

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6209985号  
(P6209985)

(45) 発行日 平成29年10月11日 (2017.10.11)

(24) 登録日 平成29年9月22日 (2017.9.22)

(51) Int. Cl.		F I			
<b>G 0 6 F</b>	<b>9/44</b>	<b>(2006.01)</b>	<b>G 0 6 F</b>	<b>9/06</b>	<b>6 2 0 A</b>
<b>G 0 6 F</b>	<b>11/36</b>	<b>(2006.01)</b>	<b>G 0 6 F</b>	<b>11/36</b>	<b>1 6 8</b>

請求項の数 8 (全 19 頁)

(21) 出願番号	特願2014-23881 (P2014-23881)	(73) 特許権者	000005223
(22) 出願日	平成26年2月10日 (2014.2.10)		富士通株式会社
(65) 公開番号	特開2015-152979 (P2015-152979A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成27年8月24日 (2015.8.24)	(74) 代理人	100074099
審査請求日	平成28年10月4日 (2016.10.4)		弁理士 大菅 義之
		(74) 代理人	100133570
			弁理士 ▲徳▼永 民雄
		(72) 発明者	菊地 英晴
			大阪府大阪市中央区城見2丁目2番6号
			株式会社富士通システムズ・ウエスト内
		審査官	坂庭 剛史

最終頁に続く

(54) 【発明の名称】 製品設計支援プログラム、製品設計支援方法および製品設計支援装置

(57) 【特許請求の範囲】

【請求項 1】

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援プログラムであって、

前記要件を定義した第1の書類と、当該第1の書類から関連付けられ前記要件の機能を定義した第2の書類と、当該第2の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部を参照し、

前記第1の書類に関連付けられた全ての前記第2の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する、

処理をコンピュータに実行させる製品設計支援プログラム。

【請求項 2】

前記要件を複数含み、前記要件ごとに前記難易度を算出する、

処理を前記コンピュータに実行させる請求項1記載の製品設計支援プログラム。

【請求項 3】

前記第2の書類は、前記要件を具体化または詳細化した第1の設計書と、当該第1の設計書を具体化または詳細化した第2の設計書と、当該第2の設計書を具体化または詳細化したソースコードと、を含み、

前記第2の書類に基づいて、前記難易度を算出する、

処理を前記コンピュータに実行させる請求項1または2記載の製品設計支援プログラム

。

【請求項 4】

前記テスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果と前記テスト書類に関連付けられる第 2 の書類の機能を実現する開発量とに基づいて、前記難易度を算出する、

処理を前記コンピュータに実行させる請求項 1 乃至 3 のうち何れか 1 項に記載の製品設計支援プログラム。

【請求項 5】

前記第 2 の書類ごとに、各第 2 の書類に関連付けられたテスト書類に含まれるテスト項目のテスト結果の障害数を前記テスト書類に含まれるテスト項目の数で除算して、前記第 2 の書類ごとの障害率を演算し、

前記第 2 の書類ごとに、前記第 2 の書類の機能を実現するソースコードのステップ数を前記障害率に乘算し、

前記第 2 の書類ごとに乘算された前記障害率を合算し、

合算した前記障害率を全ての前記ソースコードのステップ数の合計値で除算して、前記難易度を演算する、

処理を前記コンピュータに実行させる請求項 4 記載の製品設計支援プログラム。

【請求項 6】

前記テスト書類に含まれる前記テスト項目の数を合算した合計値と前記テスト結果に含まれる障害の合計値とに基づいて、前記難易度を算出する、

処理を前記コンピュータに実行させる請求項 1 乃至 3 のうち何れか 1 項に記載の製品設計支援プログラム。

【請求項 7】

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援方法であって、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部を参照し、

前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する、

製品設計支援方法。

【請求項 8】

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援装置であって、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部と、

前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する演算部と、

を備える製品設計支援装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、製品設計支援プログラム、製品設計支援方法および製品設計支援装置に関する。

【背景技術】

【0002】

ソフトウェアが制御する製品の開発を行うときに、各種のテストが行われる。各種のテ

10

20

30

40

50

ストの一例として、ソフトウェア単体のテストがある。また、各種のテストの一例として、ソフトウェアと電子回路とを組み合わせたシステムテストや制御される部品を実装した製品を対象とした実機テスト等もある。

#### 【 0 0 0 3 】

関連する技術として、設計書のレビューを行ったときのレビュー結果、およびプログラムコードのテスト結果を関連付けて記憶する技術が提案されている。また、各業務の総試験項目数を、設定したウェイトに従って割り振り、サブシステムごとの目標値を算出する技術が提案されている。また、ソフトウェア開発の過程でうみだされる各工程の成果物の評価方法であって、ソフトウェア開発における複数の工程の成果物同士の間の依存関係の状態を評価する技術が提案されている（例えば、特許文献 1 ～ 3 参照）。

10

#### 【 先行技術文献 】

#### 【 特許文献 】

#### 【 0 0 0 4 】

【 特許文献 1 】 特開 2 0 0 7 - 2 4 1 3 4 7 号 公 報

【 特許文献 2 】 特開 2 0 0 1 - 3 3 7 8 4 6 号 公 報

【 特許文献 3 】 特開 2 0 1 3 - 0 1 5 9 5 8 号 公 報

#### 【 発明の概要 】

#### 【 発明が解決しようとする課題 】

#### 【 0 0 0 5 】

ソフトウェアが制御する製品の開発を行うときには、開発対象の製品が満たす条件が定義された要件定義が行われる。要件定義は、例えば要件定義書等の書類に記述される。要件定義は、抽象的な定義であり、要件を詳細化または具体化することにより、各種の機能が定義される。機能は、設計書等の書類に記述される。

20

#### 【 0 0 0 6 】

要件や機能には、実現するための難易度がある。例えば、要件や機能が要求する条件が複雑な場合には、難易度が高くなる。要件を具体化または詳細化した機能のレベルでは、機能のテストを行ったテスト結果に基づいて、機能を実現する難易度を定量的に導き出すことは難しくない。

#### 【 0 0 0 7 】

一方、上述したように、要件は、開発対象の製品が満たす条件が抽象的に定義されたものであり、要件を実現する難易度を定量的に導き出すことは難しい。開発対象の製品を設計する上で、設計者は要件の実現性の難易度を定量的に評価できることが好ましい。

30

#### 【 0 0 0 8 】

1 つの側面では、本発明は、製品に必要な要件を実現する難易度を定量的に算出することで、製品の設計の支援を行うことを目的とする。

#### 【 課題を解決するための手段 】

#### 【 0 0 0 9 】

1 つの案として、開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援装置は、記憶部と演算部とを備える。前記記憶部は、前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する。前記演算部は、前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する。

40

#### 【 発明の効果 】

#### 【 0 0 1 0 】

製品に必要な要件を実現する難易度を定量的に算出することで、製品の設計の支援を行うことができる。

#### 【 図面の簡単な説明 】

#### 【 0 0 1 1 】

50

- 【図 1】製品設計支援装置の一例を示すブロック図である。
- 【図 2】製品設計支援装置のハードウェア構成の一例を示す図である。
- 【図 3】各書類のリレーションの一例を示す図（その 1）である。
- 【図 4】各書類のリレーションの一例を示す図（その 2）である。
- 【図 5】実施形態の処理の一例を示すフローチャートである。
- 【図 6】実施形態における要件ごとの難易度を示す表の一例である。
- 【図 7】実施形態における要件ごとの難易度を示すグラフの一例である。
- 【図 8】応用例における処理の一例を示すフローチャートである。
- 【図 9】図 8 の要件ごとの難易度の演算処理の一例を示すフローチャートである。
- 【図 10】応用例における要件ごとの難易度を示す表の一例である。

10

【発明を実施するための形態】

【0012】

<製品設計支援装置の一例>

以下、図面を参照して、実施形態について説明する。図 1 は、実施形態の製品設計支援装置の一例を示している。図 1 において、難易度算出部 1 は、表示部 2 およびデータベース 3 に接続されている。表示部 2 は、難易度算出部 1 が出力する画面を表示する。例えば、表示部 2 にはディスプレイを適用することができる。データベース 3 は、記憶部の一例である。

【0013】

データベース 3 は、種々の書類を電子データで記憶している。以下、書類の電子データを書類データと称する。データベース 3 は、機能要件書と概要機能設計書と詳細機能設計書とソースコードとを記憶している。また、データベース 3 は、実機テスト仕様書とシステムテスト仕様書とユニットテスト仕様書とプログラムテスト仕様書とを記憶している。

20

【0014】

実機テスト仕様書、システムテスト仕様書、ユニットテスト仕様書およびプログラムテスト仕様書は、それぞれテスト結果と対応付けられて、データベース 3 に記憶されている。テスト結果は、対応するテスト仕様書に基づくテストの結果である。テスト仕様書は、テスト書類の一例である。

【0015】

図 1 に示した一例では、各書類データを 1 つのデータベース 3 に記憶しているが、各書類データは異なるデータベースに記憶されていてもよい。また、各書類データのうち、対応付けられているテスト仕様書とテスト結果とは同一のデータベースに記憶され、その他の書類データは異なるデータベースに記憶されてもよい。

30

【0016】

また、図 1 に示した一例では、データベース 3 には機能要件書が複数記憶されている。要件が 1 つの場合、データベース 3 に記憶される機能要件書は 1 つであってもよい。他の書類データも同様であり、同種の書類データについて、複数の書類データがデータベース 3 に記憶されてもよいし、1 つの書類データがデータベース 3 に記憶されてもよい。

【0017】

難易度算出部 1 は、必要性記憶部 4 とデータ処理部 5 と演算部 6 と画面生成部 7 とを備えている。必要性記憶部 4 は、要件の必要性を記憶する。要件には必要性があり、要件によって、必要性は異なる。

40

【0018】

例えば、製品に必須な要件は、必要性が非常に高い。また、製品の商品力向上の要件は、安全上必要な要件よりは必要性が低い、比較的必要性が高い。また、製品の便利機能の要件は、必要性が低い。必要性記憶部 4 は、要件ごとに必要性を数値で記憶している。

【0019】

データ処理部 5 は、データベース 3 に記憶されている各書類データを取得して、各種の処理を行う。演算部 6 は、データ処理部 5 が取得した情報に基づいて、要件の難易度を得るための種々の演算を行う。画面生成部 7 は、要件の難易度を含む画面を生成して、表示

50

部 2 に出力する。表示部 2 は、入力した画面を表示する。

【 0 0 2 0 】

< 難易度算出装置のハードウェア構成の一例 >

次に、難易度算出装置のハードウェア構成の一例について説明する。図 2 の一例に示すように、難易度算出装置は、バス 2 0 に対して、プロセッサ 2 1 と R A M (Random Access Memory) 2 2 と R O M (Read Only Memory) 2 3 と補助記憶装置 2 4 と周辺機器接続部 2 5 と通信インタフェース 2 6 と可搬型記憶装置接続部 2 7 とが接続されている。

【 0 0 2 1 】

プロセッサ 2 1 は C P U (Central Processing Unit) のような任意の処理回路である。プロセッサ 2 1 は R A M 2 2 に展開されたプログラムを実行する。R O M 2 3 は R A M 2 2 に展開されるプログラムを記憶する不揮発性の記憶装置である。

10

【 0 0 2 2 】

補助記憶装置 2 4 は、種々の情報を記憶する記憶装置であり、例えばハードディスクドライブや半導体メモリ等を補助記憶装置 2 4 に適用することができる。上述したデータベース 3 は、補助記憶装置 2 4 に記憶されてもよい。周辺機器接続部 2 5 は、ディスプレイ 2 8 および入力デバイス 2 9 を接続する。

【 0 0 2 3 】

通信インタフェース 2 6 は、外部との通信を行うときのインタフェースである。例えば、データベース 3 は L A N (Local Area Network) 経由で接続される外部の記憶装置に記憶されていてよい。この場合、プロセッサ 2 1 は、通信インタフェース 2 6 を介して、データベース 3 を参照してもよい。

20

【 0 0 2 4 】

可搬型記憶装置接続部 2 7 は、可搬型記憶装置 3 0 と接続可能に設けられている。可搬型記憶装置 3 0 としては、可搬型のメモリや光学式ディスク（例えば、C D (Compact Disk) や D V D (Digital Video Disk) 等）を適用することができる。実施形態の処理を行うプログラムは、コンピュータ読み取り可能な記録媒体に記録されていてよい。この場合、記録媒体に可搬型記憶装置 3 0 を適用することができる。

【 0 0 2 5 】

R A M 2 2 、R O M 2 3 および補助記憶装置 2 4 は、何れもコンピュータ読み取り可能な有形の記憶媒体の一例である。これらの有形の記憶媒体は、信号搬送波のような一時的な媒体ではない。

30

【 0 0 2 6 】

図 1 で一例として示した難易度算出部 1 のデータ処理部 5 と演算部 6 と画面生成部 7 とは、例えばプロセッサ 2 1 で実行されるアプリケーションにより実現されてもよい。必要性記憶部 4 は、R A M 2 2 に記憶されてもよい。

【 0 0 2 7 】

< 各書類データの関連の一例 >

図 3 および図 4 は、各書類データの関連の一例を示している。機能要件は、例えば、ソフトウェアで制御する製品を開発する上で、製品が満たす条件が定義された要件のうち機能に関する要件を含む。機能要件は、単に要件と称されることもある。

40

【 0 0 2 8 】

実施形態では、要件は機能要件書で定義される。機能要件書は第 1 の書類の一例である。図 3 の一例では、機能要件書 A と機能要件書 B と機能要件書 C とを示しているが、機能要件書の数には限定されない。

【 0 0 2 9 】

図 1 の一例に示したように、データベース 3 は、実機テスト仕様書、システムテスト仕様書、ユニットテスト仕様書およびプログラムテスト仕様書を記憶している。これらの各テスト仕様書にはそれぞれテスト結果が対応付けられている。

【 0 0 3 0 】

テスト結果は、テスト仕様書のテストに基づいた結果である。テスト仕様書は、1 また

50

は複数のテスト項目を含んでいる。実施形態では、テスト仕様書は複数のテスト項目を含んでいるものとする。テスト結果はテスト項目に基づくテストの結果であり、正常または障害（不良であってもよい）を含む。テスト仕様書は、他に、テストの内容やテストの実行環境等の情報を含んでもよい。

【0031】

図3の一例では、実機テスト仕様書は1つを示しているが、実機テスト仕様書は複数であってもよい。実機テスト仕様書は、実際の製品にソフトウェアを実装したときのテストに関する仕様書である。実機テスト仕様書Aにはテスト結果が対応付けられている。

【0032】

図3の一例では、6つの概要機能設計書A～Fを示している。概要機能設計書は、機能要件書で定義される機能の概要を示す設計書である。概要機能設計書は、第1の設計書の一例である。概要機能設計書の数、6つには限定されない。システムテスト仕様書A～Fは、概要機能設計書A～Fに対応している。

【0033】

概要機能設計書は、上述したように、機能要件書で定義される機能の概要を示す設計書である。概要機能設計書に関連付けられるシステムテスト仕様書は、製品のシステム全体のテストになる。システムテスト仕様書A～Fにはそれぞれテスト結果が対応付けられている。テスト仕様書に含まれるテスト項目の数は1つであってもよく、その場合は、テスト結果も1つになる。

【0034】

図3および図4の一例では、書類データ間の関連を実線の矢印または破線の矢印で示している。実線の矢印は、第1のリレーションを示している。第1のリレーションは、詳細化・具体化リレーションとも称される。機能要件書には、開発対象の製品が満たす抽象的な条件が記述される。従って、機能要件書に第1のリレーションが設定されている概要機能設計書は、当該機能要件書を詳細化または具体化した設計書になる。

【0035】

図3の一例では、機能要件書Aと概要機能設計書Aおよび概要機能設計書Cとの間に第1のリレーションが設定されている。また、機能要件書Bと概要機能設計書Bおよび概要機能設計書Eとの間に第1のリレーションが設定されている。また、機能要件書Cと概要機能設計書Dおよび概要機能設計書Fとの間に第1のリレーションが設定されている。

【0036】

また、図3の一例において、破線で示す矢印は第2のリレーションである。第2のリレーションは検証リレーションとも称される。第2のリレーションは、テスト仕様書に設定されるリレーションである。第2のリレーションは、テスト仕様書によりテストされる対象となる書類データとの関連を示している。

【0037】

図3の一例では、概要機能設計書A～Fとシステムテスト仕様書A～Fとの間に第2のリレーションが設定されている。また、機能要件書Cと実機テスト仕様書Aとの間にも第2のリレーションが設定されている。

【0038】

第1のリレーションおよび第2のリレーションは、書類データの間に予め設定されているものである。第1のリレーションおよび第2のリレーションは、任意の手法で設定されてよい。例えば、機能要件書および概要機能設計書が共通するファイル名を有していれば、両者の間に第1のリレーションを設定することができる。また、機能要件書に含まれるデータと概要機能設計書に含まれるデータとの間に共通性があれば、両者の間に第1のリレーションを設定することができる。

【0039】

以上のように、第1のリレーションおよび第2のリレーションは予め設定することができる。予め設定された第1のリレーションおよび第2のリレーションは、例えば、RAM 22や補助記憶装置24等に記憶されてもよい。

10

20

30

40

50

## 【 0 0 4 0 】

図 3 および図 4 の一例で示すように、概要機能設計書 F と詳細機能設計書 A ~ F との間には第 1 のリレーションが設定されている。実施形態の概要機能設計書は、製品のシステム全体の概要の設計書である。概要機能設計書とシステムテスト仕様書との間に第 2 のリレーションが設定される。

## 【 0 0 4 1 】

実施形態の詳細機能設計書は、概要機能設計書を詳細化または具体化した設計書である。詳細機能設計書は、第 2 の設計書の一例である。詳細機能設計書は、個々の部品やモジュール等の条件を定義している。図 3 および図 4 に示した一例では、概要機能設計書 F を詳細化または具体化した設計書が詳細機能設計書 A ~ F になる。

10

## 【 0 0 4 2 】

概要機能設計書 F は複数の設計書に詳細化または具体化することができる。一例として、概要機能設計書 F は詳細機能設計書 A ~ F に詳細化または具体化されるものとする。従って、概要機能設計書 F と詳細機能設計書 A ~ F との間に第 1 のリレーションが設定される。

## 【 0 0 4 3 】

上述したように、1 つの概要機能設計書は複数の詳細機能設計書と第 1 のリレーションが設定されてもよいし、1 つの概要機能設計書は 1 つの詳細機能設計書と第 1 のリレーションが設定されてもよい。また、概要機能設計書 A ~ E についても、それぞれ 1 または複数の詳細機能設計書と第 1 のリレーションが設定されていてもよい。

20

## 【 0 0 4 4 】

詳細機能設計書 A ~ F は、ユニットテスト仕様書 A ~ F との間に第 2 のリレーションが設定されている。ユニットテスト仕様書は、個々の部品やモジュール等（以下、ユニットと称する）のテストの内容を定義した仕様書である。ユニットテスト仕様書 A ~ F は、それぞれユニットテスト仕様書のテストを行ったテスト結果と対応付けられている。

## 【 0 0 4 5 】

図 4 の一例で示すように、詳細機能設計書 F とソースコード A ~ C との間に第 1 のリレーションが設定されている。実施形態のソースコードは、詳細機能設計書で定義される条件をソースコードレベルまで具体化または詳細化したものである。ソースコードとプログラムテスト仕様書との間に第 2 のリレーションが設定される。

30

## 【 0 0 4 6 】

詳細機能設計書 F は複数のソースコードに具体化または詳細化することができる。一例として、詳細機能設計書 F はソースコード A ~ C に具体化または詳細化されるものとする。従って、詳細機能設計書 F とソースコード A ~ C との間に第 1 のリレーションが設定される。

## 【 0 0 4 7 】

上述したように、1 つの詳細機能設計書は複数のソースコードと第 1 のリレーションが設定されてもよいし、1 つの詳細機能設計書は 1 つのソースコードと第 1 のリレーションが設定されてもよい。また、概要機能設計書 A ~ E についても、それぞれ 1 または複数のソースコードと第 1 のリレーションが設定されていてもよい。

40

## 【 0 0 4 8 】

ソースコード A ~ C は、プログラムテスト仕様書 A ~ C と第 2 のリレーションが設定されている。プログラムテスト仕様書は、ソースコードのテストの内容を定義した仕様書である。プログラムテスト仕様書 A ~ C は、それぞれプログラムテスト仕様書のテストを行ったテスト結果と関連付けられている。テスト結果は、例えば、正常または障害として示される。

## 【 0 0 4 9 】

概要機能設計書、詳細機能設計書およびソースコードは、第 2 の書類の一例である。第 2 の書類は、第 1 の書類で定義される要件の機能を定義する書類である。従って、第 2 の書類は、要件の機能を定義する書類であれば、概要機能設計書、詳細機能設計書およびソ

50

ースコードには限定されない。

【 0 0 5 0 】

< 実施形態の処理の一例 >

次に、実施形態の処理の一例について説明する。最初に、データ処理部 5 は、データベース 3 から難易度の演算を行う要件についての第 1 の書類を取得し、第 1 の書類をリスト L 1 にセットする (ステップ S 1 )。実施形態では、第 1 の書類は、機能要件書であるため、リスト L 1 には、取得された機能要件書がセットされる。

【 0 0 5 1 】

データベース 3 には、複数の機能要件書が記憶されている。データ処理部 5 は、データベース 3 に記憶されている複数の機能要件書から対象となる一部または全部の機能要件書を取得する。データ処理部 5 は、取得した機能要件書をリスト L 1 にセットする。リスト L 1 は R A M 2 2 により実現されてもよい。この場合、データ処理部 5 は、取得した機能要件書を R A M 2 2 に記憶する。

【 0 0 5 2 】

次に、データ処理部 5 は、リスト L 1 にセットした第 1 の書類の数の分、ループ処理を開始する (ステップ S 2 )。データ処理部 5 は、リスト L 1 から 1 つの第 1 の書類を取得し、取得した第 1 の書類に設定されている第 1 のリレーションを再帰的に辿る (ステップ S 3 )。

【 0 0 5 3 】

そして、データ処理部 5 は、第 1 のリレーションを辿ったときに得られる第 2 の書類をリスト L 2 にセットする (ステップ S 4 )。実施形態では、第 2 の書類は、概要機能要件書、詳細機能要件書またはソースコードである。データ処理部 5 は、第 1 のリレーションを辿り、リスト L 2 にこれらの第 2 の書類をセットする。リスト L 2 は、R A M 2 2 に記憶されてもよい。

【 0 0 5 4 】

実施形態では、概要機能設計書は、1 または複数の詳細機能設計書との間に第 1 のリレーションが設定されている。データ処理部 5 は、リスト L 2 にセットされた全ての概要機能設計書について、第 1 のリレーションが設定されている詳細機能設計書を取得する。データ処理部 5 は、取得した詳細機能設計書をリスト L 2 にセットする。

【 0 0 5 5 】

実施形態では、詳細機能設計書は、1 または複数のソースコードとの間に第 1 のリレーションが設定されている。データ処理部 5 は、リスト L 2 にセットされた全ての詳細機能設計書について、第 1 のリレーションが設定されているソースコードを取得する。データ処理部 5 は、取得したソースコードをリスト L 2 にセットする。

【 0 0 5 6 】

以上により、データ処理部 5 は、リスト L 1 から取得した第 1 の書類を起点として第 1 のリレーションが設定されている全ての第 2 の書類をリスト L 2 にセットする。実施形態では、リスト L 1 には機能要件書がセットされる。また、リスト L 2 には、概要機能設計書と詳細機能設計書とソースコードとのうち何れか 1 つまたは全部がセットされる。

【 0 0 5 7 】

次に、データ処理部 5 は、リスト L 2 にセットされている第 2 の書類の数の分、ループ処理を開始する (ステップ S 5 )。概要機能設計書は、システムテスト仕様書との間に第 2 のリレーションが設定されている。詳細機能設計書は、ユニットテスト仕様書との間に第 2 のリレーションが設定されている。ソースコードは、プログラムテスト仕様書との間に第 2 のリレーションが設定されている。

【 0 0 5 8 】

データ処理部 5 は、リスト L 2 にセットされている第 2 の書類を 1 つ取得する。そして、データ処理部 5 は、第 2 の書類に設定されている第 2 のリレーションを辿ることにより、テスト仕様書に含まれるテスト項目を取得する (ステップ S 6 )。実施形態では、データ処理部 5 は、システムテスト仕様書、ユニットテスト仕様書およびプログラムテスト仕

10

20

30

40

50



様書のそれぞれのテスト仕様書に含まれる全てのテスト項目を取得する。

【 0 0 5 9 】

データ処理部 5 は、第 2 の書類との間に第 2 のリレーションが設定されているテスト仕様書にテスト項目が含まれているか否かを判定する（ステップ S 7）。テスト仕様書にテスト項目が含まれていなければ（ステップ S 7 で N O）、ステップ S 8 およびステップ S 9 の処理は行われない。

【 0 0 6 0 】

一方、テスト仕様書にテスト項目が含まれていれば（ステップ S 7 で Y E S）、データ処理部 5 は、テスト項目数と障害数とを取得する（ステップ S 8）。システムテスト仕様書、ユニットテスト仕様書およびプログラムテスト仕様書には、それぞれテスト結果が対応付けられている。

10

【 0 0 6 1 】

テスト結果は、テスト項目数と障害数とを含む。テスト項目数は、テスト仕様書に含まれるテスト項目の数であり、障害数はテスト項目のテストを行った結果、障害が発生した数である。

【 0 0 6 2 】

演算部 6 は、第 2 の書類に関連付けられているテスト結果のテスト項目数および障害数を要件ごとに加算する（ステップ S 9）。実施形態では、演算部 6 は、機能要件書ごとに、テスト項目数および障害数を要件ごとに加算する。テスト項目数および障害数の初期値はゼロである。

20

【 0 0 6 3 】

データ処理部 5 および演算部 6 は、リスト L 2 から取得した第 2 の書類についてステップ S 6 ～ステップ S 9 の処理を終了すると、リスト L 2 にセットされている次の第 2 の書類について、ステップ S 6 ～ステップ S 9 の処理を開始する。

【 0 0 6 4 】

従って、演算部 6 は、リスト L 2 にセットされている全ての第 2 の書類について、テスト項目があれば、テスト項目数および障害数を順次加算する。よって、1 つの第 1 の書類から第 1 のリレーションに関連付けられている全ての第 2 の書類について、テスト項目数の合計値および障害数の合計値が得られる。つまり、1 つの要件についての、テスト項目数の合計値および障害数の合計値が得られる。演算部 6 は、障害数の合計値をテスト項目数の合計値で除算することで、難易度の演算を行う（ステップ S 1 1）。

30

【 0 0 6 5 】

テスト項目数の合計値に対する障害数の合計値の割合が大きければ、要件に対する障害が大きいことになる。従って、要件を実現する難易度は高くなる。一方、テスト項目数の合計値に対する障害数の割合が小さければ、要件に対する障害が小さいことになる。従って、要件を実現する難易度は低くなる。

【 0 0 6 6 】

ステップ S 3 ～ステップ S 1 1 の処理は、1 つの要件についての処理である。ステップ S 1 の処理で、データ処理部 5 は 1 つの第 1 の書類を取得している。リスト L 1 に他の第 1 の書類があれば、データ処理部 5 および演算部 6 は、当該第 1 の書類について、ステップ S 3 ～ステップ S 1 1 の処理を行う。

40

【 0 0 6 7 】

従って、リスト L 1 にセットされている全ての第 1 の書類について、ステップ S 3 ～ステップ S 1 1 の処理を行うことで、演算部 6 は、全ての第 1 の書類についての難易度を演算することができる。リスト L 1 にセットされている全ての第 1 の書類について、ステップ S 3 ～ステップ S 1 1 の処理が終了したときに、ステップ S 2 ～ステップ S 1 2 のループ処理は終了する。

【 0 0 6 8 】

以上の処理により、演算部 6 は、第 1 の書類ごと、つまり要件ごとに難易度を演算する。これにより、抽象的な概念を定義している第 1 の書類の要件について実現の難易度を定

50

量的な数値を用いて評価することができる。

【 0 0 6 9 】

図 6 は、要件ごとの難易度を定量的に表示する表の一例を示している。図 6 において、要件 A および要件 B の難易度を数値で示している。従って、図 5 のステップ S 1 でリスト L 1 にセットされる第 1 の書類は要件 A および要件 B の書類データである。

【 0 0 7 0 】

また、機能 1、機能 2、機能 3 および機能 5 は、第 2 の書類で定義される機能の一例である。図 6 の一例では、要件 A は、機能 1、機能 3 および機能 5 との間で第 1 のリレーションが設定されている。また、要件 B は、機能 1、機能 2 および機能 5 との間で第 1 のリレーションが設定されている。

10

【 0 0 7 1 】

従って、データ処理部 5 は、要件 A について、第 1 のリレーションを再帰的に辿ることにより、機能 1、機能 3 および機能 5 のテスト結果からテスト項目数および障害数を得る。図 6 の一例では、機能 1 は、テスト項目数が「 2 0 」であり、障害数が「 2 」である。機能 3 は、テスト項目数が「 3 5 」であり、障害数が「 1 0 」である。機能 5 は、テスト項目数が「 4 8 」であり、障害数が「 3 」である。

【 0 0 7 2 】

演算部 6 は、図 5 のフローに示されるように、テスト項目数および障害数を加算していく。これにより、要件 A を実現する機能 1、機能 3 および機能 5 のテスト項目数の合計値および障害数の合計値が得られる。図 6 に示す一例の場合、要件 A のテスト項目数の合計値は「 1 0 3 ( = 2 0 + 3 5 + 4 8 ) 」になる。

20

【 0 0 7 3 】

一方、障害数の合計値は「 1 5 ( = 2 + 1 0 + 3 ) 」になる。演算部 6 は、要件 A の障害数の合計値をテスト項目数の合計値で除算する。これにより、演算部 6 は「 0 . 1 4 6 ( = 1 5 / 1 0 3 ) 」という数値を得る。なお、以下において、各数値は、小数点 4 桁を四捨五入した例を示している。

【 0 0 7 4 】

図 5 のフローで説明したように、要件 A と要件 B とは別個に難易度の演算が行われる。データ処理部 5 は、要件 B について、第 1 のリレーションを再帰的に辿ることにより、機能 1、機能 2 および機能 5 のテスト結果からテスト項目数および障害数を得る。図 6 の一例では、機能 1 および機能 5 は、上述した値である。機能 2 は、テスト項目数が「 4 0 」であり、障害数が「 4 」である。

30

【 0 0 7 5 】

図 6 に示す一例の場合、要件 B のテスト項目数の合計値は「 1 0 8 ( = 2 0 + 4 0 + 4 8 ) 」になる。一方、障害数の合計値は「 9 ( = 2 + 4 + 3 ) 」になる。よって、演算部 6 は、要件 B の障害数の合計値をテスト項目数の合計値で除算して「 0 . 0 8 3 ( = 9 / 1 0 8 ) 」を得る。

【 0 0 7 6 】

要件 A を実現するための難易度「 0 . 1 4 6 」および要件 B を実現するための難易度「 0 . 0 8 3 」は定量的な数値である。難易度の数値が高いほど、要件を実現することが困難になり、難易度の数値が低いほど、要件を実現することが容易になる。従って、画面生成部 7 は、図 6 に示すような画面を表示部 2 に出力する。

40

【 0 0 7 7 】

これにより、表示部 2 を視認する者（以下、設計者とする）は、要件ごとの難易度を定量的に評価することができる。設計者は、定量的に表示されている難易度に基づいて、要件を実現するか否かを評価することができる。このため、製品の設計を支援することができるようになる。

【 0 0 7 8 】

図 6 の画面例は、必要性の項目を有している。上述したように、製品の要件には必要性のレベルに差がある。例えば、上述した、製品に必須な要件と商品力向上の要件と便利機

50

能の要件とはそれぞれ必要性のレベルに差異がある。

【 0 0 7 9 】

そこで、各要件の必要性が数値化されて必要性記憶部 4 に記憶されている。必要性記憶部 4 は、要件ごとに数値化された必要性を記憶している。この必要性の値は、要件が開発対象の製品にどの程度必要であるかの度合いを示している。例えば、ある要件についての必要性の統計に基づいて、必要性の値が決定されてもよい。図 6 の画面例では、要件 A の必要性は「 0 . 1 5 」であり、要件 B の必要性は「 0 . 0 5 」である。従って、必要性は、要件 B より要件 A の方が高い。

【 0 0 8 0 】

表示部 2 に、要件ごとの難易度を表示することで、設計者は定量的に難易度を認識することができる。このとき、難易度と必要性とを比較できることが好ましい。これにより、設計者は、必要性に対する難易度を、要件を製品に実装するか否かを評価するときの指標とすることができる。

【 0 0 8 1 】

図 6 の画面例では、要件 A の難易度は「 0 . 1 4 6 」であり、必要性は「 0 . 1 5 」である。よって、設計者は、要件 A が必要性に見合った難易度を有していることを認識することができる。一方、図 6 の画面例において、要件 B の難易度は「 0 . 0 8 3 」であり、必要性は「 0 . 0 5 」である。よって、設計者は、要件 B が必要性に対して高い難易度を有していることを認識できる。

【 0 0 8 2 】

図 6 の画面例において、難易度の箇所別に別の数値を表示している。この数値は、難易度を必要性で除算した値である。つまり、この数値は、難易度に対する必要性の割合を示している。要件 A の場合は「 0 . 9 7 1 ( = 0 . 1 4 5 6 / 0 . 1 5 ) 」であり、要件 B の場合は「 1 . 6 6 7 ( = 0 . 0 8 3 3 / 0 . 0 5 ) 」である。

【 0 0 8 3 】

図 6 の画面例では、難易度に対する必要性の割合を示している。例えば、要件 A の数値は「 0 . 9 7 1 」になっており、その数値は「 1 」以下になっている。これにより、設計者は、括弧内の数値を視認することで、難易度と必要性とが見合っていることを認識できる。

【 0 0 8 4 】

一方、要件 B の数値は「 1 . 6 6 7 」になっており、その数値は「 1 」に対して比較的高い。設計者は、括弧内の数値を視認することで、必要性よりも難易度が比較的高いことを認識することができる。つまり、設計者は、括弧内の 1 つの数値に基づいて、難易度と必要性との関係を認識することができる。

【 0 0 8 5 】

図 7 の画面例は、グラフ形式で難易度を示している。上述したように、要件ごとの難易度は定量的な数値で得られている。従って、画面生成部 7 は、難易度をグラフ表示することができる。図 7 の画面例では、横軸を必要性とし、縦軸を難易度として、要件 R 1 ~ 要件 R 4 の難易度を示している。要件 R 1 は必要性が低く、且つ難易度も低い。従って、設計者は、要件 R 1 は、開発対象の製品に要件 R 1 を実装しても、必要性に対して難易度が見合うことを認識することができる。

【 0 0 8 6 】

一方、要件 R 2 は必要性が低く、且つ難易度が非常に高い。従って、設計者は、要件 R 2 については、必要性に対して難易度が見合わないことを認識することができる。これにより、設計者は、開発対象の製品から要件 R 2 を除外することを検討する手がかりを得ることができる。

【 0 0 8 7 】

要件 R 3 は、必要性が高く、且つ難易度も高い。従って、設計者は、要件 R 3 は、開発対象の製品に要件 R 3 を実装しても、必要性に対して難易度が見合うことを認識することができる。

10

20

30

40

50

## 【 0 0 8 8 】

一方、要件 R 4 は、必要性が高く、且つ難易度が非常に高い。従って、設計者は、要件 R 4 について、必要性に対して難易度が高いことを認識できる。ここで、必要性が高い要件は、開発対象の製品に実装が要求される要件である場合が多い。例えば、製品に必須の要件は難易度にかかわらず、要件を実現することが要求される。

## 【 0 0 8 9 】

そこで、設計者は、要件 R 4 の必要性が高いことから、要件 R 4 について、必要性に対して難易度が見合わなくても、要件 R 4 を開発対象の製品に実装することを判断することができる。

## 【 0 0 9 0 】

図 7 の画面例では、画面生成部 7 は、定量的な数値として得られた難易度に基づいて、グラフ表示をする画面を生成し、表示部 2 にグラフを表示している。よって、設計者は、要件ごとの難易度を視覚的に認識することができる。また、設計者は、必要性によっては、難易度にかかわらず、開発対象の製品に実装することを認識することができる。

## 【 0 0 9 1 】

< 応用例の処理の一例 >

次に、応用例について説明する。応用例では、演算部 6 は、開発量を加味した難易度の演算を行っている。図 8 のフローは、応用例の処理の一例を示している。図 8 のフローは、図 5 のフローに対してステップ S 8 およびステップ S 1 1 が変更されている。その他のステップは、図 5 のフローと同じである。

## 【 0 0 9 2 】

開発量は、第 2 の書類の機能を実現するために要する工数であり、一例として、開発量は、ソースコードのステップの総数とする。第 2 の書類が概要機能設計書または詳細機能設計書の場合は、これらの設計書を起点として第 1 のリレーションが設定されている 1 または複数のソースコードのステップの総数が開発量になる。

## 【 0 0 9 3 】

図 5 のステップ S 8 と同様に、データ処理部 5 は、テスト項目数と障害数とを取得する。また、データ処理部 5 は、テスト項目を含むテスト仕様書がプログラムテスト仕様書の場合は、当該プログラムテスト仕様書と第 2 のリレーションが設定されているソースコードのステップ総数を取得する（ステップ S 2 1 ）。

## 【 0 0 9 4 】

データ処理部 5 は、テスト項目を含むテスト仕様書が概要機能設計書または詳細機能設計書の場合は、第 1 のリレーションを辿って、1 または複数のソースコードを取得する。そして、データ処理部 5 は、取得した 1 または複数のソースコードのステップ総数を取得する。

## 【 0 0 9 5 】

ステップ S 5 ～ステップ S 1 0 のループ処理が終了すると、要件ごとの難易度の演算処理が行われる（ステップ S 2 2 ）。図 9 は、ステップ S 2 2 の要件ごとの難易度の演算処理の一例を示している。また、図 1 0 は、要件ごとの難易度の表の一例を示している。なお、図 1 0 の表には、開発量の項目が追加されている。

## 【 0 0 9 6 】

図 9 のフローチャートに示されるように、演算部 6 は、機能ごとに障害率を演算する（ステップ S 3 1 ）。演算部 6 は、要件 A および要件 B に含まれる機能 1、機能 2、機能 3 および機能 5 の障害率を演算する。応用例では、障害率は、障害数をテスト項目数で除算した値になる。

## 【 0 0 9 7 】

従って、機能 1 の障害率は「 $0.1 (= 2 / 20)$ 」になる。機能 2 の障害率は「 $0.1 (= 4 / 40)$ 」になる。機能 3 の障害率は「 $0.286 (= 10 / 35)$ 」になる。機能 5 の障害率は「 $0.0625 (= 3 / 48)$ 」になる。

## 【 0 0 9 8 】

10

20

30

40

50

機能 1 および機能 5 は、要件 A と要件 B とで重複している。この場合、要件ごとに機能 1 および機能 5 を演算してもよいが、重複した演算を行うことになる。従って、要件 A で演算した機能 1 および機能 5 の障害率は、要件 B では演算を行わず、その結果を利用してもよい。

#### 【 0 0 9 9 】

次に、演算部 6 は、機能ごとの障害率に開発量を乗算する（ステップ S 3 2）。図 1 0 に示す一例では、各機能はいずれもキロステップとして示している。また、機能 1 の開発量は「 1」、機能 2 の開発量は「 1 . 8」、機能 3 の開発量は「 1 . 5」および機能 5 の開発量は「 2」である。

#### 【 0 1 0 0 】

機能 1 の障害率に開発量を乗算した値は「 0 . 1 ( = 0 . 1 × 1 )」になる。機能 2 の障害率に開発量を乗算した値は「 0 . 1 8 ( = 0 . 1 × 1 . 8 )」になる。機能 3 の障害率に開発量を乗算した値は「 0 . 4 2 9 ( = 0 . 2 8 6 × 1 . 5 )」になる。機能 5 の障害率に開発量を乗算した値は「 0 . 1 2 5 ( = 0 . 0 6 2 5 × 2 )」になる。

#### 【 0 1 0 1 】

次に、演算部 6 は、各機能の障害率を合算する（ステップ S 3 3）。要件 A は機能 1、機能 3 および機能 5 との間に第 1 のリレーションが設定されている。演算部 6 は、機能 1、機能 3 および機能 5 についての上述した値（障害率に開発量を乗算した値）を合算する。

#### 【 0 1 0 2 】

従って、演算部 6 は、「 0 . 1 + 0 . 4 2 9 + 0 . 1 2 5」の演算を行い、「 0 . 6 5 4」の合算値を得る。この合算値は要件 A について障害率を合算した値になる。演算部 6 は、要件 B についての合算も行う。演算部 6 は、「 0 . 1 + 0 . 1 8 + 0 . 1 2 5」の演算を行い、「 0 . 4 0 5」の合算値を得る。この合算値は要件 B について障害率を合算した値になる。

#### 【 0 1 0 3 】

次に、演算部 6 は、合算した障害率を開発量の合計値で除算する（ステップ S 3 4）。要件 A の開発量の合計値は「 4 . 5 ( = 1 + 1 . 5 + 2 )」になる。よって、演算部 6 は要件 A について、「 0 . 6 5 4 / 4 . 5」の演算を行い、「 0 . 1 4 5」の値を得る。

#### 【 0 1 0 4 】

また、要件 B の開発量の合計値は「 4 . 8 ( = 1 + 1 . 8 + 2 )」になる。よって、演算部 6 は、要件 B について、「 0 . 4 0 5 / 4 . 8」の演算を行い、「 0 . 0 8 4」の値を得る。

#### 【 0 1 0 5 】

ステップ S 3 4 で演算部 6 が演算した値は、要件ごとの難易度となる。従って、演算部 6 は、要件 A については難易度が「 0 . 1 4 6」、要件 B については難易度が「 0 . 0 8 4」を得る（ステップ S 3 5）。

#### 【 0 1 0 6 】

要件の定量的な難易度は、図 5 のフローに示したように、テスト項目数および障害数から得ることができる。応用例では、演算部 6 は、テスト項目数および障害数に、開発量のファクターを加えて、要件の定量的な難易度の演算を行っている。これにより、より詳細な要件の難易度を得ることができる。

#### 【 0 1 0 7 】

図 6 は、テスト項目数と障害数とに基づいて演算される難易度の一例を示している。図 6 の一例に示されるように、要件 A の難易度は「 0 . 1 4 5」である。一方、図 1 0 に一例として示す難易度は「 0 . 1 4 6」になっている。この難易度は、テスト項目数と障害数と開発量とに基づいて演算される値である。よって、両者は値が異なる。

#### 【 0 1 0 8 】

図 6 の一例に示されるように、要件 B の難易度は「 0 . 0 8 4」である。一方、図 1 0 に一例として示す難易度は「 0 . 0 8 3」である。要件 B についても、開発量を加味して

10

20

30

40

50

難易度の演算を行っているか否かで、演算される値が異なる。従って、開発量を加味した演算を行うことにより、より詳細な要件の難易度を出力することができる。

【 0 1 0 9 】

応用例においても、表示部 2 は、図 1 0 のように難易度と必要性とを表により表示するだけでなく、図 7 のように難易度と必要性との関係をグラフ表示するようにしてもよい。これにより、設計者は、開発量を加味した難易度と必要性との関係を認識することができる。

【 0 1 1 0 】

実施形態と応用例とでは、演算部 6 の演算方法は異なる。ただし、第 1 の書類に関連付けられた全ての第 2 の書類についてのテスト書類に含まれるテスト項目の数と、当該テスト項目のテスト結果とに基づいて、要件の難易度を演算している点では同じである。これにより、設計者は、要件を実現する難易度を定量的に認識することが可能になる。

【 0 1 1 1 】

実施形態および応用例において、難易度算出部 1 は、データベース 3 の各書類データを参照できればよい。従って、難易度算出部 1 は、データベース 3 から各書類データを取得した後に、取得した書類データを参照してもよいし、データベース 3 に記憶されている各書類データの内容を検索するようにしてもよい。

【 0 1 1 2 】

また、実施形態および応用例において、演算部 6 は、テスト結果の障害数を用いて、要件の難易度を演算していたが、テスト結果の正常数を用いて、要件の難易度の演算を行ってもよい。正常数と障害数との合計がテスト項目数になるため、演算部 6 は、テスト結果の正常数を用いて、要件の難易度の演算を行うこともできる。

【 0 1 1 3 】

開示の実施形態とその利点について詳しく説明したが、当業者は、特許請求の範囲に明確に記載した本発明の範囲から逸脱することなく、様々な変更、追加、省略をすることができるであろう。

【 0 1 1 4 】

以上の実施形態に関し、更に以下の付記を開示する。

( 付記 1 )

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援プログラムであって、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部を参照し、

前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する、

処理をコンピュータに実行させる製品設計支援プログラム。

( 付記 2 )

前記要件を複数含み、前記要件ごとに前記難易度を算出する、

処理を前記コンピュータに実行させる付記 1 記載の製品設計支援プログラム。

( 付記 3 )

前記第 2 の書類は、前記要件を具体化または詳細化した第 1 の設計書と、当該第 1 の設計書を具体化または詳細化した第 2 の設計書と、当該第 2 の設計書を具体化または詳細化したソースコードと、を含み、

処理を前記コンピュータに実行させる付記 1 記載の製品設計支援プログラム。

( 付記 4 )

前記要件の必要性を前記難易度と共に表示する、

処理を前記コンピュータに実行させる付記 1 記載の製品設計支援プログラム。

( 付記 5 )

前記難易度を前記必要性で除算した値を表示する、  
処理を前記コンピュータに実行させる付記 4 記載の製品設計支援プログラム。

(付記 6)

前記難易度と前記必要性との関係をグラフ表示する、  
処理を前記コンピュータに実行させる付記 4 記載の製品設計支援プログラム。

(付記 7)

前記テスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果と前記テスト書類に関連付けられる第 2 の書類の機能を実現する開発量とに基づいて、前記難易度を算出する、

処理を前記コンピュータに実行させる付記 1 記載の製品設計支援プログラム。

10

(付記 8)

前記第 2 の書類ごとに、各第 2 の書類に関連付けられたテスト書類に含まれるテスト項目のテスト結果の障害数を前記テスト書類に含まれるテスト項目の数で除算して、前記第 2 の書類ごとの障害率を演算し、

前記第 2 の書類ごとに、前記第 2 の書類の機能を実現するソースコードのステップ数を前記障害率に乗算し、

前記第 2 の書類ごとに乗算された前記障害率を合算し、

合算した前記障害率を全ての前記ソースコードのステップ数の合計値で除算して、前記難易度を演算する、

処理を前記コンピュータに実行させる付記 7 記載の製品設計支援プログラム。

20

(付記 9)

前記テスト書類に含まれる前記テスト項目の数を合算した合計値と前記テスト結果に含まれる障害の合計値とに基づいて、前記難易度を算出する、

処理を前記コンピュータに実行させる付記 1 記載の製品設計支援プログラム。

(付記 10)

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援方法であって、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部を参照し、

30

前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する、

処理をコンピュータが行う製品設計支援方法。

(付記 11)

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援装置であって、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を定義した第 2 の書類と、当該第 2 の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶する記憶部と、

40

前記第 1 の書類に関連付けられた全ての前記第 2 の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する演算部と、

を備える製品設計支援装置。

(付記 12)

開発対象の製品が満たす条件が定義された要件の難易度を算出する製品設計支援装置であって、

前記製品設計支援装置は、プロセッサを備え、

前記プロセッサは、

前記要件を定義した第 1 の書類と、当該第 1 の書類から関連付けられ前記要件の機能を

50

定義した第２の書類と、当該第２の書類に関連付けられたテスト書類と、当該テスト書類に対応付けられたテスト結果と、を記憶するデータベースを参照し、

前記第１の書類に関連付けられた全ての前記第２の書類についてのテスト書類に含まれるテスト項目の数と当該テスト項目のテスト結果とに基づいて、前記要件を実現する際の難易度を算出する、

処理を行う製品設計支援装置。

【符号の説明】

【 0 1 1 5 】

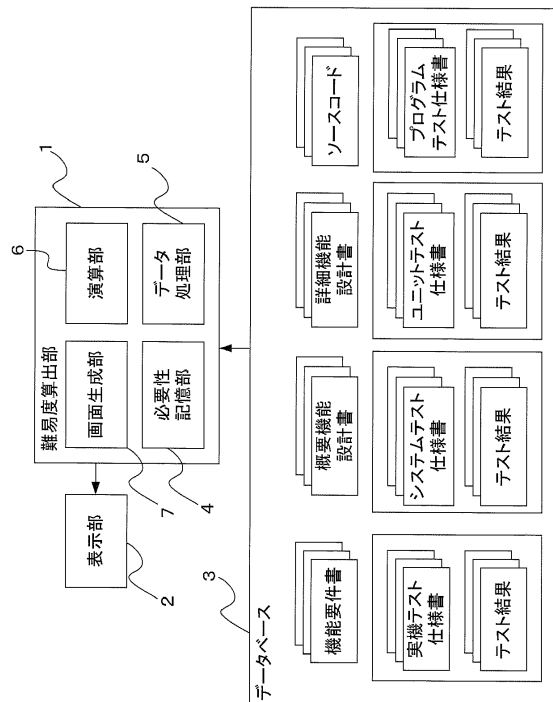
- 1 難易度算出部
- 2 表示部
- 3 データベース
- 4 必要性記憶部
- 5 データ処理部
- 6 演算部
- 7 画面生成部
- 21 プロセッサ
- 22 R A M
- 23 R O M
- 24 補助記憶装置
- 30 可搬型記憶装置

10

20

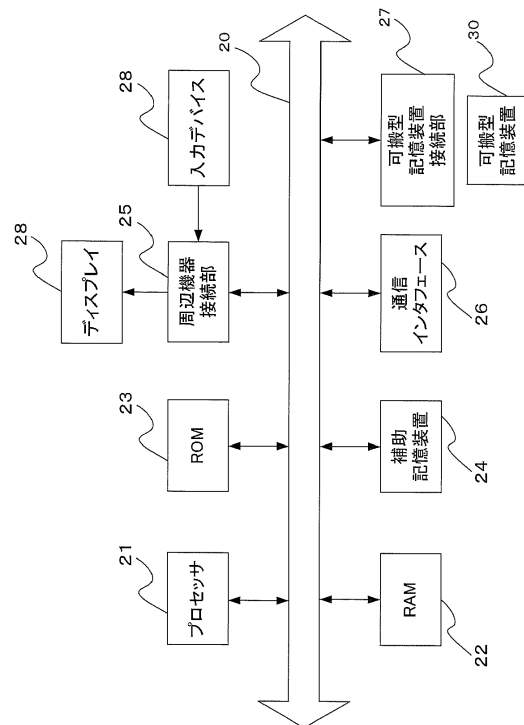
【図１】

製品設計支援装置の一例を示すブロック図



【図２】

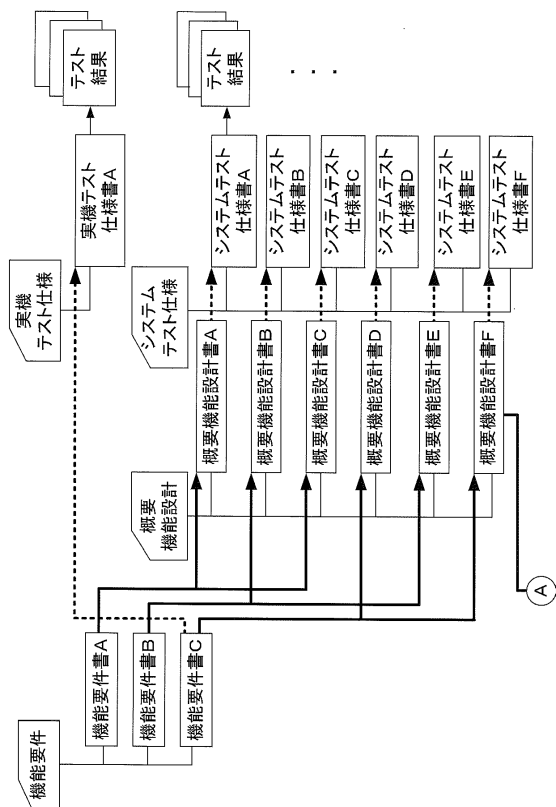
製品設計支援装置の  
ハードウェア構成の一例を示す図





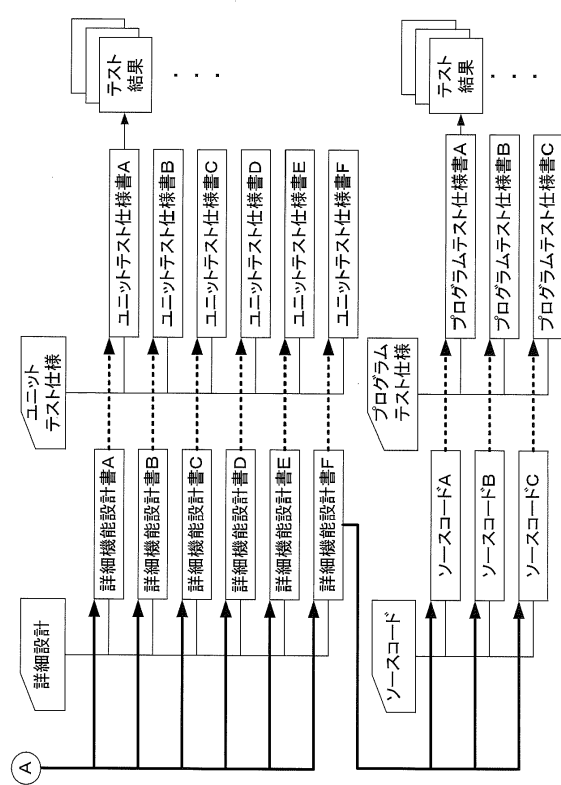
【図 3】

各書類のリレーションの一例を示す図(その1)



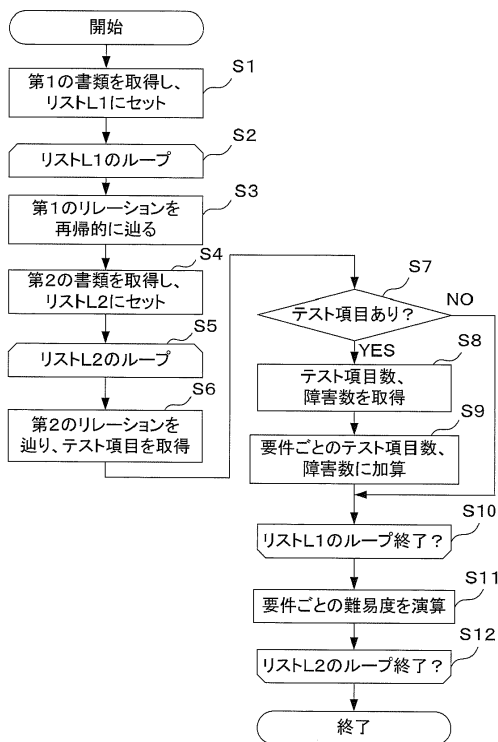
【図 4】

各書類のリレーションの一例を示す図(その2)



【図 5】

実施形態の処理の一例を示すフローチャート



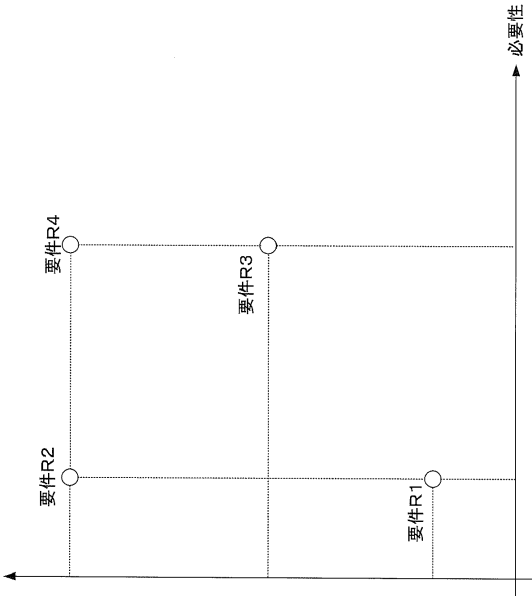
【図 6】

実施形態における要件ごとの  
難易度を示す表の一例

	機能	テスト項目数	障害数	難易度	必要性
要件A	機能1	20	2	0.146 (0.973)	0.150
	機能3	35	10		
	機能5	48	3		
要件B	機能1	20	2	0.083 (1.660)	0.050
	機能2	40	4		
	機能5	48	3		

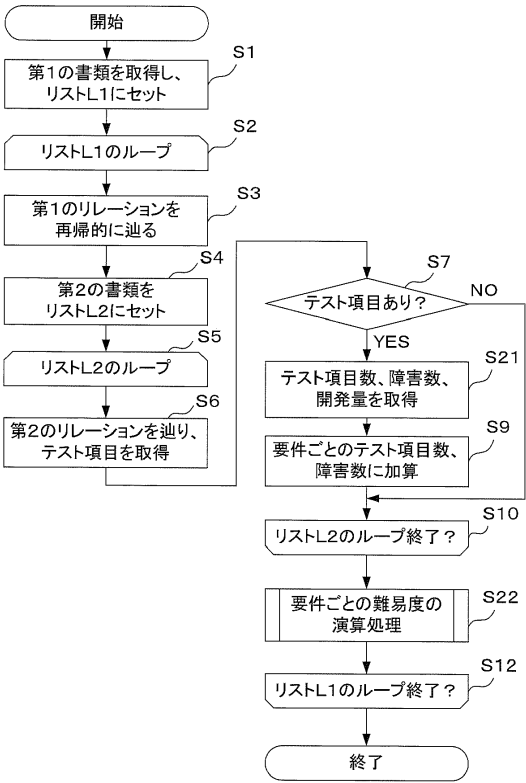
【 図 7 】

実施形態における要件ごとの  
難易度を示すグラフの一例



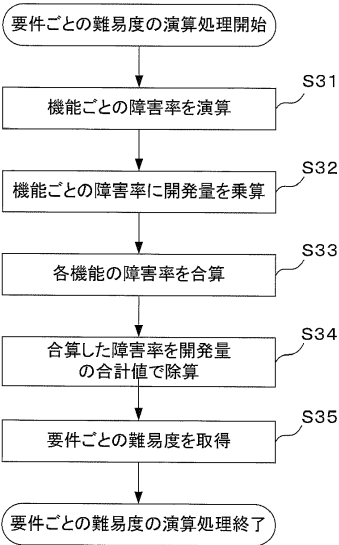
【 図 8 】

応用例における処理の一例を示すフローチャート



【 図 9 】

図8の要件ごとの難易度の演算処理の  
一例を示すフローチャート



【 図 1 0 】

応用例における要件ごとの  
難易度を示す表の一例

	機能	開発量 (Kstep)	テスト項目数	障害数	難易度	必要性
要件A	機能1	1	20	2	0.145 (0.967)	0.150
	機能3	1.5	35	10		
	機能5	2	48	3		
要件B	機能1	1	20	2	0.084 (1.680)	0.050
	機能2	1.8	40	4		
	機能5	2	48	3		

---

フロントページの続き

(56)参考文献 特開2007-207029(JP,A)  
特開2003-099283(JP,A)  
特開2001-337846(JP,A)  
特開2001-125782(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 9/44  
G06F 11/36