



Quid de la documentation?

Ressource R3.04

Qualité de développement

Informatique > Développement > Qualité de développement

Descriptif détaillé

Objectif

L'objectif de cette ressource est d'approfondir la qualité de développement.

Savoirs de référence étudiés

- ✓ Approfondissement des concepts de développement orienté objet (par ex. : SOLID, notions de dépendance et de couplage...)
- ✓ Compréhension et mise en œuvre de patrons de conception, éléments d'architecture logicielle, restructuration de code (refactoring)
- ➔ Rédaction de la documentation du code (par ex. : javadoc, doxygen...)
- ✓ Structuration de l'application (modularité)
- ✓ Gestion de version avancée (par ex. : branches...)
- ✓ Les différents savoirs de référence pourront être approfondis

Patrons de conception

Gestion de version

Documentation

Quid de la documentation du code ?

✓ Les patterns ne serait-il pas de la documentation ?

Lire et comprendre la documentation pour l'appliquer à un contexte particulier

✓ La javadoc pour documenter le code de l'API

Aider d'autres développeurs à réutiliser votre code

2.4. Javadoc at Method Level

Methods can contain a variety of Javadoc block tags.

Let's take a look at a method we're using:

```
/**
 * <p>This is a simple description of the method. . .
 * <a href="http://www.supermanisthegreatest.com">Superman!</a>
 * </p>
 * @param incomingDamage the amount of incoming damage
 * @return the amount of health hero has after attack
 * @see <a href="http://www.link_to_jira/HERO-402">HERO-402</a>
 * @since 1.0
 */
public int successfullyAttacked(int incomingDamage) {
    // do things
    return 0;
}
```

Pour approfondir, rendez-vous sur
<https://www.baeldung.com/javadoc>

com.baeldung.javadoc

Class SuperHero

java.lang.Object

com.baeldung.javadoc.Person

com.baeldung.javadoc.SuperHero

```
public class SuperHero
extends Person
```

Hero is the main entity we will be using to ...

Author:

Captain America

Field Summary

Fields

Modifier and Type	Field and Description
private int	defense
private int	health
private java.lang.String	heroName

En réalité... 4 types de documentations différentes avec des objectifs différents ...

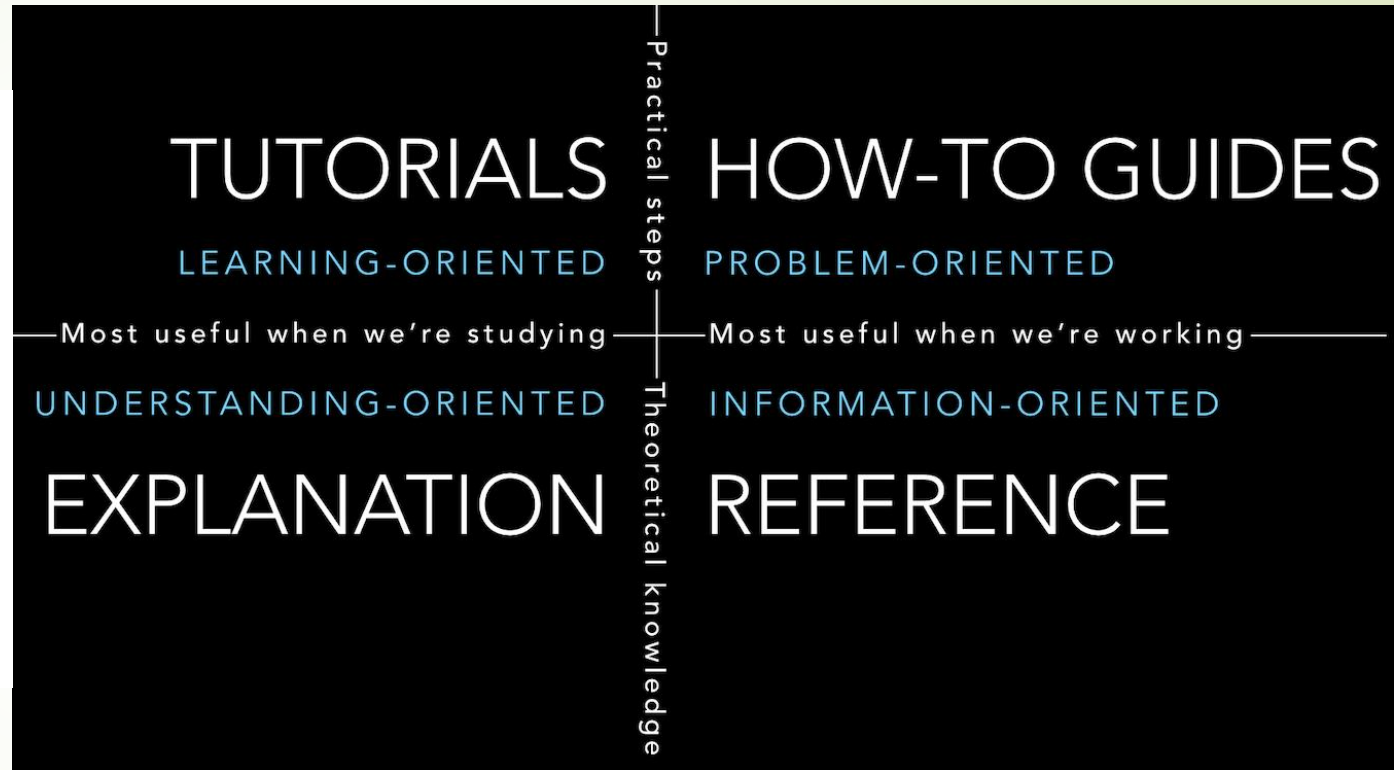
The documentation system

The Grand Unified Theory of Documentation

- David Laing

There is a secret that needs to be understood in order to write good software documentation: there isn't one thing called *documentation*, there are four.

They are: *tutorials*, *how-to guides*, *technical reference* and *explanation*. They represent four different purposes or functions, and require four different approaches to their creation. Understanding the implications of this will help improve most documentation - often immensely.



Extraits de <https://documentation.divio.com/> où vous trouverez des **bonnes pratiques** autour de la documentation pour chacune des 4 types de documentations

⇒ Ainsi qu'une vidéo : **What nobody tells you about documentation (Daniele Procida - Pycon 2017)**

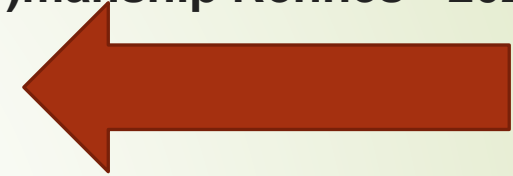
<https://www.youtube.com/watch?v=t4vKPhjcMZg>

Plus de liens à explorer autour de la documentation ...

- ❑ Living Documentation (Sébastien Fauvel– Software Crafts(wo)manship Rennes - 2020)

<https://www.youtube.com/watch?v=LG4SADs2nf8>

⇒ <https://fr.slideshare.net/SbastienFauvel/livingdocumentation>



- ❑ Documentation as Code (expliqué à mon père) (Hubert Sablonière – DevFest Nantes 2017)

⇒ AsciiDoc et AsciiDoctor (et son écosystème)

<https://www.youtube.com/watch?v=T6YJlaY0Dpw>

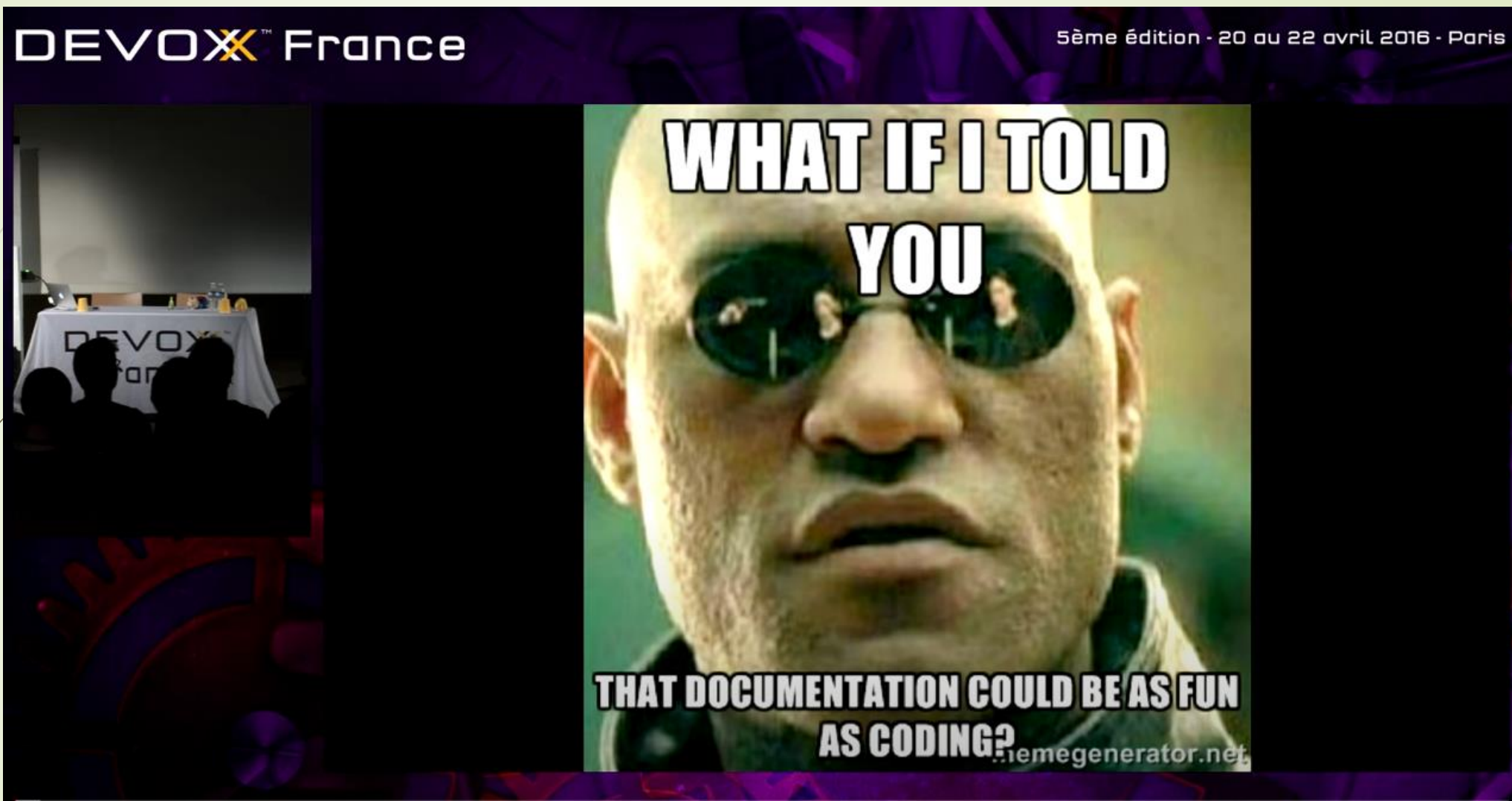
- ❑ Boîte à outils – living Documentation (Benoît Prioux - Bdxio 2016)

<https://binout.github.io/bdxio-2016/livingdoc-toolbox.html>

- ❑ Leaving Documentation, or Living Documentation (Cyrille Martaire – Socrates 2021)

<https://www.youtube.com/watch?v=T6YJlaY0Dpw>

Living documentation



Living Documentation : vous allez aimer la documentation ! (Cyrille Martraire)

<https://www.youtube.com/watch?v=Tw-wcps7WqU>

Isabelle BLASQUEZ