

研究タイトル

プロジェクトマネジメントコース 矢吹研究室 1342097 浜野太豪

1. 序論

ドキュメントの検査にプログラムやツールによるサポートが必要である。なぜなら、システム開発の現場では、様々なドキュメントを作成する必要があるからだ。例えばプロジェクト計画書や要件定義書、マニュアルなどがある。このような文書では、読み手に誤解を与えてはいけない。さらに、わかりやすい文書を書くには一定のルールを守る必要がある。短い文で書くこと、正しい表現方法で書くこと、フォーマットを統一することなどである。このようなルールで、大量のドキュメントを人の目によってチェックすることには限界がある。

そこで継続的インテグレーションを活用する方法がある。継続的インテグレーションとは、プログラム全体を常に統合して動く状態にしておくことである。最近では、ビルド（プログラムのコンパイルや自動テスト、アーカイブ化、ソースコードへのタグ付け、実行環境へのデプロイの一連の手順）を自動化するツールが数多くある [1]。

このような継続的インテグレーションで活用されている自動化ツールを用いて、大量のドキュメントをチェックできるツールを構築する。

2. 目的

文書チェックを自動的に行うシステムを構築する。研究室ではドキュメントの変更履歴をバージョン管理システム GitHub を用いて管理している。GitHub に文書を提出した際に、自動で文書チェックプログラムが実行され、実行結果の表示、通知を行う。

3. 手法

構築の手法について以下に示す。

1. 文書チェックプログラムの設定を行う。
2. 自動化ツールの設定を行う。
3. 文章チェックプログラムと自動化ツールと GitHub の連携を行う。

4. 結果

GitHub に課題研究の概要を提出した場合、文章チェックプログラムによって自動的に文章チェックが行えるようになった。課題研究の概要を文書チェックにかけてみたところ以下の指摘が表示された。

1. SentenceLength - 文長を検査する。
2. InvalidExpression - 不正な表現が利用されていないか検査する。
3. KatakanaSpellCheck - カタカナ単語のスペルチェックを行う。
4. JapaneseAmbiguousNounConjunction - 助詞の「の」が連続しているか検査する。

5. 考察

文書チェックプログラムは設定ファイルを変更することで、出力結果も変わる。そのため設定を変更し、添削に有効な設定を検討する必要がある。例えば添削前の文書を文書チェックプログラムにかけ、どの範囲まで文書の誤りをチェックすることができののか調べることで、文書チェックプログラムが添削できる範囲を調べることができると考えた。

6. 結論

自動的に文書チェックを行う環境を構築することができた。文書チェックプログラムの設定に関しては詳しく調べることができなかったため、検証が必要である。

参考文献

- [1] 平鍋健児, 野中郁次郎. アジャイル開発とスクラム. 翔泳社, 2013.