|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Extraction Form** | | | | | | | | | | | | | | | | | | | | | | | |
| **Title** | Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews | | | | | | | | | | **Authors(s)** | | Walid Maalej, Hadeer Nabil | | | | | | | | | | |
| **Year** | 2015 | | | | | | | | | | **Venue** | | * **Journal** | | * Conference | | | | | * Other \_\_\_\_\_\_\_\_\_\_ | | | |
| **Quality Assessment criteria** | | | | * QC1 | | | | * **QC2** | | | | | * **QC3** | | * QC4 | | | | | * **QC5** | | | * QC6 |
| **Inclusion Criteria** | | | | * IC1 | | | | * IC2 | | | | | * **IC3** | | | * **IC4** | | | | * IC5 | | | |
| **Exclusion Criteria** | | | | * EC1 | | | * EC2 | | * EC3 | | | | * EC4 | * EC5 | | | | | * EC6 | | * EC7 | | |
| **Approach Used**   * introduces several probabilistic techniques to classify app reviews into four types: bug reports, feature requests, user experiences, and ratings. | | | | | | | | | | **Type of Solution** | | | | | | | | **Yes** | | **No** | | **Unclear** | |
| Novel Technique (Method, Tool, Technique) | | | | | | | | Check mark, Wingdings font, character code 252 decimal. | |  | |  | |
| Evaluation of existing techniques  (Evaluation framework, tool, platform) | | | | | | | |  | | Check mark, Wingdings font, character code 252 decimal. | |  | |
| Supporting techniques | | | | | | | |  | | Check mark, Wingdings font, character code 252 decimal. | |  | |
| **Review dataset** | | **Total number of apps** | | | | App Store application reviews | | | | **Evaluation Method Used** | | | | | | | | Runtime monitoring architecture for video games (Game Loop) | | | | | |
| **Total number of crawled reviews** | | | | 1126453 | | | |
| **Year** | | | | 2013 | | | |
| **Research Type Facet**   * Validation Research * **Evaluation Research** * **Solution Proposal** * Philosophical Papers * Opinion Papers * Experience Papers | | | | | **Solution Type**   * **Single** * Hybrid/Integrated | | | | | | | **Contribution**   * Technique * **Tool** * Comparison * Model * Framework * Prototype * Taxonomy | | | | | **Evaluation Strategy**   * Case Study * Controlled Experiment * **Survey** * Questionnaire | | | | | | |
| **Features used**   * Categorical * **Textual** * Both | | | | | | |
| **Factors Considered** | | | Test ready models | | | | | | | | | | | | | | | | | | | | | |
| **Notes** | | |  | | | | | | | | | | | | | | | | | | | | | |
| **Limitations** | | | * Focused only on functional testing of the game * Developers have to develop separate test ready models * Test ready models requires time and cost | | | | | | | | | | | | | | | | | | | | | |
| **Description / Summary** | | | The paper provides a detailed modeling methodology to support automated system-level game testing. As part of the methodology,  they provide guidelines for modeling the platform games for testing using our proposed game test modeling profile. They have used domain modeling for representing the game structure and UML state machines for behavioral modeling. They presented the details related to automated test case generation, execution, and oracle generation. Demonstrate their model-based testing approach by applying it on two cases studies, a widely referenced and open source implementation of Mario brothers game and an industrial case study of an endless runner game. | | | | | | | | | | | | | | | | | | | | | |