229

*Keywords:*
Software product forums
Content analysis
User feedback
Software quality
Contextual information

ABSTRACT

User feedback on software usage is utilised by developers to improve their software. Software product forums are platforms rich in software-related user feedback, such as forum threads containing bug reports or requests for new features. However, previous studies have mainly focused on analysing user feedback from software product forums as individual sentences, which can lead to missing insights and a lack of understanding of the overall context of forum posts. To fill this gap in research, this work examines user feedback found in software product forum posts to investigate the differences between content classifications found in forum sentences and posts. We manually evaluated software product forum posts collected from two open-sourced software product forums and discovered five new types of user feedback that can only be identified when examining user feedback in the form of forum posts. Additionally, we examined the association between sentence classifications found within software product forums. Our results indicate that contextual information complimenting product improvement insights can be found in software product forums, with a confidence of 0.75 and 0.69 for the association between apparent bug and application usage sentences. This information can be used to reduce manual efforts required to chase up missing contextual information when attempting to understand or fix software issues. We also provide insights into the progression of posts in software product forums at the thread-level, and our progression flowchart can be used to summarise the sequence of events in software product forum threads. Our findings reveal the importance of looking at user feedback within software product forums in the format of forum posts to identify new insights on user feedback for software improvements.

*Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.*

## 1. Introduction

User feedback for software often reflects the software users' opinions of the software product. Nowadays, this user feedback can be found all around the internet, whether it is on app stores or within online conversations (Mezouar et al., 2018; Dabrowski et al., 2022). Many existing studies have examined the content of user feedback, mainly focused on the feedback from app stores (de Araújo and Marcacini, 2021), Twitter(now X) (Martens and Maalej, 2019), and recently, Reddit (Ali Khan et al., 2020). These prior studies have shown that the feedback on these channels contains potential product improvement insights and should be collected and examined by developers. On the other hand, software product forums have not been studied as extensively as the other channels despite forums containing valuable discussions between developers and software users (Wang et al., 2021).

Software product forums are online discussion forums built to discuss a specific software product (Holtz et al., 2012). The forums are usually maintained by the developers of that software product, and the main purpose of using the forum is to provide support and gather feedback for the software (Tizard et al., 2019). Threads are the main format of a discussion within a software product forum. Threads consist of an initial title post containing a title and body text and posts written in reply to that initial post. Forum users can discuss a new topic by creating a thread, and discussions around that topic would be posted as replies within the forum thread. Such conversations between forum users and software developers themselves can offer valuable insights into user needs and potential software insights, whether it is from a question asking about a specific feature of the software or a post identifying possible bugs, suggesting the need to improve the software (Tizard et al., 2022a). While some initial work was done

☆ Editor: Nicole Novielli.

* Corresponding author.
*E-mail address:* hwan531@aucklanduni.ac.nz (H. Wang).

https://doi.org/10.1016/j.jss.2024.112108
Received 22 June 2023; Received in revised form 13 March 2024; Accepted 20 May 2024
Available online 24 May 2024

to automatically identify such insights as possible requirements from user feedback contained in software product forums, prior work has analysed only individual sentences within the posts or only the title post for a thread (Tizard et al., 2019; Iqbal et al., 2021). Forum discussions consist of multiple posts in threads. By not considering the full posts, valuable information related to not just requirements but potential product improvement insights might be missed.

One main difference between feedback channels such as the Google Play Store and software product forums is the length of the feedback. Feedback collected from Google Play Store reviews is often shorter sentences with a rating, while user feedback within software product forums contains multiple paragraphs of discussion, usually with multiple people contributing to each thread (Noei and Lyons, 2019; Gottipati et al., 2011). In the context of product improvement insights, such as possible bug reports or feature requests, contextual information is often needed for developers to fully understand the request or to reproduce the issue (Bettenburg et al., 2008). Yet, feedback on channels such as Twitter and the Google Play Store can often miss out on providing contextual information due to the shorter length of feedback provided (Martens and Maalej, 2019). The lack of contextual information in feedback often requires additional manual effort by the software development team to request more details in order to fully understand the potential development oversight and the environment when it happened (Haering et al., 2021). On the other hand, feedback found in software product forums is often longer and contains more details (Tizard et al., 2019). Research from general online forums indicates that forums tend to contain more contextual information (Assimakopoulos and Yan, 2006), but it is not known if this is also true for software product forums.

Prior research has also examined the structure and interactions in general online forums (Bickart and Schindler, 2001), but there have not been studies focusing on how software product forum users interact in threads within the forums, especially when discussing potential product improvement insights. The forum format encourages bidirectional communication between different users, but little is known about the content in replies and how developers can use such information to improve their products. Tizard et al.'s work revealed that forum users were replying to forum posts providing software support for other users that were facing issues, and developers also replied to confirm issues as known bugs or limitations (Tizard et al., 2019). Knowing more about the discussion patterns in threads on software product forums can help software teams understand emerging product improvement insights. For example, if a forum thread receives no support reply from other users or if the potential issue is still unresolved after support is given, it can indicate changes are needed in the software product or its documentation to solve the potential issue.

To address the gaps in research, we look to analyse user feedback from software product forums at a post-level, where we examine user feedback found in posts instead of breaking them down into sentences, as has been done in previous works. We examined the content of two large open-source software product forums: VLC media player and Firefox web browser. We analysed the content of individual forum posts, the associations between the sentences within the forum posts, and the sequence of forum posts within threads. Our analysis was guided by the following research questions:

**RQ1:** *What new product improvement insights can be discovered by examining forum user feedback at a post-level?*

**RQ2:** *How often can we find product improvement insights and contextual information in software product forum posts?*

**RQ3:** *How do potential product improvement insights get treated in software product forum threads?*

The main contributions of this paper are as follows: (1) We identified new, unique types of information which provide new insights that can only be found when examining software product forum data as posts instead of sentences. Our findings also confirm that the majority of posts within software product forums are valuable to software development teams. (2) We identified patterns of sentences that often appear together within software product forum posts. The identified patterns provide additional insight into user habits when posting in software product forums, as well as revealing associations between different sentences that help researchers better understand how contextual information is posted in software product forums, compared to other feedback channels where contextual information is often missing from informal user feedback (Martens and Maalej, 2019). (3) We examined the sequence of events of software product forum threads. We identified common patterns of discussions found within software product forum threads. The patterns can help developers understand the status of a software product forum thread, whether it requires the developers' immediate attention, as well as how a forum thread evolves from the initial title post to eventually getting marked as resolved.

This paper first examines the previous studies related to extracting software development insights from user feedback in Section 2. In Sections 3 and 4, we present our research methodology and results for each research question. Our findings are discussed in Section 5. Section 6 discusses the limitations and threats to the validity of this paper, and finally, Section 7 concludes the paper.

## 2. Related work

### 2.1. User feedback and software product forums

Software users are providing more and more feedback on software products, often in the form of app reviews, tweets, and online discussions on social media platforms (Nayebi et al., 2018; Harman et al., 2012; Pagano and Maalej, 2013; Maalej and Nabil, 2015). Studies have continuously shown that such feedback often contains information that can be used to help developers, whether it is reporting bugs, asking for potential features, or describing unintuitive features (Tizard et al., 2019; Devine et al., 2021; Seyff et al., 2014). Because of this, many techniques and tools have also been proposed to help developers automatically identify and extract potential product improvement insights, such as requirements, from online user feedback (Devine et al., 2021; Mekala et al., 2021; Hey et al., 2020).

More recently, research has also begun to examine software product forums as a source for product improvement insights. Tizard et al.'s work examined sentences within software product forums and identified insight-relevant information in forum sentences, then proposed techniques to automatically classify those sentences using supervised learning (Tizard et al., 2019). Rahimi and Cleland-Huang proposed using personas within software product forums to better collect feature requests using clustering and association rule mining (Rahimi and Cleland-Huang, 2014). Reddit, a discussion-based platform similar to software product forums, has recently been studied to gather potential software insights. Khan et al.'s work collected and classified title posts from Reddit's software-related forums and classified them into insight-related categories (Khan et al., 2019). StackOverflow is another platform that contains user feedback and discussions. Compared to software product forums, StackOverflow's target audiences are more often developers rather than general software users (Nasehi et al., 2012). Devine and Blincoe proposed methods to automatically extract the topics from StackOverflow posts (Devine and Blincoe, 2022). These studies have been focused on reducing the manual effort when reviewing feedback for potential software development insights in forums while also demonstrating that software product forums contain valuable information to developers (Wang et al., 2021; Tizard et al., 2019).

Some studies also examined forums as a community to understand how the community interacts or proposed feature improvement ideas for the forums. Gottipati et al.'s work tackled the issue of finding relevant answers in forums and created a search engine for users

looking to find solutions to their issues (Gottipati et al., 2011). Frith's work explored how different forum structures can affect forum users' behaviour, from adding sub-forums to providing possible guidelines for user submissions (Frith, 2017). Nugroho et al. showed that senior community members in software product forums are the most active participants in responding to bug and non-bug-related threads in forums (Nugroho et al., 2021). Yet, to our knowledge, no studies have examined the contents of all the posts in software product forums at a post-level with a focus on identifying potential product improvement insights. Also, there have not been studies that examine how posts in software product forums work together in a thread. Our work addresses these gaps in research by providing a fine-grained analysis of software product forum posts, as well as examining how forum threads in software product forums progress from the initial title post.

### 2.2. Contextual information

However, not all feedback is useful to developers, as studies have also shown that negative feedback often misses out on contextual details that help make it understandable to developers (Mezouar et al., 2018; Martens and Maalej, 2019; Bettenburg et al., 2008; Breu et al., 2010; Martin et al., 2016). Martens and Maalej's work listed out three key issues developers face when working with bug reports included in informal user feedback: Missing contextual information when writing the feedback, issue reported unable to be reproduced, and the manual efforts required to analyse the feedback (Martens and Maalej, 2019).

When contextual information is missing from feedback, extra effort is required from developers to follow up on these informal reviews to better understand the issue that users are facing. Studies have been done to try to reduce the effort required to collect such contextual information from user feedback. Martens and Maalej's work created a chatbot to ask users to include the version of the device if it is not included in the initial feedback (Martens and Maalej, 2019). Tizard et al.'s work automatically connected forum threads to issue tracker issues, reducing the manual effort required to link the two together (Tizard et al., 2022a). This study also addresses the lack of contextual information in forums by examining the association between sentences within software product forum posts to understand how contextual information in forums is written.

## 3. Research methods and data

In this study, we examined the forum posts of two software product forums: the VLC Media Player and the Firefox Browser forum. We conducted a manual content analysis on the different types of posts found in software product forums. We also examined the association between sentences and posts found within software product forums. Lastly, we examined the software product forum posts at a thread-level, where we consider the context of posts within the full forum thread that it was collected from to summarise the sequence of events in a software product forum thread.

### 3.1. Data collection

For this study, we used the existing dataset from Tizard et al.'s work on software product forum sentences (Tizard et al., 2019). The dataset contains a total of 3,810 sentences from VLC Media Player forum and Firefox browser forum that were manually labelled with software requirement relevant labels. The study was initially published in 2019 with the data collected and labelled in 2018, which included forum data from November 2003 to May 2018 for the VLC forum and data from September 2010 to November 2018 for the Firefox forum. The labels were created based on the type of sentences typically found within software product forums such as *apparent bug*, *question on application*, or *application guidance* that represent software requirement relevant discussions within the dataset. We based our study on this dataset of labelled sentences because of its relevance to software product forum based studies, as all of the sentences were extracted from software product forums. The sentences from this dataset were initially extracted from randomly selected threads from the two forums, including the title post for each of the threads, allowing us to trace each sentence back to the post from which it was extracted.

For each of the sentences used in Tizard et al.'s work, we found the original thread in the forum and collected all of the posts within that thread, as not all sentences within threads were included in the original dataset (Tizard et al., 2019). This was done by examining all of the title sentences within the dataset and then searching for the thread title on both the VLC and Firefox forums. Once the threads were found, we built a web scraper to automatically collect the content of each post into a dataset to be used for this study. Overall, 126 VLC forum threads and 71 Firefox threads were collected for this study, resulting in 875 posts that were associated with Tizard et al.'s sentence dataset. For each post, we collected the body of the post, the username of the user that wrote the post, as well as the order of the posts within the thread (title post as 1, first reply as 2, second reply as 3, etc.).

### 3.2. Forum post content analysis

**RQ1:** *What new product improvement insights can be discovered by examining forum user feedback at a post-level?*

Current studies on software product forum content are either based on individual sentences (Tizard et al., 2019), or they are focused mainly on the title post of a thread (Iqbal et al., 2021). Both methods have shown that user feedback in software product forums is valuable to software development, but there is still a gap in research when it comes to understanding all of the posts within forum threads for software development insights. Users participating in software product forums construct user feedback in the format of posts, and breaking posts into sentences or specifically looking at title posts does not cover all aspects of the user feedback in software product forums. In the case of forum sentences, the context associated with each sentence within paragraphs can be missed, and in the case of title posts, the replies where additional insights could be found can be missed. We address this gap in research by analysing the content of all types of posts in software product forums at a post-level as to whether or not the feedback is related to product improvement, following Tizard et al.'s approach on forum sentences, where different types of sentences were classified as to whether they contained project improvement insights (Tizard et al., 2019). For this study, we consider a software product improvement insight as relevant user feedback found in software product forums that provides useful information to developers to improve the quality of the software.

To evaluate the content of the posts, we conducted a manual content analysis for the 875 posts collected from the VLC and Firefox forums (Krippendorff, 2018). We first removed 114 VLC posts and 26 Firefox posts that only contained one sentence from the dataset, as a post that only contains a singular sentence is the same as sentence level analysis that was already classified in Tizard et al.'s work (Tizard et al., 2019). After removing the single sentence posts, we evaluated a total of 369 VLC posts and 366 Firefox posts.

To create a set of classifications to be used for the analysis, we started with Tizard et al.'s forum sentence labels, which contained 18 sentence classification labels that are relevant to software requirements (Tizard et al., 2019). Three of the authors (two senior PhD students and a researcher, all with experience in requirements engineering) were given 50 random posts for a preliminary classification process using the sentence labels. If no suitable labels were available for a post during the labelling process, then a new post-level label was created. The newly created label was then discussed among the three authors to determine its suitability for the post. For existing labels mapped from the sentence level classification, we redefined some of the existing labels and definitions from Tizard et al.'s work to describe the content

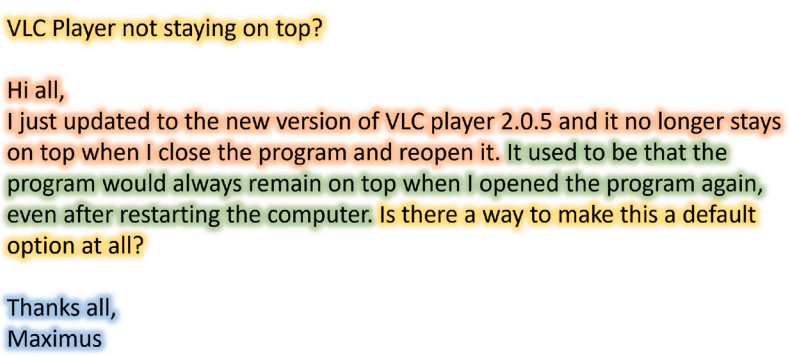


**Fig. 1.** Example of a forum post with sentences classified into different labels. Yellow = Question on Application, Red = Apparent Bug, Green = Application Usage, Blue = Non-informative. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the post more accurately when used at a post-level (Tizard et al., 2019).

For this preliminary post labelling process, the authors had an initial discussion after their set of 50 classifications was completed independently. Prior to the discussion, we counted the number of posts that had the same labels from all three authors and recorded the value as our initial-preliminary agreement rate. The remaining posts that contained different labels from authors were then discussed to see if the difference could be reconciled and labels could be changed. Once the discussion was completed, we recorded the final number of posts where all three authors agreed on its label as our final-preliminary agreement rate. For this task, our initial-preliminary agreement rate was 86%, and after discussions about the labels, we reached a 100% agreement rate for all 50 posts.

We initially allowed the coders to classify each post with multiple labels since some posts can contain multiple motivations or ideas, but after the initial categorisation of the 50 posts, we found that all the posts had one central theme of discussion despite containing sentences from different sentence level classifications. Therefore, we chose to only allow one label per post for the rest of the process. Fig. 1 is an example post from our dataset where the post itself contains multiple topics from a sentence classification perspective, but when examining the content at a post-level, we classified this post as a **Question on Application** post since the post as a whole raised a question about an existing feature in the software, despite having *Apparent Bug* and *Application Usage* sentences within the post. Table 1 presents all the sentence labels used for post labelling and the new labels created for post-level categorisation.

The preliminary labelling process of 50 posts resulted in a set of post-level classifications, with descriptions for each of the classification labels. Two of the authors (Ph.D. students) then used these classification labels to perform the manual content analysis of the remaining software product forum posts. They independently analysed and classified all of the posts using the classification labels and their descriptions as guidelines. If a post could not be classified with an existing classification, a new label was created and discussed later by the authors. After the remaining 685 posts were classified by both authors independently, we recorded the classification agreement rate between the two authors as our Initial Agreement%. Then, the two authors discussed the differences in the labels to try and reconcile the differences, as well as discuss the new labels. This happened for several rounds until no more differences could be reconciled, and the Final Agreement% was recorded for the manual content analysis for that round. The inter-coder reliability score for the classification process was calculated using the ReCal[1] tool, and the results are presented in Table 2, where the Cohen's Kappa was calculated using our final set of classifications across both rounds. As seen in this table, we obtained a high inter-coder reliability score for both of the datasets. In our final dataset, there were 14 VLC posts and 5 Firefox posts that the two authors couldn't agree on and were discarded from the dataset. The discarded posts were ambiguous and could be interpreted in different ways, for example, the post: *"Now here is my openion: VLC is an all in one media player solution. there are very few formats it can't support. but the biggest mistake being done here is that instead of concentrating on stabilising/improving existing features, other features are being added too quickly. As this is a nightly, it does not matter, but I hope that the final VLC 0.9 removes all those problems".* was discarded because the authors couldn't decide whether the poster was dispraising the software or confirming that a feature/issue could not be resolved at that moment. In total, we classified 355 VLC and 361 Firefox posts for our manual content analysis of software product forum posts. The classified posts are available in our replication package along with our coding guideline used for this task[2].

**Table 1**
Sentence to post classification mapping.

| Tizard's sentence labels | Post labels |
|---|---|
| Application Usage | |
| Non-Informative | Non-Informative |
| Apparent Bug | Apparent Bug |
| Application Guidance | Application Guidance |
| Question on Application | Question on Application |
| Feature Request | Feature Request |
| Help Seeking | |
| User Setup | |
| Question on Background | |
| Attempted Solution | Issue Not Resolved |
| Requesting more Information | Clarification Question |
| Praise Application | Praise Application |
| Dispraise for Application | Dispraise for Application |
| Acknowledgement of resolution | Issue Resolution |
| Agreeing with the problem | |
| Limitation Confirmation | Limitation Confirmation |
| Bug Confirmation | Bug Confirmation |
| Agreeing with Feature Request | |
| | Additional Context |
| | Confirmation of Issue Resolution |
| | Guidance not working |
| | Partial Bug Resolution |
| | Unclear Guidance |

[1] http://dfreelon.org/utils/recalfront/

[2] https://zenodo.org/record/8058229

**Table 2**
Inter-coder Reliability for preliminary and final round of classification. Cohen's Kappa was calculated with the final dataset.

| Forum | Preliminary round (total n = 50) | | Final round (total n = 685) | | Overall (n = 735) |
|---|---|---|---|---|---|
| | Initial agreement | Final agreement | Initial agreement | Final agreement | Cohen's Kappa |
| VLC | 92.0% | 100% | 91.0% | 96.2% | 95.9% |
| Firefox | 80.0% | 100% | 94.4% | 98.2% | 98.3% |

### 3.3. Association between forum posts and sentences for insights

**RQ2:** *How often can we find product improvement insights and contextual information in software product forum posts?*

Currently, studies on software product forums have indicated that these forums contains user feedback that is valuable to developers (Tizard et al., 2019; Wang et al., 2021; Devine et al., 2021). However, there is a lack of understanding of the contextual information found in user feedback on software product forums, as well as how sentences work together within forum posts. Understanding how often contextual information appears within forum posts and what types of information are most likely to appear in forum posts can help developers better understand user behaviour when writing informal user feedback and also quickly identify the contextual information used to replicate potential bugs or development oversights. For this process, we used the sentence classifications *User Setup* and *Application Usage* and the post classification *Additional Context* as contextual information related classifications for examining contextual information.

We conducted two sets of analysis for classified software product forum sentences and software product forum posts. As our dataset of posts is based on the labelled sentences from Tizard et al.'s work, we were able to analyse directly between posts and sentences since both sets of data were extracted from the same set of forum threads (Tizard et al., 2019). For this RQ, we analysed the 735 posts that were classified in RQ1.

First, we examined the distribution of sentence labels between different types of post classifications for both forums. We investigated how each post within our dataset was constructed from a set of sentence classifications. Sentence level classifications such as *Application Usage* and *User Setup* are valuable contextual information that can help developers better understand the situation surrounding potential issues to software (Tizard et al., 2019). By examining the post composition in terms of sentence classifications, we aim to identify whether or not there are associations between sentence-level classifications and post-level classifications and where contextual information is found within software product forum posts.

We then conducted association rule mining on all of the sentences to reveal how often different sentence classifications appear together within software posts. We used the Apriori association rule mining algorithm for this task (Agrawal et al., 1994). Each post's collection of sentence labels was used as inputs for the algorithm, and the algorithm examines how often a pair of sentence labels appear together within posts. Based on the results of the previous step, we found that the sentence dataset consists of multiple sentence categories below the 10% minimum support cutoff that were often used for the algorithm (Jafarzadeh et al., 2017; Swamy et al., 2015). However, many of these sentence categories, such as *User Setup* and *Requesting more Information* contain valuable information to developers, and it would be counterintuitive to remove them from the association rule mining process because of the category's sparsity (Soni et al., 2020). For our implementation, we set the minimum support to 0.03 compared to the standard minimum of 0.1 when the algorithm is implemented in other fields (Jafarzadeh et al., 2017; Swamy et al., 2015). By setting our minimum support for the algorithm to 0.03, we can ensure that even smaller, emerging associations between sentence categories can be picked up by the algorithm since most of the smaller sentence categories appear within 6% to 10% of the posts (AlZu'bi et al., 2018). The minimum confidence was set to 0.5, following Diamantopoulos and Symeonidis' work (Diamantopoulos and Symeonidis, 2020). The parameters for this task ensure that association rules will be discovered as long as they are present in at least 3% of the whole dataset (0.03 minimum support), and the occurrence rate of such associations happens more than 50% of the time (0.5 minimum confidence), allowing us to remove noisy associations with lower appearance rates. This parameter setup is also supported by Blatter and Einsele's work on user comment associations using the algorithm, where they identified that the minimum support should be set lower than 0.1 to ensure all associations could be picked up by the algorithm (Blatter and Einsele, 2022). The association rule mining task evaluates the dataset by collecting support for both antecedents and consequents, which are then used to calculate the support of each association rule and its confidence. The terms and formulas are defined as follows:

**Antecedent:** An item in the dataset, and in the context of this study, the item is a singular sentence category.

**Consequent:** An item in the dataset that is found in combination with the antecedent item. In the context of this study, the item is a singular sentence category.

**Support:** The percentage of how frequently the item appears in the dataset. The support of each association rule is the percentage of how frequently both the Antecedent and Consequent appear together in the dataset.

**Confidence:** Confidence is the probability that consequent Y will follow antecedent X, where X and Y are the support for each item, and it is given as:

$$\sigma(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

### 3.4. Progression of software product forum threads

**RQ3:** *How do potential product improvement insights get treated in software product forum threads?*

As of now, all of our categorisations on software product forums and sentences are fine-grained classifications. While such fine-grained classifications are useful in identifying specific trends within user feedback in software product forums, a high-level overview of the different types of posts would also be helpful to developers to quickly identify the status of a forum thread at a glance, judging by the type of post and replies. More specifically, it would be beneficial for developers to understand how a typical software product forum thread gets developed from the initial title post that contains development insights to eventually getting the issue resolved, features implemented, or development oversight identified and used to improve the software. Based on our classifications created in earlier steps and our findings on associations between sentence classification and post-level classification, we grouped our fine-grained post-level classifications into high-level categories of the major post types found within software product forums. We examined the post position, the distribution of sentence classifications, and the definition of each post classification to determine if there are similar post-level classifications that could be grouped into high-level post categories.

Once the high-level post categories were created, we looked to identify how the post-level classifications and high-level groupings we created in previous steps were connected in software product forum threads. Our dataset of posts used for this study was originally collected from 126 VLC and 71 Firefox forum threads. For our thread-level analysis, we individually examined all 197 threads by identifying how

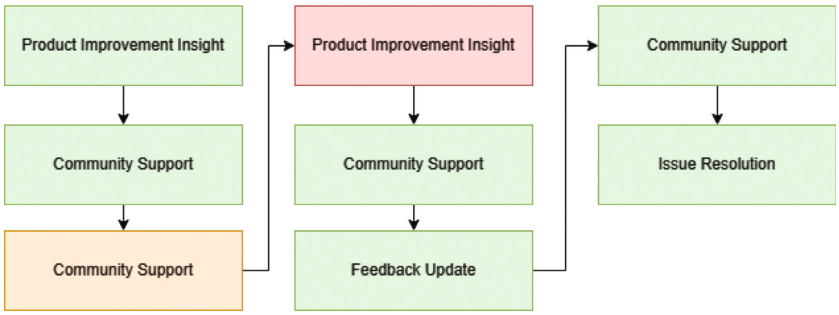


**Fig. 2.** Example of a forum thread consist of multiple classified posts being verified against our proposed flowchart. Green = post is following the flowchart. Yellow = duplicate post type within the same step. Red = post classification not following the flowchart. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the classified posts from our dataset were connected within the thread. The 197 threads contained 875 posts in total, with 716 of them used as our classified post dataset for RQ1 and RQ2. There were also 140 forum posts that only contained one sentence, which were initially removed from our analysis in RQ1. However, as the purpose of this RQ is to examine how forum posts work together in a thread setting, we added these 140 forum posts back into the dataset for this RQ by converting the individual sentence classifications that were initially labelled by Tizard et al. into our post-level classifications created in RQ1, much like our classification mapping process in the previous RQ (Tizard et al., 2019). Then, based on the grouping of our high-level post categories in the previous step, the 140 posts were grouped into their appropriate high-level categories if deemed necessary. For the remaining 19 posts that we had trouble classifying in RQ1, we removed them from the dataset used for this RQ, as we stated earlier that the authors could not come to an agreement when it comes to classifying the posts into a topic. Overall, for this RQ, we examined 856 software product forum posts found in 197 threads in VLC and Firefox forums.

For each post, we examined the adjacent post classifications and categories (previous post and next post) found within the thread that the post was initially collected from to see if we could identify common sequences of events found in software product forum threads. This gives us an understanding of how the post-level classifications and high-level categories are connected within forum threads and identify insights as to when a forum thread would require a developer's attention. Based on our findings, we created a flowchart on the sequence of events within a typical software product forum thread and identified which steps would require the developer's attention.

After creating the flowchart by examining adjacent posts for each post, we examined each full thread to see if the flowchart captured the full flow correctly. For each thread, we examined the percentage of forum posts that followed the sequence of events from our flowchart by manually cross-checking the classification and categories for each post within the thread and the events in our flowchart. For every post found in the thread that is part of our labelled post dataset, we checked whether the post within the thread was following the sequence of events found in our flowchart. Fig. 2 presents an example of our cross-checking process. For each post, we examined the post classification itself, whether or not it matched with the flowchart's sequence for the current step, and whether or not the post classification contains the same classification as the previous post from the same thread. If the post contained the same classification/category as the previous post, then we marked the post as a duplicate, and it was not counted towards our analysis since the state of the thread did not change if consecutive posts within the same thread are discussing the same type of user feedback, such as when multiple forum users are providing different methods of support to an issue. The results are measured as percentages of sequences of steps that follow the flowchart in each thread, and individual post classifications that do not follow the flowchart were recorded.

## 4. Results

### 4.1. What new product improvement insights can be discovered by examining forum user feedback at a post-level (RQ1)

The result of the manual content analysis on software product forum posts can be seen in Table 3. The results describe each of the classifications used to identify the topic of discussion for software product forum posts found in VLC and Firefox forums. We also included the sentence proportion for sentence-level classification from Tizard et al.'s work if the post classification was mapped from the sentence level classification (Tizard et al., 2019).

Overall, the manual content analysis of 355 VLC and 361 Firefox posts resulted in 17 forum post classification categories. Within the categories, 12 were mapped from Tizard et al.'s forum sentence classifications (Tizard et al., 2019), and five are unique to post-level analysis. *Additional Context*, *Confirmation of Issue Resolution*, *Guidance Not Working*, *Partial Bug Resolution* and *Unclear Guidance* type posts are only found when examining forum user feedback as posts instead of sentences.

While the remaining 12 classifications were mapped from previous sentence classifications, the definitions of those classifications were modified to fit more with post-level classifications. For example, *Apparent Bug* sentences at a sentence level indicate a sentence that has a strong possibility to discuss a possible bug, like with the sentence *"I have a problem with VLC Player"*. But when it comes to post-level classifications, **Apparent Bug** posts contain more than just an issue statement, but most of the time, also include other sentences such as contextual information that serves as complimentary statements to the main issue statement, like the post *"Green lines. Hello to everyone. I have a problem with VLC Player. Most of the movies i try to play with VLC have green lines in the middle of the screen. Can someone please tell me what's happening??"* where the majority of the posts are either *Application Usage* or *Question on Application* sentences, but the main focus of the post is on an apparent bug. This redefinition is true for all of the post-level classifications that were mapped from sentence-level classifications.

For the five new classification labels, ***Additional Context*** type posts are forum posts that contain only contextual information within the post. Compared to other post classifications such as *Apparent Bug*, where issues are being stated in the post and contextual information is provided along with the issue statement, *Additional Context* type posts mostly just provide additional contextual information instead of stating the problem explicitly. For example, a typical *Apparent Bug* post, such as: *"2.0.5 audio messed up. After upgrade from 2.0.4 to 2.0.5, volume in VLC is very low. Tried various audio settings without success. Also, audio "pops" occasionally while playing dvd movie - cuts out for a split second".*, often states the issue explicitly, then describes the context surrounding it. In the case of *Additional Context* posts, such as *"I should have also mentioned that when playing WMA Pro files none of the features in WMP*

**Table 3**
Post-level classification for software product forum posts and the data composition between sentence dataset from (Tizard et al., 2019) and our post dataset.

| Post classifications | Descriptions | Sentence% from (Tizard et al., 2019) | | Post% | |
|---|---|---|---|---|---|
| | | VLC | Firefox | VLC n = 355 | Firefox n = 361 |
| Apparent Bug | Posts with strong indication that the software is not working as intended. | 12% | 8% | 16.34% | 12.19% |
| Application Guidance | Posts that provides guidance to other users requesting help. | 10% | 33% | 20.56% | 33.80% |
| Bug Confirmation | Confirmation of an issue to be a bug for the software. | <1% | <1% | 3.10% | <1% |
| Clarification Question | Requesting more information from other users to better understand their requirement. | 2% | 6% | 4.51% | 9.70% |
| Dispraise for Application | Expresses negative sentiment towards the software. | <1% | 3% | <1% | 0.00% |
| Feature Request | Requesting new features for the software. | 6% | <1% | 8.73% | <1% |
| Issue Not Resolved | Stating that the issue is still unresolved despite attempts to resolve it. | 3% | 3% | 4.51% | 3.05% |
| Issue Resolution | Issue that the user is facing is now resolved. | <1% | 4% | 2.82% | 10.53% |
| Limitation Confirmation | Confirms that the issue the user is facing is due to limitations of the software. | <1% | <1% | 3.10% | 1.66% |
| Non-Informative | Posts which does not discuss about software related topics. | 15% | 8% | 1.41% | <1% |
| Praise Application | Praising the application. | <1% | <1% | 1.41% | <1% |
| Question on Application | Questions on software, including "how to" questions, but can also indicate unintuitive features. | 7% | 7% | 17.75% | 8.59% |
| Additional Context | The user provides additional context to an existing issue that they are facing. This includes set up information, steps to reproduce the issue. | – | – | 13.24% | 8.31% |
| Confirmation of Issue Resolution | Confirming that the issue the user is facing is now resolved. | – | – | <1% | 3.60% |
| Guidance Not Working | Explicitly stating that specific guidance provided by other users did not resolve their problem on the software. | – | – | <1% | 4.43% |
| Partial Bug Resolution | Some parts of the issue is resolved, but still requires more guidance to fix the remaining issues. | – | – | <1% | <1% |
| Unclear Guidance | Stating that the guidance provided by other users are unclear and hard to follow. | – | – | <1% | <1% |

*are operable ie. eq, SRS, Quiet Mode etc. Also it outputs as a line level not a volume - all volume controls are inoperable".*, do not describe the issue but simply provide additional contextual information so others can better understand the situation. The main difference between the **Additional Context** post label and the *Application Usage* sentence label is that an *Additional Context* post is strictly adding context to an existing issue, whereas *Application Usage* sentences can appear in title posts when introducing the issue and become part of the initial issue statement. Within our post dataset, 13.24% and 8.31% of the posts are *Additional Context* posts, making it the fourth largest type of post found in software product forums.

***Confirmation of Issue Resolution*** type posts confirm that the issue has been solved by the user. The post type is similar to *Issue Resolution*, but instead of being posted by the user who was facing the issue, the confirmation is often posted by other forum users to validate that the issue was solved using specific methods discussed in the thread. For example, an *Issue Resolution* post that states *"This was the answer for me. I only found this after doing the clean reinstall, though. I wonder if cancelling the "Restore previous session" would have cured the problem without the clean reinstall. This is something that I cannot test now".* suggesting that the issue has been resolved, and another person confirms that the method works by stating *"This happened to me again. I can confirm that going to options and disabling "restore previous session", restart firefox, and enable it again if you want, fixes the issue".*, confirming that the issue was indeed solved using the methods described in the previous replies of the thread. The *Confirmation of Issue Resolution* acts as a validation that the issue has truly been resolved and the thread does not require further attention.

***Guidance Not Working*** type posts strictly describe that the guidance provided by other community members or developers does not solve the issue. An example of this can be seen in the sample post *"Hello, Thank you for your prompt reply. Unfortunately it did not help. I tested it on our Xenapps, local windows and macbooks. Each with and without virus protection. It did not make a difference. Proxy, cache has also made no difference. I also tested it with different versions of Firefox (50, 53, 55.58) that makes no difference Patrick".* This type of post cannot be

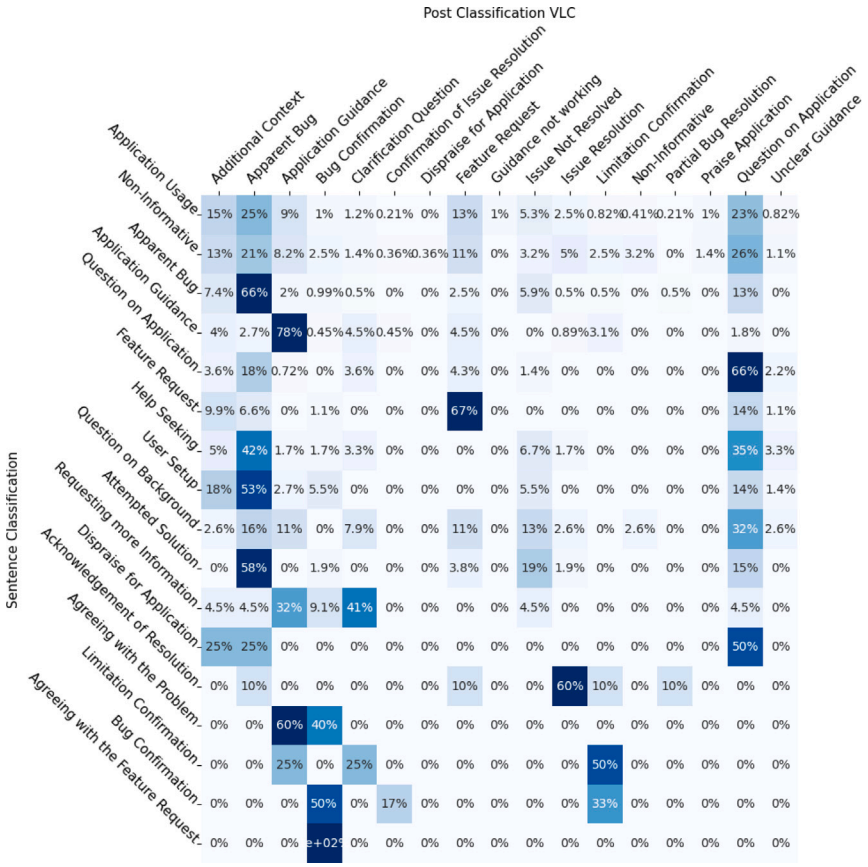


**Fig. 3.** Distributions of sentence classification within VLC posts.

identified by looking just at sentences because, without the surrounding context of other sentences in the post, the sentence *"Unfortunately it did not help* would be hard to make sense on its own, but after reading it in the context of the post, we can understand that it is talking about the help received in previous replies.

On the other hand, ***Unclear Guidance*** type posts are similar to *Guidance Not Working* type posts found in software product forums, except that in this case, it is the user stating the guidance provided either did not make sense, or it was hard to follow, to the point that they had trouble reproducing it on their own device. For example, the post *"Not that into this so have absolutely no idea how to do what you have suggested. Thanks, but I guess Im stuck with acute slowness"*. describes the issue of not understanding all the instructions given to the person.

***Partial Bug Resolution*** type post indicate that there is progress made towards solving the problem, but either it did not fix all of the problems, or a new problem showed up after solving the previous issue. The post *"Yeah I eventually figured it out, trouble was I had seen people say that in my google searches but no one said where in the preferences that option was. I hadn't thought about it being listed under subtitles so I looked pretty much everywhere but there. But no matter, got it running now in 16:9, 30 fps and 75 ms, pretty smooth except for a split second delay from the live feed, oh yeah, and it won't record video while still displaying the live feed"*. encapsulates the situation where the previous issue was solved, but another issue showed up to bother the user again. This type of post is hard to identify when looking at a sentence level because often times, the individual sentence would describe the issue being solved, and the other sentence would describe that an issue has occurred, which makes sense in isolation, but when looking at it as a whole post, it is easy to spot that there is still work to do to resolve the issue that the person is facing.

When comparing sentence-level classifications and post-level classifications, there are differences between the data composition. While *Issue Resolution* type sentences only accounted for <1% and 4% of the overall sentences when conducting sentence-level analysis, the percentage jumped up to 2.82% and 10.53% when examining the forum posts at a post-level. This is also true for most of the post-level classifications, where the percentages of the posts are higher than their respective sentence percentages. The only classification this rule does not apply is the *Non-Informative* classification, where only 1% of the posts are considered non-informative, but 15% and 8% of the sentences within both forums were considered non-informative. The compositions of posts between forums are also consistent, with *Application Guidance*, *Apparent Bug*, *Question on Application*, and *Additional Context* type posts taking up more than 60% of all the posts used in this study.

> **Answer to RQ1:** 17 classification labels were identified by examining the topic of discussion with software product forum posts. We identified 5 new classification labels and redefined the 12 classifications labels that were mapped from sentence-level classification that are relevant to product improvement insights. For the data examined in this study, only 2% of the forum posts were unrelated to product improvement insights.

### *4.2. How often can we find product improvement insights and contextual information in software product forum posts? (RQ2)*

The distributions of sentence classifications across the two forums are very similar. Fig. 3 and Fig. 4 describe the distribution of sentence classifications for each of the post classifications created by our manual content analysis in VLC and Firefox forums. For both forums, post classifications that were mapped from the sentence classifications showed a strong overlap between topics, with *Apparent Bug*, *Application Guidance*, *Feature Request*, *Question on Application* all being the majority

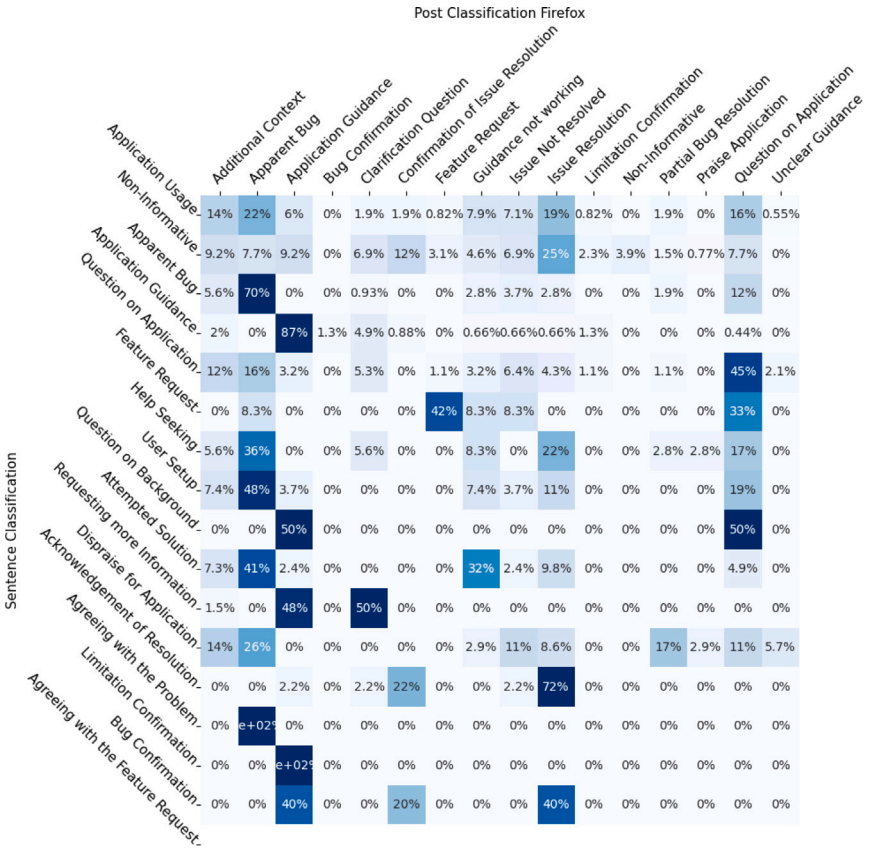


**Fig. 4.** Distributions of sentence classification within Firefox posts.

sentence category in their corresponding post categories. For sentences providing context without explicitly stating a request or an issue, *User Setup* sentences are most likely to appear in *Apparent Bug* posts, with 53% and 48% of all the *User Setup* sentences being found in *Apparent Bug* posts. More than 50% of *Application Usage* sentences appear in *Additional Context*, *Apparent Bug*, and *Question on Application* posts for both forums.

Table 4 presents the association rules discovered by the Apriori algorithm for VLC and Firefox forums. Explicit requirement sentences such as *Apparent Bug* and *Question on Application* are associated with contextual information sentences in *Application Usage* or *User Setup*. There are 75% and 69% chances that a forum post containing *Apparent Bug* sentences would also contain at least one *Application Usage* sentence in both forums. A similar association can also be found between *Question on Application* and *Application Usage* sentences. In the case of *User Setup* sentences, there are 78% and 54% chance that an *Apparent Bug* sentence would appear within the same post with at least one *User Setup* sentence. *Attempted Solution* sentences also appear frequently next to *Apparent Bug* sentences for both forums, with 73% and 52% confidence.

Associations that only exist in one forum are due to the difference in data composition between the two forums. As shown in Table 3, for VLC Associations, *requesting more information*, *dispraise application*, *problem resolution*, *dispraise application* sentences each only accounted for less than 3% of the sentences found in VLC forums, compared to the data composition of these sentence classifications in Firefox forums (6%, 3%, 4%). For Firefox Associations, *help seeker* only accounted for 2% of the sentences in Tizard et al.'s original work, compared to the 6% of sentences found to be *help seeker* in VLC forum. This is also true for *feature request* and *question on background* sentences, where there are simply more sentences of these types found in VLC forums compared to Firefox forums, making it difficult to pick up the associations due to data sparsity.

**Answer to RQ2:** Forum users post contextual information within forum posts that are presenting product improvement insights. More than half of *Application Usage* and *User Setup* sentences that represent contextual information can be found within *Apparent Bug*, *Question on Application*, and *Additional Context* posts in both VLC and Firefox forums. For the association between sentences, contextual information also appears frequently with *Apparent Bug* and *Question on Application* sentences in both VLC and Firefox forums.

### 4.3. How do potential product improvement insights get treated in software product forum threads? (RQ3)

#### 4.3.1. High-level categories

We analysed all of the forum posts found within each thread to understand how a typical software product forum thread progressed from the initial title post and how our post-level classifications can be grouped into high-level categories. Fig. 5 and Fig. 6 indicated the post position for different post-level classifications used in this study at the thread-level. Based on manual analysis of the threads and the post position, we identified three common types of posts within the threads:

**Product Improvement Insight:** *Apparent Bug*, *Question on Application* and *Feature Request* posts are the most common types of posts to start a thread, as shown in Fig. 5 and Fig. 6. These types of posts all describe information that can be useful for product improvements. The post content is either potential bug reports, asking questions on the software, or potential feature requests. The post often contains contextual information, including the usage of the software and the user setup for the software environment, such as application version number or OS version number, as shown in Fig. 3 and Fig. 4 where *Application Usage* and *User Setup* sentences frequently appear within

**Table 4**
Association Rules for VLC and Firefox Forum sentences. A-support and C-support are the support for antecedents and consequents.

| Antecedents | Consequents | VLC association | | | | Firefox association | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | a-support | c-support | support | confidence | a-support | c-support | support | confidence |
| Apparent bug | application usage | 0.247 | 0.534 | 0.186 | 0.752 | 0.168 | 0.450 | 0.116 | 0.692 |
| Attempted solution | apparent bug | 0.090 | 0.247 | 0.066 | 0.732 | 0.075 | 0.168 | 0.039 | 0.517 |
| Help seeker | apparent bug | 0.123 | 0.247 | 0.079 | 0.643 | – | – | – | – |
| Apparent bug | non-informative | 0.247 | 0.389 | 0.125 | 0.504 | – | – | – | – |
| User setup | apparent bug | 0.107 | 0.247 | 0.083 | 0.776 | 0.062 | 0.168 | 0.034 | 0.542 |
| Attempted solution | application usage | 0.090 | 0.534 | 0.061 | 0.683 | 0.075 | 0.450 | 0.052 | 0.690 |
| Feature request | application usage | 0.107 | 0.534 | 0.077 | 0.714 | – | – | – | |
| Help seeker | application usage | 0.123 | 0.534 | 0.096 | 0.786 | 0.083 | 0.450 | 0.070 | 0.844 |
| Non-informative | application usage | 0.389 | 0.534 | 0.247 | 0.635 | 0.264 | 0.450 | 0.155 | 0.588 |
| Question on application | application usage | 0.208 | 0.534 | 0.153 | 0.737 | 0.194 | 0.450 | 0.152 | 0.787 |
| Question on background | application usage | 0.072 | 0.534 | 0.050 | 0.697 | – | – | – | – |
| User setup | application usage | 0.107 | 0.534 | 0.083 | 0.776 | 0.062 | 0.450 | 0.049 | 0.792 |
| Help seeker | non-informative | 0.123 | 0.389 | 0.083 | 0.679 | – | – | – | – |
| Question on application | non-informative | 0.208 | 0.389 | 0.123 | 0.589 | – | – | – | – |
| Question on background | non-informative | 0.072 | 0.389 | 0.039 | 0.545 | – | – | – | – |
| User setup | non-informative | 0.107 | 0.389 | 0.068 | 0.633 | – | – | – | – |
| Requesting more information | application guidance | – | – | – | – | 0.129 | 0.406 | 0.090 | 0.700 |
| Dispraise application | application usage | – | – | – | – | 0.059 | 0.450 | 0.041 | 0.696 |
| Problem resolution | application usage | – | – | – | – | 0.101 | 0.450 | 0.052 | 0.513 |
| Dispraise application | question on application | – | – | – | – | 0.059 | 0.194 | 0.031 | 0.522 |
| Problem resolution | non-informative | – | – | – | – | 0.101 | 0.264 | 0.054 | 0.538 |
| User setup | question on application | – | – | – | – | 0.062 | 0.194 | 0.031 | 0.500 |

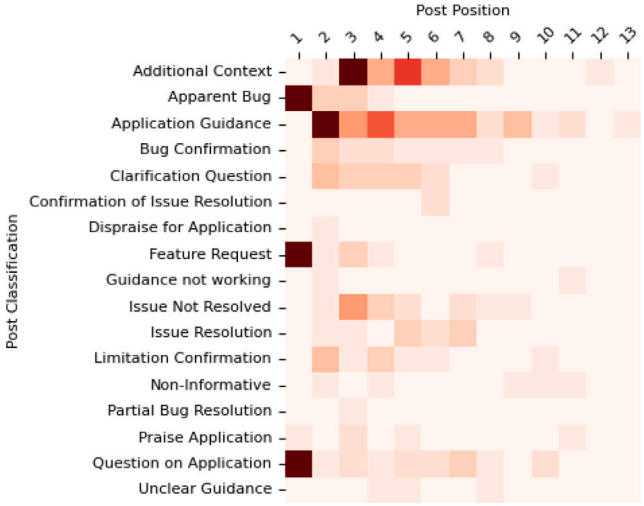


**Fig. 5.** Post positions for VLC forum posts.

these post types. E.g. *"update not working. One of my PCs updated to version 55 yesterday. Today my other PC says 54.0.1 is update and will not update to 55. Both are running windows 7"*.

**Community Support:** In most cases, after a Product Improvement Insight post has been made, a post providing support occurs in the reply section. Both *Application Guidance* and *Clarification Question* type posts provide support to users facing issues with the software product. The support often contains information such as instructions or guidance for users to try and solve their issues. If the user did not provide enough details with their initial report, the support poster would often ask for additional clarification on the issue to better diagnose the problem. Most of the community support posts can be found as the first reply to a software product forum thread, but additional support is often given throughout the thread until the issue is either solved or confirmed to be a development oversight. E.g. *"If 56 is working for you, you may want to stay there. In Firefox support run a search on Quantum or Quantum slow and you will see there appear to be many problems with memory and unhappiness in layout. 56 wouldn't work for me so I went back to 55.0.3 and am going to stay there for awhile"*.

**Feedback Update:** *Additional Context*, *Guidance Not Working*, *Issue Not Resolved*, *Partial Bug Resolution* and *Unclear Guidance* are post classifications that provide an additional update to existing Product Improvement Insight thread starters. The update acts as a reply to provide additional detail to an existing insight thread. The additional details would include contextual information, such as what caused the issue or providing an update on the support given earlier in the thread. Feedback updates also act as an indicator that the issue has not been resolved; hence, there is a need for an additional update to provide more information. E.g. *"I have an Intel(R) Core(TM) I7-3770 CPU @ 3.40 GHz 3.90 GHz 64-bit Operating System. I've just installed Firefox 51.0.1 today. I've removed it from Kaspersky's firewall's list of trusted programs, it didn't help"*.

### 4.3.2. Thread analysis

We analysed 197 software product forum threads to examine how our classifications and categories were connected in software product forum threads. Figs. 7 and 8 show the adjacent post types for all of the 856 forum posts found in our thread analysis. We provide the overall post proportion for each of our post classifications and categories for the post dataset, the percentage of the post being the first post or the last post of the thread, as well as how often each post classification/category is connected by identifying the type of post found connecting to the post that was examined. For **Product Improvement Insight** type posts, 83.8% of the posts found in threads act as a thread starter post, and 65% of the posts found after the **Product Improvement Insight** posts are **Community Support** type posts. For **Feedback Update** type posts, 69% of the post type can be found as a reply after a **Community Support** type post. Once a **Feedback Update** type post was posted, 52% of the first replies after that post are also **Community Support** type posts.

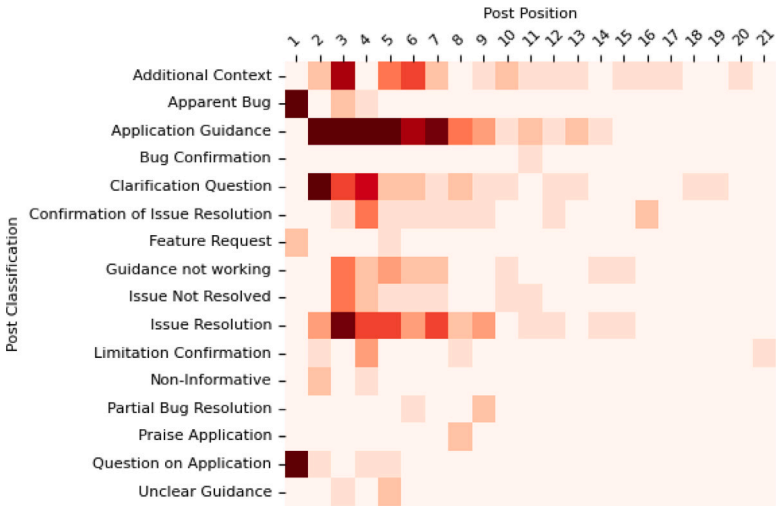


**Fig. 6.** Post positions for Firefox forum posts.

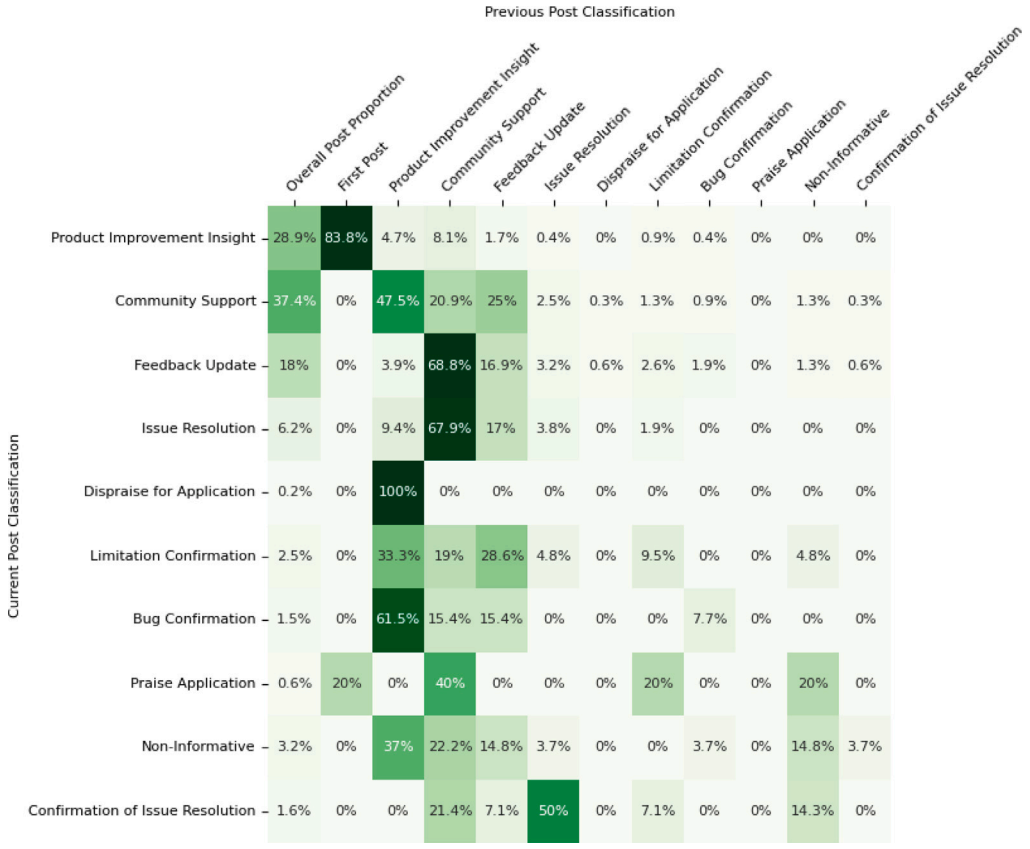


**Fig. 7.** Previous post classification/categories for each of the post classification/categories found in the thread analysis.

*4.3.3. Sequence of events in software product forum threads*

To illustrate the common patterns of user activities found within software product forum threads, Fig. 9 was created based on our previous step of thread analysis. We formulated the flowchart based on our results from previous steps on identifying how the posts were connected in forum threads, as well as our analysis of the post positions for each type of post. Based on our results, we identified that 81% of all **Product Improvement Insight** type posts act as a thread starter post, followed by a range of different types of replies. **Community Support** type posts are the most common type of reply found after a Product Improvement Insight post (65%), followed by **Feedback Update** (33%) type posts. **Community Support** posts are also given after **Feedback Update** type posts, as 52% of the immediate reply after the update are **Community Support** type posts, indicating that, often times, multiple rounds of update and support were happening within a software product forum thread. **Issue Resolution** type posts are most commonly found after **Community Support** type posts (68%), and **Confirmation of Issue Resolution** type posts are most likely found after the issue has been resolved (50%). Both **Bug Confirmation** and **Limitation Confirmation** type posts are most likely to be found after either **Product Improvement Insights** or **Feedback Update** type posts, and as Tizard et al. identified in their work, many of these confirmations of bugs and limitations are then added to the software team's issue tracker (Tizard et al., 2022b).


**Fig. 8.** Next post classification/categories for each of the post classification/categories found in the thread analysis.

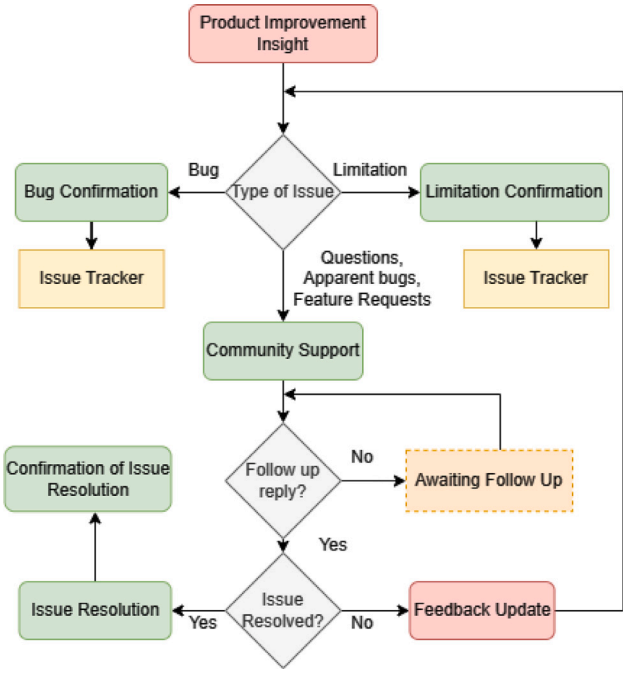


**Fig. 9.** Events in software product forum threads. Red/Green: Attention required/not required. Yellow: Linking to issue tracker. Orange: Waiting for follow-up. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For each type of post used in our flowchart, we examined the definition of the post type, as well as the percentage of posts that appeared as the last post in threads during our thread analysis. For the flowchart, the events coloured in red indicate that they require the developer's attention, whether it is a new product improvement insight thread or a new feedback update appearing in an existing thread. These two types of posts also have a low chance of being the last posts within forum threads (13.7% and 14.3%), suggesting that additional replies or attention is given to these posts in most cases. The events in green indicate that it does not require the developer's attention as of now because the thread itself was either recently given support by the community or the issue was resolved. Events marked in yellow indicate that the event is outside the scope of the content found within a typical forum thread, in this case, directing the user to submit their bug report in the forum to the development team's issue tracker (Tizard et al., 2022a). Finally, after community support was given, there is a chance that no updates were given by other users to indicate whether or not the issue was solved (23%). To highlight this, we marked the waiting period for feedback update as orange and named it *Awaiting Follow Up*. *Non-Informative* type posts were not included in the flowchart because the post is not informative for potential product improvement insights. Similarly, *Praise Application* and *Dispraise for Application* type posts were also not included in the flowchart because the content of these posts only provides sentimental feedback towards the software instead of technical insight such as requirements.

We also examined how effective our flowchart can be when navigating through our dataset of 197 forum threads. Fig. 10 shows how the threads adhere to the sequence shown in the flowchart we created in Fig. 9. Out of the 197 threads analysed with an average of 4.3 posts in each thread, the sequence of events in 144 threads were completely following the sequence in our flowchart, with 53 forum threads containing at least one post not following our sequence flowchart. Out of the 53

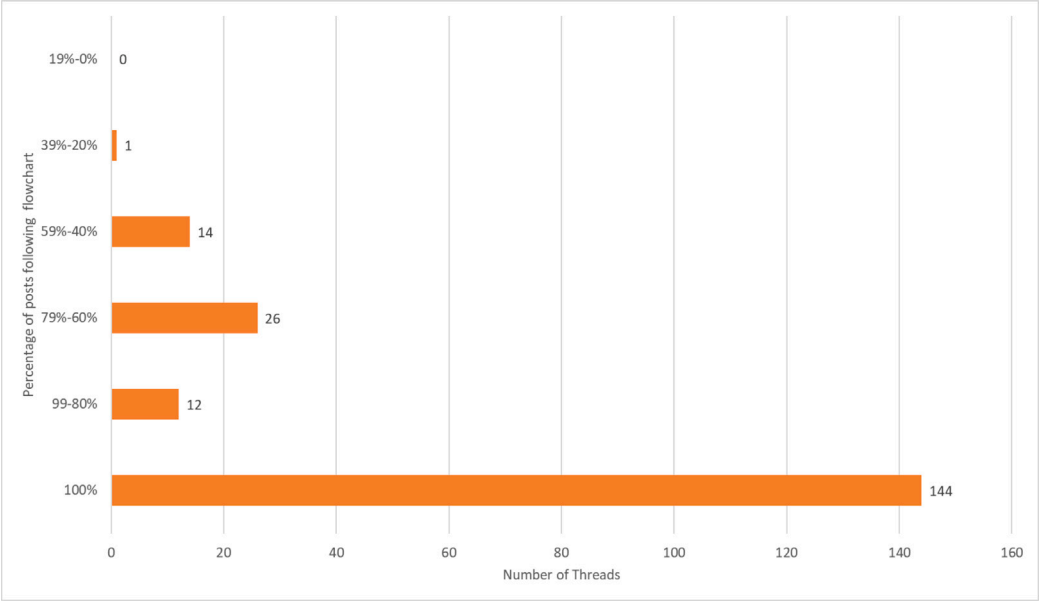


**Fig. 10.** Percentage of posts following the sequence of our flowchart within each thread.

forum threads, We provide three main reasons for 45 of these threads not following the flowchart, along with the number of threads affected for each reason. The remaining eight threads that did not follow our sequence flowchart were either long threads that had multiple rounds of discussion and ended up discussing topics that were unrelated to the main topic or shorter threads that had no structure and noisy replies.

**New insight in existing thread (20/53):** While discussing methods to resolve issues in forum threads, sometimes new product improvement insights, unrelated to the initial post, are added to the thread, causing **Product Improvement Insight** type posts to appear after either **Community Support** or **Limitation Confirmation** type posts. For example, after being provided with a workaround to an issue the user was facing, the user could request a feature to improve the software. This new insight is then tacked onto the existing thread where the discussion originally occurred, essentially starting a new sequence for a new **Product Improvement Insight**, but in an existing thread with existing replies to other insights.

**Additional discussion after insight attended (14/53):** After a product improvement insight was either treated and resolved or confirmed as a bug or feature, additional support might still be provided in forum threads, causing **Community Support** type posts to appear as replies after **Issue Resolution**, **Limitation Confirmation** and **Bug Confirmation** type posts. When a conclusion was reached in forum threads regarding a potential product improvement insight, some users would still provide additional support in the forms of alternative solutions to an issue or clarifications as to when the limitation or bug occurred. In other cases, additional **Feedback Update** type posts were provided as replies after **Issue Resolution**, **Limitation Confirmation** and **Bug Confirmation** type posts to provide more details to what was actually causing the issue that was resolved or identified as bugs or limitations for the software.

**Thread updated despite no support given (11/53):** Some forum threads would receive no support from other users, but the user who originally raised the issue would sometimes still update the thread with more information on the issue, or eventually provide their solution to the problem, causing **Feedback Update** and **Issue Resolution** type posts to appear as replies after **Product Improvement Insight** and **Feedback Update** type posts.

**Answer to RQ3:** We identified that there are common conversational patterns in the different types of product improvement insight threads. We summarise the flow of events in software product forum threads in Fig. 9, starting from the initial statement on **Product Improvement Insights**. We also identified **Product Improvement Insights** and **Feedback Update** posts as events which would require the developer's attention in a typical software product forum thread. Our flowchart were able to 100% track the sequence of events in 73% of the threads we analysed, and less than 10% of the forum posts found in our thread analysis were not following our flowchart.

## 5. Discussion

In this section, we discuss the implication of our results and future work that can be inspired by this study.

### 5.1. Post-level analysis

Our analysis shows that new insights can be found from software product forum posts by examining the whole post and understanding the context of all of the sentences in the post. For example, consider the post *"Thank you for your prompt reply. Unfortunately it did not help. I tested it on our Xenapps, local windows and macbooks. Each with and without virus protection. It did not make a difference. Proxy, cache has also made no difference.I also tested it with different versions of Firefox (50, 53, 55.58) that makes no difference"*. This post has seven individual sentences, and using Tizard et al.'s classification scheme (Tizard et al., 2019), they were classified as *Non-Informative, Application Usage, Application Usage, Application Usage, Application Usage, and Attempted Solution*. While these sentence classifications help with understanding and identifying sentences relating to software product improvements, they do not accurately describe the theme of the forum post when combining them together. When reading the post as a whole, the post indicates that the user has tested out the solution given to them earlier with key phrases such as *"Thank you for your prompt reply. Unfortunately it did not help"*., and that the user still requires additional assistance since the issue is still unresolved. This post would

be classified in our scheme as *Guidance not working*. Such posts, as well as *Partial Bug Resolution* or *Unclear Guidance* posts can help the software developers identify new issues that are not resolved by previous workarounds, indicating product improvements might be required or new workarounds must be derived. While prior work, such as the study by Tizard et al. (2019), created classifiers that extracted software requirement relevant information from forum posts at the sentence level, our results indicate that there is a need to create solutions at the post-level to identify more insights. While Iqbal et al. (2021) created a classifier that could handle full title posts from Reddit, they considered only a small set of classifications (bug, feature, or irrelevant). Our findings indicate that more fine-grained post-level analysis is needed. As with the advancement in natural language processing techniques such as BERT (Devlin et al., 2018),NoRBert (Hey et al., 2020) and generative AI such as ChatGPT (Wu et al., 2023), future work can look into whether or not post-level classifications for forums can be automatically classified using these new techniques as well as transitional machine learning algorithms such as Naive Bayes (Tizard et al., 2019).

We also identified the difference in context when examining the same type of classification between posts and sentences. In a sentence setting, an *Apparent Bug* label indicates a singular sentence that strictly discusses a potential bug, whereas in a post setting, *Apparent Bug* posts not only include the bug sentence itself but also often include other sentences to provide more context to the issue. For example, the post *"update not working. One of my PCs updated to version 55 yesterday. Today my other PC says 54.0.1 is update and will not update to 55. Both are running windows 7".* not only describes the issue but also provides contextual information such as the OS version and software version. Such differences in the amount of content and context between the two types of classifications despite sharing the same classification label should be noted for future research, where a potential hierarchy of classifications can be created for forum-based user feedback. Developers could potentially quickly identify the topic of discussion for posts with our post-level classification and, if required for more in-depth examination, could use sentence-level classifications from Tizard et al.'s work to identify not only fine-grained topics of discussion relevant to software product improvement but also identify the contextual information within the post itself (Tizard et al., 2019).

Our results also indicate that there are correlations between sentence classifications and post-level classifications. The correlations can be further studied to explore which sentences within a post can help software practitioners identify the main topic of discussion. One of the many issues we face when looking to automatically classify text into categories is the classification of longer documents due to the complexity of the document itself, as well as the time required to collect and train documents for classification purposes. With our results indicating promising overlap between the sentence classifications and the post classifications, we can leverage that to design systems that focus on identifying and classifying sentences to predict the topics of longer documents using sentence classifications. More research could also be done to understand when sentence-level classification would be sufficient for the identification of product improvement insights over the high level labels such as our post-level classifications.

### 5.2. Contextual information in forum posts

We found that users often include contextual information when submitting their informal software feedback in software product forums. Martens and Maalej's work described the issue of lacking contextual information when users are submitting their software feedback on other channels, like App Store reviews or tweets from Twitter (Martens and Maalej, 2019). For forum users, this does not seem to be the case. Through association mining, we found that contextual information sentences like *Application Usage* are associated with many explicit requirement statements such as *Apparent Bug* and *Question on Application* sentences in software product forum posts with promising confidence. This indicates that users often provide contextual information when creating a product improvement insight thread and providing feedback updates with replies. This association is consistent across both forums despite the differences in structure. A software product forum strictly focuses on the discussion of software; it does not invite users to rate the software like app stores, nor does it have a word limit to keep the discussion short like Twitter, which can encourage users to provide more details about their issue, as more details can lead to a faster resolution. Recent studies have focused on the classification and identification of statements explicitly stating potential software insights and requirements in various channels for the purpose of automatic classification (Dabrowski et al., 2022). With the results of our study and in the context of user feedback from software product forums, we suggest designing new tools to extract not only those explicit statements but also the contextual information that comes along with the statement in forum posts. More work can also be done to examine the variation of contextual information found within forum posts, such as different user setups, steps to reproduce the issue, and problem statements to determine the extent of issues.

Software teams that maintain software product forums could also consider providing templates for forum posts to encourage users to provide the contextual information that is needed for the team to fully understand the issue or feature request. This would save the effort needed to manually ask for additional contextual information, allowing quicker resolution of issues. While bug trackers such as Bugzilla[3] provide guidelines to encourage users to submit contextual information when reporting a bug, this practice should also be implemented by software product forums to improve the quality of discussions found in forums. As Tizard et al.'s work identified that bugs confirmed in forum threads are often transferred to issue trackers, forum guidelines can help speed up this process with the addition of contextual information in their forum posts. Future work can also examine in detail why forum users more commonly add contextual information to their forum posts compared to other feedback channels such as Twitter and App Stores. Also, by examining how contextual information and product improvement insights are written in forums, future work design tools to find contextual information on other platforms when the same issue has been identified in multiple platforms.

### 5.3. Patterns in forum threads

Lastly, we examined the sequence of discussions that happen in a typical software product forum thread, starting from the initial **Product Improvement Insight** statement, and we illustrated this using a flowchart. As there are no guidelines for forum design, almost no forums are the same in features (Wang et al., 2021). For the Firefox forum, issues that have been solved would require the initial poster of the issue to mark the issue as solved to end the case. This can be inconsistent as many forum users either do not indicate whether or not their issues were solved or do not revisit the forum after their issue has been solved. Constantly checking whether or not the issue has actually been solved on forums would require additional effort from the development team. In the case of the VLC forum, there are no features that can indicate whether or not the issue was solved, which means that developers would manually have to scan each post to see the progress. Our findings suggest that there are five types of events in forums that would not require the immediate attention of developers. Once a forum thread has been given community support, or if the issue was discussed as solved or confirmed as a bug or limitation by the thread participants, developers can choose to focus on other threads that describe more urgent issues, such as new threads or existing threads that still require assistance. Our summary flowchart can be used to describe both the process in VLC forums and Firefox

[3] https://bugzilla.mozilla.org/home

forums despite the differences between the forums. By understanding these common patterns, developers can be more aware of when support is needed when issues require attention, or when product improvements are needed. Developers can also use this flowchart to design automated systems that can flag threads as either attention required or not to improve their efficiency at processing informal user feedback. As Nugroho et al.'s work stated that senior community members of software product forums are the most active participants in providing community support, future work can leverage this information along with our flowchart to design recommender systems for these active community members, recommending them with forum threads that require attention to improve the efficiency which they can provide community support (Nugroho et al., 2021). On the other hand, if the same type of improvement insight constantly comes up, or a thread constantly gets new **Feedback Update** replies, it can also indicate that the software itself is due for an update since the existing features might no longer satisfy the need of software users. This is also visible in our summary flowcharts as *Bug Confirmations* and *Limitation Confirmations*, where developers or senior community members would remark on the need to either solve the bug or update the software to address current limitations instead of having to constantly provide workarounds to solve issues raised by product users.

## 6. Limitations and threats to validity

One threat to conclusion validity relates to whether or not our post-level classification labels created in RQ1 accurately describe the content of the posts. In the case of RQ1, we conducted manual analysis on the content of software product forum posts, where manual analysis is prone to potential bias. Our method to mitigate this threat was to provide coding guidelines to the coders to better guide them when conducting the analysis. The manual analysis was also done by two coders independently, with the initial analysis done by three coders independently to reduce possible bias created by just having one coder. The three coders all had previous experiences working with user feedback in the field of requirements engineering. We were able to achieve a high level of inter-coder reliability, and the cases of disagreements were excluded from our post dataset. Despite this, a threat still exists where bias could have been introduced during the manual content analysis, impacting our findings.

Another threat to the conclusion validity of our research was our decision to give each post only one label during the manual analysis. Compared to forum sentences, which often contain just a single topic of discussion, posts can be complex and consist of a mixture of different ideas. By only allowing one classification per post, there is a concern that we were eliminating potential product improvement insights with our analysis as we focused on a single topic for each post. We decided to only allow one classification based on the result of our initial analysis of 50 posts. None of these posts required more than one classification when individually analysed by three authors, indicating that the posts used in our study can be classified using just one classification. While there were sentences that discussed attempted solutions or contextual information in requirement related posts, when authors read through the post as a whole, they were able to understand that such sentences were providing additional information to the main topic of discussion instead of creating an entirely new topic within the same post. Our high agreement rate across both forums also helped to strengthen this case, as the authors agreed on the majority of posts being focused on a single topic. Nonetheless, a single classification per post may have affected the result of our analysis for posts that might have contained two topics.

A possible threat to external validity is the ability to transfer our findings to other forums. We tried to mitigate this by using a dataset on previously published work that was collected from two distinct forums. However, the demographic information of the forum users in VLC and Firefox forums is not readily available. Therefore, it is possible that the difference in community demographics between VLC and Firefox forums could influence the user feedback found in these forums, affecting the results of this study. Our results indicate that despite the difference in data composition and possible demographic differences between the two forums, there are key similarities in our results across the two forums. All of our classification labels for software product forums are created as non-explicit labels that could be applied to both forums we studied, and during our manual analysis, we did not identify noticeable differences between the content of user feedback found between forums, as we had a high consistent agreement rate across both forums. The associations found between the sentence classifications and post classifications in this study were also consistent across two different forums, and the patterns in the threads were consistent across both of the forums used in this study. However, we cannot claim that our results generalise beyond the two forums examined in this study. Future work can further validate our findings on additional forums and different software user demographics.

There is a threat that software product forum posts from 2019 onwards could be different from the posts in our analysis. The dataset used for this study was initially collected in 2018 and contains data from 2003 to 2018 (Tizard et al., 2019). Therefore, we cannot claim that the results and the findings of this study apply to newer user feedback. It is possible that in the last five years, new trends in user behaviour and types of user feedback could be found in software product forums. For our post dataset, when we examined the labelled posts, we did not identify any trends in user feedback over time. However, the number of posts we examined in this study was relatively small, so if trends did exist, we may have missed them. Future work can examine more recent user feedback found in VLC and Firefox forums to further validate our findings.

## 7. Conclusion

In this study, we examined software product forums at the post and thread-level and found that new product development insights could be discovered through this higher-level analysis. While sentence-level classification is useful in identifying software product improvement insights in forums, some context and insights can only be found when analysing forum posts as a whole to paint a better picture of the insight. We investigated the type of user feedback found in software product forum posts from the VLC and Firefox forums. We identified 17 types of software product forum posts, 12 of which were aligned with the types identified through sentence-level analysis and five that can only be identified through examination of a full forum post. Through the use of association rule mining, we show that contextual information is commonly included in software product forum posts. We also examined discussion patterns at the thread-level, and we identified the common patterns that occur within a software product forum thread. The knowledge uncovered through this analysis can inform future work to create additional support for software teams to extract the most useful product improvement insights from the rich software product forum data.

## CRediT authorship contribution statement

**Hechen Wang:** Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Peter Devine:** Data curation, Validation, Writing – original draft. **James Tizard:** Data curation, Validation, Writing – original draft. **Seyed Reza Shahamiri:** Project administration, Writing – review & editing. **Kelly Blincoe:** Conceptualization, Project administration, Resources, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

https://zenodo.org/record/8058229.

## References


Agrawal, R., Srikant, R., et al., 1994. Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, Vol. 1215. Citeseer, pp. 487–499.

Ali Khan, J., Liu, L., Wen, L., Ali, R., 2020. Conceptualising, extracting and analysing requirements arguments in users' forums: The CrowdRE-Arg framework. J. Softw.: Evol. Process 32 (12), e2309.

AlZu'bi, S., Hawashin, B., ElBes, M., Al-Ayyoub, M., 2018. A novel recommender system based on apriori algorithm for requirements engineering. In: 2018 Fifth International Conference on Social Networks Analysis, Management and Security. Snams, IEEE, pp. 323–327.

Assimakopoulos, D., Yan, J., 2006. Sources of knowledge acquisition for Chinese software engineers. R&D Manage. 36 (1), 97–106.

Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., Zimmermann, T., 2008. What makes a good bug report? In: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering. pp. 308–318.

Bickart, B., Schindler, R.M., 2001. Internet forums as influential sources of consumer information. J. Interact. Market. 15 (3), 31–40.

Blatter, P., Einsele, F., 2022. Mining association rules in commuter feedback comments from facebook of swiss national railways (SBB) using apriori algorithm. In: International Conference on Data and Information in Online. Springer, pp. 230–241.

Breu, S., Premraj, R., Sillito, J., Zimmermann, T., 2010. Information needs in bug reports: improving cooperation between developers and users. In: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work. pp. 301–310.

Dabrowski, J., Letier, E., Perini, A., Susi, A., 2022. Analysing app reviews for software engineering: a systematic literature review. Empir. Softw. Eng. 27 (2), 43.

de Araújo, A.F., Marcacini, R.M., 2021. RE-BERT: automatic extraction of software requirements from app reviews using BERT language model. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. pp. 1321–1327.

Devine, P., Blincoe, K., 2022. Unsupervised extreme multi label classification of stack overflow posts. In: 2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering. NLBSE, IEEE, pp. 1–8.

Devine, P., Koh, Y.S., Blincoe, K., 2021. Evaluating unsupervised text embeddings on software user feedback. In: 2021 IEEE 29th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 87–95.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Diamantopoulos, T., Symeonidis, A.L., 2020. Mining software requirements. In: Mining Software Engineering Data for Software Reuse. Springer, pp. 75–97.

Frith, J., 2017. Forum design and the changing landscape of crowd-sourced help information. Commun. Des. Q. Rev. 4 (2), 12–22.

Gottipati, S., Lo, D., Jiang, J., 2011. Finding relevant answers in software forums. In: 2011 26th IEEE/ACM International Conference on Automated Software Engineering. ASE 2011, IEEE, pp. 323–332.

Haering, M., Stanik, C., Maalej, W., 2021. Automatically matching bug reports with related app reviews. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering. ICSE, IEEE, pp. 970–981.

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

Hey, T., Keim, J., Koziolek, A., Tichy, W.F., 2020. NoRBERT: Transfer learning for requirements classification. In: 2020 IEEE 28th International Requirements Engineering Conference. RE, IEEE, pp. 169–179.

Holtz, P., Kronberger, N., Wagner, W., 2012. Analyzing internet forums. J. Media Psychol.

Iqbal, T., Khan, M., Taveter, K., Seyff, N., 2021. Mining reddit as a new source for software requirements. In: 2021 IEEE 29th International Requirements Engineering Conference. RE, IEEE, pp. 128–138.

Jafarzadeh, A.A., Mahdavi, A., Jafarzadeh, H., 2017. Evaluation of forest fire risk using the Apriori algorithm and fuzzy c-means clustering. J. Forest Sci. 63 (8), 370–380.

Khan, J.A., Xie, Y., Liu, L., Wen, L., 2019. Analysis of requirements-related arguments in user forums. In: 2019 IEEE 27th International Requirements Engineering Conference. RE, IEEE, pp. 63–74.

Krippendorff, K., 2018. Content Analysis: An Introduction to Its Methodology. Sage publications.

Maalej, W., Nabil, H., 2015. Bug report, feature request, or simply praise? on automatically classifying app reviews. In: 2015 IEEE 23rd International Requirements Engineering Conference. RE, IEEE, pp. 116–125.

Martens, D., Maalej, W., 2019. Extracting and analyzing context information in user-support conversations on twitter. In: 2019 IEEE 27th International Requirements Engineering Conference. RE, IEEE, pp. 131–141.

Martin, W., Sarro, F., Jia, Y., Zhang, Y., Harman, M., 2016. A survey of app store analysis for software engineering. IEEE Trans. Softw. Eng. 43 (9), 817–847.

Mekala, R.R., Irfan, A., Groen, E.C., Porter, A., Lindvall, M., 2021. Classifying user requirements from online feedback in small dataset environments using deep learning. In: 2021 IEEE 29th International Requirements Engineering Conference. RE, IEEE, pp. 139–149.

Mezouar, M.E., Zhang, F., Zou, Y., 2018. Are tweets useful in the bug fixing process? an empirical study on firefox and chrome. Empir. Softw. Eng. 23, 1704–1742.

Nasehi, S.M., Sillito, J., Maurer, F., Burns, C., 2012. What makes a good code example?: A study of programming Q&A in StackOverflow. In: 2012 28th IEEE International Conference on Software Maintenance. ICSM, IEEE, pp. 25–34.

Nayebi, M., Cho, H., Ruhe, G., 2018. App store mining is not enough for app improvement. Empir. Softw. Eng. 23 (5), 2764–2794.

Noei, E., Lyons, K., 2019. A survey of utilizing user-reviews posted on google play store. In: Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering. pp. 54–63.

Nugroho, Y.S., Islam, S., Nakasai, K., Rehman, I., Hata, H., Kula, R.G., Nagappan, M., Matsumoto, K., 2021. How are project-specific forums utilized? A study of participation, content, and sentiment in the eclipse ecosystem. Empir. Softw. Eng. 26, 1–33.

Pagano, D., Maalej, W., 2013. User feedback in the appstore: An empirical study. In: 2013 21st IEEE International Requirements Engineering Conference. RE, IEEE, pp. 125–134.

Rahimi, M., Cleland-Huang, J., 2014. Personas in the middle: Automated support for creating personas as focal points in feature gathering forums. In: Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering. pp. 479–484.

Seyff, N., Ollmann, G., Bortenschlager, M., 2014. AppEcho: a user-driven, in situ feedback approach for mobile platforms and applications. In: Proceedings of the 1st International Conference on Mobile Software Engineering and Systems. pp. 99–108.

Soni, A., Saxena, A., Bajaj, P., 2020. A methodological approach for mining the user requirements using apriori algorithm. J. Cases Inf. Technol. (JCIT) 22 (4), 1–30.

Swamy, K.R., Babu, G.H., Venkatasubbaiah, R., 2015. Identification of frequent item search patterns using APRIORI algorithm and WEKA tool. Int. J. Innov. Technol. Res. 3 (5), 2401–2403.

Tizard, J., Devine, P., Wang, H., Blincoe, K., 2022a. A software requirements ecosystem: Linking forum, issue tracker, and FAQs for requirements management. IEEE Trans. Softw. Eng..

Tizard, J., Rietz, T., Liu, X., Blincoe, K., 2022b. Voice of the users: an extended study of software feedback engagement. Requir. Eng. 27 (3), 293–315.

Tizard, J., Wang, H., Yohannes, L., Blincoe, K., 2019. Can a conversation paint a picture? mining requirements in software forums. In: 2019 IEEE 27th International Requirements Engineering Conference. RE, IEEE, pp. 17–27.

Wang, H., Devine, P., Tizard, J., Shahamiri, S.R., Blincoe, K., 2021. The use of sub-forums in software product forums. In: 2021 IEEE 29th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 293–297.

Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q.-L., Tang, Y., 2023. A brief overview of ChatGPT: The history, status quo and potential future development. IEEE/CAA J. Autom. Sin. 10 (5), 1122–1136.



**Hechen Wang** is a Ph.D. candidate at the University of Auckland, studying under the HASEL research lab (Human Aspects in Software Engineering). His research mainly revolves around understanding Software requirements within user feedback found on internet platforms such as Software forums. His work involves the use of ML and NLP techniques to classify user feedback into requirement relevant categories.

**Peter Devine** is a Ph.D. from The University of Auckland. His main research interest is in using machine learning to classify, filter, cluster, and summarise user feedback in the service of making better requirements. His other interests include using deep learning for source code search and investigating Software industry diversity.

**James Tizard** is a Ph.D. working in the HASEL research lab (Human Aspects in Software Engineering), at the University of Auckland. His research focuses on leveraging ML and NLP techniques to automatically elicit Software requirements from online user feedback. His recent work also investigates the users who give online feedback, and their motivation to give it. James has several published papers in these areas.

**Seyed Reza Shahamiri** is an experienced deep learning Software engineer mainly interested in smart assistive technologies there he builds intelligent Software platforms and solutions. He holds BE and ME qualifications in Software Engineering and a Ph.D. in Computer Science. He is currently a Senior Lecturer in Software Engineering with the Department of Electrical, Computer, and Software Engineering, Faculty of Engineering, The University of Auckland, New Zealand. His research interest deals with developing novel, intelligent health care systems using deep learning technologies, speech and

speaker identification, and Software test oracle automation. More details can be found at http://www.rezanet.com.

**Kelly Blincoe** is an Associate Professor of Software Engineering at the University of Auckland, New Zealand in the Department of Electrical, Computer, and Software Engineering. Her research is mainly in the human and social aspects of Software engineering, and her current research topics include: Software dependencies, Software ecosystems, collaborative Software development, Software requirements engineering, and Software developer diversity and inclusion. She serves on the Editorial Board of the IEEE Transactions on Software Engineering, Empirical Software Engineering, and the Journal of Systems and Software. More details can be found at https://kblincoe.github.io/. Contact Kelly at k.blincoe@auckland.ac.nz.