

【特許請求の範囲】

【請求項 1】

ソフトウェアの要件定義から、開発見積、設計、運用までを支援するソフトウェア設計・運用統合管理システムであって、

業務案件を定義する要件情報と、前記要件情報に基づいて作成された見積情報と、前記要件情報に基づいて作成された設計情報と、前記設計情報に基づいて製造されたソフトウェアのリリース後の運用情報を格納するデータベースを備え、

前記設計情報には、少なくとも、機能仕様書、外部インターフェース設計書、画面設計書、帳票設計書、DB設計書、プログラム設計書を設計要素として含み、前記設計情報を管理する設計情報管理部と、

前記見積情報には、過去にリリースした案件の開発工数、開発期間の実績値を見積要素として含み、前記見積要素に基づいて、新規開発案件の見積値を算出する見積情報管理部と、

前記運用情報には、前記ソフトウェアが稼動する先の顧客情報を運用要素として含み、前記運用情報を管理する運用情報管理部と、

前記設計要素、前記見積要素、前記運用要素それぞれに対して、互いに影響範囲にある要素をリンク付けして前記データベースに格納する手段と、

前記リンク付けされた情報を各情報管理部間で登録・修正・検索・照会する手段とを、備えることを特徴とするソフトウェア設計・運用統合管理システム。

【請求項 2】

前記顧客情報は、顧客担当者の連絡先情報を含み、前記連絡先情報は、前記ソフトウェアを構成するプログラムまたはプログラムの各出力に対応した設計要素にリンク付けされており、前記ソフトウェアの障害発生時に、障害が検知された設計要素にリンク付けされた前記顧客情報を表示することを特徴とする、請求項 1 に記載のソフトウェア設計・運用統合管理システム。

【請求項 3】

前記顧客情報は、障害時の対処方法を記述した情報を含み、前記障害時に、障害が検知された設計要素にリンク付けされた前記対処方法を表示することを特徴とする、請求項 2 に記載のソフトウェア設計・運用統合管理システム。

【請求項 4】

前記対処方法には、障害復旧の完了必達時間を含み、前記障害の復旧完了予定時間が前記完了必達時間よりも遅れる場合には、顧客に通知することを前記運用情報管理部の操作員に促すことを特徴とする、請求項 3 に記載のソフトウェア設計・運用統合管理システム。

【請求項 5】

前記見積情報管理部は、案件の画面数、帳票数、機能数、外部ファイル項目数を設計パラメータとしたとき、

$$FP = k + a_1 \times [\text{画面数}] + a_2 \times [\text{帳票数}] + a_3 \times [\text{機能数}] + a_4 \times [\text{外部ファイル項目数}] + \dots$$

で定義される F P 算出式において、過去実績の F P 値と前記設計パラメータの実績値から、 k 、 a_1 、 a_2 、の各種定数を回帰分析によって算出することを特徴とする、請求項 1 に記載のソフトウェア設計・運用統合管理システム。

【請求項 6】

前記見積情報管理部は、前記各種定数の求めた前記 F P 算出式に、新規開発案件の画面数、帳票数、機能数、外部ファイル項目数の予測値を入力することにより、F P 予測値を算出することを特徴とする、請求項 5 に記載のソフトウェア設計・運用統合管理システム。

【請求項 7】

前記見積情報管理部は、前記新規開発案件の案件リリース後に、前記各種定数を定める前の前記 F P 算出式に、リリース後の案件の画面数、帳票数、機能数、外部ファイル項目

10

20

30

40

50

数の実績値を入力することにより、各種定数を再度求めることにより前記FP算出式にフィードバックをかけることを特徴とする、請求項6に記載のソフトウェア設計・運用統合管理システム。

【請求項8】

前記見積情報管理部は、開発担当者のスキル値の実績値を含み、前記算出されたFP見積値と前記スキル値から、新規案件の作業期間を見積もることを特徴とする、請求項6乃至7に記載のソフトウェア設計・運用統合管理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、要件情報、設計情報、プログラム情報、運用情報を結び付けたソフトウェア設計・運用統合管理システムに関する。

【背景技術】

【0002】

従来から、ソフトウェアの設計や運用、あるいは開発作業の見積もりを支援するためのシステムが多数知られている。例えば、特許文献1には、入力された構成情報からシステム構成図を作成し、システム要件を入力させ、これと運用ポリシーデータを作成し、構成情報に反映させる設計支援システムが記載されている。また、特許文献2には、運用作業項目、運用条件、サービスレベルを格納する運用要件DBと、運用要件DBの内容を画面表示し、設計項目を入力する手段と、入力した運用設計項目をもとに、コストの算出、設計ドキュメントの作成をおこなう運用設計支援システムが記載されている。また、特許文献3には、ソフトウェア規模および開発工数の見積もりの指定を作業者に促す画面を表示し、作業者は順次表示される画面に情報を入力していく見積もり作業支援システムが記載されている。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2009-251672号公報

【特許文献2】特開2009-075961号公報

【特許文献3】特開2003-263320号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、上記特許文献の記載のシステムは、設計、運用、見積もりをそれぞれ個々に支援するものであるが、それらを一環してサポートするものではない。要件情報、見積情報、設計情報、プログラム情報、運用情報（障害情報を含む）は互いに密接に関連しており、一部の変更が他に与える影響範囲を確実に管理しなければならない。詳細設計段階やプログラミングの段階では、プログラム・モジュールの関連性や変更の管理をサポートするシステムは多く存在するが、要件定義から、見積もり、設計、製造（プログラミング及びテスト）、運用、保守に至るまで一環してサポートできる仕組みはまだ少ない。したがって、各段階での詳細レベルの情報を有機的に結合して、総合管理できるシステムが望まれている。

【0005】

本発明では、上記のような課題に鑑み、要件定義から、見積もり、設計、製造、運用、障害時にいたるまで、ソフトウェアの各コンポーネントレベルで有機的に詳細情報を関連付け、統合されたソフトウェア設計・運用のための統合管理システムを提供することを目的とする。

【課題を解決するための手段】

【0006】

上記課題を解決するため、本発明のソフトウェア設計・運用統合管理システムは、以下

10

20

30

40

50

のような解決手段を提供する。

ソフトウェアの要件定義から、開発見積、設計、運用までを支援するソフトウェア設計・運用統合管理システムであって、業務案件を定義する要件情報と、前記要件情報に基づいて作成された見積情報と、前記要件情報に基づいて作成された設計情報と、前記設計情報に基づいて製造されたソフトウェアのリリース後の運用情報を格納するデータベースを備え、前記設計情報には、少なくとも、機能仕様書、外部インターフェース設計書、画面設計書、帳票設計書、DB設計書、プログラム設計書を設計要素として含み、前記設計情報を管理する設計情報管理部と、前記見積情報には、過去にリリースした案件の開発工数、開発期間の実績値を見積要素として含み、前記見積要素に基づいて、新規開発案件の見積値を算出する見積情報管理部と、前記運用情報には、前記ソフトウェアが稼動する先の顧客情報を運用要素として含み、前記運用情報を管理する運用情報管理部と、前記設計要素、前記見積要素、前記運用要素それぞれに対して、互いに影響範囲にある要素をリンク付けして前記データベースに格納する手段と、前記リンク付けされた情報を各情報管理部間で登録・修正・検索・照会する手段とを、備えることを特徴とする。

10

【0007】

このような構成によれば、設計情報管理部、見積情報管理部、運用情報管理部がそれぞれ管理する情報を各要素単位で互いに影響を及ぼす情報をリンク付けしてデータベース上で管理し、見積情報管理部、設計情報管理部、運用情報管理部等のサブシステム間で、各段階の必要な場面で、それぞれの情報を作成・変更したり、参照したりすることができるので、要件定義から、見積もり、設計、運用までを一貫したシステムでサポートすることができる。

20

【0008】

本発明のシステムは、下記のような特徴をさらに備えることができる。

前記顧客情報は、顧客担当者の連絡先情報を含み、前記連絡先情報は、前記ソフトウェアを構成するプログラムまたはプログラムの各出力に対応した設計要素にリンク付けされており、前記ソフトウェアの障害発生時に、障害が検知された設計要素にリンク付けされた前記顧客情報を表示することを特徴とする。

【0009】

このような特徴によれば、本システムの障害支援機能として、ソフトウェアを構成するプログラム（プログラム・モジュール）またはプログラムの出力（ファイル、画面、帳票等）に対応する詳細レベルの設計情報に対して、プログラムまたはプログラムの出力に障害が発生した時点で連絡が必要な顧客の連絡先をリンク付けする。このようにすることで、設計情報の各要素ごとに、障害時に連絡が必要な連絡先を表示することができ、障害時の迅速な連絡を支援することができる。例えば、エンドユーザに対して、帳票出力を行うプログラムであったとし、その帳票のデータに誤りが発見された場合、顧客のエンドユーザに多大な影響があるので、緊急の連絡が必要な場合があるからである。

30

【0010】

本発明のシステムは、下記のような特徴をさらに備えることができる。

前記顧客情報は、障害時の対処方法を記述した情報を含み、前記障害時に、障害が検知された設計要素にリンク付けされた前記対処方法を表示することを特徴とする。

40

【0011】

このような特徴によれば、障害時には各要素単位で連絡先を表示する他、予め定められた障害時の対処方法が運用状況監視者等（オペレータ）に表示できるようになる。例えば、ジョブがアベンドした場合、後続の連携するジョブを停止する等の必要な処置をとることができる。このようにすることで、障害時の判断を迅速に行うことができる。

【0012】

本発明のシステムは、下記のような特徴をさらに備えることができる。

前記対処方法には、障害復旧の完了必達時間を含み、前記障害の復旧完了予定時間が前記完了必達時間よりも遅れる場合には、顧客に通知することを前記運用情報管理部の操作員に促すことを特徴とする。

50

【0013】

このような特徴によれば、復旧完了予定時間を保守員等が入力し、復旧が早期に見込まれる場合等には、不要な連絡をしない等の処理をとることができる。このようにすることで、障害時の判断を的確に行うことができる。

【0014】

また、本発明のシステムは、下記のような特徴をさらに備えることができる。

前記見積情報管理部は、案件の画面数、帳票数、機能数、外部ファイル項目数を設計パラメータとしたとき、 $FP = k + a_1 \times [\text{画面数}] + a_2 \times [\text{帳票数}] + a_3 \times [\text{機能数}] + a_4 \times [\text{外部ファイル項目数}] + \dots$ 、で定義されるFP算出式において、過去実績のFP値と前記設計パラメータの実績値から、 $k_1, a_1, a_2,$ の各種定数を回帰分析によって算出することを特徴とする。

10

【0015】

このような特徴によれば、本システムの見積支援機能として、見積情報管理部が保存している過去の案件のFP値と上記の各種パラメータ実績値を上記算出式に代入し、上記算出式の各種定数(係数)を回帰分析によって求めることで、FP(ファンクション・ポイント)法における、ソフトウェアの持つ機能の数をもとに、そのソフトウェアの規模を測定する手法の考えにのっとり、見積もりを定量化するための一つのFP算出式を求めることができる。

【0016】

上記の見積支援機能は、下記のような特徴をさらに備えることができる。

20

前記見積情報管理部は、前記各種定数の求まった前記FP算出式に、新規開発案件の画面数、帳票数、機能数、外部ファイル項目数の予測値を入力することにより、FP予測値を算出することを特徴とする。

【0017】

このような特徴によれば、新規案件のパラメータ数を見積もり、上記の回帰分析により定数(係数)が求まったFP算出式に入力することで、新規案件に対するFP予測値が得られる。

【0018】

上記の見積支援機能は、下記のような特徴をさらに備えることができる。

前記見積情報管理部は、前記新規開発案件の案件リリース後に、前記各種定数を定める前の前記FP算出式に、リリース後の案件の画面数、帳票数、機能数、外部ファイル項目数の実績値を入力することにより、各種定数を再度求めることにより前記FP算出式にフィードバックをかけることを特徴とする。

30

【0019】

このような特徴によれば、新規案件をリリースした後に、FP実績値、各種パラメータ実績値を、入力することで、上記のFP算出式の各種定数を再度求め、FP算出式自体にフィードバックをかけることができる。このようにすることで、案件情報が蓄積すればするほど、より正確な見積もりが行えるようになる。

【0020】

上記の見積支援機能は、下記のような特徴をさらに備えることができる。

40

前記見積情報管理部は、開発担当者のスキル値の実績値を含み、前記算出されたFP見積値と前記スキル値から、新規案件の作業期間を見積もることを特徴とする。

【0021】

このような特徴によれば、FP予測値を人日でなく、見積日数(見積作業期間)として求めることができる。

【発明の効果】

【0022】

本発明によれば、要件定義から、見積もり、設計、製造、運用、障害時にいたるまで、ソフトウェアのコンポーネントレベルで有機的に関連付け、統合されたソフトウェア設計・運用のための統合管理システムを提供することができる。

50

【図面の簡単な説明】

【0023】

【図1】本発明の一つの実施形態に係るソフトウェア設計・運用統合管理システムの全体像を示す図である。

【図2】設計用語の定義を示す図である。

【図3】サブシステム情報管理の画面遷移を示す図である。

【図4】要件情報と設計情報のリンクを示す図である。

【図5】設計情報とプログラムのリンクを示す図である。

【図6】障害連絡時の判断支援(Case 1)を示す図である。

【図7】障害連絡時の判断支援(Case 2)を示す図である。

10

【図8】障害連絡時の判断支援画面イメージを示す図である。

【図9】見積支援の概要を示す図である。

【図10】見積支援画面イメージを示す図である。

【図11】開発工数見積ロジックを示す図である。

【図12】開発工数見積ロジックの例を示す図である。

【図13】開発期間見積ロジックを示す図である。

【図14】開発期間見積ロジックの例を示す図である。

【発明を実施するための形態】

【0024】

以下、添付図面を参照して、本発明を実施するための形態(以下、実施形態)について詳細に説明する。なお、実施形態の説明の全体を通して同じ要素には同じ番号を付している。

20

【0025】

図1は、本発明の一つの実施形態に係るソフトウェア設計・運用総合管理システム(以下、本システム100と呼ぶ)の全体像を示す図である。本システムは、図示するように、典型的な構成としては、設計情報DBを備えた設計情報管理部110、変更管理部120、ワークフロー管理部130、リリース管理部140、見積支援ツールと見積情報DBを備えた見積情報管理部150、運用管理ツールと運用情報DBを備えた運用情報管理部160、リソース管理ツールとプログラムソース管理DBを備えたリソース管理部170から構成される。それぞれの管理部は、本システム100のサブシステムとして機能する。

30

【0026】

設計情報管理部110は、各種のデザイン支援ツール(画面デザインツール、帳票デザインツール等)と連動し、様々な設計書を格納する設計情報DBを備えた設計情報管理サブシステムである。設計情報DBには、設計書として、ビジネスフロー定義書、要件定義書、及び要件定義書に基づいた機能設計書が格納される。さらに、この機能設計書に基づいた画面設計書、帳票設計書、外部I/F設計書をはじめとして、データベース(以下、DB)設計書、レイアウト設計書、コード設計書、パラメータ設計書、メッセージ設計書、プログラム設計書、ジョブネット設計書等が格納される。これらの設計書は、設計要素とも呼ばれ、後述の図2の表で定義した内容を含んだファイルである。これらの設計書は、お互いの依存関係を示すため、互いに紐付け(リンク付け)されて、設計情報DBに格納される。本システム100では、1または複数のサーバ装置が、複数のユーザ端末とネットワークで接続され、設計情報DBを中核とする各DBを用いて、各ファイルのバージョン管理、影響範囲管理、変更履歴管理を行う。

40

【0027】

変更管理部120は、設計情報が基づいている案件情報に変更があった場合や設計情報に変更があった場合に、その変更を管理するサブシステムである。ワークフロー管理部130は、上記設計書を新規作成、変更、照会する場合に、作成者、変更者、照会者、承認者等及びその承認経路を定義したDB(図示せず)を備えた承認ワークフローのためのサブシステムである。リリース管理部140は、単体テスト後のプログラムを結合してシス

50

テムテストを行う際、及び本番リリース時のバージョン、レビジョン等のリリース管理や配布管理、貸出管理、自動コミット等を行うためのサブシステムである。

【 0 0 2 8 】

見積情報管理部 1 5 0 は、新規案件の見積もりや既存案件の変更見積もりが必要な場合に、見積者が、類似する過去の案件の見積情報を参照したり、現在登録されている開発担当者（SE やプログラマ等）のスキルや単価等を格納した見積情報 DB を備えたサブシステムである。本システム 1 0 0 の見積情報は、案件情報、設計情報とも密接に関係付けられる。

【 0 0 2 9 】

運用情報管理部 1 6 0 は、ソフトウェアが実際のユーザの環境で稼働後の運用状況を監視する運用ツールと、障害時に障害内容や障害を関係者に連絡するための連絡先情報を格納する運用情報 DB を備えたサブシステムである。本システム 1 0 0 の運用情報は、案件情報、設計情報とも密接に関係付けられる。

10

【 0 0 3 0 】

リソース管理部 1 7 0 は、開発中及びリリース後のプログラムのソースファイルをソース解析し、プログラム間の関連性を管理したり、ソースファイルの C R U D（Create（生成）、Read（読み取り）、Update（更新）、Delete（削除））を管理するリソース管理ツールと、それらの情報を格納したプログラムソース管理 DB を備えたサブシステムである。なお、本システム 1 0 0 では、上記のサブシステム以外にも、製造・単体テストを管理する単体テスト管理部、プログラム設計書に基づいてプログラムを自動生成する P G M 生成部等が存在するが、ここでは省略している。

20

【 0 0 3 1 】

本システム 1 0 0 では、これらのサブシステム内、サブシステム間の情報の関連を確実に紐付けるため、要件定義段階、見積段階、設計段階、製造段階（プログラミング段階）、テスト段階、運用段階のドキュメントやデータの各コンポーネント（これらを、それぞれ「要件情報要素」、「見積情報要素」、「設計情報要素」、「プログラム要素」、「運用情報要素」と呼ぶことにする）おのにおに I D（識別子）を定義し、互いにリンク付けした上で DB に格納することを特徴としている。言い換えれば、I D をふられたファイルや情報を、「要素」と呼ぶことにする。設計要素には、上述した各種の設計書が含まれるし、見積情報要素には、過去の案件の情報や開発担当者や運用担当者の情報が含まれる。また、運用情報要素には、運用状況のデータや顧客情報が含まれる。

30

【 0 0 3 2 】

なお、上記図示した構成は、あくまでも一例であり、一つの機能部（サブシステム）を更に分割したり、複数の機能部をまとめて一つの機能部として構成してもよい。各機能部は、サーバ装置に内蔵された C P U（Central Processing Unit）が、R O M（Read Only Memory）またはハードディスク等の記憶装置に格納されたコンピュータ・プログラムを読み出し、C P U により実行されたコンピュータ・プログラムが、記憶装置に格納されたデータベース（D B ;Data Base）やメモリ上の記憶領域からテーブル等の必要なデータを読み書きし、場合によっては、関連するハードウェア（例えば、入出力装置、表示装置、通信インターフェース装置）を制御することによって実現される。また、本発明の実施形態におけるデータベース（D B）は、商用データベースであってよいが、単なるテーブルやファイルの集合体をも意味し、データベースの内部構造自体は問わない。

40

【 0 0 3 3 】

図 2 は、設計用語の定義を示す図である。この表は、主に設計時に用いられる用語の定義を表にまとめたものである。例えば、「パラメータ設計」とは、「システムの挙動を簡単に変えることができるような業務に関するデータを定義する作業である」のように定義され、その作業のアウトプットのドキュメントがパラメータ設計書である。設計情報 DB に格納される設計書（設計情報ファイル）は、この表の内容を定義したファイルであり、本システム 1 0 0 の画面デザインツールや帳票デザインツール等に沿った所定のフォーマットに従って作成されるものと、ビジネスフロー定義書、要件定義書、機能定義書等、任

50

意のファイル形式であってよいものが存在する。また、この表は一例であり、特に表には示していないような設計用語も存在する。

【0034】

図3は、サブシステム情報管理の画面遷移を示す図である。本発明のソフトウェア設計・運用統合管理システム100は、要件定義から、見積もり、設計、プログラミング、テスト、リリース、運用、障害対処まで、様々な局面で利用されるが、それぞれの局面をサポートするため、すでに述べた複数のサブシステムで構成される。本システム100に登録された情報は、互いに紐づけされ、関連するので、サブシステム内ではもちろん、サブシステム間でも共用される必要がある。ここでは、複数のサブシステム間の情報管理のために画面遷移について説明する。また、本システム100では、単に情報共有だけでなく、上記の様々な業務を支援するようなツールが提供され、本システムのユーザ、例えば、プロジェクトリーダー、設計者、プログラマ、テスト、運用監視者、保守要員等に対して、それぞれの役割にあった情報を提供することは言うまでもない。

10

【0035】

ユーザは、本システムにアクセスするために、まず、ログイン画面からログインする必要がある。ログイン画面は他のサブシステムと共通であってもよいし、サブシステム毎に別々の画面であってもよい。ここでは、あるユーザが、通常、Aサブシステムを使用するユーザであり、場合によって、他のサブシステム(Bサブシステムとする)に情報照会をする場合を考える。ユーザがAサブシステムのログインに成功すると、Aサブシステムがサポートするメニュー画面が表示される。ここでは、情報連携のため、大きく分けて、「登録・修正画面」、「検索画面」、「照会画面」を取り上げている。

20

【0036】

システムに登録された情報の検索・照会時においては、ユーザが検索入力画面(SF-A21-01)から必要な情報をキーワード等で検索し、その結果を検索結果画面(SF-A21-02)に表示させる。検索結果表示画面(SF-A21-02)で、必要な情報のID(対象定義ID)が見つかったら、そこに表示されたリンク等をクリックすることで、照会画面(SF-A21-03)に移り、その情報を参照することができる。照会画面からさらに変更履歴照会画面(SF-B02-03)に移り、変更履歴も参照することができる。図示するように、照会画面から登録・修正画面へは、複数のTabキー等を押すことにより、いつでも切り替えることができる。

30

【0037】

登録・修正時においては、入力画面(SF-A21-04)から新規にシステムに登録する情報の入力や、登録した情報を修正することができる。登録・修正入力画面(SF-A21-04)で、「実行」キーを押すことで、確認画面(SF-A21-05)が表示され、登録・修正が完了すると完了画面(SF-A21-06)が表示される。

【0038】

ユーザが、別のサブシステムであるBサブシステム管理下にある情報にアクセスする必要がある場合は、Bサブシステムにアクセスするための申請検索を申請検索入力画面(SF-B01-01)から行う。この申請は、プロジェクト関係者であればそのまま処理されるが、部外者がアクセスする場合等、必要な場合には承認ワークフローサブシステムを利用するようにしてもよい。承認ワークフローサブシステムのその他の典型的な利用例としては、重要な変更と考えられる「設計情報の変更」などがある。この申請が処理されると、申請検索結果画面(SF-B01-02)が表示され、そこからAサブシステムの照会画面(SF-A21-03)に戻り、Bサブシステム下の情報に、以後アクセスできるようになる。この申請により、Bサブシステムの照会画面(SF-A16-03)に対象定義IDによってAサブシステムの照会画面(SF-A21-03)からリンク付けがされるからである。このようにすることで、Bサブシステムに再度ログインする等の手間がなくなる。このような情報管理画面を提供することで、ユーザは、複数のサブシステム間のアクセスが容易になり、各情報の連携が進むことになる。

40

【0039】

50

図4は、要件情報と設計情報のリンクを示す図である。既に述べたように、設計情報DBに格納される各ファイルは、互いに紐付け（リンク付け）されるが、ここでは、一例として、要件情報と設計情報のリンク付けを示す。要件情報とは、ビジネスフロー定義と、要件定義を記述したファイルである。ビジネスフローとは、図2で示したように、顧客の業務について明確化する作業（アクティビティ）であり、要件定義とは、顧客が漠然と考えているシステムに求める役割を明確化する作業である。図示するように、一つのビジネスフロー定義から複数の要件定義が生まれることもある。それぞれを定義したファイルには、アクティビティID（BA01）、要件ID（RQ01, RQ02）のような変更管理IDがふられ、それぞれの変更管理IDがリンク付けされて（図では太い実線で示す）、DB上で管理される。

10

【0040】

設計情報レポジトリ（設計情報DB）には、機能設計書、画面や帳票等の人間とのインターフェースであるI/F設計書、DBの詳細（テーブル等）を定めたDB設計書、プログラム（PGM）ファイルが格納され、それぞれのファイルに対応するID（機能ID、画面ID、帳票ID、テーブルID、プログラムID）がふられて、各IDがリンク付けされる。また、プログラムIDは、単体テストシナリオまたは単体テストケースのIDとリンク付けされテスト管理で使用される。それぞれのリンク付けは、情報の作成者が、多くの場合、その情報を作成する際に参照した情報のIDを指定することによってなされるが、プログラム設計書等、所定のフォーマットで作成する場合は、フォーマット中にリンク先のIDを含めるようにする。また、プログラムソース・ファイルの場合は、ソース

20

【0041】

図5は、設計情報とプログラムのリンクを示す図である。この図では、設計情報とジョブ及びプログラムとのリンク付けの簡単な一例を示す。図示するように、データ連携ジョブネット10（ID：JN11）は、後続のデータ連携ジョブネット20（ID：JN12）を従えている。ジョブネットとは、実行順序を指定した、一つ以上のジョブの集まりのことである。この図では、データを連携する2つのジョブネットを示している。

【0042】

データ連携ジョブネット10は、データ連携ジョブ11（ID：JB11）と帳票出力ジョブ12（ID：JB12）の2つのジョブから構成されている。データ連携ジョブ11は、データ連携のためのメインプログラムであるデータ連携メイン13（ID：PG11）とリンク付けされ、データ連携メイン13は、ファイル出力サブルーチン15（ID：PG01）と帳票出力サブルーチン16（ID：PG02）を呼び出している。ファイル出力サブルーチン15は、外部のサブシステムとデータ連携するためのファイルを出力するプログラムである。出力したファイルは、I/F IDとしてIF01がふられ、このIDを、ファイル連携用のID、例えば、HULFT（登録商標）；（Host Unix Linkage File Transfer）IDとして、出力サブルーチンと紐付けて管理する。帳票出力サブルーチン16は、帳票出力メイン14（ID：PG12）からも呼び出される共通サブルーチンでもあり、顧客確認用の帳票（帳票フォーム18（ID：RP01）、及び帳票

30

40

【0043】

以上、図4、図5の例で説明したように、本システム100では、要件定義情報から、機能仕様書、ジョブ定義書、プログラム、及び最終的に出力されるファイルや帳票にいたるまで、一つ一つのコンポーネント（要素）にIDがふられ、それぞれの関係が紐付けされてDBで管理されるので、それぞれのコンポーネントが他のコンポーネントに及ぼす影響の範囲（これを影響範囲と呼ぶ）を詳細に捉えて管理することができる。この影響範囲の管理は、単に開発時の設計情報の間だけでなく、開発以前の見積段階や開発リリース後の運用保守情報の影響の管理にも役立てることができる。そのため、以降では、特に、障害時の判断支援や見積支援にスポットを当てて、具体例を説明する。

50

【 0 0 4 4 】

図 6 は、障害連絡時の判断支援 (C a s e 1) を示す図である。この図では、設計情報と運用情報をリンク付けすることによる影響範囲管理の具体例として、障害連絡時の判断支援を説明する。ここでは、C a s e 1 として、実環境で運用開始後、帳票データが誤っていることが発覚した場合を想定している。図で示した各コンポーネント間のリンク関係を表すツリー構造は、基本的には図 5 と同じであるとする。

【 0 0 4 5 】

設計情報と運用情報のリンクとして、ここでは、各ジョブの出力であるインターフェース 1 7、2 2、及び帳票フォーム 1 8、1 9 に、それぞれ顧客情報 1 7 a、1 8 a、1 9 a、2 2 a が紐付けられている。「顧客情報」とは、対象ソフトウェアが稼動するシステムを運用する顧客の連絡先等 (部署名、担当者名、電話番号)、及び障害時に顧客管理のため必要な対処法の情報を記述したファイルである。顧客情報は、設計時に定義されることもあるが、運用時に設定されることが普通であるので、通常は、設計情報 D B でなく、運用管理情報 D B に格納される。

【 0 0 4 6 】

ここで、帳票フォーム 1 8 (I D : R P 0 1) でデータの誤りが発覚した場合を考える。この場合、帳票フォーム 1 8 に誤りが発覚した時点で、保守要員等 (オペレータ) が運用管理ツール上で所定の操作をすると、帳票フォーム 1 8 にリンクされた顧客情報 1 8 a が表示される。また、システムがリンク情報を参照することで、影響範囲の対象であるコンポーネント (ここでは網掛けで示している) をすべて表示し、同じ帳票出力サブルーチン 1 6 が出力した帳票フォーム 1 9 についても横並びのチェックが必要なことが直ちに判断できる。さらに、このリンク情報から、帳票そのものでなく、ファイル出力サブルーチン 1 7 が出力したファイル 1 7 (I D : I F 0 1) にも誤りがある可能性が容易に判断がつくので、後続のデータ連携のため、例えば、H U L F T キックジョブを停止したりする必要性や、原因究明に時間がかかりそうな場合は、顧客に連絡が必要であることも直ちに判断できる。

【 0 0 4 7 】

顧客情報 1 8 a には、顧客連絡先の他、直ちに連絡すべきかどうかを判断するための情報も含まれていてもよい。例えば、無条件に直ちに報告を必要とする場合を定めた情報や、所定時間以内に解決すれば事後報告のみでよいとする場合を定めた情報等である。また、顧客情報には、顧客の連絡先だけでなく、開発担当者やプロジェクトリーダーの連絡先等を含ませてもよいし、担当者情報を格納した D B 中の担当者 I D をリンク付してもよい。このようなリンク付によって、たとえ障害が発生しても、誰に、どういう手順で、どのくらいのタイミングで連絡すべきかを、障害が発生したコンポーネントごとに詳細に定義することができる。このことにより、逆に、無駄な連絡や報告を減少させることも可能である。

【 0 0 4 8 】

図 7 は、障害連絡時の判断支援 (C a s e 2) を示す図である。この図では、設計情報と運用情報をリンク付けすることによる影響範囲管理として、障害連絡時の判断支援の別の例を説明する。ここでは、C a s e 2 として、実環境で運用開始後、ジョブがアベンド (A b e n d) した場合を想定している。ここでも、図で示した各コンポーネント間のリンク関係を表すツリー構造は、基本的には図 5 と同じであるとする。

【 0 0 4 9 】

帳票出力ジョブ 1 2 (I D : J B 1 2) がアベンドしたことが検知されると、システムが、影響範囲のコンポーネントのリンク情報を参照することで、網掛けで図示するように関連するコンポーネントが保守員等に表示される。まず、同じジョブ内では、帳票フォーム 1 8 と帳票フォーム 1 9 が、このジョブの出力であるので、それらにリンク付けされた顧客情報 1 8 a、1 9 a が表示される。その顧客情報には、帳票が出ないことで即座に顧客に伝える必要がある場合には、オペレータから顧客に連絡させるための必要な情報が表示される。また、この場合、帳票出力ジョブ 1 2 がアベンドして停止しているため、後続

10

20

30

40

50

のデータ連携ジョブネット20にも影響範囲が及ぶことになる。そのため、後続のデータ連携ジョブネット20には、これにリンク付けされた復旧の完了予測時間、完了必達時間の情報が表示される。また、データ連携ジョブ21には、前ジョブがアベンドしたことをいつまでに原因究明しないと顧客に影響が出るかを表示するようにする。不要な顧客連絡をしないようにもできる。また、次のデータ連携のジョブネットがある場合には、インターフェース22(ID:IF02)にHULFT送信時刻が過ぎそうであれば、顧客にオペレータから連絡が必要とのリンク情報を表示することもできる。

【0050】

以上、図6、図7で説明したように、設計時の各コンポーネントに顧客情報までもリンク付けることで、より柔軟な運用や保守が可能となる。また、即座に顧客に連絡すべきか否か、障害の完了予測時間や完了必達時間等の情報は、通常は、設計時には予測できないものであるため、運用管理DBで管理されるが、設計情報DBと運用情報DBとの間で各コンポーネントがリンク付けされ、情報の作成・変更が管理されるのはいうまでもない。

10

【0051】

図8は、障害連絡時の判断支援画面イメージを示す図である。この図では、保守担当者等が使用する運用管理ツールの一つの画面イメージを表している。図の中央の横線30は時間軸を表している。ここで、例えば、ジョブがアベンドしたことを何らかの手段でシステムが検知すると、現在時刻(12月1日 7:45)の時間と共に、アベンドしたジョブネットのID、この例ではJN12のアイコンが「×アベンド」(図中の白抜きの文字)として表示される。保守員等は、このアベンドしたジョブネットJN12のアイコン31をクリックすると、そのジョブネットの詳細情報32が表示される。また、ジョブネットJN12に関連して、完了要のジョブネットがある場合は、それらのアイコンが符号33で示すように表示される。特に、ジョブネットJN12の影響範囲内にあるジョブネットは、図のように網掛けで表示され、「要注意」であることが示される。

20

【0052】

ここで、「要注意ジョブネット」のうち、例えば、JN82のアイコンをクリックすると、ジョブネットJN82の詳細情報34がさらに表示される。この詳細情報34の中には、ジョブネットJN12の説明の他、システム状況として、障害復旧担当者等が入力した完了予定時刻の他、システムによって予め定められた完了必達時刻が表示され、このままでは完了必達時刻に間に合わない可能性がある旨が表示されている。また、ジョブネットJN82の詳細情報34の中には、出力対象帳票としてRP022のアイコンが表示され、このアイコンをクリックすると、帳票RP022の詳細情報35(ここでは対応する顧客連絡先等も含んでいる)がさらに表示される。これを見た保守員等は、オペレータに顧客に連絡することを指示することになる。また、アベンドしたJN12の影響範囲内にあるインターフェースであり、要注意のIF11のアイコン36をクリックすると、そこにリンク付けされた顧客連絡先37(この例では、開発を担当した会社の連絡先)が表示され、ここにも連絡が必要とわかる。このように、本システムの影響範囲管理の仕組みは、障害時の状況把握や必要な連絡先の表示等、リアルタイムに障害連絡時の判断をサポートすることができる。

30

【0053】

図9は、見積支援の概要を示す図である。この図では、本システムの影響範囲管理の別の具体例として、開発着手前の見積もり段階を支援する場合を示す。図で示す見積管理システムは、図1で説明した見積情報管理部150が設計情報管理部110と連携したイメージを表している。

40

【0054】

通常、新規開発案件の見積もりには要件定義、既存システムの変更の見積もりでは変更外形要素の定義と変更影響を管理する機能が必要である。また、開発方針や前提条件を定めた開発スコープ等が必要となる場合もある。さらに、開発実績の工数や障害時の対応工数等、過去の類似案件を実績分析することも必要である。また、開発要員の稼働状況や個人スキル値等も考慮する必要がある。これらの情報を本システムでは、主として、設計情

50

報DBと見積情報DBに格納している。個々の情報は、変更管理IDで管理され、互いにリンク付けされているのは言うまでもない。

【0055】

見積管理システムは、上記の各インプットから求められるデータを、ファンクション・ポイント(FP)算出式に入力し、新規案件の見積もりを行うことができる(これを改善FP法と呼ぶことにする)。また、要件、関連設計情報、変更外形要素、個人スキル値等を入力することで、新規案件の見積もりだけでなく、既存案件の変更の見積もりにも対応することができる。また、図9の右端で示すように、改善FP法のアウトプットは、開発の総人日として求められるので、それを個人スキル値で割ることにより、作業期間の見積もりをすることもできる。改善FP法による見積もり支援については以降の図で具体的に説明する。

10

【0056】

図10は、見積支援画面イメージを示す図である。この図は、見積管理システムが、見積者に対して表示する画面のイメージを示したものである。見積者は、左上の検索窓40から、検索したい種類(要件説明、ブロック説明等)を選択し、プロジェクトのキーワード等を入力して、関連する過去の案件を検索することができる。検索の結果は、左中央の符号41で示すように、要件ID、要件概要、FP、外部設計項目数(ブロック数、画面数・・・)等が一覧で表示される。そして、例えば、案件IDのRQ01のアイコンをクリックすると、右の符号42で示すように、各要件に紐づく情報がツリー構造で表示される。また、右下の符号43で示すように、その案件を担当した担当者ごとに、階層(スキルレベル)、担当した工程、その実績工数(人日)が表示される。

20

【0057】

図11は、開発工数見積ロジックを示す図である。この図では、改善FP法を用いて開発工数を見積る際の手順を説明する。開発工数の見積手順としては、(1)〈準備〉、(2)〈利用方法〉、(3)〈フィードバック〉の3段階で実施する。ただし、(2)〈利用方法〉には、「FP見積式を利用する方法」と、「似たような案件から規模感を乗じて算出」の2通りがある。以下、実施方法を順に説明する。

【0058】

まず、(1)〈準備〉として、図示するようなFPの算出式(式(1))の各種定数(k_1, a_1, a_2, \dots)の初期値を求める。そのためには、入力パラメータである画面数、帳票数、機能数、外部ファイル項目数等を、過去の実績値からインプットし、回帰分析を行い、上記の各種定数(係数)の大きさを推計する。上記のFP算出式では、FPが回帰分析における従属変数であり、各入力パラメータが独立変数である。回帰分析には様々な手法があるが、最小2乗法のような代表的な手法を用いてよい。実際には、見積者がFP実績値や各パラメータをいちいち入力せずとも、案件IDを選択するだけで、見積管理システムが、FP実績値と各パラメータの実績値とから、各種定数を算出する。

30

【0059】

上記の各種定数の算出が終わると、次に、(2a)〈利用方法1〉において、「FP見積式を利用する方法」画面数、帳票数等の各パラメータに値を代入してFPを算出することでFP見積を求める。ここでのインプットは、(1)〈準備〉で算出した各種定数と、今回見積もりを行う案件の画面数、帳票数、機能数、外部ファイル項目数等の各パラメータの予測値である。これを、上記のFP算出式に代入することで、FP見積値(人日)が求まる。

40

【0060】

上記の(2a)〈利用方法1〉では、システムが過去の実績値から予測パラメータからFP見積もりを算出した「FP見積式を利用する方法」であるが、(2b)〈利用方法2〉では、見積者が、検索結果から、似たような要件を持つ案件があれば、今回の案件がその案件と比較して規模感を決め、それをインプットとしてFP見積もりを算出する。規模感とは、似たような案件を持つ案件との規模比較(今回、何%の作業量がありそうか?)といった抽象的なもので、見積者の過去の経験に依存することになるが、逆に実践的な手法

50

でもある。また、(2a) <利用方法1>と併用することで、機械による単純計算と人間の経験に基づいた直感が組み合わさることで、より正確な見積もりをすることも可能である。

【0061】

上記のようにして求めたFP見積値は、案件リリース後に、(3) <フィードバック>として、その案件のFP実績値と、各種パラメータの実績値をインプットとして、FP算出式の精度を上げるために利用される。すなわち、FP見積値とFP実績値の差が最小になるように各種定数(k_1, a_1, a_2, \dots)を算出しなおす。このように求めた各種定数は、以後の案件のFP算出式で使用されることになる。

【0062】

図12は、開発工数見積ロジックの例を示す図である。この図では、図11で説明した開発工数見積ロジックの簡単な具体例を示す。まず、(1) <準備>では、過去の案件ごとに、テーブル50で示すような実績値が与えられているとし、これを回帰分析することで、FP算出式が式(2)のように求まる。ただし、ここでは、簡略化するために、設計パラメータをテーブル項目数、修正画面数だけに限定している。また、この例では、FPの単位として金額(千円)を使用しているが、標準工数等、様々な指標であってもよい。

【0063】

(2a) <利用方法1>においては、今回の案件A101の各種パラメータをテーブル51のように予測し、(1) <準備>で求めた式(2)に代入すると、 $FP見積 = 80.6 + 12.8 \times 2 + 22.2 \times 10 = 328.2$ [千円]とFP見積が求まる。

【0064】

(2b) <利用方法2>においては、案件A102と最も似ている案件がテーブル50のA002であったとし、A102はA002と比べ、機能が若干複雑になりそうのため120%くらいの作業量とすると、単純に、 $FP見積 = 300$ [千円] $\times 120\% = 360$ [千円]と予測できる。あるいは、A002を変更することでA102が実現可能と予測できる場合は、A002からの変更量を50%とすれば、 $FP見積 = 300$ [千円] $\times 50\% = 150$ [千円]と予測できる。このような場合、どちらの開発スコープを取るかで見積もりが大きく変わってくるので注意が必要であることが本見積管理システムを利用することで容易に判断できる。

【0065】

(3) <フィードバック>においては、案件A101が来たときにかかった実績工数も算出式のインプットとして利用する。実績値がテーブル52で示すような結果であった場合、FP算出式を再び回帰分析を利用して作成すると、 $FP見積 = 71.3 + 14.7 \times$ [修正対象テーブル項目数] $+ 8.2 \times$ [修正画面数] [千円] \dots 式(2)'となる。

【0066】

図13は、開発期間見積ロジックを示す図である。この図では、図12で説明した開発期間見積ロジックの簡単な具体例を示す。開発期間見積では、過去実績として、データベースに保持した社員(担当SE、プログラマ等)の作業履歴も利用する。これらの情報を利用することで、新規案件の開発工数だけでなく、作業期間の見積もりも可能となる。以下、その手順を説明する。

【0067】

まず、(1) <準備>では、図示するような式(3)に当てはめ、回帰分析にて個人能力値(スキル値)を見積もる。この場合のインプットは、FP(過去の1案件あたりの特定業務・特定工程あたりの工数(人日))と、社員*i*の作業工数(社員*i*が1案件・特定行部・特定工程に費やした工数)である。アウトプットは、社員*i*のスキル値(社員のスキルを係数化した値)である。

【0068】

(2) <利用方法>では、図11の開発工数ロジックで算出した「FP見積」を、対応する社員のスキル値で割ることで、見積作業期間(見積日数)を算出する(式(4)参照

10

20

30

40

50

）。この場合のインプットは、FP見積、社員*i*のスキル値、社員*i*の作業比率（社員がその案件に力を入れる割合（0～100%））である。アウトプットは、見積日数（見積作業期間）である。

【0069】

（3）＜フィードバック＞では、案件リリース後に、FP実績値と、各種パラメータをインプットとして、FP算出式（3）の精度を上げる。この場合のインプットは、FP実績値、社員*i*の作業工数であり、アウトプットは、社員のスキル値である。

【0070】

図14は、開発期間見積ロジックの例を示す図である。この図では、図13で説明した開発期間見積ロジックの簡単な具体例を示す。ここでは、簡略化のため、工程や業務知識は無視して、同一工程・同一業務に対する案件であるとする。まず、（1）＜準備＞において、各案件における担当者の作業工数がテーブル60のようであったとする。このテーブルをインプットとして、前図の式（3）から、各社員の能力がテーブル61のように求まる。

10

【0071】

（2）＜利用方法＞においては、図11で説明した開発工数見積ロジックを利用し、FP=200（千円）となるような案件が発生したとして、これを社員2と社員3の2人体制で作業するとした場合、図示するように、 $FP = 200 \div (1.3 +) = 50$ 日がかかることになる。同じ案件を社員1のみで作業したとすると、 $FP = 200 \div 6.4 = 32$ 日間で開発完了する見込みになる。

20

【0072】

以上説明したように、本システム100は、ソフトウェアの設計開発・運用支援を行うものであるが、単にソフトウェア開発だけに留まらず、ハードウェアを含んだ様々な製品やサービスの開発においても応用することが可能である。

【0073】

以上、実施形態を用いて本発明を説明したが、本発明の技術的範囲は上記実施形態に記載の範囲には限定されないことは言うまでもない。上記実施形態に、多様な変更または改良を加えることが可能であることが当業者に明らかである。またそのような変更または改良を加えた形態も本発明の技術的範囲に含まれ得ることが、特許請求の範囲の記載から明らかである。

30

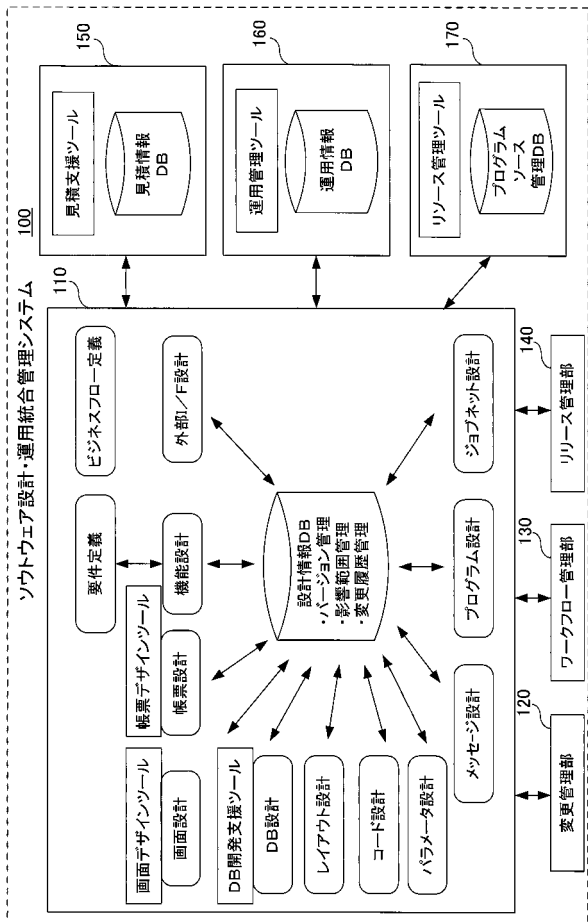
【符号の説明】

【0074】

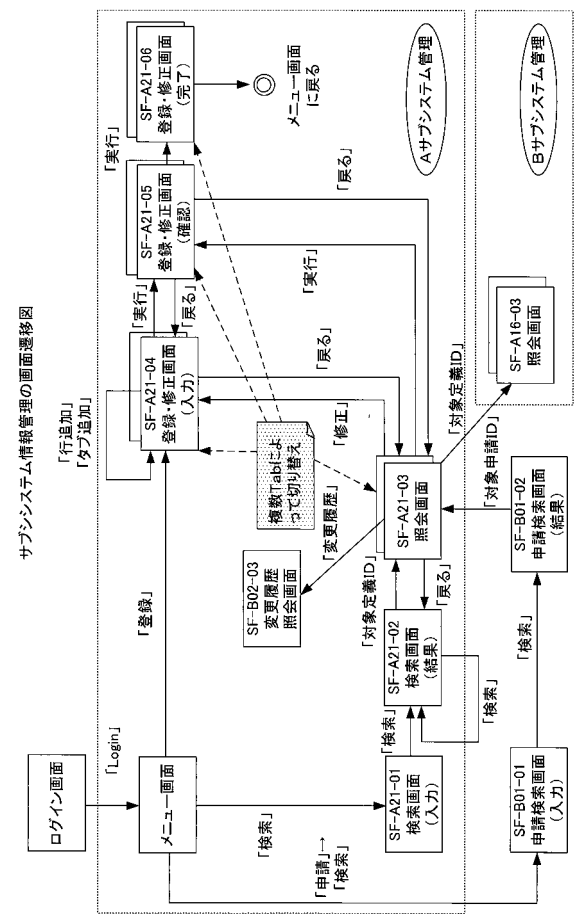
- 10、20 データ連携ジョブネット
- 11、21 データ連携ジョブ
- 12 帳票出力ジョブ
- 13 データ連携メイン
- 14 帳票出力メイン
- 15 ファイル出力サブルーチン
- 16 帳票出力サブルーチン
- 17、 IF ID
- 18、19 帳票フォーム
- 100 ソフトウェア設計・運用統合管理システム
- 110 設計情報管理部
- 120 変更管理部
- 130 ワークフロー管理部
- 140 リリース管理部
- 150 見積情報管理部
- 160 運用情報管理部
- 170 リソース管理部

40

【図1】



【図3】

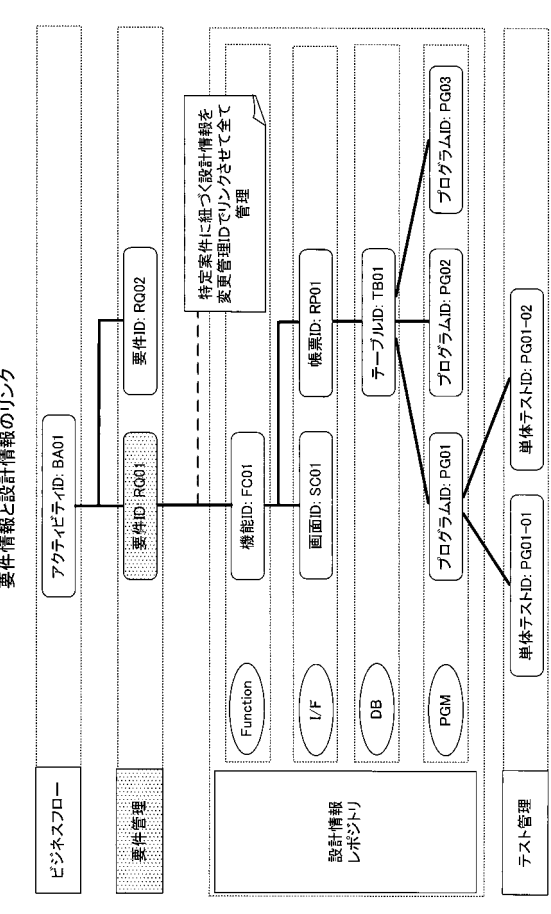


【図2】

設計用語の定義

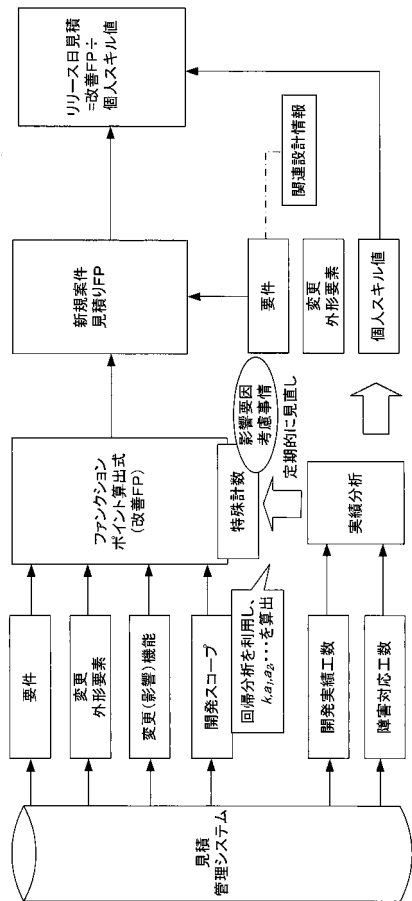
ビジネスフロー定義	顧客の業務について明確化する作業
ビジネスアクティビティ	顧客の業務の詳細なフローに関する定義
ビジネスルール	顧客の業務の詳細なルールに関する定義
要件定義	顧客が当然と考えているシステムに求める役割を明確化する作業
機能設計	システムが果たす役割や振る舞いの詳細を明確に定義する作業
機能定義	システムで果たす役割や振る舞いの詳細に関する定義 (例) 処理フロー、テーブルアクセス、処理優先度等
画面設計	システムで利用する画面の詳細を定義する作業
画面定義	画面に関する説明や、画面出力するために必要な情報の取得の定義
画面タブ定義	画面内に存在するタブに関する定義
画面項目定義	画面で表示する項目や入力させる項目に関する定義 (内容例) 入力許可文字設定、サイズ、入力チェック内容等
画面アクション定義	画面上から操作する内容に関する定義 (内容例) 画面から他画面への遷移、ファイルのダウンロード等
帳票設計	システムから出力される帳票のレイアウト、利用明細書、資格対照表などの内容の紙であり、電子ファイルの場合も含む)の強固な定義する作業
帳票定義	帳票に関する説明や、帳票を出力するために必要な情報の取得方法を定義
帳票項目定義	帳票に出力する項目に関する定義 (内容例) フォント、サイズ、罫線の有無等
外部インターフェース設計	対外システムとのシステム連携の仕組みを定義する作業
外部インターフェース定義	対外システムへのファイル、電文の送信や対外システムからのファイル、電文の受信に関する定義 (内容例) 対外システムIPアドレス、ファイル名、ファイル形式、圧縮有無、暗号化方式等
DB設計	データベースの仕組みを定義する作業
テーブル定義	データベースに格納する情報に関する定義
テーブル項目定義	データベースに格納するテーブルの詳細に関する定義 (内容例) 項目名、文字種別、桁数、項目の内容説明等
索引定義	データベースで使用する索引の詳細に関する定義
ビュー定義	データベースで使用するビューの詳細に関する定義
シークエンス定義	データベースで使用するシークエンス(シークエンス)の詳細に関する定義
コード設計	システム内で簡略化した書や文字列に置き換えて汎用的に利用できるように準備する作業
コード定義	コードとして必要な情報を定義 (内容例) コード種別(男性、女性、9その他)や説明等
パラメータ設計	システムの挙動を簡単に変更することができるような業務に関するデータを定義する作業 [急参考]「パラメータ設計」シートを参照
パラメータ定義	システムの挙動を変える対象に関する定義
パラメータ項目定義	システムの挙動を変えるに切り、制御を振り分けるキーとなる情報とそのパラメータの適用期間、各ケースにおける変更履歴に関する定義
メッセージ設計	システムでエンドユーザーや保守員に対して表示するメッセージを定義する作業
メッセージ定義	メッセージに関する定義 (例) メッセージレベル、メッセージ内容、対処方法等
プログラム設計	プログラムの詳細、ロジック、フローを定義する作業
プログラム定義	プログラムの詳細、ロジック、フローに関する定義 (内容例) アクセス対象テーブル、利用パラメータ、呼び出しプログラム等
レイアウト設計	ファイルの構造を定義する作業
レイアウト定義	ファイルの構造のテンプレートに関する定義
レイアウト項目定義	ファイル内のデータの項目に関する定義 (内容例) 各項目の意味、サイズ、型(文字、数値、日付、...)等
ジョブネット設計	システム内部で自動実行させるジョブの詳細を定義する作業
ジョブグループ定義	実行するジョブネットのスケジューリングを定義
ジョブネット定義	実行順序を指定した、一つ以上のジョブの集まりを定義
ジョブ定義	コンピュータにおける処理の最小単位を定義

【図4】



【図9】

見積支援の概要



【図11】

開発工数見積ロジック

①<準備> 過去実績をINPUTとして、回帰分析を行い定数の初期値を算出する

$$FP = k + a1 \times \text{【画面数】} + a2 \times \text{【帳票数】} + a3 \times \text{【機能数】} + a4 \times \text{【外部ファイル項目数】} + \dots \quad \text{式(1)}$$

- ・FP: 過去の1案件あたりの総工数(人月)
- ・【画面数】、【帳票数】、【機能数】、【外部ファイル項目数】等
- ・k, a1, a2, ...等各種定数

②a <利用方法1> 【画面数】、【帳票数】等、パラメータに値を代入してFPを算出することでFP見積りを求める

- ・【画面数】、【帳票数】、【機能数】、【外部ファイル項目数】等
- ・①で算出したk, a1, a2, ...等各種定数
- ・FP見積

②b <利用方法2> 似たような要件を持つ案件を探し、その案件と比較して規模感を決め、それをINPUTとして、FP見積りを算出する

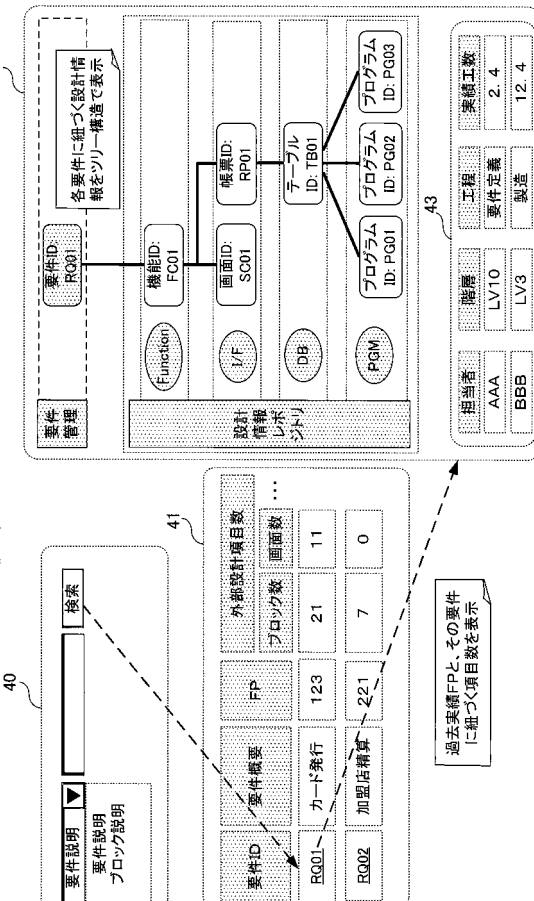
- ・似たような案件を持つ案件との規模比較(何%の作業量がありそうか?)
- ・FP見積

③ <フィードバック> 案件リリース後に、FP、各種パラメータをINPUTとして、FP算出式の精度を上げる

- ・FP: 今回の案件の実工数(人月)
- ・【画面数】、【帳票数】、【機能数】、【外部ファイル項目数】等
- ・k, a1, a2, ...等各種定数

【図10】

見積支援画面イメージ



【図12】

開発工数見積ロジックの例

①<準備>

案件番号	A001	A002	A003	A004
修正対象テーブル項目数	10	15	5	10
修正画面数	2	1	0	0
FP[千円]-実績	250	300	150	200

回帰分析することで、
 $FP_{見積} = 80.6 + 12.8 \times \text{【修正対象テーブル項目数】} + 22.2 \times \text{【修正画面数】}$ [千円] ...式(2)
 が得られる。

②a <利用方法1>

案件番号	A101
修正対象テーブル項目数	2
修正画面数	10
FP[千円]	???

案件A101が来たときにかかる工数を、式(2)を用いて予測すると、
 $FP_{見積} = 80.6 + 12.8 \times 2 + 22.2 \times 10 = 328.2$ [千円]

②b <利用方法2>

案件番号	A102
修正対象テーブル項目数	???
修正画面数	???
FP[千円]	???

案件A102が来たときにかかる工数を、A002と似ている案件でかつ機能が若干複雑になりそうなので120%くらいの作業量とすると、
 $FP_{見積} = 300 \text{ [千円]} \times 120\% = 360 \text{ [千円]}$

③ <フィードバック>

案件番号	A101
修正対象テーブル項目数	2
修正画面数	10
FP[千円]-実績	180

案件A101来たときにかかった実績工数も算出式のINPUTとして利用する。
 $算出式を再び回帰解析を利用して作成すると、$
 $FP_{見積} = 711.3 + 14.7 \times \text{【修正対象テーブル項目数】} + 8.2 \times \text{【修正画面数】}$ [千円]
 ...式(2)

【 図 1 3 】

開発期間見積ロジック

①<準備> 下記式の当てはめ、回帰分析にて個人能力を算出する

$$FP = \sum_{i=1}^n \frac{\text{社員iのスキル値} \times \text{社員iの作業工数}}{\text{社員iの作業期間}} \dots \text{式(3)}$$

INPUT

OUTPUT

- ・FP: 過去の1案件あたりの特定業務・特定工程あたりの工数(千円)
- ・【社員iの作業工数】: 社員が1案件・特定業務・特定工程に費やした工数
- ・【社員iのスキル値】: 社員のスキルを係数化した値

②<利用方法> 開発工数見積ロジックで算出した「FP見積」を、対応する社員のスキル値で割ることで、見積作業期間を算出する

$$\text{【見積日数】} = \sum_{i=1}^n \frac{\text{見積FP}}{\text{社員iのスキル値}} \times \text{社員iの作業比率} \dots \text{式(4)}$$

INPUT

OUTPUT

- ・FP見積
- ・【社員iのスキル値】: 社員のスキルを係数化した値
- ・【社員iの作業比率】: 社員がその案件に力を入れる割合(0~100%)
- ・【見積日数】: 見積作業期間

③<フィードバック> 案件リリース後に、FP、各種要素をINPUTとして、FP算出式(3)の精度を上げる

INPUT

OUTPUT

- ・FP: 過去1案件あたりの総工数(千円)
- ・【社員iの作業工数】: 社員が1案件に費やした工数
- ・【社員iのスキル値】: 社員のスキルを係数化した値

【 図 1 4 】

開発期間見積ロジックの例

①<準備>

INPUT: 各案件における各担当者の作業工数

案件番号	A201	A202	A203	A204	A205
社員1工数(人日)	10	0	30	0	5
社員2工数(人日)	20	10	0	0	0
社員3工数(人日)	0	10	5	5	5
社員4工数(人日)	0	10	0	5	0
FP[千円]	100	200	200	150	50

OUTPUT: 各社員の能力

社員1能力(千円/人日)	6.4
社員2能力(千円/人日)	1.3
社員3能力(千円/人日)	2.7
社員4能力(千円/人日)	18.3

②<利用方法>

FP=200(千円)となるような案件

社員2と社員3の2人体制で作業するとした場合

$$FP = \frac{200}{1.3 + 2.7} = 50[\text{日}]$$

社員1のみの作業体制の場合

$$FP = \frac{200}{6.4} \approx 32[\text{日間}]$$