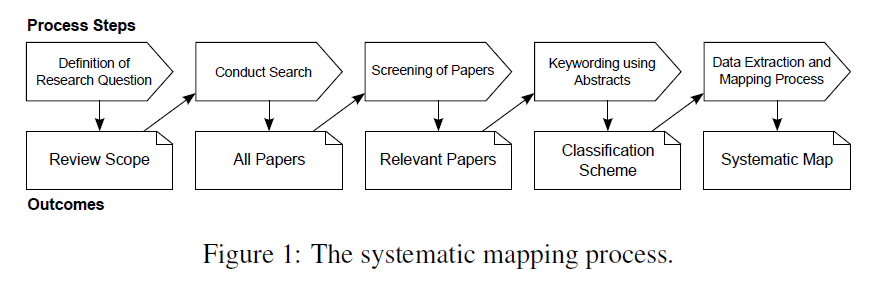
Classification systématique : utilisabilité des APIs

**Auteurs :** Selsabil Dbouba, Constantin Masson, Khaled Fayala, Maxime Gallais-Jimenez



# Définition des hypothèses de recherche

Cette classification systématique nous a été demandé dans le cours de maîtrise « Méthode empirique en génie logiciel » de Houari Sahraoui. Le sujet de notre revue est l’utilisabilité des APIs. L’utilisabilité est un sujet complet que nous allons étudier sous deux aspects. D’une première part qu’est ce qui définit l’utilisabilité et dans un second temps comment l’améliorer. Nous tenterons de répondre aux deux problématiques suivantes.­­­

RQ1 : Quels sont les **critères** d’une utilisabilité efficace d’API ?

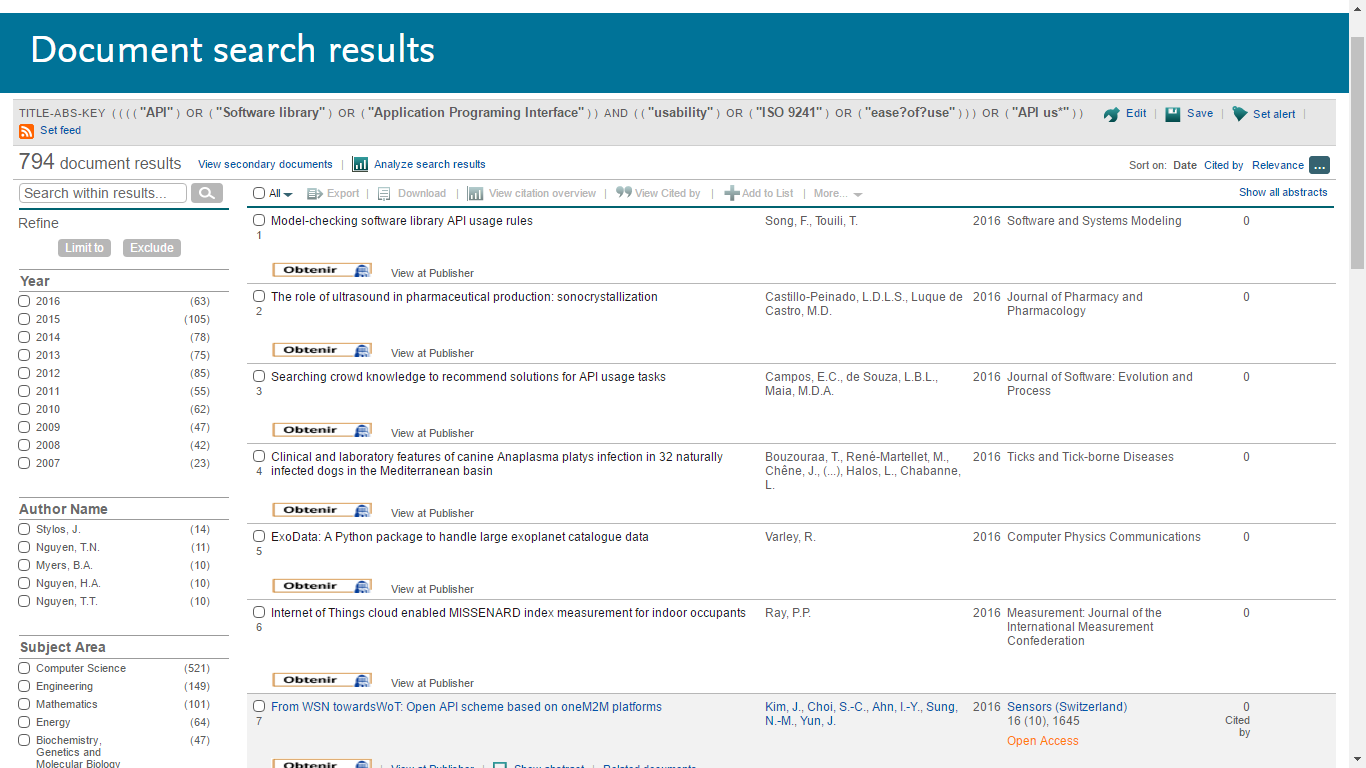
RQ2 : Comment **améliorer** l’utilisabilité d’une API ?

# Choix des mots clés

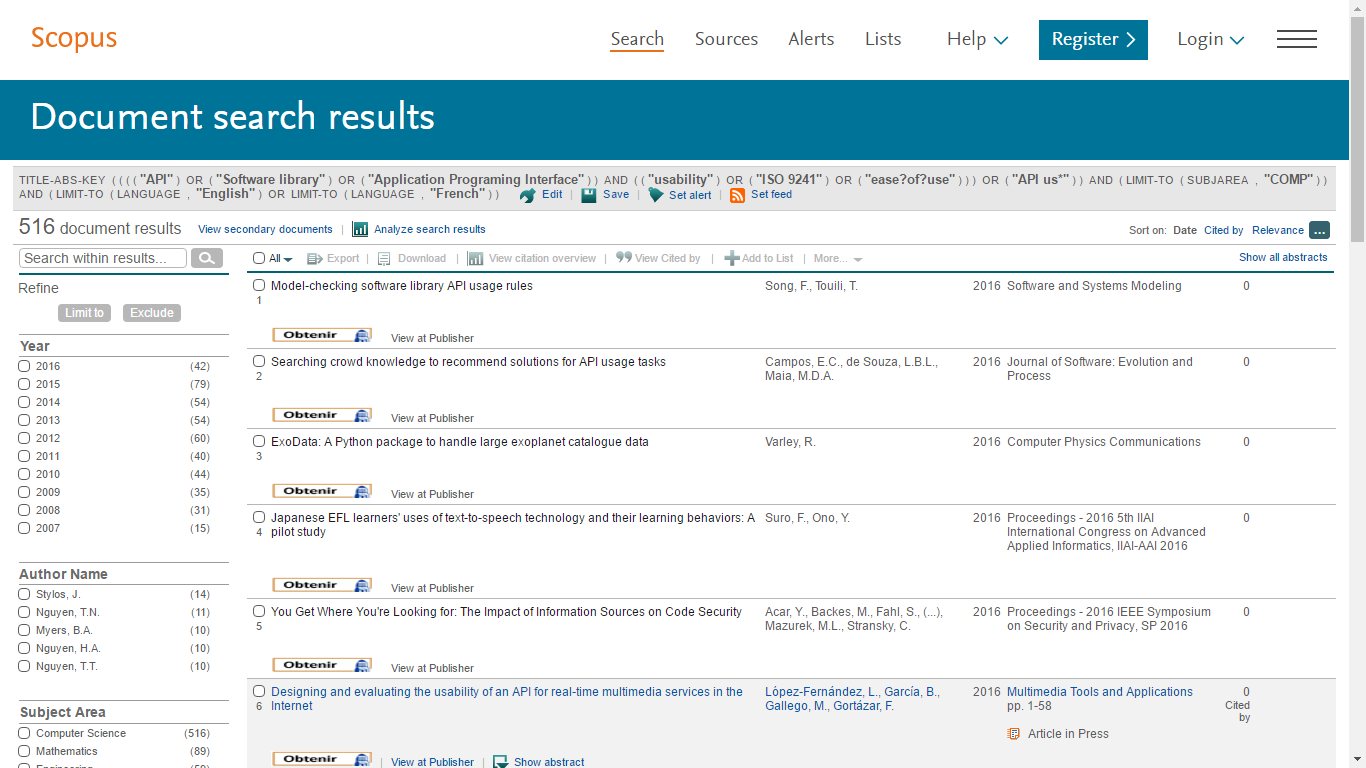
Pour les mots clés nous avons donc choisi les mots *API* et *usability* qui sont les termes de notre sujet. Puis nous avons décidé de les dériver afin de rechercher les termes : *software library* et *Application Programing Interface* en dérivé d’API et *ISO 9241* qui est la norme pour l’utilisabilité et *ease-of-use* qui est un concept qui définit bien l’utilisabilité. Nous avons ajouté à la requête *API us\** afin de pouvoir avoir les articles en fonction de de l’action d’utiliser une API. Cette recherche a été effectuée seulement dans le titre, le résumé et les mots clés de l’article.

**Requête** : *TITLE-ABS-KEY((((‘’API’’) OR (‘’Software library’’) OR (‘’Application Programming Interface’’)) AND ((‘’usability’’) OR (‘‘ISO 9241’’) OR (‘’ease ?of ?use’’)) OR (‘’API us\*’’)*

Cette requête nous a permis d’avoir une sélection de 794 articles.



Puis nous avons appliqué des filtres à cette requête. Le premier filtre fût sur le domaine d’application que nous avons réduit à l’informatique (‘’computer sciences’’). Le second filtre a été mis sur la langue que nous avons réduite au Français et l’Anglais. Après filtrage il nous restait 516 documents pour l’étape suivante, le screening.



**Requête :** *TITLE-ABS-KEY((((‘’API’’) OR (‘’Software library’’) OR (‘’Application Programming Interface’’)) AND ((‘’usability’’) OR (‘‘ISO 9241’’) OR (‘’ease ?of ?use’’)) OR (‘’API us\*’’) AND (LIMIT-TO(SUBJAREA, ‘’COMP’’)) AND (LIMIT-TO(LANGUAGE, ‘’English’’) OR LIMIT-TO (LANGUAGE, ‘’French’’)*

# Screening

Suite au screening nous avons obtenu 56 papiers validés, 196 rejetés et 253 ambigus. Dans les 253 ambigus 91 avait été validés par un membre de l’équipe et refusé par un autre. Après réunion de l’équipe nous avons gardé 17 articles. Nous allons travailler sur 73 articles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Acceptés | Ambigus | Rejetés | A discuter | Après conflit |
| 56 | 253 | 196 | 91 | 73 |

# Liste des articles

1. A Case Study of Web API Evolution
2. A case study of API redesign for improved usability
3. A cooperative approach for combining client-based and library-based API usage pattern mining
4. A dataset for API usage
5. A framework for extending usability engineering: API usability essentials: Extending usability via component-based platform
6. A study of modern Linux API usage and compatibility: What to support when you're supporting
7. ACUA: API change and usage auditor
8. An API design process in terms of usability: A case study on building more usable apis for smart TV platform
9. An exploratory study of API usage examples on the web
10. An observational study on API usage constraints and their documentation
11. API peer reviews: A method for evaluating usability of application programming interfaces
12. API usability peer reviews: A method for evaluating the usability of application programming interfaces
13. API usability: CHI'2009 special interest group meeting
14. Applying the cognitive dimensions of API usability to improve API documentation planning
15. Approximate structural context matching: An approach to recommend relevant examples
16. An empirical comparison of static and dynamic type systems on API usage in the presence of an IDE: Java vs. Groovy with eclipse
17. An empirical study of API usability
18. An exploratory study of api changes and usages based on apache and eclipse ecosystems
19. Automated measurement of API usability: The API Concepts Framework
20. Automatic evaluation of API usability using complexity metrics and visualizations
21. Automatic parameter recommendation for practical API usage
22. Approximate structural context matching: An approach to recommend relevant examples
23. Building more usable APIs
24. Constructing usage scenarios for API redocumentation
25. Context-sensitive code completion tool for better API usability
26. CriticAL: A critic for APIs and libraries
27. Designing and evaluating the usability of an API for real-time multimedia services in the Internet
28. Detecting inefficient API usage
29. Evaluating the reference and representation of domain concepts in APIs
30. Generating API-usage example for project developers
31. Improving API documentation usability with knowledge pushing
32. Improving API documentation using API usage information
33. Improving API usability
34. Influence of code completion methods on the usability of APIs
35. Influencing factors on the usability of API classes and methods
36. Informing API design through usability studies of API design choices: A research abstract
37. Linking usage tutorials into API client code
38. MAPO: Mining API usages from open source repositories
39. Mapping the space of API design decisions
40. Measurable concepts for the usability of software components
41. Measuring API Usability
42. Methods towards API usability: A structural analysis of usability problem categories
43. Mining API patterns as partial orders from source code: From usage scenarios to specifications
44. Mining API usage examples from test code
45. Mining API usage patterns by applying method categorization to improve code completion
46. Mining multi-level API usage patterns
47. Neural networks for sentiment analysis on Twitter
48. Obstacles in using frameworks and APIs: An exploratory study of programmers' newsgroup discussions
49. Prose + test cases = specifications
50. Recommending API usages for mobile apps with hidden Markov model
51. Recommending posts concerning API issues in developer Q&A sites
52. Recommending proper API code examples for documentation purpose
53. Recommending reference API documentation
54. Searching the State Space: A Qualitative Study of API Protocol Usability
55. Some structural measures of API usability
56. Statistical learning approach for mining API usage mappings for code migration
57. Synthesizing API usage examples
58. Temporal analysis of API usage concepts
59. The API walkthrough method: A lightweight method for getting early feedback about an API
60. The concept maps method as a tool to evaluate the usability of APIs
61. The factory pattern in API design: A usability evaluation
62. The implications of method placement on API learnability
63. The role of conceptual knowledge in API usability
64. Type names without static type checking already improve the usability of APIs (As Long as the Type Names are Correct): An Empirical Study
65. Unit tests as API usage examples
66. Usability evaluation of configuration-based API design concepts
67. Usability implications of requiring parameters in objects' constructors
68. Usability of educational technology APIs: Findings and guidelines
69. Usable results from the field of API usability: A systematic mapping and further analysis
70. Useful, but usable? Factors affecting the usability of APIs
71. Using concept maps to evaluate the usability of APIs
72. Visualization based API usage patterns refining
73. What makes APIs hard to learn? answers from developers

# Schéma de classification

Le screening des articles et une lecture de tous les articles nous a permis d’élaborer un tri des papiers en fonction de 4 catégories.

Etude / type de validation : (Ajouter comment la validation a été faite en commentaire)

* Etude de cas
* Etude empirique
* Etude exploratoire
* Etude observatoire
* Etude qualitative
* Etude quantitative
* Etude utilisateur

Type de contribution :

* Détection d’artéfact
  + Pattern constraint
  + Usage exemple
  + Data Mining
  + Code completion
  + Documentation
  + Pattenr usage
  + Pattern mining
  + Unit test
* Mesure d’API (Définir les différents types de mesure)
* Etude empirique

Stade de contribution :

* Avant développement
* Pendant développement
* Après développement

Sources de données :

* Données industrielles
* Données hybrides
* Données Open Source
* Réponse d’une enquête

Répartition des articles :

Maxime : 1 à 18

Khaled : 19 à 36

Selsabil : 37 à 55 (47 exclus)

Constantin : 56 à 73

Précision sur les articles :

Nous avons décidé de retirer l’article 47 qui est hors contexte. L’article 4 n’est pas disponible gratuitement.