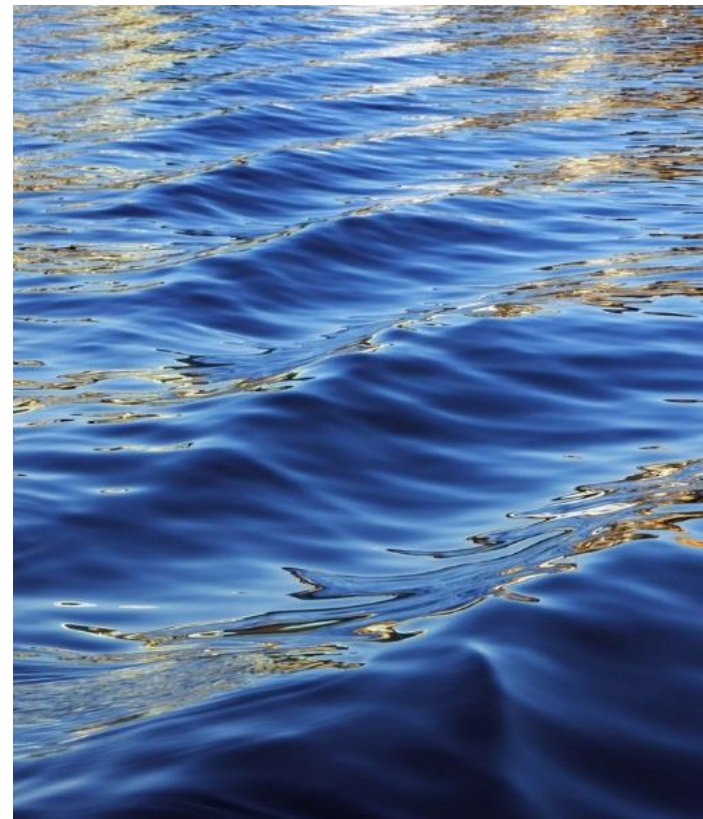




プログラミング講座

番外編：PDCA と OODA (1)



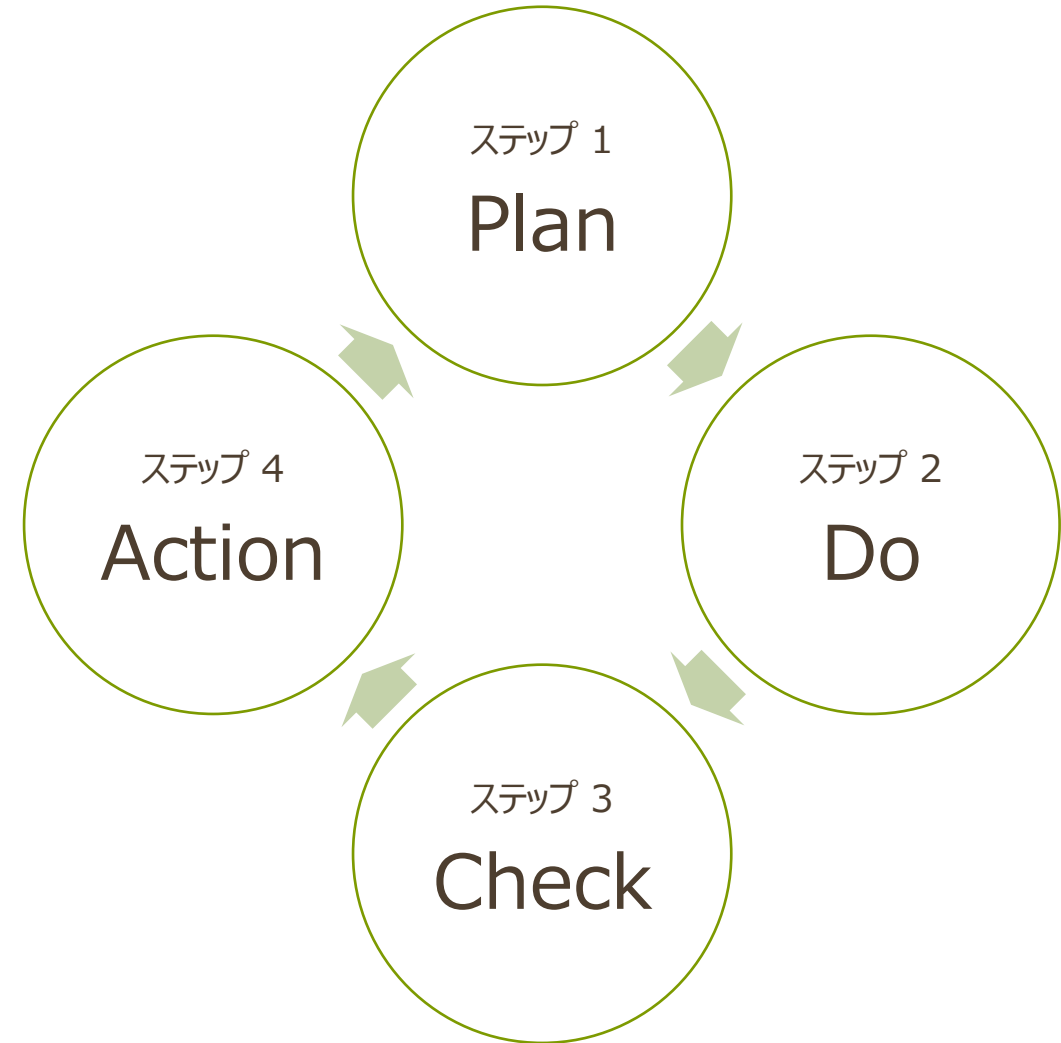
PDCA と OODA を理解し業務効率の改善を図ろう！

- PDCA と OODA を覚えましょう
- プログラミング講座 番外編(1) PDCA とは
- プログラミング講座 番外編(2) OODA とは

OFTチームのPDCA

- Plan : 計画
- Do : 実行
- Check : 評価(検査・テスト)
- Action : 対策・改善

OFTプロジェクトにおける
PDCA
を考えてみます。



Plan : 計画

指摘(指示)内容の確認 / 納期(スケジュール) / 工数(見積)

- 指摘(指示)内容の確認
 - OFT指摘リスト の内容を確認する。
 - ※指摘に記載されている単語の意味を理解出来ているか？
 - ※自衛官目線で内容の確認が出来ているか？
 - ※正しい仕様を理解できているか？
 - ※FCPMプロジェクトのどのクラスに関連しているか？

Plan : 計画

指摘(指示)内容の確認 / 納期(スケジュール) / 工数(見積)

- 納期(スケジュール)
 - 検査・リリースまで含めた納期(スケジュール)で考える
 - ※リリース後、現地確認まで実施(依頼)まで考えているか？ ->依頼した方にもスケジュールがあります。
 - ※リリースしたモジュールで品質保証までとして納期を守れることができるか？
 - ※単体テスト(Debug) / 結合テスト(Release) まで納期に入っているか？

Plan : 計画

指摘(指示)内容の確認 / 納期(スケジュール) / 工数(見積)

- 工数(見積)

- 確認した内容の全容が見えているか？

- ※影響範囲(他の操作時など)を理解し、全操作に対して対応する工数を算出しているか？

- ※コードレビュー・検査工数は算出されているか？

- ※指摘された際の再対応の工数まで算出されているか？

- ※覚えよう！安く(工数を短く)算出しても誰も得はしません！

- 安く(早く)見積もると、回答している時に上長から褒められるでしょう！でも実は本当は誰も得はしません。むしろ間違えた算出をリカバリーする方が大変です。

- また時間だけで判断する上長でしたら、上長失格ですので、更に御上の方に密告しましょう！

精度の良い見積ほど、出来るエンジニアです。

Do : 実行

プログラミング詳細設計 / プログラミング / コードレビュー

- プログラミング詳細設計

- 明確なビジョンが見えているか？

- ※後輩は先輩からの指示が理解できているか？

- ※先輩は後輩に正しく指示が出来ているか？ また理解したことを確認できているか？

- ※(1)正しく理解できることを確認するため、コードレビューの目的欄の記載から始めたら？

- ※(2)忙しいからといって先輩がコードレビューを怠っていませんか？ 怠っている様をみかけたら更なる上長に密告しましょう！

Do : 実行

プログラミング詳細設計 / プログラミング / コードレビュー

- プログラミング

- 誰が見ても理解できるプログラミングになっているか？

- ※無駄な変数をつくっていませんか？

- ※ネストが深くなっていませんか？

- ※コード規約通りのプログラミング(コーディング)になっているか？

Do : 実行

プログラミング詳細設計 / プログラミング / コードレビュー

- コードレビュー
 - 目的 -> 対応までを記載
 - ※プログラミングの変更箇所の理由を説明できているか？
 - ※初期化からプログラミング構成まで説明できているか？
 - ※見やすさまで考慮できているか？

Check : 評価 (検査・テスト)

単体テスト (Debug) / 結合テスト (Release) / 品質保証 / 結合検査

- 単体テスト (Debug)
 - Visual Studio ステップ送りで全てのパターンの確認をする
 - ※条件分岐で全ての分岐の確認ができていますか？
 - ※考慮した条件分岐と同じ動作をしているか？
 - ※わかりやすいコメントになっているか？

Check : 評価 (検査・テスト)

単体テスト (Debug) / 結合テスト (Release) / 品質保証 / 結合検査

- 結合テスト (Release)
 - Releaseビルドで全てのパターンを確認できているか？
 - ※プログラミング変更による影響範囲を理解できているか？
 - ※当初の目的からずれたプログラミングになっていないか？
 - ※リファクタリングなどにより、他の影響を確認できているか？
 - ※全ての動作パターンをテスト内容に反映できているか？

Action : 対策・改善

テスト不具合対策 /コードレビュー改善

- テスト不具合対策
 - なぜテスト不具合が発生したか？
 - ※プログラミング (コーディング) ミスが発生したのか？
 - ※詳細設計の段階で問題が発生していたのならば、フィードバックはできているか？

Action : 対策・改善

テスト不具合対策 /コードレビュー改善

- コードレビュー改善
 - なぜ指摘されたか？
 - ※プログラミング構造として問題があったのか？
 - ※初期化など、C++の基本を理解できているか？
 - ※コーディング規約を把握できているか？

参加者 全員で発表しましょう！

今回の講義内容を聴いて、全員発表をしましょう

- PDCAについて

- PDCAを理解できたか？

- ※先輩は、業務が怠慢になっていないか？←今回の講義で反省点を述べましょう！
(対象：鈴木)

- ※後輩たちは、今、自信を持っている項目を発表しましょう！←自信をもって発表しましょう！
(対象：能見T, 吉野T, 川口T, 山口T)

先輩は日々の反省をすること。
後輩たちは自信を持って発表できることを増やすこと！

次回は OODAについて講義していきます。楽しみに！