



ドキュメントの勘所  
～後続のエンジニアに向けたドキュメント編～

2023年9月13日

一般社団法人BOSS-CON JAPAN  
PHP技術者認定機構  
エバンジェリスト 三雲 勇二

# アジェンダ

- 自己紹介
- 【本編】ドキュメントの勘所  
～後続のエンジニアに向けたドキュメント編
  1. 共通認識を持つ
  2. 文章を組み立てる
  3. 自動化を進める
- まとめ



## 自己紹介

---

# 自己紹介

## ■ 三雲 勇二 (みくも ゆうじ)



## ■ 所属：

- プライム・ストラテジー株式会社  
WordPress などCMS高速化技術  
KUSANAGI の会社です 😊
- PHP技術者認定協会 エバンジェリスト

## ■ PHP 8 上級 / PHP 5 上級 / 徳丸実践 / KUSANAGI for WordPress ほか





# ドキュメントの勘所～後続エンジニアに向けて

# ドキュメントの勘所～後続エンジニアに向けて

今回の「後続エンジニア」は、「新入社員」をイメージして下記のブロックに分けてお話します。

1. 共通認識を持つ
2. 文章を組み立てる
3. 自動化を進める



# ドキュメントの勘所～後続エンジニアに向けて

## 1. 共通認識を持つ

# 共通認識を持つ

このブロックでは、次の項目に分けてお話しします。

1. 前提の提示
2. 用語の共通認識を確認
3. 社内の共通認識を整備

## 1. 前提の提示

# 1. 前提の提示

先程、

今回の「後続エンジニア」は、「**新入社員**」をイメージして下記のブロックに分けてお話します。

と、始めました。

では「**新入社員**」とは、どんなイメージとなるでしょうか？

# 1. 前提の提示

経験はどこまでのことでしょうか？

- 新社会人: エンジニア方面の学習をしていた？していなかった？
- 中途採用: エンジニア経験あり？なし？

職種はどこまでを対象にしましょう？

- エンジニア職
- バックオフィス職

# 1. 前提の提示

つまり、今回は

今回の「後続エンジニア」は、~~「新入社員」~~  
「エンジニア職の新入社員」をイメージして下記の  
ブロックに分けてお話しします。

と、書いてあると、読む対象者が明確になります。

# 1. 前提の提示

エンジニア職と前提を提示してあれば、基本的なエンジニアの知識を持っていることが求められます。

このときに、「PHP8初級」程度の知識を持っていることとするなど、具体的な指標がイメージできているとなお良いでしょう。

# 1. 前提の提示

書き手は、「どのような人が読むことを想定しているドキュメントなのか？」ということ、前提を提示することが望ましいです。

明示的に前提を記述しなくても、ドキュメントを書くときには、「前提」を意識するとよいでしょう。

## 2. 用語の共通認識を確認

## 2. 用語の共通認識を確認

次の用語の意味はなんでしょう？

1. デフォルト
2. PHP



## 2. 用語の共通認識を確認

次の用語の意味はなんでしょう？

1. デフォルト

2. PHP

[エンジニアの回答]

1. 標準の設定値
2. PHP 言語



[金融機関関係エンジニアの回答]

1. 債務不履行
2. フィリピン・ペソ



## 2. 用語の共通認識を確認

エンジニアとして一般的ではない用語を使う場合、用語の共通認識を合わせておくことも重要です。

一般的でない使い方の場合はドキュメント内で一度、正式名称を書いておくことが良いでしょう。

例えば、PHP が PHP 言語の意味でない場合、共通認識を揃えるため「PHP(フィリピン・ペソ)」のように書いておきます。

## 3. 社内の共通認識を確認

### 3. 社内の共通認識を確認

ある社内ドキュメントの例です。

「BE もしくは PE が適用されているか確認するには、`dnf list --installed` でパッケージ一覧を確認してみてください。」

### 3. 社内の共通認識を確認

BE や PE というのは、実は社内で使われている略語の話でした。

「BE もしくは PE が適用されているか確認するには、`dnf list --installed` でパッケージ一覧を確認してみてください。」

これを新人が把握するのは大変です。

### 3. 社内の共通認識を確認

そこで「社内用語集」を作成してみましょう！

用語	読み方	解説
BE	ビーイー	KUSANAGI の Business Edition
PE	ピーイー	KUSANAGI の Premium Edition
WEXAL	ウェクサル	WEXAL Page Speed Technology (PST) のことで、KUSANAGI Premium Edition の機能

※私の勤務先(プライム・ストラテジー)の社内用語集を一部抜粋。

### 3. 社内の共通認識を確認

社内用語集を作るときのポイントです。

- 「新入社員から質問されたタイミングで更新する」くらいの気持ちで、まずは作成してみましよう。
- 「読み方」は、口頭で会話するときに重要です。
- 「技術用語」や「社内業務用語」など表を分けてあるとなお良いです。



# ドキュメントの勘所～後続エンジニアに向けて

---

## 2. 文章を組み立てる

# 文章を組み立てる

このブロックでは、次の項目に分けてお話します。

1. ドキュメントの目的
2. 情報を書く優先順位
3. 「事実」と「予想」
4. 5W1H / 6W2H

# 共通認識を持つ

## 1.ドキュメントの目的



# 1. ドキュメントの目的

いちばん大事なことです。

ドキュメントを作成する「目的」を明確にしましょう。

- 仕様書 → 成果物をその通りに作って欲しい
- コーディング規約 → コードの書き方を統一したい
- 報告書 → 問題から原因を見つけ、対応方法を確認したい

# 1. ドキュメントの目的

目的を明確化することとは、

- 読み手に伝えたいこと
- 読み手に行動してほしいこと

これらを明確化することでもあります。

## 2. 情報を書く優先順位

## 2. 情報を書く優先順位

ドキュメントの目的をはっきりさせると、読み手に読ませたい優先順位、つまりは書くべき優先順位が決まります。

例えば、サーバーで障害が発生し、対応した報告書を作成するとします。

## 2. 情報を書く優先順位

[時系列でそのまま報告]

問題 → 予想 → 調査内容 → 原因 → 再発防止策 → **結論**

これだと、最後まで読まないで結論がわかりません。

[読み手に重要な情報の順序に入れ替えて報告]

問題 → **結論** → 原因 → 再発防止策 → 調査内容 → 予想

報告書では読み手はまず結論が知りたいものです。

必要に応じて、重要な情報を読み進めていただければよいはず  
です。

## 2. 情報を書く優先順位

すべて読まなくても伝わるのであれば、それは良いドキュメントと言えます。

そのような結論に至ったのかなど、優先度が低い情報は後ろに回しても良いでしょう。

## 3. 「事実」と「予想」

### 3. 「事実」と「予想」

「事実」と「予想」は分けて書きます。  
事実には証拠を明示します。

例【事実】

アイスの売上が良かったのは、気温が30度超えていたため。  
過去のアイスの売上と気温の相関関係は、別資料に添付。

【予想】

湿度が高いこともアイスの売上に影響しているかもしれない。

4. 5W1H / 6W2H

## 4. 5W1H / 6W2H

事実として情報の精度を上げる場合は、「5W1H」や「6W2H」の観点を付け加えると良いでしょう。

- When (いつ)
- Where (どこで)
- Who (だれが)
- Whom (だれに)
- What (なにを)
- Why (なぜ)
- How (どうやって)
- How much (いくらで)



# ドキュメントの勘所～後続エンジニアに向けて

## 3. 自動化を進める

### 3. 自動化を進める

文書はメンテナンスが必要です。メンテナンスはできる限り自動化しておくことが好ましいので、そのヒントとなる内容を軽くご紹介したいと思います

この後の古庄親方の発表に詳しく出てくる項目となるとおもいます 😊

## 3. 自動化を進める

### 1. 文書の書き方テンプレートを作る

FAQ 用のテンプレートや、報告書用のテンプレートなどを前もって用意しておくの良いです。

読む側の負担も減ります。

### 3. 自動化を進める

#### 2. コードからドキュメントを自動でつくる

docstring からコード仕様書を作成することができます。

(Doxygenなど)

## 3. 自動化を進める

### 3. 変更履歴を Git などで管理する

ソースコードでなくてもバージョン管理システムは使えるので、文書のバージョン管理しておくとも元に戻すことも簡単です。

### 3. 自動化を進める

4. 文書にも「コーディング規約」のようなものを整備する

表記ゆれしそうな用語は、予め社内で決めておいて一括で検知すると良いです。

例) nginx/Nginx → nginx

### 3. 自動化を進める

#### 5. 情報は1か所にまとめる

ドキュメントは読まれてこそです。探しやすいように1か所にまとめておきましょう。

昨今はAIで収集することもできるので、そちらを検討することもあります。

## 3. 自動化を進める

### 6. 文章の校正ツールを使う

辞書ツールや校正ツールを使いましょう。

例えば Word にコピーして、日本語の表記ゆれや英語のスペルチェックをするだけでも、文章の品質がぐっと上がります。



# ドキュメントの勘所～後続エンジニアに向けて まとめ

# まとめ

特にお伝えしたい内容

1. まずは社内用語集を作ってみましょう
2. ドキュメントは読み手にどのような行動を促したいのか？という目的を決めましょう
3. 校正はツールを使って自動化しましょう



---

PHP技術者認定機構