0.0.1. プログラムのテストと統合

- ・デバッガ
- ・ブレークポイント
- ・ステップ実行

0.0.2. 周辺装置やネットワークの利用

- ・ハードウェアを使用したプログラム
 - ▶ バーコードリーダー・IC タグ・スキャナ・カメラ
- ・ネットワークを利用したプログラム
 - ▶ Web サーバー・クラウド
- データベースを利用したプログラム
 - ▶ DBMS · SQL 言語

<API の活用>

プログラム同士が機能や管理をするデータなどを相互にやり取りするために使われる「WebAPI」

- ・Web サーバー上の URL にアクセス -> 処理結果
- 事前の利用登録が必要なものもある(API キー)

0.1. プログラムの統合

0.1.1. 単体テスト

0.1.2. 結合テスト

- ・トップダウンテスト
 - ▶ 上位のモジュールから順に結合していく(スタブ要)
- ・ボトムアップテスト
 - ▶ 会のモジュールから順に結合していく(ドライバ要)
- ・サンドイッチテスト
 - トップダウンの両方から行う(時間短縮)
- ・インターフェーステスト
 - ▶ 個々のモジュールが正しく連携するかのテストをする
- ・シナリオテスト
 - システムで使われるシナリオをもとにテストをする
- 負荷テスト
 - ▶ 処理能力の有無をテストする
- ・総合テスト
 - ▶ 要求定義所、外部設計書の内容が実現できているかどうかを確認する

1. 情報システムの開発管理と運用・保守

1.1. 情報システムの開発工程の管理

1.1.1. 開発手法の種類と選択

大型汎用機→クライアントサーバーシステム、パソコン ダウンサイジング、高機能、開発 スピードが求められる

<開発手法>

プロトタイプ型、スパイラル型、アジャイル型、ウォーターフォール型

開発手法	メリット	デメリット
プロトタイプ型	仕事が明確でなくても開始 ができる、手戻りが起こり にくい	利用者の要望が増加しやす い、開発全体の納期や費用 を見積もりにくい
スパイラル型	仕様変更に柔軟に対応でき る、計画変更しやすい	開発初期に全体を把握しづ らい、開発の終息に時間が かかることがある
アジャイル型	仕様が明確でなくても開始 ができる、要望変更、市場 や環境の変化にも対応でき る	従来型の開発のほうが適し ている場合がある、利用者 の協力が得られない場合は 効果が出づらい
ウォーターフォール型	作業内容が明確であるため 計画が立てやすい、開発事 例が多く参考にしやすい	開発終盤まで実際に動くシ ステムを確認できない、不 具合対応や仕様変更の影響 が大きい

1.1.2. 開発工程の管理手法

プロジェクトチームでの開発※プロジェクトリーダーによる工程管理が必要

- 1. 開発工程名
- 2. 作業名
- 3. 担当者名
- 4. 作業予定期間
- 5. 作業実績期間

1.2. チームにおける開発手法

1.2.1. ソースコードレビュー

- ・作成担当者以外のメンバーがソースコードをレビューする
- ・プログラミング言語の知識 + ミスを生まない記述方法の知識

チェック内容 : コーディング規約に沿って書いてあるかどうか : 正しいロジックで書いてあるかどうか

1.2.2. コーディング規約

プログラミング言語ごとに作る(記述ルールが異なるため)

- ・プログラムの冒頭に着けるコメント
- ・変数名のつけ方や宣言方法
- ・プログラムロジックの記述
- ・変数の型 etc...

1.2.3. プログラム開発の履歴管理

チームでの開発 -> 生活部の一元管理、共有が必要

分散型バージョン管理システム

1.3. 情報システムの運用と保守

1.3.1. 運用と保守の違い

運用:通常業務、利用可能な状態を維持する

保守:業務内容や環境の変化に対応した機能追加、トラブルや機能低下発生時、修理や復旧などの障害対策

契約不適合担保責任

...開発者が契約通りの機能を提供できなかった場合、修正や再開発を行う責任。

1.3.2. 担当部門の役割

- ・業務手順書の作成と更新:ユーザーの操作手順書(マニュアル)ではなく、システムの操 作方法
- · 利用者教育
- ・システム監視:状態を監視し、以上を検知することでトラブルを未然に防ぐ
- ・定例報告と改善案の提案:内容はすべて記録し、ノウハウを蓄積(5W3H)

SLA

…サービスレベル契約(Service Level Agreement)の略。 サービス提供者と利用者との間で、サービスの品質や提供内容について合意した契約。

1.4. 情報システムのセキュリティ

1.4.1. 情報システムに対する脅威

・IPA による「情報セキュリティの 10 大脅威」

種類

- 1. 外部要因
- 2. 内部要因
- 3. 環境的脅威
- 4. 物理的脅威
- 5. 技術的脅威
- 6. 人為的脅威

順位

- 1. 標的型攻撃による機密情報の摂取
- 2. 内部不正による情報漏えい
- 3. ビジネスメール詐欺による金銭被害
- 4. サプライチェーンの弱点を悪用した攻撃
- 5. ランサムウェアによる被害
- 6. 予期せぬ IT 基盤の障害に伴う業務停止
- 7. 不注意による情報漏えい
- 8. インターネット上サービスからの個人情報の摂取
- 9. IoT 機器の不正利用
- 10. サービス妨害攻撃によるサービスの停止

1.4.2. 情報セキュリティ管理

- ・情報セキュリティの定義 機密性 完全性 可用性
- ・規格: JIS で制定
- ・管理体制の構築
 - 1. 統括責任者が基本方針を策定 -> 周知
 - 2. 担当する部署やメンバーを決める
 - 3. 担当部署が運用手順と緊急時の対策手順を定める
 - 4. 適宜改善する

1.4.3. セキュリティ対策とソフト

	外部要因	内部要因
環境的脅威	台風、洪水、噴火、火災、地震、 電磁波	火災、漏水
物理的脅威	電力障害、通信障害、破壊	経年劣化、故障
技術的脅威	マルウェア、ランサムウェア、サ イバー攻撃	バグ、セキュリティホール、脆弱 性
人為的脅威	ソーシャルエンジニアリング、盗 聴	操作ミス、データ紛失・破棄、情 報漏えい、データ横領

1.4.4. 技術者倫理とコンプライアンス