Conduite de projet : Documenter son projet

Mo Foughali
(IRIF, U-Paris) – foughali@irif.fr

17 décembre 2021

Plan

Plan

Documenter ?

On vous demande toujours de documenter vos projets. Mais de quoi parle-t-on ?

- Qu'est-ce qu'une documentation ?
- Que faut-il documenter (de quoi parle-t-elle ?) ?
- Qui en est le destinataire ?
  (En tout cas, ce n'est pas nécessairement la personne qui vous a donné le projet…)

Mais qu'obtient-on ?

- Souvent on vous demande au début de rédiger des documents dans les phases initiales d'un projet (cahier des charges, spécification fonctionnelle, spécification détaillée ...)
  Si vous l'avez fait, sont-ils à jour ?
  (Est-ce que les joindre à votre rendu a une valeur ajoutée ?)
- Inversement, si vous rédigez quelques documents à la hâte la veille de la deadline.
  La qualité sera-t-elle là ?

L'envers du décors

- Quand c'est vous l'utilisateur et/ou l'installateur, quelle documentation vous est-elle utile ?
  (Et puis, lisez-vous vraiment sérieusement la documentation ?)
- Question adressée à ceux qui sont arrivés en cours de semestre : qu'est-ce qui vous a le plus aidé à vous intégrer, la documentation ou bien le code ?

De l'utilité et de la destination de la documentation

Certaines réponses peuvent faire penser que la documentation est inutile.

→ Une documentation utile est une documentation rédigée en pensant à qui va la lire ! (ça vous rappelle quelque chose ?)

Documenter

De quoi on parle

Que comporte la documentation d'un projet ?

Documenter

De quoi on parle

Que comporte la documentation d'un projet ?

- la Big picture (vision d'ensemble) : descriptif très court du logiciel, destiné à fixer son périmètre en une phrase. (Question : c'est destiné à qui ?)
- documentation utilisateur
- documentation de suivi de projet
- documentation technique

Documentation utilisateur

Celle-ci peut prendre des formes diverses. Pas forcément un manuel !

Regardez les logiciels autour de vous, est-ce que la plupart du temps vous avez vraiment besoin d'ouvrir un document séparé pour avoir de l'aide ?

(Question : quelles formes de documentation utilisateur rencontrez-vous souvent ?)

Documentation de suivi de projet

(Ah, c'est destiné à qui, au fait ?)

Elle est généralement produite pendant le déroulement du projet.

Dans notre cas : le contenu de votre site GitLab fournit cette documentation si vous avez effectivement utilisé correctement la plateforme tout au long du projet.

Documentation technique

Même question, qui est-ce que ça intéresse ?

À quoi peut-elle leur servir ?

Documentation technique

Même question, qui est-ce que ça intéresse ?

À quoi peut-elle leur servir ?

→ c'est pour les développeurs et uniquement eux, qu'ils viennent du projet ou d'ailleurs.

→ cette documentation doit fournir le plus rapidement possible des réponses claires, précises et fiables afin de les aider à ajouter plus vite de nouvelles fonctionnalités.

On peut généralement distinguer :

- la documentation d'architecture
- la documentation exhaustive des API

La première doit généralement être écrite « à la main » et s'illustre de divers diagrammes et schémas.

La seconde est idéalement générée depuis le code source annoté.

Plan

Documenter son projet

Quoi ? Pour qui ? Comment ?

La documentation des API

Documentation des API

API : application programming interface C'est l'« interface »[1] de votre code, la partie avec laquelle on interagit pour utiliser votre code dans un autre projet.

Quelles qualités pour une bonne documentation des API ?

- elle doit être synchrone (tout le temps à jour)
  → elle doit être écrite en même temps que le code
- lisible depuis le code et depuis l'extérieur (multidiffusion)
  → elle doit être écrite dans le code (commentaires)... sous une forme qui peut être traitée automatiquement pour générer un document séparé

1. entendre « partie émergée de l'iceberg »

Javadoc

En Java, le système de documentation des API s'appelle Javadoc.

Il consiste en :

- une syntaxe adéquate de commentaire pour générer la documentation
- une commande « `javadoc` » qui génère la documentation sous forme de pages HTML depuis les commentaires au format Javadoc.

Javadoc – Syntaxe

Un exemple de commentaire Javadoc

```
1  /**
2    * Returns an expression equivalent to current expression, in which
3    * every occurrence of unknown symbol was substituted by the
4    * expression specified by parameter by.
5    *
6    * @param symbol symbol that should be substituted in this expression
7    * @param by     expression by which the symbol should be substituted
8    * @return       the transformed expression
9    */
10 Expression subst(UnknownExpr symbol, Expression by);
```

Javadoc – Syntaxe

Plus précisément

Tout commentaire de Javadoc est de cette forme :

```
1       /**
2         * Un texte descriptif.
3         * Celui-ci peut contenir <b>des tags HTML</b>.
4         * On y fait une présentation générale,
5         * on y exprime les contrats,
6         * mais on peut aussi ajouter tout élément technique
7         * nécessaire à la compréhension du bon fonctionnement.
8         *
9         * @tag1 un premier tag javadoc dont voici le descriptif.
10        * @tag2 bla bla bla
11        * @tag4 à noter qu'on peut aussi insérer des @tagenlignes
12        * @tag3 etc.
13        */
```

Javadoc – Syntaxe

Plus précisément

Tags les plus courants :

- `@author` : auteur de l'élément
- `@param` (suivi du nom du paramètre) : description d'un paramètre de méthode
- `@return` : description de la valeur de retour pour une méthode
- `@throws` (suivi du nom de l'exception) : description d'une exception susceptible d'être déclenchée par la méthode
- `@see` : mettre en relation avec la doc d'un autre élément
- `@deprecated` : signale que l'élément est obsolète (on doit expliquer par quoi le remplacer)

Javadoc

intégration dans les IDE

Exemple dans l'IDE Eclipse :

- Pour générer la documentation HTML : menu « Project », « Generate Javadoc... ».
- Quand on est sur une définition de classe/interface/enum/méthode/... , la combinaison de touches `Alt+Shift+J` créée un « squelette » de Javadoc à compléter.

Par ailleurs, le survol à la souris de tout identificateur affiche au bout d'environ une seconde une infobulle avec le contenu de la Javadoc associée à cet identificateur.

(Cela est vrai dans Eclipse, mais aussi dans IntellIJ IDEA ou Visual Studio Code et probablement d'autres...)

Javadoc

Quelles entités faut-il documenter ?

- Il est **indispensable** de documenter tout ce qui est public.
- Il est **fortement recommandé** de documenter tout ce qui n'est pas privé (car utilisable par d'autres programmeurs, qui n'ont pas accès au code source).
- Il est **utile** de documenter ce qui est privé, pour soi-même et les autres membres de l'équipe.

On peut documenter les types (classes, interfaces, enums, ...), les méthodes, les attributs... mais aussi les packages (créer un fichier `packageinfo.java` dans le répertoire du package, et y insérer le commentaire Javadoc.)