Résumé

# Réponse à la RQ1 : Quels sont les critères d’une utilisabilité efficace d’API ?

## Définition stricte

L’organisation internationale de standardisation définit l’utilisabilité ainsi : « the extent to which product can be used by specified users to achieve specified goals with **effectiveness**, **efficiency** and **satisfaction** in specified context of use. » [5]. Cela permet d’avoir quelques critères à étudier pour définir l’utilisabilité.

## Utilisateur

Le **ressenti** de l’utilisateur par rapport à une API est un critère très important [2]. De plus le code produit par l’utilisateur est important à étudier. Un **code fiable** peut définir l’utilisabilité de l’API [11]. Critères très importants liés à l’utilisateur est la **compréhensibilité** et le niveau **d’apprentissage** permis par l’API [12].

## Temps d’exécution d’une tâche

Le **temps** passé pour exécuter une tâche est très souvent étudier. L’amélioration du temps d’exécution entre deux versions permet de décider laquelle version d’une API est la meilleure [2]. Une des tâches qui peut être étudiée et la rapidité à traiter les **erreurs** [11].

## Maintenance

L’utilisabilité peut avoir un lien avec la **maintenance**. Lorsque l’API est mise à jour par exemple, le client ne doit pas souffrir de cette situation [7]. Donc l’utilisabilité peut être **évolutive** au cours du temps. Le niveau **d’abstraction** du code est un critère qui peut être pris en compte [12]. Un code au niveau est plus facile à comprendre. Autre point la **réutilisabilité** qui est important si on veut qu’une API soit utilisé et utilisable [12].

## Complétion de code et pratiques

La **complétion** de code est un thème très lié à l’utilisabilité, mais aussi le nombre de **changements** de fichiers, le nombre de **recherches**, le nombre de **complétion** et le temps de complétion peuvent être des facteurs à analyser [11].

## Exemples

Les **exemples** ont une part majeure dans l’utilisabilité d’une API [14]. Avoir des exemples va permettre de faciliter l’apprentissage d’une API. Mais les exemples doivent être le plus exhaustifs possible. C’est pour cela qu’un des critères important est le contexte structural de l’API [18]. Il faut savoir comment l’API va être utilisée.

### 

# Réponse à la RQ2 : Comment améliorer l’utilisabilité d’une API ?

## La documentation

La documentation a une part prépondérante dans l’utilisabilité. Même si le but ultime d’une API est de ne pas avoir besoin d’API tellement elle serait claire.

### La place de la documentation

La documentation est la première chose qu’un utilisateur regarde et a accès. Avoir une documentation solide, et complète est donc important. Beaucoup d’efforts sont mis en œuvre afin de redocumenter les APIs existantes afin de combler les lacunes de celles-ci par exemples en terme de contraintes d’utilisation [15]. Cette documentation doit être initiée avant le développement de l’API [16].

### La documentation dans le temps

Une API évolue avec le temps. Ce qui est important d’améliorer est la documentation au cours du temps. Les changements doivent absolument être notifiés dans la documentation afin que l’utilisateur puisse adapter son utilisation et sa compréhension des exemples [1].

## L’utilisateur

L’utilisateur est la personne centrale en terme d’utilisabilité. Il faut être attentif à ses attentes, ces besoins peuvent être différents de ce que le développeur imaginait. C’est pour cela qu’il est important de faire des études sur les APIs.

Donc l’API se doit de fournir suffisamment de fonctionnalités afin que l’utilisateur puisse réaliser sa tâche. Cependant elle doit rester minimale avec un minimum de contrainte. Chose importante il faut que le **contexte** soit bien assimilé par le développeur afin qu’il puisse répondre aux attentes [16].

### L’enquête

Il est très important d’écouter l’utilisateur. Un des critères à améliorer peut être le niveau d’abstraction des méthodes proposés [2]. Les utilisateurs ont tendances à avoir une pensée haut niveau. Lorsqu’ils utilisent une API ils veulent se simplifier la tâche et donc être le plus efficace possible. C’est pour cela qu’il faut bien étudier comment est utilisée l’API [6, 8].

### Étude cognitive

Le type d’enquête qui essaye d’apporter de la nouveauté dans le domaine est l’étude cognitive. La dimension cognitive est un secteur de plus en plus étudié. Le problème est que les critères de la dimension cognitive sont nombreux et donc le test de dimension cognitif ne peuvent pas s’attacher à tous les points. Cependant ce type d’étude permet de d’être plus proche de l’utilisateur et mettre de la lumière sur les différentes zones d’ombre de l’API [12].

## Les études

Plusieurs types d’études existent à propos de l’utilisabilité. Une des méthodes les plus courantes et les plus documentées est le test empirique. Problème de ce test c’est qu’il est couteux en terme de moyen et de temps. Il peut être complété ou substitué par une « API Peer Review » [8,10]. L’API Peer Review se révèle être un bon complément à l’étude empirique permettant de révéler des problèmes différents et cela plus rapidement. Donc combiner les deux méthodes permet détecter tous les problèmes d’utilisabilité.

## Les exemples

Pour améliorer l’apprentissage et la compréhension d’une API les exemples sont primordiaux. Comme dit Einstein « L’exemple n’est pas une autre façon d’enseigner, c’est la seule. ». La variété des exemples est très importante, elle permet de couvrir de multiples cas. Les exemples doivent aussi être bien décrits avec leur explication en langage naturel de préférence [14]. Les places où trouver des exemples étant variées les développeurs des APIs ne doivent pas hésiter de répondre aux questions posées sur les sites de question réponse afin d’aider les utilisateurs et pouvoir corriger la documentation, les exemples et l’API en elle-même [14]. Il est important de prendre en compte le contexte afin de fournir l’exemple le plus pertinent [18]. Car il existe une multitude d’exemples plus ou moins correctes, sur des plateformes différentes, mais le principal problème de l’utilisateur est de trouver l’exemple qui correspond à son contexte. Des approches sont donc mises en place afin de proposer des exemples plus pertinents et répondant aux attentes

# En bref

L’utilisabilité se définit sous plein de critères différents et elle peut être améliorée sous plusieurs aspects. Les trois piliers de l’amélioration sont donc la documentation, les exemples et les études utilisateur.

# Des critères en vrac

Les critères d’utilisabilité sont :

* Avantage du langage
* Avantage de la terminologie
* Avantage de la métaphore
* Avantage des entrées
* Fonctionnalité
* Simplicité
* Consistance
* Minimum de mémoire chargée
* Navigabilité
* Minimum d’entrainement

Différents critères définissent une bonne API :

* Une API doit fournir suffisamment de fonctionnalités pour que l’appelant puisse exécuter sa tâche.
* Une API doit être minimale, elle ne doit imposer des inconvénients excessifs à l’appelant.
* Les APIs ne peuvent être conçues sans comprendre leur contexte.
* L’objectif des APIs générales devraient d’être sans règles, alors que les spécifiques devraient être dotées de beaucoup de règles.
* Les APIs devraient être conçues dans la perspective de l’appelant.
* Les bonnes APIs ne causent pas de problème
* Les APIs devraient être documentées avant d’être implémentées.
* Les bonnes APIs sont ergonomiques.