

ドキュメント勘所  
～お客様に向けたドキュメント編

by 古庄道明(がる / gallu)

# 自己紹介

---



- ▶ 古庄道明と申します
  - ▶ 「がる(gallu)」というハンドルでふらついております
  - ▶ 本職は技術者です。現役プログラマーやっています
    - ▶ バックエンド系なので、インフラとかDBとか運用とかも一通り
    - ▶ 最近はPM業も多いですねえ
    - ▶ 「ヒアリングして技術提案」なんてこともさせていただいています
  - ▶ 教育も色々と携わっております(専門学校講師とか)
  - ▶ 「PHP8技術者認定上級試験」「PHP8技術者認定初級試験」の作成を担当させていただいています
  - ▶ コラムとかblogとかeラーニングの教材とかで技術系の文章を書くこともちらほら
- 



# 今日のお題

---

- ▶ 「お客様に向けたドキュメント」の作成について学んでいきましょう
  - ▶ 基本的には「エンジニアさんに向けた」内容を想定しています
  - ▶ なので、お話がわりと「エンジニア目線」になります



# そのドキュメントは「何のために」？

---

- ▶ そもそも、ドキュメントは「必要」なのでしょうか？  
もし「必要」だとしたら、それは「何のために」必要なの  
でしょうか？
  - ▶ 納品物の一環
  - ▶ 必要な情報を共有する手段



# 「納品物の一環」としてのドキュメント

---

- ▶ 「契約書」に「納品物の1つとして」ドキュメントが指定されていること、があります
    - ▶ 頻度は「会社さん(と契約形態)にも拠る」のですが、請負の開発だと割と「指定されている事が多い」ように思います
  - ▶ 細かいところまで規定がある、なら「それに従いましょう」
  - ▶ 細かいところの規定がない時は「一般的なフォーマット」を踏襲すると、お互いに苦労が減る事が多いです
    - ▶ この辺は「開発ドキュメント」と、それに加えて適切な単語を加えてググってみるのが早いです
      - ▶ 「IPA」とか「標準」とか「経産省」とか
      - ▶ ただ割と種類が「こってり」している事も多いので、「その中からある程度必要なものを抜き出す」といいでしょう
- 



# ポピュラーな「納品物」としての一例

---

- ▶ RFP(提案依頼書)
- ▶ 要求/要件定義
  - ▶ 要求定義は「やりたいこと」、要件定義は「どう作るか」
- ▶ 技術的な諸々(「基本設計」「詳細設計」の枠組みが多い)
  - ▶ インフラ周り(インフラ設計書、ネットワーク構成図など)
  - ▶ DB周り(ER図、テーブル定義書など)
  - ▶ 設計周り(クラス図、シーケンス図など)
  - ▶ 画面周り(UI設計書など)
  - ▶ テスト周り(テスト仕様書、テスト報告書など)
  - ▶ 運用周り(運用マニュアル、操作マニュアルなど)



- 
- ▶ 各ドキュメントには「タイトル」「目次」と「版(バージョン)情報」を入れておきましょう
    - ▶ 目次は特に「そこを中心に読む(そこしか読まない)」人もいる可能性を考慮しておくといいでしょ
  - ▶ 「納品物の一環」としてのドキュメントは、高確率で「エンジニアではない人も読む」ので、技術以外の所にも気を配っておくといいでしょ
    - ▶ 先方にもよりますが、ある程度の技術用語には「説明」を入れておくのも一案です
    - ▶ n文字略語は特に「正式名称と説明」を入れておいたほうが齟齬が減ります
- 
- ▶

# 気にしておきたいポイント

---

## 特に「エンジニアが割と失念しやすい」ポイント

- ▶ 誤字脱字に気をつける
  - ▶ プログラムの「SyntaxError」は気になりますよね???
- ▶ 語尾や文体を統一する
  - ▶ 「だ、である調」と「ですます調」が混ざると違和感があるので、どちらかにそろえるようにしましょう
- ▶ 用語を統一する
  - ▶ システム用語は大体統一出来ていますが、ビジネス用語の統一が抜けやすいので、気をつけましょう
- ▶ 明文にする(後述)





# 明文とかテクニカルライティングとか

---

- ▶ 情報伝達物である以上「誤解なく齟齬なく伝わる」事が第一です
- ▶ 全体として「MECEであること」も大切ですが、その1文が「誤解を生み得る文章」だと、全体が台無しになります
  - ▶ 「漏れなく」は必須ですが、「ダブリなく」は最悪、少し優先度を下げてもよいかと思います
- ▶ そのために、まずは「正確」であり、できるだけ「わかりやすく」「読みやすい」とよいでしょう
- ▶ そういった書き方を「テクニカルライティング」や「明文」といった言い方をします



## 明文「ではない」例

---

- ▶ この入力項目は100文字以下で入力してください。100文字以上はエラーになります
  - ▶ 「ちょうど100文字」がOKなのかエラーなのか分からない
  - ▶ 100文字がOKなら「100文字を超えるならエラー」、100文字がNGなら「100文字未満で入力」となる
- ▶ ハードウェアの性能によっては、納入ソフトの全ての機能を使用できない場合があります
  - ▶ 「全部、全く使えない」のか「(一部は使えるが)全部は使えない」のかが分からない
  - ▶ 全部NGなら「性能が不足している場合、納入ソフトの機能は一切使用できなくなります」、一部NGなら「納入ソフトの一部の機能を使用できない場合があります」等



- 
- ▶ ソフトウェアの動作には十分なマシン性能が必要です
    - ▶ 「十分」が不明なので、数値等で示しましょう
  - ▶ 純正製品をお使いにならないと、データの保存が出来ない場合があります
    - ▶ 二重否定は誤解を生みやすいので避けましょう
    - ▶ 「純正品をお使いください。他社製品をお使いになった場合、データの保存が出来ない場合があります」等
  - ▶ ここではきものをぬいでください
    - ▶ 割と有名な文ですね。「漢字を使う」「句読点を適切に使う」
    - ▶ 「ここで、履き物を脱いでください」「ここでは、着物を脱いでください」
- 



# それ以外に気をつけたいことの例

---

## ▶ 短い文章を書く

- ▶ 1文あたり50文字以内を目標にする

## ▶ 1文に複数の情報を詰め込まない

## ▶ 1文で同じ言葉を重複させない


- ▶ hogeコマンドは、運用の時に使うコマンドです

- ▶ 「hogeコマンドは、運用の時に使います」

## ▶ 言葉の順番に気をつける

- ▶ 「白いかごの中の小鳥」だと、「かごが白い」のか「小鳥が白い」のかわからない。「かごの中の、白い小鳥」なら小鳥が白くなる(かごが白いなら「白いかご、の小鳥」「白塗りのかごの中の小鳥」等、工夫が必要になる)



- 
- ▶ 「正確さ」のためには、数値なども使いながら、実際に確認をしていくとよいでしょう
  - ▶ 「読みやすさ」「わかりやすさ」については、「テクニカルライティング」「明文」といったワードで検索をした書籍などを読んでみるとよいでしょう
  - ▶ 「他人に読んでもらう」「一晩間を空けて、翌日に読み返す」なども有効な事が多いです
- 
- 

# 「必要な情報を共有する手段」としてのドキュメント

---

- ▶ 技術情報などは実際に「必要な情報を伝える」ことが重要になります
  - ▶ その辺りは前の三雲さんのセッションで語られているかと思うので省略します(笑)
  - ▶ 「コードやテーブルレイアウトなどを読んでもわからない」事などは特に重要です
- ▶ 一方「非技術者に見てもらう」技術的なドキュメントとしてよくある1つが「操作マニュアル」になるかと思えます
- ▶ これも「自動化」出来るツールが最近増えてきているので、積極的に使うとよいでしょう
  - ▶ 画面デザインがちょっと変わる毎に「大事になる」のは避けたいものです



改めて、そのドキュメントは「何のために」？

---

- ▶ そもそも、ドキュメントは「必要」なのでしょうか？
- ▶ 特に最近「アジャイル開発(スクラム)」が流行ってます
- ▶ アジャイル開発では「ドキュメントは不要」なのでは？

**違います**

---



# 「アジャイルソフトウェア開発宣言」

---

プロセスやツールよりも個人と対話を、  
包括的なドキュメントよりも動くソフトウェアを、  
契約交渉よりも顧客との協調を、  
計画に従うことよりも変化への対応を、

- ▶ ほら。「ドキュメント不要」って書いてある  
……訳ではないのです。





---

価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。

- ▶ とあります。つまり「左記のことがらに価値がある」事を認めた上で「右記により価値をおく」だけなんです
- ▶ 過去に「ドキュメントはしっかりとしているんだけど、実際のソフトウェアが全く動かない」なんて事が、あったとか無かったとかあったとか以下略
- ▶ でも「全くドキュメントがない、動くソフトウェアだけ」だと、後の保守がムリゲーになってしまうんです!!!



- 
- ▶ エンジニアが「面倒に思う」気持ちが分からないわけでは  
ありません
  - ▶ でも一方で「ビジネス側」や「引き継いだエンジニア側」に  
とって、(ちゃんとした)ドキュメントは割と「生命線」です
  - ▶ なので「自動化」などしつつ、「必要なものをちゃんと伝達  
する」必要があります
    - ▶ 「更新のたびにドキュメントを再提出するかどうか」は契約次  
第ですが、「再提出出来る」のは強みと言えます
  - ▶ そんなドキュメントを「ちゃんと準備できる」と、エンジニア  
にとって「**+1の武器**」になり得ます!!
    - ▶ 最後に主題が出てきました!!
- 



# まとめ

---

- ▶ 「動くソフトウェアを作る」のは労力がかかる、とても大変で大切なお仕事です
  - ▶ でも、ドキュメントの不足のために「画竜点睛を欠く」「竜頭蛇尾に終わる」状況は、とてももったいないものです
  - ▶ であれば、ドキュメントも適切に(できればお手軽に)出す事が出来るようになると「有終の美が飾れる」のではないのでしょうか?
  - ▶ ドキュメントに対する気持ちとか苦手意識とかが少しでも軽減れば幸いです
- 

