5.I.11. 筒居 稔

1 ソフトウェアの性質と開発の課題

1.1 ソフトウェアの特徴

1.1.1 ソフトウェアの構成

プログラム, 手法・技法, ドキュメントの三つから構成される.

1.1.2 ソフトウェア開発の特徴

- 実態がつかみにくい.
- 発工程に作業が集中する.
- 運用・保守期間が長い.
- 再利用が少ない.

1.1.3 ソフトウェアの概念

- 要求仕様の満足度.
- 高い操作性.
- 適切な開発コストと開発期間.
- わかりやすさと保守性.

1.2 ソフトウェアの分類

1.2.1 応用ソフトウェア

業種別ソフトウェア

金融業, 製造業, 流通業, サービス業.

業務別ソフトウェア

販売, 生産管理, 購買, 人事, 経理等.

共通応用ソフトウェア

表計算, 統計処理,CAD,Web 制作ツール等.

1.2.2 ミドルウェア

データベース管理, ネットワーク管理,GUI 等.

1.2.3 基本ソフトウェア

OS, 言語処理, サービスプログラム等.

1.3 ソフトウェアのライフサイクル

要求分析, 外部設計, 内部設計, プログラミングテスト, 保守・運用.

1.4 ソフトウェア開発現場における課題

- 要求仕様決定の困難性
- 再利用の困難性
- プロジェクトトラブルの発生可能性
- ソフトウェア開発規模・工数見積もりの誤り

2 ソフトウェア開発プロセス

要求分析	要求仕様書
外部設計	外部設計書 (外部仕様書)
内部設計	内部設計書 (内部仕様書)
プログラミング	ソースコード, プログラム仕様書
テスト	テスト仕様書, 成績書
保守・運用	運用・保守マニュアル, 操作説明書

2.1 プロセスモデル

2.1.1 ウォーターフォールモデル

築時的なて銃ですすめる方法. 要求分析を明らかにしてから外部設計, 内部設計, プログラミング, テストを順番に行い, 次に保守・保守に移っていく.

特徴:築時的に開発を勧められ, 各段階の成果はドキュメントによって引き継がれるので管理が容易. 各段階ごとの作業分担がしやすい. 大規模システムの開発に向いている. 開発方法として定着しており, 開発要員の教育がしやすい.

問題点:要求定義の結果がコンピュータ上の動作で確認されるまで長期間を要する. 作業中のある段階で遅れが生じると, それ以降の段階に遅れが波及し, 次々にコウテイ遅れが生じる. 作業中の段階で上流段階の不具合が見つかった場合, 重龍にさかのぼって修正するのに多くの量力要する.

問題点への対応作:要求定義の内容をプロトタイピングで確認する. サブシステムごとに分割開発できる場合には, サブシステム単位にスパイラルモデルを用いる.

要求分析＝運用テスト, 外部設計＝システムテスト, 内部設計＝結合テスト, プログラミング＝単体テストの相互で認証・検証の関係を図で表し v モデル,v カーブと呼ぶ.

2.1.2 プロトタイピングモデル

プロトタイピングモデルは, 要求分析の内容の主要部分を試作し, ユーザとの使用確認結果をフィードバックして, プロトタイプを繰り返し修正しつつ要求内容の確認を行うモデルである.

プロトタイピングモデルには使い捨てプロトタイピングと進化型プロトタイピングがある. 進化型プロトタイピングモデルはプロトタイプを作成, 使用確認, 機能追加を繰り返し実用化の間で持っていく.

2.1.3 スパイラルモデル

スパイラルモデルは, ウォータフォールモデルと, プロトタイピングモデルの発展的進化を組み合わせたモデルである. このモデルは目標・対策・制約決定の区域, 対策評価の区域, 開発検証の区域, 計画の区域, の四つの区域を渦巻状に開発を進めていく. この四つの区域を一サイクルとして, ソフトウェア開発をいくつかのサイクルを繰り返して進めていく. スパイラルモデルは, これらの区域を繰り返し通ることにより, 仕様の確認, リスクの分析と回避, 代替案の取り入れなどを含めて開発を進めていくことが特徴である.

3 要求分析

ソフトウェア開発の初期段階で行われるもので, 何をこのソフトウェアで実現すべきかを明らかにする作業. ユーザの要求を把握し, 満足してもらえる実現可能な要求モデルを作成し, ソフトウェア設計者が正確に判断できる要求仕様書にまとめること.

機能仕様:システムが実現する機能要件

非機能仕様:システムの持つべき性能, 保守性, 安全性, 信頼性, セキュリティ, 相互運用性等で品質特性に大きく関連する.

3.1 要求分析における課題

- ユーザ要求のあいまいさ
- ユーザ要求の多様性
- ユーザと分析者間の相互理解の難しさ
- ユーザ要求の頻繁な変化

3.2 要求分析の技法

ユーザ要求の獲得	ユーザから直接, 情報を収集整理し, 知識として理解する作業
ユーザ要求の表現	得た知識を要求モデルあるは 自然言語により表現し, 機能やその他の仕様として記述する作業
ユーザ要求の妥当性確認	表現された要求仕様が ユーザの意図とあっているか確認, 曖昧や抜け, 矛盾がないか確認を行う作業

3.3 ユーザ要求の獲得

獲得する知識・情報	獲得の方法
解決スベき問題と関連情報	インタビューによる分析
組織構成	目的・目標の分析
実現システムの利害関係者	シナリオの利用
システム運用環境	プロトタイピング
個別業務内容	業務関連文書の利用
問題領域の範囲	

3.4 ユーザ要求の妥当性確認

3.4.1 要求モデル検証の基準

正当性:ユーザが望んでいること

完全性:重要な情報が欠落していないこと

一貫性:記述が矛盾していないこと

非曖昧性:二つ以上の意味に解釈されないこと

最小性:最小の記述量であること

検証可能性:テストできること

実現可能性:実現できること