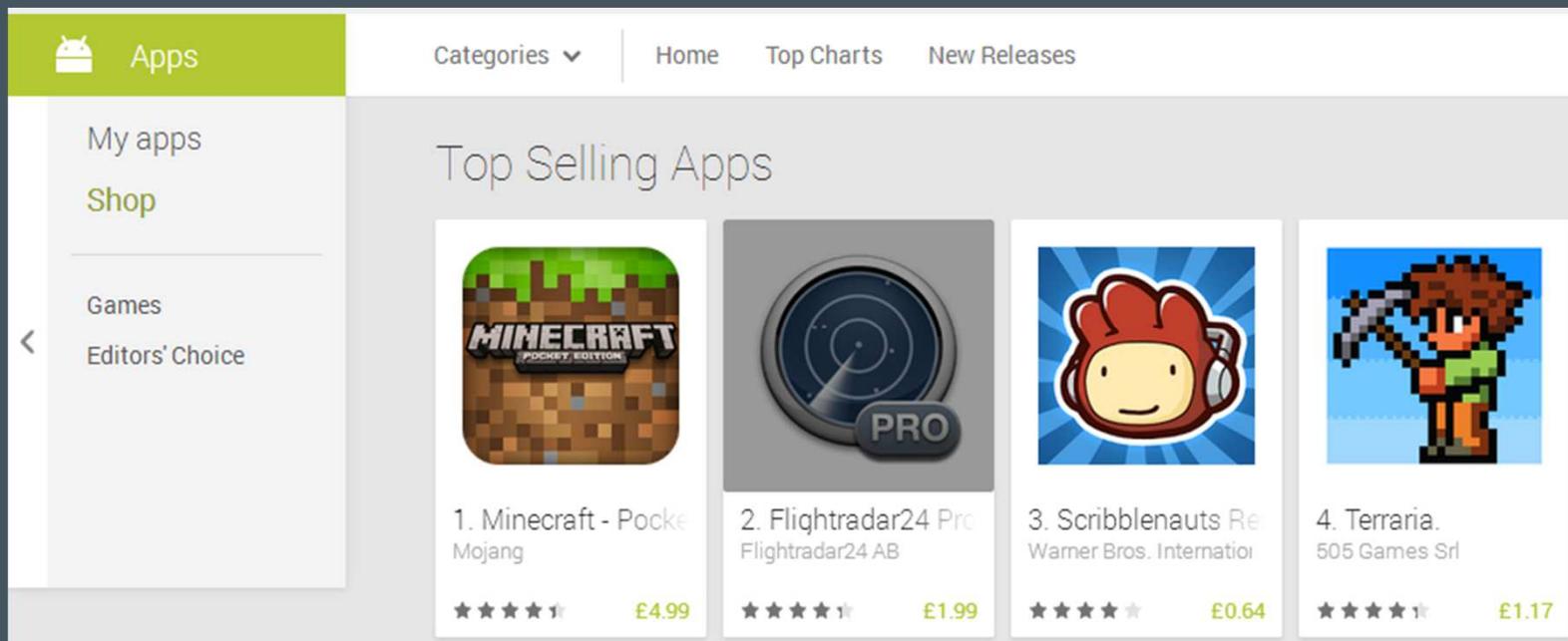# ANALYSIS OF REVIEWS FROM THE GOOGLE PLAY STORE

Prof. Rachel Harrison
Oxford Brookes University
rachel.harrison@brookes.ac.uk

CONTEXT AND MOTIVATION

- 18% of all apps downloaders say ratings and reviews are "*extremely important*", 36% say they are "*very important*," and 34% say they are "*somewhat important*." (Nielsen, 2010)

- The number of customer reviews an app receives tends to grow exponentially (Especially for very popular apps)

- An ever-increasing volume of data to sieve through for useful information
  - critical reviews
  - recurring issues
  - trends reported

www.jolyon.co.uk

STUDY DESIGN – DATA COLLECTION
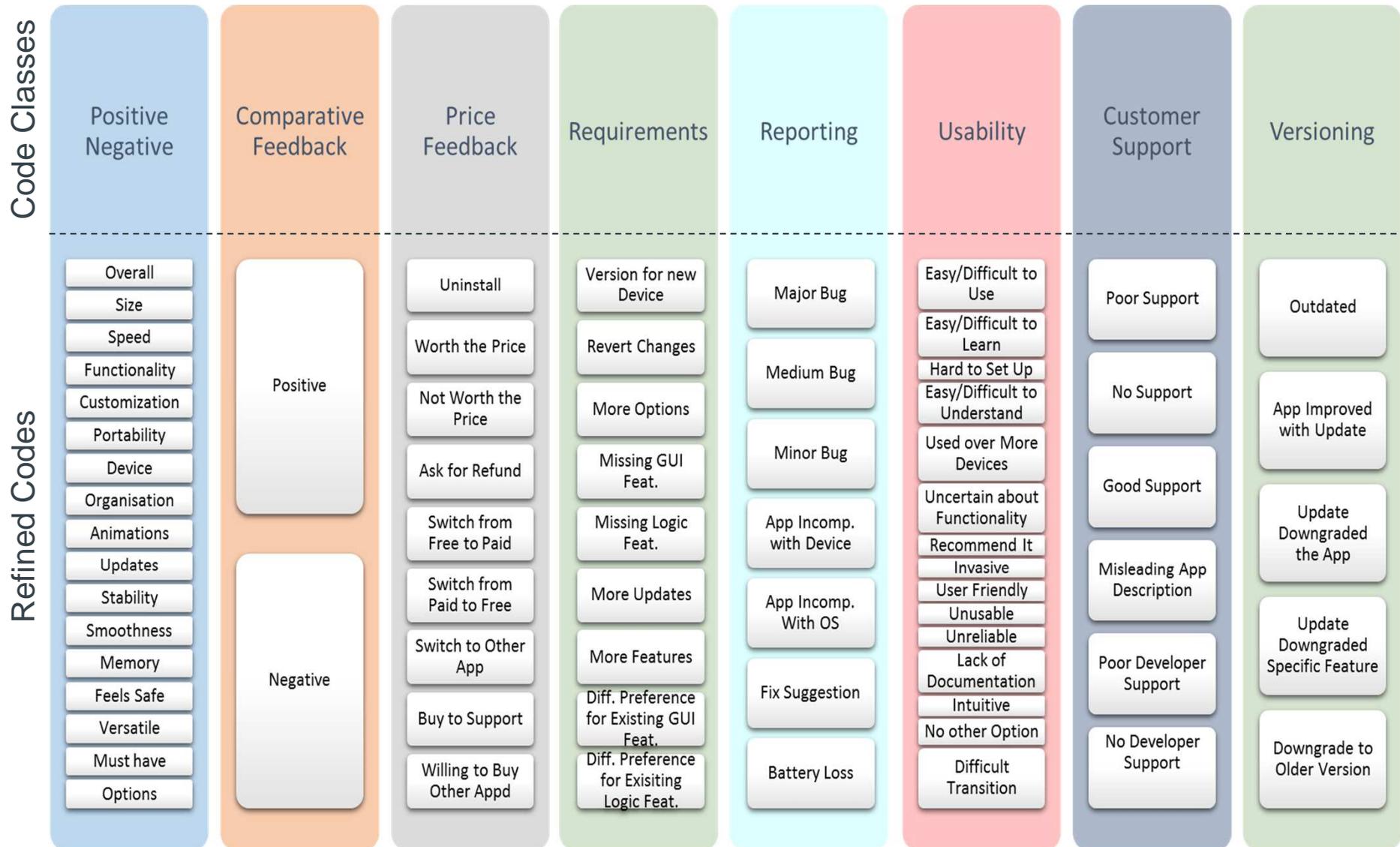
- Google App store*

- 6 most popular categories
    - Personalization
    - Tools
    - Books and references
    - Education
    - Productivity
    - Health and fitness

* Iacob, Claudia, Varsha Veerappa, and Rachel Harrison. "What are you complaining about?: a study of online reviews of mobile applications." *Proceedings of the 27th International BCS Human Computer Interaction Conference.* British Computer Society, 2013.

STUDY DESIGN – DATA COLLECTION

- For each app:
  - Rating, number of ratings, price, size, number of installs, last update, current version, reviews

- For each review:
  - Date, rating, device, version of the app, title, text

- 169 apps & 3279 reviews
  - 4.27 avg. rating
  - 326.83 avg. number of ratings/app
  - £1.92 avg. price

CLASSIFICATION SCHEME

OXFORD
BROOKES
UNIVERSITY

Code Classes

Positive Negative	Comparative Feedback	Price Feedback	Requirements	Reporting	Usability	Customer Support	Versioning

Refined Codes

Positive Negative	Comparative Feedback	Price Feedback	Requirements	Reporting	Usability	Customer Support	Versioning
Overall	Positive	Uninstall	Version for new Device	Major Bug	Easy/Difficult to Use	Poor Support	Outdated
Size		Worth the Price	Revert Changes	Medium Bug	Easy/Difficult to Learn	No Support	App Improved with Update
Speed		Not Worth the Price	More Options	Minor Bug	Hard to Set Up	Good Support	Update Downgraded the App
Functionality		Ask for Refund	Missing GUI Feat.	App Incomp. with Device	Easy/Difficult to Understand	Misleading App Description	Update Downgraded Specific Feature
Customization		Switch from Free to Paid	Missing Logic Feat.	App Incomp. With OS	Used over More Devices	Poor Developer Support	Downgrade to Older Version
Portability	Negative	Switch from Paid to Free	More Updates	Fix Suggestion	Uncertain about Functionality	No Developer Support	
Device		Switch to Other App	More Features	Battery Loss	Recommend It		
Organisation		Buy to Support	Diff. Preference for Existing GUI Feat.		Invasive		
Animations		Willing to Buy Other Appd	Diff. Preference for Exisiting Logic Feat.		User Friendly		
Updates					Unusable		
Stability					Unreliable		
Smoothness					Lack of Documentation		
Memory					Intuitive		
Feels Safe					No other Option		
Versatile					Difficult Transition		
Must have							
Options							

CLASSIFICATION SCHEME

- Agreed on by 2 reviewers

- How it works:

"*Works good prefer over swype. Wish it had a smiley face button and issues with typing a single letter, doesn't automatically space.*"

Snippet	Code Class	Refined Code
"*Works good*"	"Positive Feedback"	"Overall"
"*prefer over swype*"	"Comparative Feedback"	"Positive"
"*wish it had a smiley face button*"	"Requirements"	"Missing gui feature"
"*issues with typing a single letter*"	"Reporting"	"Minor bug"
"*doesn't automatically space*"	"Requirements"	"Missing logic feature"

RESULTS 1 – HOW DO USERS RATE APPS?

- Lower ratings are often accompanied by concerns about customer support with no concern at about versioning

- Middle ratings are mostly likely to be linked to bug reports

- High ratings occur with requirements requests


Distribution of Code Classes by Star Rating
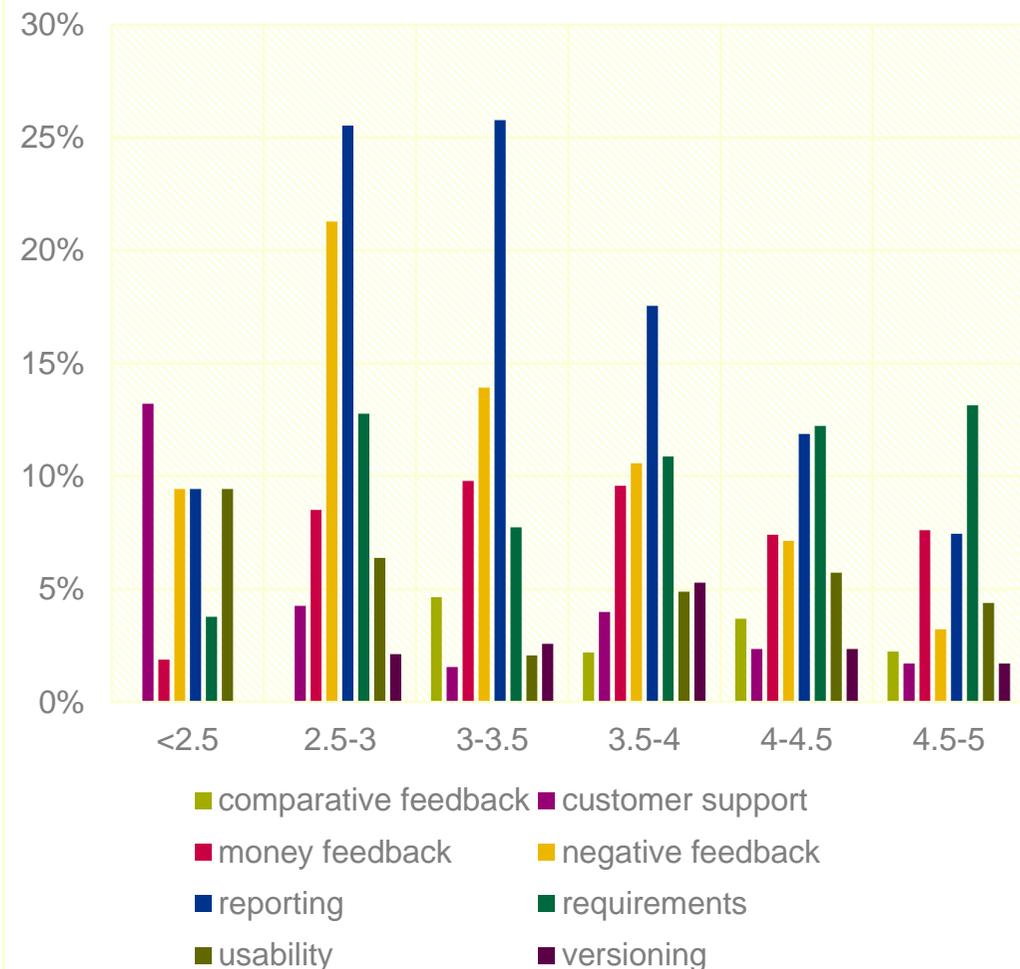
RESULT 2- HOW DO REVIEWS VARY WITH PRICE?

OXFORD BROOKES UNIVERSITY

- The main concern for reviews for cheapest apps was mostly about requirements while that more expensive ones was bug reporting.

- Price and money feedback was positively correlated

- There is a weak negative correlation between price and requirements


Distribution of Class Codes by Price
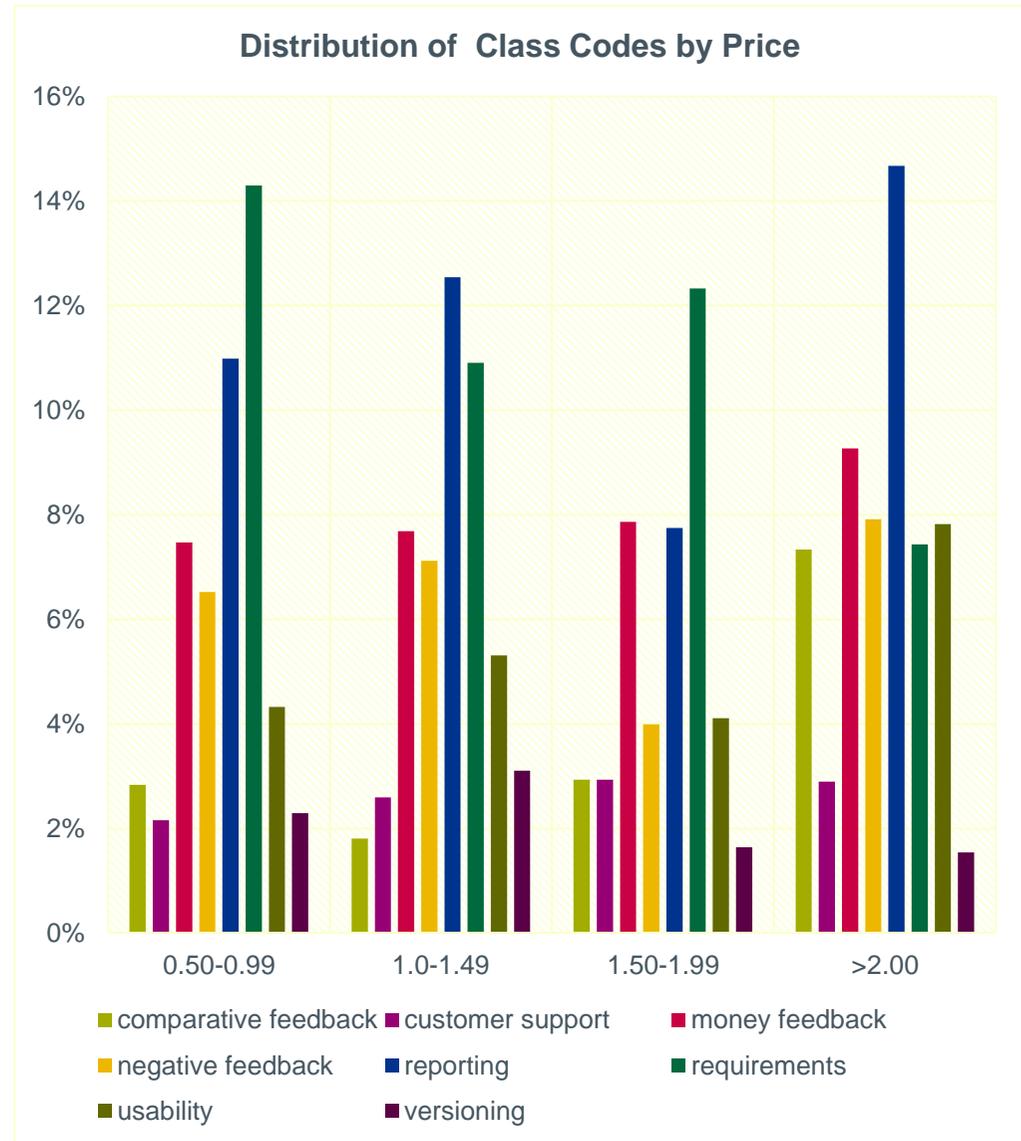
RESULT 3(A) – DISTRIBUTION OF REVIEWS ACROSS CLASS OF CODES

- Users tend to provide positive feedback.

- Reviews are used for expressing requirements and reporting bugs.

- Users are least concerned with issues related to versioning

No. of Reviews by Code Class
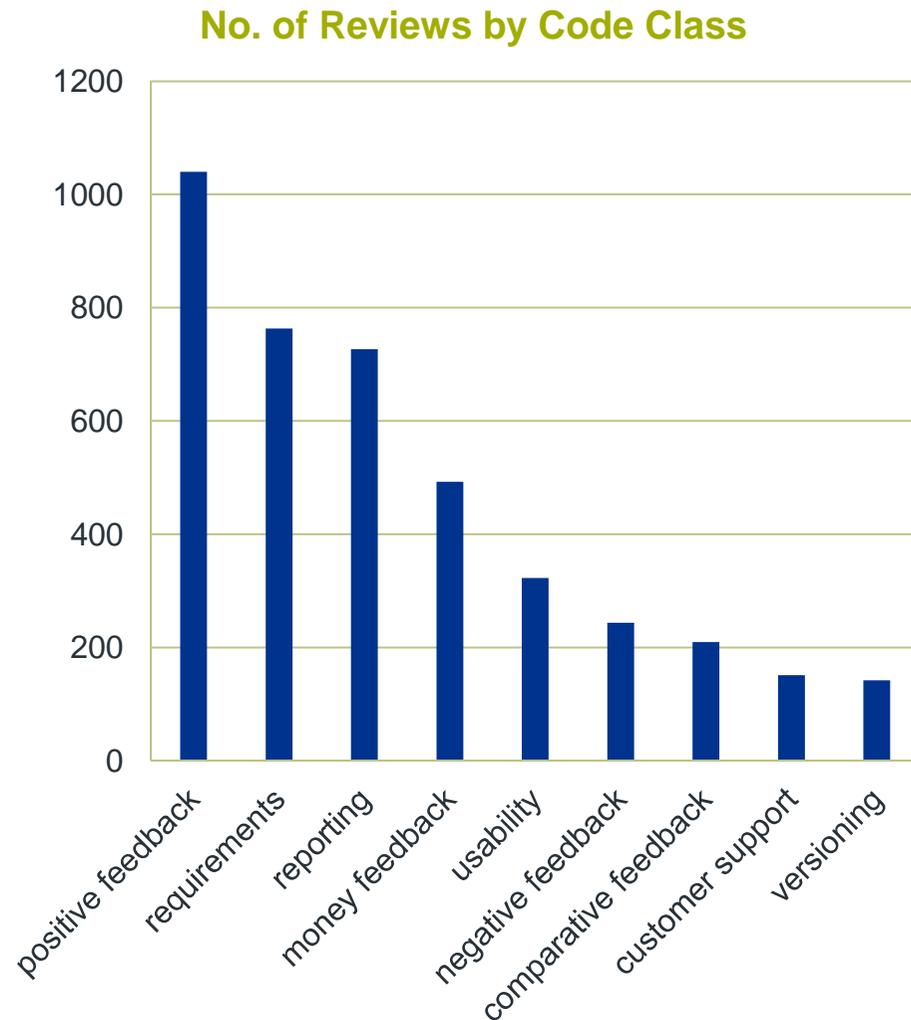
RESULT 3(B) – DISTRIBUTION OF REVIEWS ACROSS REFINED CODES

- Omitted "Overall" which dominates the distribution with 2219 reviews.

- Users write mostly about the functional aspect of apps.

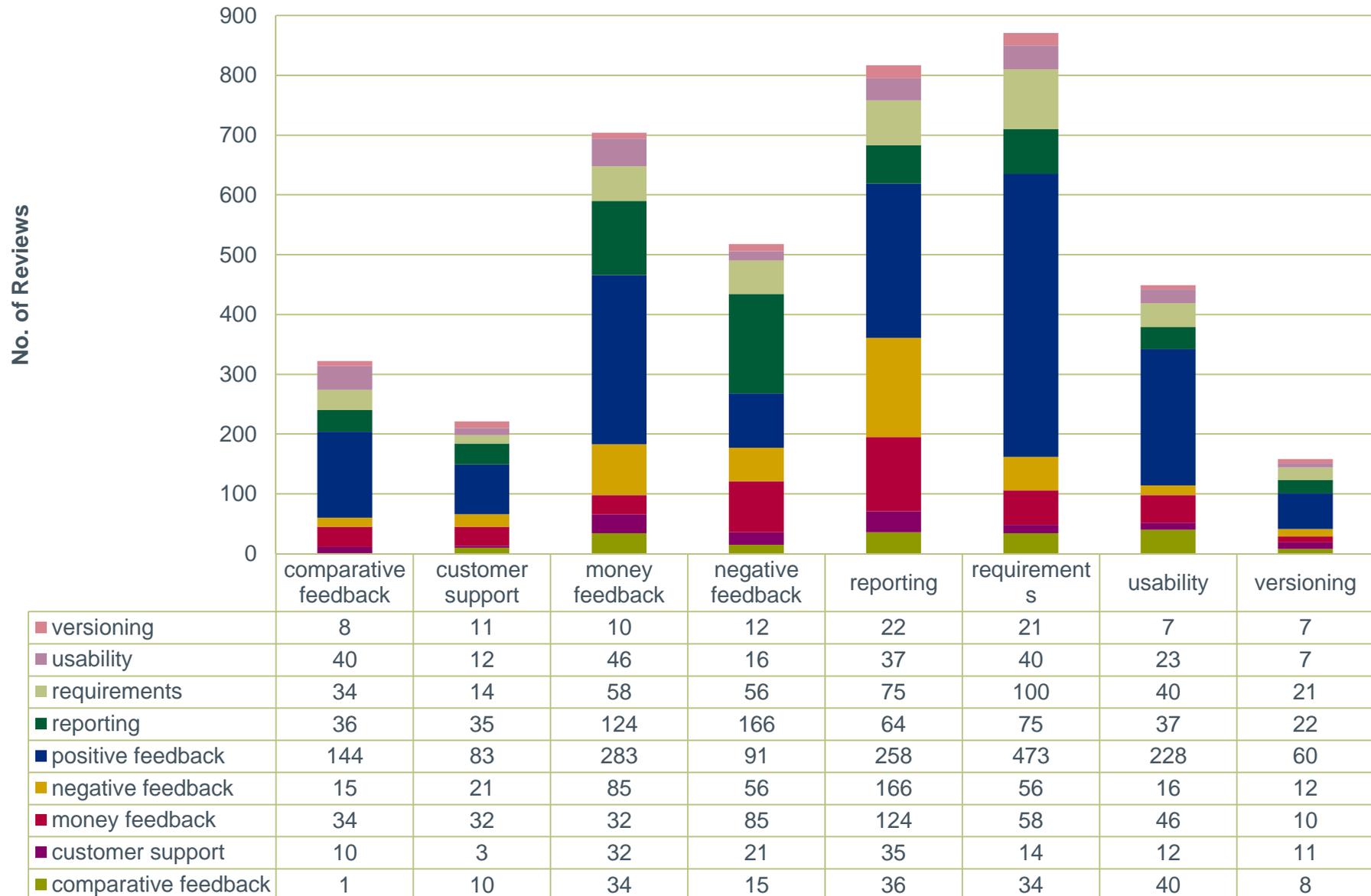**Distribution of Reviews by Refined Code**
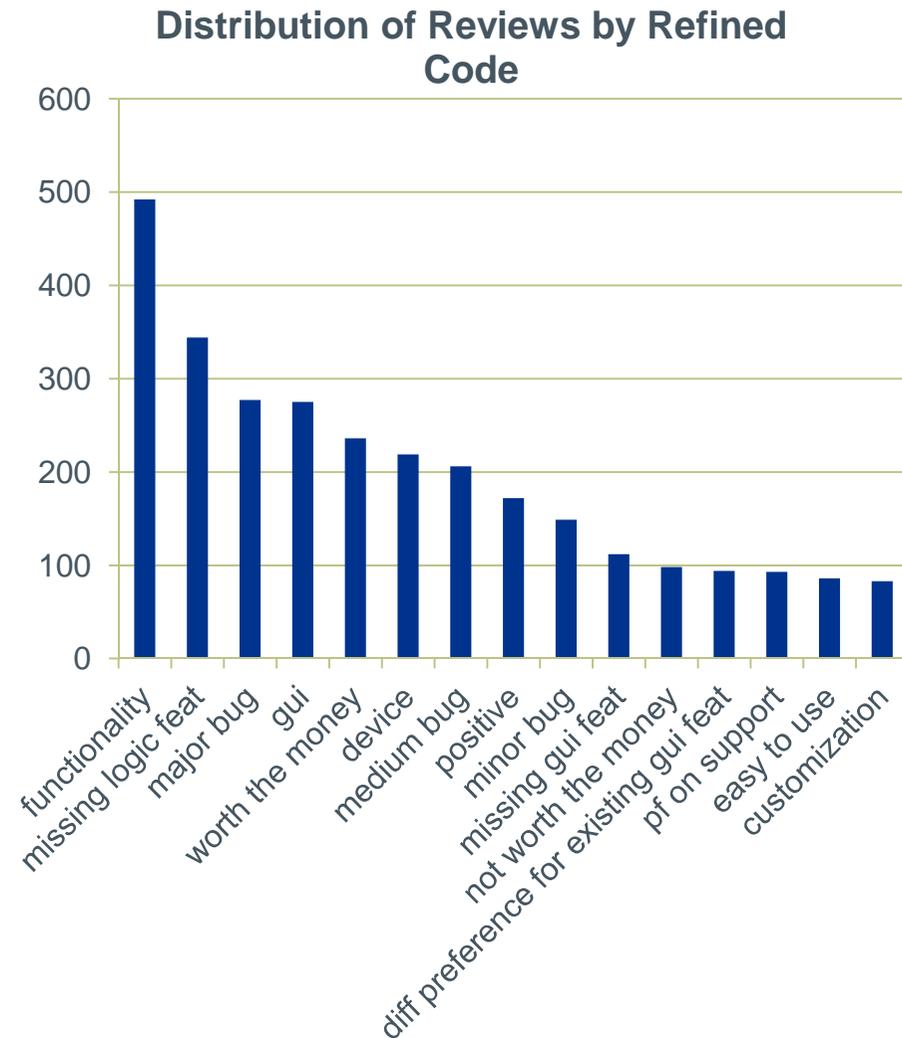
RESULT 4 – COMMONLY OCCURRING PAIRS OF CLASS OF CODE IN REVIEWS

- Main Observations

  - Positive feedback is dominant across the reviews

  - Users tend to provide more than one type of feedback in a review

  - Money feedback occurred mostly with Reporting and Negative feedback.

  - Users tend to group multiple Requirements related issues in a review

  - The main measure of comparative feedback is usability

RESULT 4 – COMMONLY OCCURRING PAIRS OF CLASS OF CODE IN REVIEWS

No. of Reviews

|	comparative feedback	customer support	money feedback	negative feedback	reporting	requirements	usability	versioning
versioning	8	11	10	12	22	21	7	7
usability	40	12	46	16	37	40	23	7
requirements	34	14	58	56	75	100	40	21
reporting	36	35	124	166	64	75	37	22
positive feedback	144	83	283	91	258	473	228	60
negative feedback	15	21	85	56	166	56	16	12
money feedback	34	32	32	85	124	58	46	10
customer support	10	3	32	21	35	14	12	11
comparative feedback	1	10	34	15	36	34	40	8

RESULT 5 – COMMONLY OCCURRING PAIRS OF (CODE CLASS, REFINED CODE) TUPLES IN REVIEWS

- Interesting observations:

  - (Positive feedback, overall) and (Requirements, Missing logic feat) appeared together in 188 reviews (most commonly occurring pair of tuples in dataset)
    - Users are always looking for improvements in apps

  - (Positive feedback, overall) and (Positive feedback, GUI) appeared together in 176 reviews (2nd most commonly occurring pair of tuples in dataset)
    - A good GUI makes users happy

  - (Positive feedback, functionality) was paired with (Positive feedback, overall) , (money feedback, worth the money), (positive feedback, gui) and (comparative feedback, positive) in 172, 51, 50 and 49 reviews respectively.
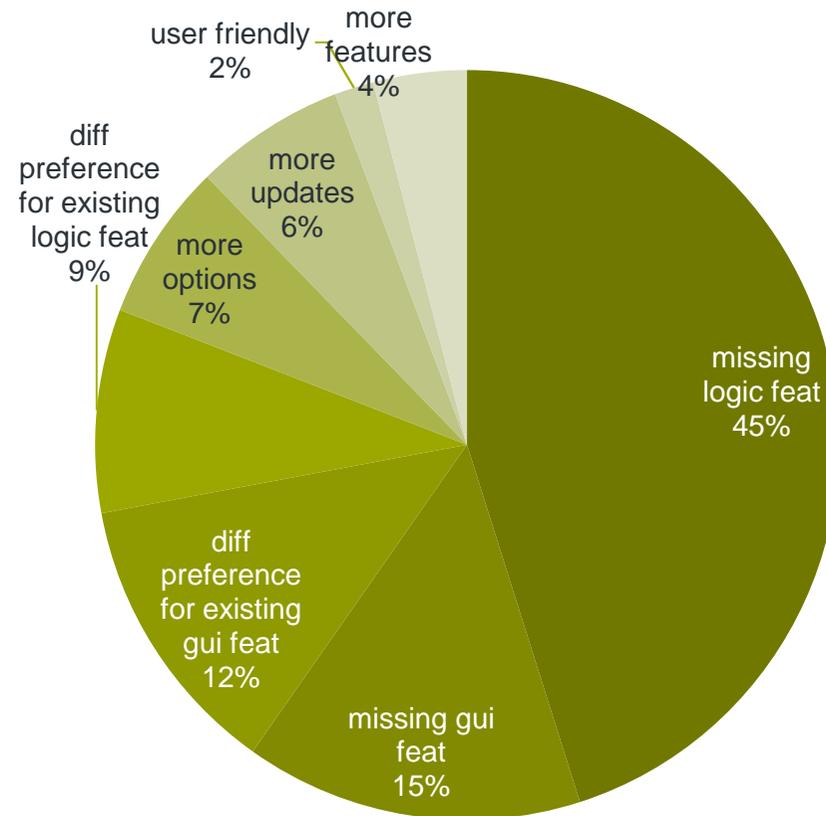   - Good functionality made users feel that they are getting value for money

RESULT 5 – COMMONLY OCCURRING PAIRS OF (CODE CLASS, REFINED CODE) TUPLES IN REVIEWS
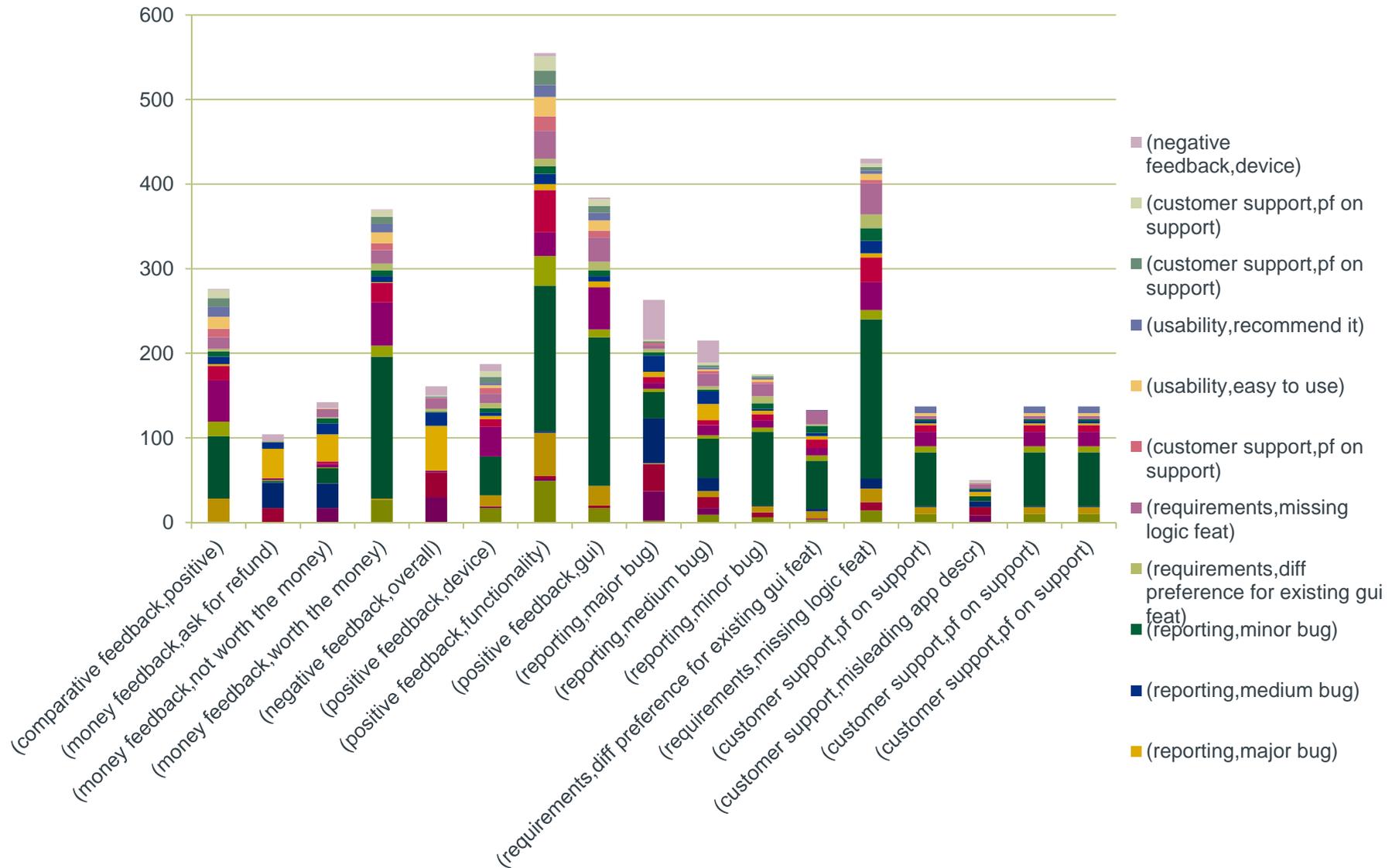
MORE ABOUT REQUIREMENTS

- Users report what needs to be added to the app to make it more useful to them

- How can we use this to extract feature requests?

Requirements

Pie chart: Requirements

- missing logic feat 45%
- missing gui feat 15%
- diff preference for existing gui feat 12%
- diff preference for existing logic feat 9%
- more options 7%
- more updates 6%
- more features 4%
- user friendly 2%

WHAT DO THESE FEATURE REQUESTS LOOK LIKE?

"simply the best keyboard there ever is! thx for this little piece of magic dev! :) keep up the good work. btw *could we pls have mandarin language pack ?*"

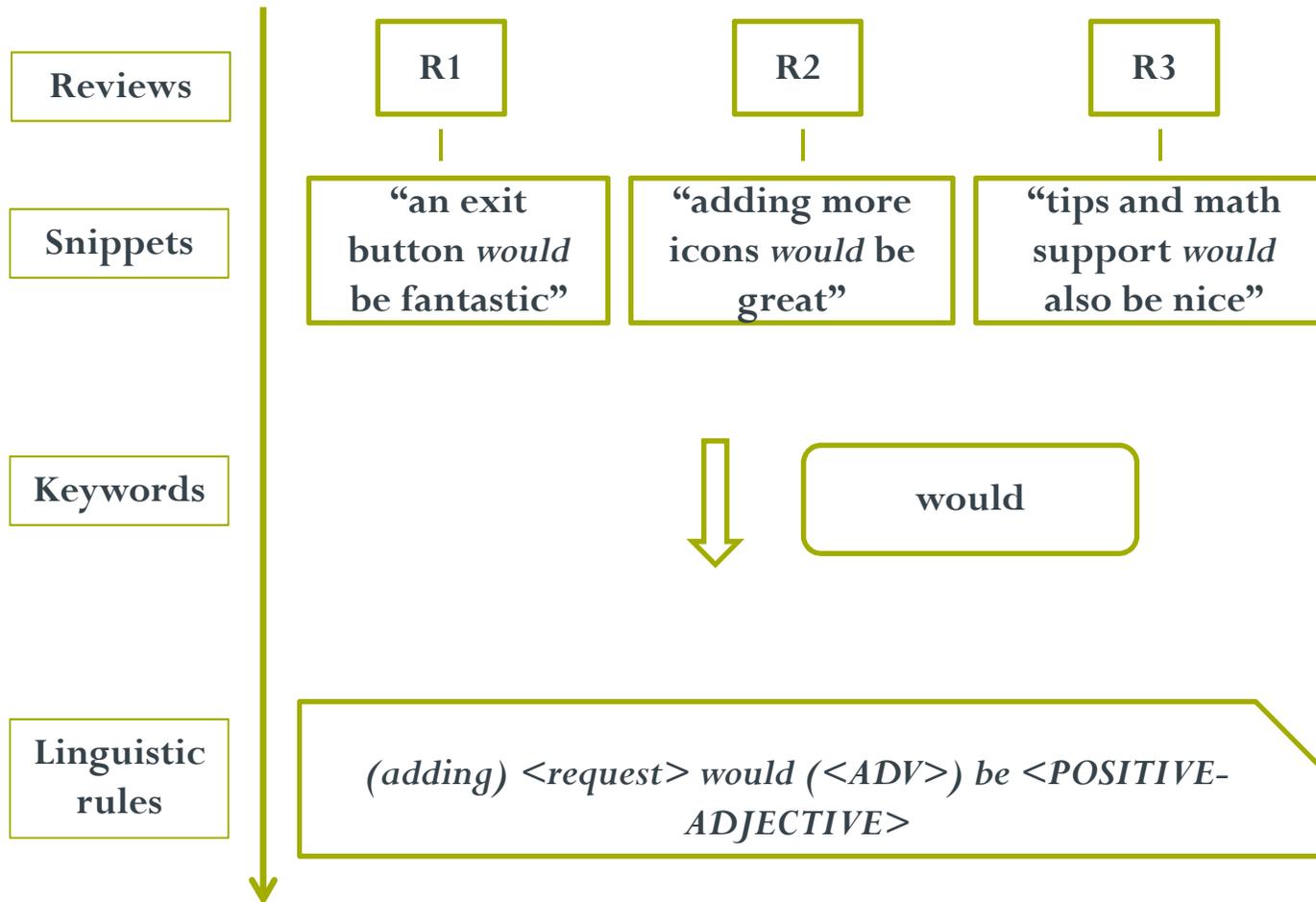"This "FaceBook" Application is the best!   Just one major flaw, which needs to be fixed IMMEDIATELY!   This "FaceBook" Application *NEEDS TO HAVE the Features: Bold, Underline, and Italics*!   PLEASE FIX IMMEDIATELY!   Is there any way that this "FaceBook" Application can please *be upgraded as soon as possible to include the Features: Bold, Underline, and Italics??*"

MEET MARA*
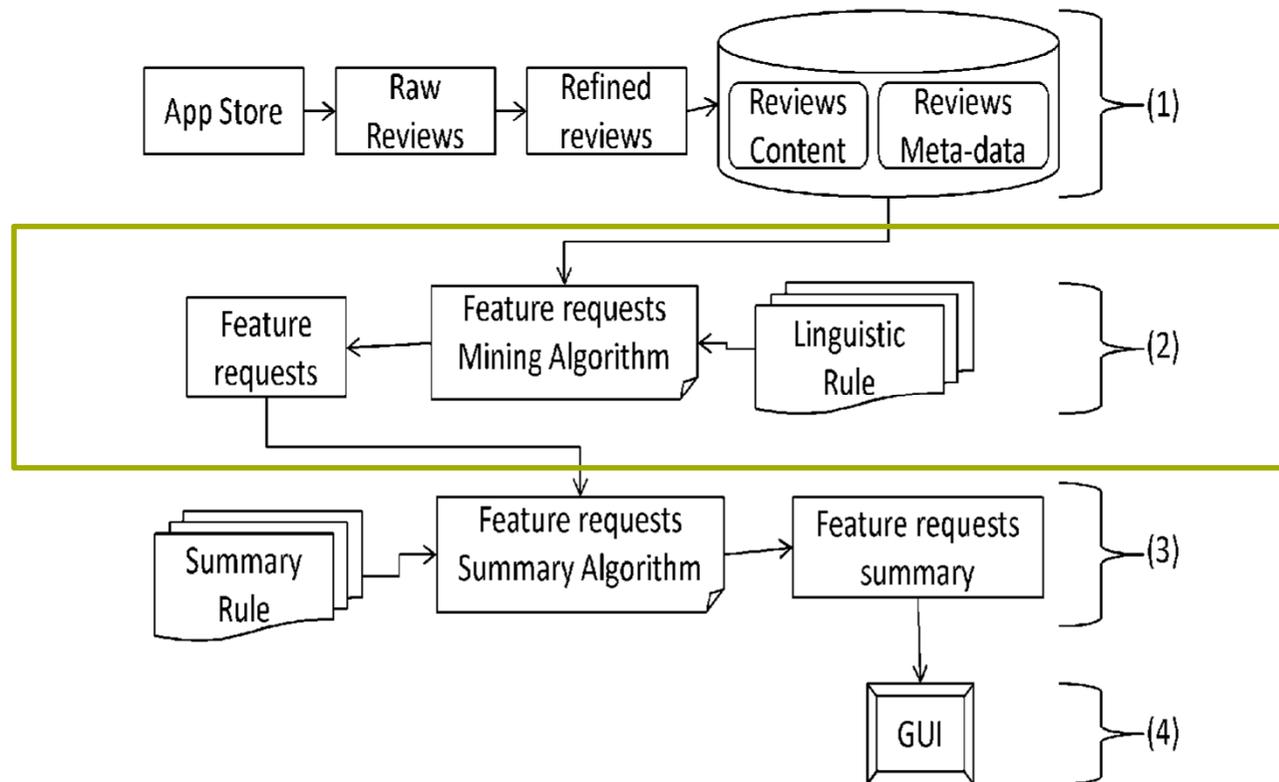
* Iacob, Claudia, and Rachel Harrison. "Retrieving and analyzing mobile apps feature requests from online reviews." *Mining Software Repositories (MSR)*, 2013 10th IEEE Working Conference on. IEEE, 2013.

DERIVING LINGUISTIC RULES

Reviews

| R1 | R2 | R3 |

Snippets

"an exit button *would* be fantastic"

"adding more icons *would* be great"

"tips and math support *would* also be nice"

Keywords

would

Linguistic rules

(adding) <request> would (<ADV>) be <POSITIVE-ADJECTIVE>

EVALUATION

136,998 reviews

Pre-processing

237 Linguistic Rules

P = TP/(TP+FP)

R = TP/(TP+FN)

$$MCC = \frac{TPXTN - FPXFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Feature Request Mining Algorithm

Feature Requests

Sample 1, size 3000

Sample 2, size 480

Randomly selected 3000 feature requests returned and checked whether they were TPs.

Randomly selected one app and chose its reviews as a sample for counting the FNs and TNs.

P = 0.85

R = 0.87
MCC = 0.90

LINGUISTIC RULES

Linguistic Rule	Example Context
<request> would make it <COMPARATIVE-ADJ>	"support for VTODO would make it much cooler"
(<SB>) (<ADV>) wish there was <request>	"I just wish there was the smiley editor ability"
<request> should be <COMPARATIVE-ADJ > than <existing-feature>	"the long press should be shorter than 0.25 seconds"
wish < request> instead of <existing feature>	"Wish the 2 add-ons were in a bundle pack instead of doing two transactions"
please include <-request>	"Next update please include a journaling feature with a keyword search"
could use (more) < request>	"Could use more icons"; "could use zoom and horizontal layouts"
add the ability to <request>	"Add the ability to create walls so they don't go off screen and to make cool mazes"
(the only thing) missing <request>	"The only thing missing is font customizations"
needs the ability to <request>	"Needs the ability to set custom wall paper"

CONCLUSION AND FUTURE WORK

- Similar work on the App Store reviews (currently)

- More detailed analysis of reviews stores

- More precise definition of classification scheme

- Strategies to extract useful information for mobile app developers

- THANKS!

	comparative feedback	customer support	money feedback	negative feedback	reporting	requirements	usability	versioning
comparative feedback	1	10	34	15	36	34	40	8
customer support	10	3	32	21	35	14	12	11
money feedback	34	32	32	85	124	58	46	10
positive feedback	144	83	283	91	258	473	228	60
negative feedback	15	21	85	56	166	56	16	12
reporting	36	35	124	166	64	75	37	22
requirements	34	14	58	56	75	100	40	21
usability	40	12	46	16	37	40	23	7
versioning	8	11	10	12	22	21	7	7

	(comparative feedback,positive)	(money feedback,ask for refund)	(money feedback,not worth the money)	(money feedback,worth the money)	(negative feedback,overall)	(positive feedback,overall)	(positive feedback,device)	(positive feedback,functionality)	(positive feedback,gui)	(reporting,major bug)	(reporting,medium bug)	(reporting,minor bug)	(requirements,diff preference for existing gui feat)	(requirements,missing logic feat)	(customer support,pf on support)	(usability,easy to use)	(usability,recommend it)	(customer support,pf on support)	(customer support,pf on support)	(negative feedback,device)	(requirements,missing gui feat)	(negative feedback,functionality)	(negative feedback,gui)
(comparative feedback,positive)	1	0	0	27	0	74	17	49	17	2	9	6	3	14	10	14	12	10	10	1	3	2	0
(money feedback,ask for refund)	0	0	17	0	30	2	1	2	0	35	8	0	1	0	0	0	0	0	0	8	1	2	5
(money feedback,not worth the money)	0	17	0	0	29	18	1	4	3	32	13	6	1	10	0	1	0	0	0	7	1	5	4
(money feedback,worth the money)	27	0	0	1	0	168	13	51	23	1	7	7	8	16	8	13	10	8	8	1	3	2	0
(negative feedback,overall)	0	30	29	0	0	0	0	2	0	53	16	1	3	12	1	0	1	1	1	11	2	16	15
(positive feedback,device)	17	1	1	13	0	46	0	35	9	4	4	5	6	11	7	3	3	7	7	8	2	3	1
(positive feedback,functionality)	49	2	4	51	2	172	35	28	50	7	12	9	9	33	17	23	14	17	17	4	7	7	2
(positive feedback,gui)	17	0	3	23	0	176	9	50	0	7	6	7	10	29	8	12	9	8	8	2	9	3	1
(reporting,major bug)	2	35	32	1	53	31	4	7	7	6	19	4	4	5	2	0	0	2	2	47	0	3	1
(reporting,medium bug)	9	8	13	7	16	46	4	12	6	19	15	2	4	15	3	2	2	3	3	26	3	10	3
(reporting,minor bug)	6	0	6	7	1	87	5	9	7	4	2	7	8	15	2	3	2	2	2	0	7	1	3
(requirements,diff preference for existing gui feat)	3	1	1	8	3	57	6	9	10	4	4	8	2	16	0	0	1	0	0	0	5	1	3
(requirements,missing logic feat)	14	0	10	16	12	188	11	33	29	5	15	15	16	37	4	7	4	4	4	6	17	11	4
(customer support,pf on support)	10	0	0	8	1	64	7	17	8	2	3	2	0	4	0	3	8	0	0	0	1	1	0
(customer support,misleading app descr)	0	8	10	0	7	6	0	0	0	5	2	2	0	5	1	0	0	1	1	2	0	1	1
(customer support,pf on support)	10	0	0	8	1	64	7	17	8	2	3	2	0	4	0	3	8	0	0	0	1	1	0
(customer support,pf on support)	10	0	0	8	1	64	7	17	8	2	3	2	0	4	0	3	8	0	0	0	1	1	0
(money feedback,uninstall)	0	7	5	0	12	8	0	1	3	7	6	2	2	2	0	0	0	0	0	0	1	5	6
(money feedback,buy to support)	0	0	0	1	0	15	1	5	1	0	0	2	1	2	0	0	1	0	0	0	0	0	0
(money feedback,switch from free to paid)	1	0	0	2	1	18	0	3	1	0	1	2	0	2	1	0	0	1	1	0	2	0	1
(negative feedback,device)	1	8	7	1	11	13	8	4	2	47	26	0	0	6	0	0	0	0	0	0	2	3	1
(negative feedback,functionality)	2	2	5	2	16	12	3	7	3	3	10	1	1	11	1	1	0	1	1	3	2	5	5
(negative feedback,gui)	0	5	4	0	15	10	1	2	1	1	3	3	3	4	0	0	0	0	0	1	3	5	1
(negative feedback,speed)	1	3	0	1	7	12	2	8	1	5	6	4	2	5	0	0	0	0	0	3	3	3	2
(positive feedback,customization)	6	0	0	7	1	58	2	11	21	1	2	2	5	5	6	6	7	6	6	1	0	0	1