(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2015-526820
(P2015-526820A)
(43)公表日　平成27年9月10日(2015.9.10)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/36	***(2006.01)***	G06F	9/06	620M	5B376

審査請求 未請求　予備審査請求 未請求　（全 22 頁）

(21)出願番号	特願2015-528542 (P2015-528542)
(86)(22)出願日	平成25年8月16日 (2013.8.16)
(85)翻訳文提出日	平成27年3月30日 (2015.3.30)
(86)国際出願番号	PCT/US2013/055226
(87)国際公開番号	WO2014/031452
(87)国際公開日	平成26年2月27日 (2014.2.27)
(31)優先権主張番号	13/589,180
(32)優先日	平成24年8月20日 (2012.8.20)
(33)優先権主張国	米国 (US)

(71)出願人　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100101373
弁理士　竹内　茂雄
(74)代理人　100118902
弁理士　山本　修
(74)代理人　100153028
弁理士　上田　忠

最終頁に続く

(54)【発明の名称】ソフトウェアビルドエラーの予測

(57)【要約】
本明細書では、ソフトウェアビルドエラーを予測するためのシステムおよび方法が説明される。一例では、方法は、ソフトウェアにおける複数の変更を検出するステップを含む。また、本方法は、複数の変更リストを識別するステップであって、変更リストが、ソフトウェアにおける複数の変更ごとに識別されるステップを含む。さらに、本方法は、複数の変更リスト内の変更リストごとに特性を識別するステップを含む。さらに、本方法は、複数の変更リストのそれぞれの特性に少なくとも部分的に基づいて、複数の確率を計算するステップであって、確率のそれぞれが、複数の変更リストのうちの１つがソフトウェアビルドエラーを発生させる可能性を示すステップを含む。また、本方法は、ソフトウェアビルドエラーの複数の確率を報告するステップを含む。

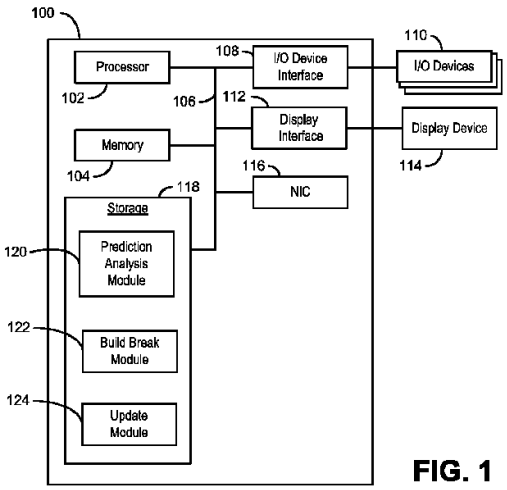

FIG. 1

【特許請求の範囲】
【請求項１】
ソフトウェアビルドエラーを予測するための方法であって、
ソフトウェアにおける複数の変更を検出するステップと、
複数の変更リストを識別するステップであって、１つの変更リストが、前記ソフトウェアにおける前記複数の変更のそれぞれについて識別されるステップと、
前記複数の変更リスト内の変更リストごとに特性を識別するステップと、
前記複数の変更リストのそれぞれの前記特性に少なくとも部分的に基づいて、複数の確率を計算するステップであって、前記確率のそれぞれは、前記複数の変更リストのうちの１つが前記ソフトウェアビルドエラーを発生させる可能性を示すステップと、
前記ソフトウェアビルドエラーの前記複数の確率を報告するステップと
を含む方法。
【請求項２】
請求項１に記載の方法であって、
前記ソフトウェアをビルドするステップと、
前記ソフトウェアビルドエラーを検出するステップと、
前記ソフトウェアビルドエラーをもたらした前記ソフトウェアにおける前記複数の変更のうちの変更を決定するステップと、
前記ソフトウェアビルドエラーをもたらした前記変更を報告するステップと
を含む方法。
【請求項３】
請求項２に記載の方法であって、前記ソフトウェアビルドエラーをもたらした前記変更に基づいて、前記ソフトウェアビルドエラーを予測するために使用される予測生成器をアップデートするステップを含む方法。
【請求項４】
請求項２に記載の方法であって、前記ソフトウェアから前記ソフトウェアビルドエラーをもたらした前記変更を除去するステップを含む方法。
【請求項５】
請求項１に記載の方法であって、前記複数の確率を計算するステップが、前記複数の確率のそれぞれが前記ソフトウェアビルドエラーもたらす可能性を示す回帰を計算するステップを含む、方法。
【請求項６】
請求項１に記載の方法であって、
前記複数の確率に基づいて、複数の高リスク変更リストを識別するステップと、
前記複数の高リスク変更リストのそれぞれを、当該１つの高リスク変更リストに基づいて前記ソフトウェアをビルドする命令とともに、別々のコンピューティングシステムに送信するステップと、
前記ソフトウェアビルドエラーを引き起こす前記高リスク変更リストを検出するステップと
を含む方法。
【請求項７】
ソフトウェアビルドエラーを予測するためのシステムであって、
複数の確率を表示するためのディスプレイデバイスと、
プロセッサ実行可能コードを実行するためのプロセッサと、
プロセッサ実行可能コードを格納する記憶デバイスであって、前記プロセッサ実行可能コードが、前記プロセッサによって実行されると、前記プロセッサに、
ソフトウェアにおける複数の変更を検出することと、
複数の変更リストを識別することであって、１つの変更リストが、前記ソフトウェアにおける前記複数の変更のそれぞれについて識別される、ことと、
前記複数の変更リスト内の変更リストごとに特性を識別することと、

回帰を識別することと、
前記回帰を使用して、前記複数の変更リストのそれぞれの前記特性に少なくとも部分的に基づいて前記複数の確率を計算することであって、前記確率のそれぞれは、前記複数の変更リストのうちの１つが前記ソフトウェアビルドエラーを発生させる可能性を示す、ことと、
前記ソフトウェアビルドエラーの前記複数の確率を報告することと、
前記ソフトウェアビルドエラーの前記確率を減らす動作を勧告することと
をさせる記憶デバイスと
を備えたシステム。
【請求項８】
請求項７に記載のシステムであって、複数の前記特性が、複雑性メトリクス、ソフトウェア開発者決定、コンピューティングシステム決定、時間決定、レビュー決定、アクティビティ決定、開発者ビルド決定、および変更決定の任意の組合せを備える、システム。
【請求項９】
複数の命令を備えた１つまたは複数のコンピュータ可読記憶媒体であって、前記複数の命令は、プロセッサによって実行されると、前記プロセッサに、
ソフトウェアにおける複数の変更を検出することと、
複数の変更リストを識別することであって、１つの変更リストが、前記ソフトウェアにおける前記複数の変更のそれぞれについて識別される、ことと、
前記複数の変更リスト内の変更リストごとに特性を識別することと、
前記複数の変更リストのそれぞれの前記特性に少なくとも部分的に基づいて複数の確率を計算することであって、前記確率のそれぞれが、前記複数の変更リストのうちの１つがソフトウェアビルドエラーを発生させる可能性を示す、ことと、
前記ソフトウェアビルドエラーの前記複数の確率を報告することと、
前記ソフトウェアビルドエラーの前記確率を減らす動作を勧告することと
をさせる、１つまたは複数のコンピュータ可読記憶媒体。
【請求項１０】
請求項９に記載の１つまたは複数のコンピュータ可読記憶媒体であって、
前記複数の命令は、前記プロセッサによって実行されると、前記プロセッサに、
前記ソフトウェアをビルドすることと、
前記ソフトウェアビルドエラーを検出することと、
前記ソフトウェアビルドエラーをもたらした前記ソフトウェアにおける前記複数の変更のうちの変更を決定することと、
前記ソフトウェアビルドエラーをもたらした前記変更を報告することと
をさせる、１つまたは複数のコンピュータ可読記憶媒体。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001]ソフトウェア開発は、機械実行可能コードに変換されるべきソフトウェアコードの開発を含み得る。開発者によって書かれたソフトウェアコードを、機械実行可能コードに変換することは、ソフトウェアビルドと呼ぶことができる。ソフトウェアを開発する間、ソフトウェアのビルド中に発生したエラーは、ソフトウェアを開発するための時間量を増加させる場合がある。たとえば、何らかの組織が、開発者のチームとともにソフトウェアを開発する。いくつかの例では、開発者の第１のチームは、ソフトウェアアプリケーションの一部のビルドを、開発者の第２のチームがソフトウェアアプリケーションの別の部分をビルドするまで待つことができる。開発者の第２のチームがソフトウェアビルドエラーに直面すると、開発者の第１のチームは、ソフトウェアアプリケーションの一部のビルドが遅延する場合がある。したがって、ソフトウェアビルドエラーを最小化することによって、ソフトウェア開発プロセスにおける遅延を防止することができる。
【発明の概要】

【０００２】
[0002]以下で、本明細書に記載のいくつかの態様の基本的な理解を提供するために、簡単な概要を提示する。この概要は、特許請求される主題の広範な概要ではない。この概要は、特許請求される主題の主要な、または重要な要素を特定することを意図するものではなく、また特許請求される主題の範囲を説明することを意図するものでもない。この概要の唯一の目的は、特許請求される主題のいくつかの概念を、後に提示されるより詳細な説明の前置きとして、簡単な形態で提示することである。
【０００３】
[0003]ある実施形態は、ソフトウェアビルドエラーを予測するための方法を提供する。本方法は、ソフトウェアにおける複数の変更を検出するステップを含む。また、本方法は、複数の変更リストを識別するステップであって、変更リストが、ソフトウェアにおける複数の変更ごとに識別されるステップを含む。さらに、本方法は、複数の変更リスト内の変更リストごとに特性を識別するステップを含む。さらに、本方法は、複数の変更リストのそれぞれの特性に少なくとも部分的に基づいて、複数の確率を計算するステップであって、確率のそれぞれが、複数の変更リストのうちの１つがソフトウェアビルドエラーを発生させる可能性を示すステップを含む。また、本方法は、ソフトウェアビルドエラーの複数の確率を報告するステップを含む。
【０００４】
[0004]別の実施形態は、ソフトウェアビルドエラーを予測するためのシステムである。本システムは、複数の確率を表示するためのディスプレイデバイスと、プロセッサ実行可能コードを実行するためのプロセッサと、プロセッサ実行可能コードを格納する記憶デバイスとを含む。本システムは、ソフトウェアにおける複数の変更を検出する。また、本システムは、複数の変更リストを識別し、変更リストが、ソフトウェアにおける複数の変更ごとに識別される。さらに、本システムは、複数の変更リスト内の変更リストごとに特性を識別する。さらに、本システムは、ロジスティック回帰を識別する。また、本システムは、ロジスティック回帰を使用して、複数の変更リストのそれぞれの特性に少なくとも部分的に基づいて複数の確率を計算して、確率のそれぞれが、複数の変更リストのうちの１つがソフトウェアビルドエラーを発生させる可能性を示す。さらに、本システムは、ソフトウェアビルドエラーの複数の確率を報告する。
【０００５】
[0005]別の実施形態は、複数の命令を備える、１つまたは複数の有形のコンピュータ可読記憶媒体を提供する。本命令は、プロセッサに、ソフトウェアにおける複数の変更を検出させて、複数の変更リストを識別させ、ただし、変更リストが、ソフトウェアにおける複数の変更ごとに識別される。また、本命令は、プロセッサに、複数の変更リスト内の変更リストごとに特性を識別させる。さらに、本命令は、プロセッサに、複数の変更リストのそれぞれの特性に少なくとも部分的に基づいて複数の確率を計算させ、ただし、確率のそれぞれが、複数の変更リストのうちの１つがソフトウェアビルドエラーを発生させる可能性を示す。また、本命令は、プロセッサに、ソフトウェアビルドエラーの複数の確率を報告させる。
【０００６】
[0006]以下の詳細な説明は、添付の図面を参照することによって、よりよく理解されよう。添付の図面は、開示された主題の多数の特徴の具体例を含む。
【図面の簡単な説明】
【０００７】
【図１】[0007]ソフトウェアビルドエラーを予測するコンピューティングシステムの例のブロック図である。
【図２】[0008]ソフトウェアビルドエラーを予測するための方法の例を示すプロセスフロー図である。
【図３】[0009]ソフトウェアビルドエラーを予測するために使用される予測分析モジュールの例を示すブロック図である。

【図４】[0010]ソフトウェアビルドエラーを分析するために使用されるビルドブレークモジュールの例を示すブロック図である。
【図５】[0011]予測分析モジュールをアップデートするために使用されるアップデートモジュールの例を示すブロック図である。
【図６】[0012]ソフトウェアビルドエラーを予測する、有形のコンピュータ可読記憶媒体の例を示すブロック図である。
【発明を実施するための形態】
【０００８】
[0013]ソフトウェアビルドエラーに関連付けられる遅延を最小限にするために、ソフトウェアビルドエラーを予測するための様々な方法が開発されてきた。いくつかの方法は、最後のソフトウェアビルド以降に変更されたソフトウェアコードの行数などの、ソフトウェアコードの特定の側面に関する情報を収集するステップを含む。これらの方法は、収集された情報に基づいて、ソフトウェアビルドが成功する可能性を決定することを試みることができる。しかしながら、これらの方法の多くは、ソフトウェアをビルドするプロセス、およびソフトウェアに行われた実際の変更ではなく、ソフトウェアコードから得られた情報に焦点を当てている。他の方法は、ソフトウェアビルドエラーを識別することができる変数のセットを識別するステップを含む。しかしながら、これらの方法の多くは、ソフトウェアビルドが失敗する可能性があるときを識別するために変数の固定セットを使用する決定木に依存する。
【０００９】
[0014]本明細書に記載の技法は、ソフトウェアビルドエラーの任意の適切な数の確率に基づいて、ソフトウェアビルドエラーを予測することができる。いくつかの実施形態では、本明細書に記載の技法は、最後のソフトウェアビルド以降の一連の変更を識別して、各変更がソフトウェアビルドエラーを作成し得る確率を計算することができる。ソフトウェアビルドは、ソフトウェアをビルドする状態を指すことができ、ソフトウェア（本明細書では、ソフトウェアコードとも呼ばれる）を機械実行可能ファイルにコンパイルして、アプリケーションを形成するために機械実行可能ファイルをリンクするステップを含む。ソフトウェアビルドエラーは、ソフトウェアコードが実行可能ファイルにコンパイルされることを防止する、またはソフトウェアコードがリンクされることを防止する、ソフトウェアコードにおけるエラーを含み得る。ソフトウェアビルドエラーは、ソフトウェアコードが機械実行可能コードに変換されることを防止することができ、それはソフトウェアコードがアプリケーションに組み込まれることを防止することができる。
【００１０】
[0015]前付けとして、図面のうちのいくつかは、機能、モジュール、特徴、要素等と呼ばれる、１つまたは複数の構造的コンポーネントのコンテキストで概念を説明する。図面に示される様々なコンポーネントは、たとえば、ソフトウェア、ハードウェア（たとえば、個別論理コンポーネント等）、ファームウェア等、またはこれらの実装形態の任意の組合せによって、任意の方法で実装され得る。一実施形態では、様々なコンポーネントは、実際の実装形態における、対応するコンポーネントの使用を反映し得る。他の実施形態では、図面に示される任意の単一のコンポーネントは、いくつかの実際のコンポーネントによって実装され得る。図面における任意の３つ以上の別々のコンポーネントの説明は、単一の実際のコンポーネントによって実行される異なる機能を反映し得る。以下で説明する図１は、図面に示される機能を実装するために使用され得る１つのシステムに関する詳細を提供する。
【００１１】
[0016]他の図面は、フローチャート形式で概念を説明する。この形式では、特定の動作が、特定の順序で実行される別個のブロックを構成するものとして説明される。そのような実装形態は例示的なものであり、非限定的なものである。本明細書に記載の特定のブロックは、グループ化して、単一の動作で実行することができ、特定のブロックは複数のコンポーネントブロックに分割することができ、また、特定のブロックは、ブロックの実行

の並列方法を含む、本明細書に示されている順序とは異なる順序で実行することができる。フローチャートに示されるブロックは、ソフトウェア、ハードウェア、ファームウェア、手動処理等、またはこれらの実装形態の任意の組合せによって実装され得る。本明細書で使用されるように、ハードウェアは、コンピュータシステム、特定用途向け集積回路（ＡＳＩＣ）などの個別論理コンポーネント等、ならびにそれらの任意の組合せを含み得る。

【００１２】

[0017]用語については、「ように構成される（ｃｏｎｆｉｇｕｒｅｄ　ｔｏ）」という語句は、任意の種類の構造的コンポーネントが、識別された動作を実行するように構築され得る、任意の方法を包含する。構造的コンポーネントは、ソフトウェア、ハードウェア、ファームウェア等、またはそれらの任意の組合せを使用する動作を実行するように構成され得る。

【００１３】

[0018]「論理（ｌｏｇｉｃ）」という用語は、タスクを実行するための任意の機能を包含する。たとえば、フローチャートに示される各動作は、その動作を実行するための論理に対応する。動作は、ソフトウェア、ハードウェア、ファームウェア等、またはそれらの任意の組合せを使用して実行され得る。

【００１４】

[0019]本明細書で使用されるように、「コンポーネント（ｃｏｍｐｏｎｅｎｔ）」、「システム（ｓｙｓｔｅｍ）」、「クライアント（ｃｌｉｅｎｔ）」等の用語は、ハードウェア、ソフトウェア（たとえば、実行中の）、および／またはファームウェア、あるいはそれらの組合せのいずれかの、コンピュータ関連エンティティを指すことを意図する。たとえば、コンポーネントは、プロセッサ上で実行しているプロセス、オブジェクト、実行ファイル、プログラム、関数、ライブラリ、サブルーチン、および／またはコンピュータ、あるいはソフトウェアとハードウェアとの組合せでよい。例として、サーバ上で実行しているアプリケーションとサーバは、両方ともコンポーネントでよい。１つまたは複数のコンポーネントはプロセス内に常駐することができ、コンポーネントは１つのコンピュータ上でローカライズされてもよく、および／または２つ以上のコンピュータ間で分散されてもよい。

【００１５】

[0020]さらに、特許請求される主題は、開示された主題を実装するためにコンピュータを制御するために、ソフトウェア、ファームウェア、ハードウェア、またはそれらの任意の組合せを生成するべく、標準プログラミング、および／またはエンジニアリング技法を使用する方法、装置、または製品として実装され得る。本明細書で使用される「製品（ａｒｔｉｃｌｅ　ｏｆ　ｍａｎｕｆａｃｔｕｒｅ）」という用語は、任意の有形のコンピュータ可読デバイスまたは媒体からアクセス可能なコンピュータプログラムを包含することを意図する。

【００１６】

[0021]コンピュータ可読記憶媒体は、これに限定されないが、磁気記憶デバイス（たとえば、特にハードディスク、フロッピー（登録商標）ディスク、および磁気ストリップ）、光ディスク（たとえば、特にコンパクトディスク（ＣＤ）、およびデジタル多用途ディスク（ＤＶＤ））、スマートカード、ならびにフラッシュメモリデバイス（たとえば、特にカード、スティック、およびキードライブ）を含むことができる。それに対して、コンピュータ可読媒体は、一般的に（すなわち、必ずしも記憶媒体ではない）ワイヤレス信号などのための送信媒体などの通信媒体をさらに含み得る。

【００１７】

[0022]図１は、ソフトウェアビルドエラーを予測するコンピューティングシステムの例のブロック図である。コンピューティングシステム１００は、たとえば、特にモバイル電話、ラップトップコンピュータ、デスクトップコンピュータ、またはタブレットコンピュータでよい。コンピューティングシステム１００は、格納された命令を実行するように適

合されたプロセッサ１０２、ならびにプロセッサ１０２によって実行可能な命令を格納するメモリデバイス１０４を含み得る。プロセッサ１０２は、シングルコアプロセッサ、マルチコアプロセッサ、コンピューティングクラスタ、または任意の数の他の構成でもよい。メモリデバイス１０４は、ランダムアクセスメモリ（たとえば、ＳＲＡＭ、ＤＲＡＭ、ゼロキャパシタＲＡＭ、ＳＯＮＯＳ、ｅＤＲＡＭ、ＥＤＯ　ＲＡＭ、ＤＤＲ　ＲＡＭ、ＲＲＡＭ（登録商標）、ＰＲＡＭ等）、読出し専用メモリ（たとえば、Ｍａｓｋ　ＲＯＭ、ＰＲＯＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ等）、フラッシュメモリ、または他の任意の適切なメモリシステムを含み得る。プロセッサ１０２によって実行される命令は、ソフトウェアビルドエラーを予測するために使用され得る。

【００１８】

[0023]プロセッサ１０２は、システムバス１０６（たとえば、ＰＣＩ、ＩＳＡ、ＰＣＩ－Ｅｘｐｒｅｓｓ、ＨｙｐｅｒＴｒａｎｓｐｏｒｔ（登録商標）、ＮｕＢｕｓ等）を通じて、コンピューティングシステム１００を１つまたは複数のＩ／Ｏデバイス１１０に接続するように適合された入力／出力（Ｉ／Ｏ）デバイスインターフェース１０８に接続され得る。Ｉ／Ｏデバイス１１０は、たとえば、キーボード、ジェスチャ認識入力デバイス、音声認識デバイス、およびポインティングデバイスを含むことができ、ポインティングデバイスは、特にタッチパッドまたはタッチスクリーンを含み得る。Ｉ／Ｏデバイス１１０は、コンピューティングシステム１００の内蔵コンポーネントでもよく、コンピューティングシステム１００に外部接続されたデバイスでもよい。

【００１９】

[0024]プロセッサ１０２は、システムバス１０６を通じて、コンピューティングシステム１００をディスプレイデバイス１１４に接続するように適合されたディスプレイインターフェース１１２にリンクされてもよい。ディスプレイデバイス１１４は、コンピューティングシステム１００の内蔵コンポーネントであるディスプレイスクリーンを含み得る。また、ディスプレイデバイス１１４は、コンピューティングシステム１００に外部接続された、特にコンピュータモニタ、テレビ、またはプロジェクタを含み得る。ネットワークインターフェースカード（ＮＩＣ）１１６も、システムバス１０６を通じてコンピューティングシステム１００をネットワーク（図示せず）に接続するように適合され得る。ネットワーク（図示せず）は、特にワイドエリアネットワーク（ＷＡＮ）、ローカルエリアネットワーク（ＬＡＮ）、またはインターネットでよい。

【００２０】

[0025]記憶装置１１８は、ハードドライブ、光ドライブ、ＵＳＢフラッシュドライブ、ドライブのアレイ、またはそれらの任意の組合せを含み得る。記憶装置１１８は、予測分析モジュール１２０、ビルドブレークモジュール１２２、およびアップデートモジュール１２４を含み得る。予測分析モジュール１２０は、ソフトウェアコードへの任意の数の変更を検出して、ソフトウェアコードがソフトウェアビルドエラーを含む可能性を予測することができる。予測分析モジュール１２０は、ソフトウェアコードへの各変更がソフトウェアビルドエラーを引き起こし得る確率を計算することによって、ソフトウェアコードがソフトウェアビルドエラーを含む可能性を予測することができる。ビルドブレークモジュール１２２は、ソフトウェアをビルドして、ソフトウェアビルドエラーを検出することができる。ビルドブレークモジュール１２２がソフトウェアビルドエラーを検出すると、ビルドブレークモジュール１２２も、ソフトウェアビルドエラーを引き起こしたソフトウェアコードへの変更を検出することができる。ビルドブレークモジュール１２２は、ソフトウェアビルドエラーを引き起こす変更を、アップデートモジュール１２４に送信することができる。アップデートモジュール１２４は、ソフトウェアコード変更、および対応するビルドエラーについての履歴情報を格納することができる。アップデートモジュール１２４は、履歴情報を予測分析モジュール１２０に提供することができ、予測分析モジュール１２０がソフトウェアビルドエラーの正確な予測を計算することを可能にする。

【００２１】

[0026]図１のブロック図は、コンピューティングシステム１００が図１に示されるすべ

てのコンポーネントを含むべきであることを示すことを意図するものではないことが理解されるべきである。むしろ、コンピューティングシステム１００は、より少数の、または図１に示されていないさらなるコンポーネント（たとえば、さらなるアプリケーション、さらなるメモリデバイス、さらなるネットワークインターフェース等）を含み得る。たとえば、コンピューティングシステム１００は、特にユーザ、アプリケーション、または別のハードウェアデバイスにソフトウェアビルド情報を報告することができる報告モジュールを含み得る。さらに、予測分析モジュール１２０、ビルドブレークモジュール１２２、またはアップデートモジュール１２４の機能のうちのいずれかは、部分的に、または全体的に、ハードウェアおよび／またはプロセッサ１０２に実装され得る。たとえば、機能は、特定用途向け集積回路、プロセッサ１０２に実装された論理、または他の何らかのデバイスに実装され得る。

【００２２】

[0027]図２は、ソフトウェアビルドエラーを予測するための方法の例を示すプロセスフロー図である。方法２００は、図１のコンピューティングシステム１００などのコンピューティングシステムで実装され得る。コンピューティングシステム１００は、ソフトウェアコードへの変更、ならびに以前のソフトウェアコード変更および以前のソフトウェアビルド結果の履歴情報に基づいてソフトウェアビルドエラーを予測することができる、予測分析モジュール１２０も含み得る。

【００２３】

[0028]ブロック２０２で、予測分析モジュール１２０はソフトウェアコードにおける変更を検出する。一実施形態では、予測分析モジュール１２０は、ソフトウェアコードと、ソフトウェアコードの以前のバージョンとを比較することによって、ソフトウェアコードへの変更を検出することができる。たとえば、予測分析モジュール１２０は、ソフトウェアコードにおける違いを識別することによって、ソフトウェアコードの２つの異なるバージョンにおける変更を検出することができる。他の実施形態では、予測分析モジュール１２０は、ソフトウェアコードへの変更に対応するソフトウェアコード内のインジケータを識別することによって、ソフトウェアコードにおける変更を検出することができる。たとえば、予測分析モジュール１２０は、ソフトウェアコードにおける変更に対応するソフトウェアコードに含まれるコメントに基づいて、ソフトウェアコードにおける変更を検出することができる。

【００２４】

[0029]ブロック２０４で、予測分析モジュール１２０は、ソフトウェアコードへの変更ごとに変更リストを識別する。いくつかの実施形態では、予測分析モジュール１２０は、各変更リスト内の複数の変更を含み得る。たとえば、開発者は、予測分析モジュール１２０が１つの変更リスト内に含むことができる、ソフトウェアコードのいくつかの行を変更することができる。他の実施形態では、予測分析モジュール１２０は、開発者ごとに、または作業セッションごとに、変更リストを識別することができる。たとえば、予測分析モジュール１２０は、特定の開発者からソフトウェアコードに行われた変更を識別して、その変更を変更リストに格納することができる。他の例では、予測分析モジュール１２０は、ソフトウェアコードへの変更を含む各作業セッションを識別して、開発者ごとの作業日ごとに変更リストを識別することができる。

【００２５】

[0030]ブロック２０６で、予測分析モジュール１２０は、変更リストごとに特性を識別する。変更リストの特性は、ソフトウェアコードにおける変更に関連付けられる任意の情報を含み得る。いくつかの実施形態では、特性は、ソフトウェアコードから得られた情報を含み得る。たとえば、変更リストの特性は、特に、修正されたソフトウェアコードファイルの数、または修正されたソフトウェアコードの行数を含み得る。いくつかの実施形態では、特性は、ソフトウェアコードにおける変更に関連する要因から得られた情報も含み得る。たとえば、変更リストの特性は、特に、ソフトウェアコードに変更を加えた開発者の決定、変更によって影響を受けたプロジェクト、ソフトウェアコードをコンパイルする

ために開発者が使用したコンピューティングシステムの決定、ソフトウェアコードへの変更をレビューした個人の数または名前（本明細書では、レビュー決定とも呼ばれる）、変更が提出された時間（本明細書では、時間決定とも呼ばれる）、ソフトウェアコードにおける変更に関連する複雑性メトリクス、および変更されたソフトウェアコードに基づく依存性を含み得る。複雑性メトリクスは、ソフトウェアコードの行における特徴の数、ソフトウェアコードの行を囲むネストされたループの数、または、ソフトウェアコードの複雑性を示す他の任意の要因を含むことができる。

【００２６】

[0031]変更リストの特性のさらなる例は、他の開発者によって、ソースコードファイル、または修正された、もしくは所与の変更リストに関連する、ソースコードファイルの行に実行される任意のアクティビティ（本明細書では、アクティビティ決定と呼ばれる）を含み得る。変更リストの特性は、開発者によって行われた変更の表示も含み得る。たとえば、開発者によって行われた変更は、導入されたソースコードフラグメント、参照された識別子、またはソースコードに行われた他の任意の変更の記述（本明細書では、変更決定と呼ばれる）を含み得る。さらなる特性は、開発者のコンピューティングシステム上でソフトウェアがビルドされたときに、開発者のコンピューティングシステムが同期されたソースコードリポジトリの状態（本明細書では、開発者ビルド決定とも呼ばれる）、開発者のコンピューティングシステム上で実行されたテスト、およびソフトウェアビルドの一部として含まれたプロジェクトも含み得る。予測分析モジュール１２０は、変更リストに対応する任意の数の特性を考慮することによって、より正確な予測を提供することができる。

【００２７】

[0032]ブロック２０８で、予測分析モジュール１２０は、ソフトウェアコードにおける変更ごとに確率を計算する。いくつかの実施形態では、変更ごとの確率は、ソフトウェアコードにおける変更がソフトウェアビルドエラーをもたらし得る可能性を表すことができる。確率は、ロジスティック回帰などの回帰を使用して計算され得る。たとえば、予測分析モジュール１２０は、変更リストに関連する特性ごとに係数を生成することができる。予測分析モジュール１２０は、履歴データに基づいて係数を決定することができる。たとえば、履歴データは、特定のソフトウェア開発者がソフトウェアビルドエラーを引き起こす確率が２０％であることを示すことができる。この例では、予測分析モジュールは、ソフトウェア開発者に関連する特性に、係数として２０％の値を割り当てることができる。

【００２８】

[0033]いくつかの実施形態では、予測分析モジュール１２０は、ソフトウェアビルドエラーの確率を結合することができる。たとえば、予測分析モジュール１２０は、変更リストごとに、ソフトウェアビルドエラーを引き起こす個々の確率（Ｐ１、Ｐ２、．．．ＰＮ）を計算することができる。予測分析モジュール１２０は、式１で確率を結合することができる。

【００２９】

１－（１－Ｐ１）（１－Ｐ２）．．．（１－ＰＮ）＝Ｐ（Ｅｒｒｏｒ）　式（１）

[0034]式１では、Ｐ１からＰＮは、変更リストがソフトウェアビルドエラーを引き起こし得る確率を表している。Ｐ（Ｅｒｒｏｒ）という用語は、ソフトウェアビルドエラーがソフトウェアコードへのＮ個の変更に基づいて発生し得る可能性を表している。

【００３０】

[0035]いくつかの実施形態では、予測分析モジュール１２０は、個々の変更リストの特性を集約して、複数の変更リストのソフトウェアビルドエラーの結合された確率を計算することができる。たとえば、予測分析モジュール１２０は、ソフトウェアビルド内の変更リストの数を検出することができる。また、予測分析モジュール１２０は、変更リストに関連付けられる任意の適切な数の集約値を検出することができる。たとえば、３つの変更リストは、ソースコードの１００、１５０、および２００行が変更されたことを示すことができる。コードの変更された行の数は、合計により集約されて、コードの変更された行

の結合された数、すなわち上述の例では４５０のコードの変更された行になり得る。上述のソフトウェアビルドが、ソフトウェアビルドエラーをもたらした４００を上回る変更されたソースコードの行を有する場合、ソフトウェアビルドは、ソフトウェアビルドエラーを引き起こすより高いリスクが割り当てられる場合がある。他の例では、予測分析モジュール１２０は、特にソフトウェアビルドエラーの集約最大確率、ソフトウェアビルドエラーの集約最小確率、ソフトウェアビルドエラーの集約平均確率、ソフトウェアビルドエラーの集約中央値確率、ソフトウェアビルドエラーの確率の合計、ソフトウェアビルドエラーの集約されたパーセンタイル、またはソフトウェアビルドエラーの確率の標準偏差を検出することによって、変更リストの特性を集約することができる。

【００３１】

[0036]ブロック２１０で、予測分析モジュール１２０は、ソフトウェアコード内の各変更がソフトウェアビルドエラーを引き起こす確率を報告する。いくつかの実施形態では、予測分析モジュール１２０は、ソフトウェアビルドエラーの確率を、特にユーザ、アプリケーション、または他のハードウェアデバイスに報告することができる。たとえば、予測分析モジュール１２０は、ソフトウェアコードへの変更がソフトウェアビルドエラーをもたらし得る確率は２０％であると計算することができる。この例では、予測分析モジュール１２０は、変更がソフトウェアビルドエラーを引き起こし得る可能性が２０％であるという予測をユーザに報告することができる。このプロセスはブロック２１２で終了する。

【００３２】

[0037]図２のプロセスフロー図は、方法２００のステップが任意の特定の順序で実行されるべきであること、または方法２００のすべてのステップがあらゆる場合において含まれるべきであることを示すことを意図するものではない。いくつかの例では、ソフトウェアコードにおける変更は徐々に検出され得る。たとえば、予測分析モジュール１２０は、ソフトウェアコードへの変更ごとに新しい変更リストを生成して、ソフトウェアビルドエラーの確率を再計算することができる。他の例では、特性は、ソフトウェアコードへのさらなる変更が検出されるにつれて変わる場合がある。たとえば、予測分析モジュール１２０は、統合された開発環境についての、特に複雑性メトリクスまたは変更決定などの様々な特性を検出することができる。次いで、予測分析モジュール１２０は、開発者がソフトウェアコードへの変更を完遂した後に、さらなる特性を検出することができる。さらに、特定の用途に応じて、方法２００に任意の数のさらなるステップが含まれ得る。いくつかの実施形態では、予測分析モジュール１２０は、変更リストをビルドブレークモジュール１２２に送信することができ、ビルドブレークモジュール１２２は、ソフトウェアコードへの変更がソフトウェアビルドエラーを引き起こすかどうかを決定することができる。ビルドブレークモジュール１２２は、以下で図４に関連してより詳細に説明する。他の実施形態では、予測分析モジュール１２０は、変更リストをアップデートモジュール１２４に送信することができる。アップデートモジュール１２４は、変更リストおよびソフトウェアビルドエラーに対応する履歴データをアップデートすることができる。アップデートモジュール１２４は、以下で図５に関連してより詳細に説明する。

【００３３】

[0038]図３は、ソフトウェアビルドエラーを予測するために使用される予測分析モジュールの例を示すブロック図である。予測分析モジュール１２０は、図１のコンピューティングシステム１００などのコンピューティングシステムに実装され得る。いくつかの実施形態では、予測分析モジュール１２０は、特徴抽出コンポーネント３０２、予測生成器３０４、トリガコンポーネント３０８、および報告コンポーネント３０６を含み得る。予測分析モジュール１２０のコンポーネントは、ソフトウェアビルドエラーの可能性を識別して分析することができる。

【００３４】

[0039]いくつかの実施形態では、予測分析モジュール１２０は、任意の適切な数の変更リストを入力として受け入れることができる。上述のように、変更リストは、ソフトウェアコードへの任意の適切な数の変更を含むことができる。予測分析モジュール１２０は、

変更リストを特徴抽出コンポーネント３０２に送信することができる。特徴抽出コンポーネント３０２は、ソフトウェアコードにおける変更に関連付けられる任意の適切な数の特性を決定することができる。たとえば、特徴抽出コンポーネント３０２は、各変更リストに関連付けられる特性を識別することができる。いくつかの例では、特徴抽出コンポーネント３０２は、特に、修正されたソフトウェアコードファイルの数、修正されたソフトウェアコードの行数、ソフトウェアコードに変更を加えた開発者、変更によって影響を受けたプロジェクト、ソフトウェアコードをコンパイルするために開発者が使用したコンピューティングシステム、ソフトウェアコードへの変更をレビューした個人の数、変更が提出された時間、ソフトウェアコードにおける変更に関連する複雑性メトリクス、および変更されたソフトウェアコードに基づく依存性などの特性を識別することができる。特徴抽出コンポーネント３０２は、変更リストおよび対応する特性を予測生成器３０４に送信することができる。

【００３５】

[0040]予測生成器３０４は、変更リストの特性に基づいてソフトウェアビルドエラーの確率を計算することができる。図２に関連して上述したように、ソフトウェアビルドエラーの確率は、特に、あるタイプの回帰を使用して、または、これに限定されないが、サポートベクターマシン、単純ベイズ（Ｎａｉｖｅ　Ｂａｙｅｓ）、もしくは決定木を含む機械学習モデルを使用して計算することができる。いくつかの実施形態では、ソフトウェアビルドエラーの確率は、線形回帰またはロジスティック回帰に基づいて計算することができる。他の実施形態では、確率は、ソフトウェアビルドエラーを引き起こす変更リストごとに結合された確率に基づいてソフトウェアビルドエラーの可能性を計算するために結合され得る。いくつかの実施形態では、予測生成器３０４は、ソフトウェアビルドエラーの確率を報告コンポーネント３０６およびトリガコンポーネント３０８に送信することができる。

【００３６】

[0041]報告コンポーネント３０６は、任意の適切な数の方法を使用して、フィードバックをユーザ、アプリケーション、またはハードウェアデバイスに提供することができる。いくつかの実施形態では、フィードバックは、変更リストがソフトウェアビルドエラーを引き起こし得る確率を含み得る。報告コンポーネント３０６は、特に、ディスプレイデバイスに送信されたメッセージ、ＩＤＥで生成されたダイアログボックス、電子メール通知、またはニュースフィードを通じてフィードバックを提供することができる。

【００３７】

[0042]トリガコンポーネント３０８は、ソフトウェアビルドエラーの確率に基づいて、さらなる動作を開始または要求することができる。たとえば、トリガコンポーネント３０８は、報告コンポーネント３０６を通じてフィードバックを開発者に提供することができる。フィードバックは、ソフトウェアビルドエラーを発生させる確率が高いソフトウェアコードのさらなるレビューを要求することができる。いくつかの例では、フィードバックは、変更リストの特定の特性、および対応するソフトウェアビルドエラーを発生させる確率を識別することができる。フィードバックは、特に、変更をソースコードに提出する前にさらなるプロジェクトをビルドするために、または提出前にさらなるテストを実行するために、特定の変更リストのさらなるレビューを勧告する（ｒｅｃｏｍｍｅｎｄ；リコメンドする、勧める、推奨する）ことができる。いくつかの実施形態では、トリガコンポーネント３０８は、ビルドプロセス中に、さらなるチェックインまたは品質論理ゲートを要求することもできる。たとえば、トリガコンポーネント３０８は、機械実行可能アプリケーションを形成するためにソフトウェアコードをコンパイルしてリンクする、ビルドコンポーネント３１０を含み得る。トリガコンポーネント３０８は、ソフトウェアビルドエラーを防止することができるさらなる品質論理ゲートを含むようビルドコンポーネント３１０に命令することができる。また、さらなる品質論理ゲートは、ソフトウェアビルドエラーの原因を決定する際に支援することができる。

【００３８】

[0043]いくつかの実施形態では、トリガコンポーネント３０８は、ソフトウェアビルドを制御する命令をビルドコンポーネント３１０に提供することもできる。たとえば、トリガコンポーネント３０８は、ビルドコンポーネント３１０に、ソフトウェアビルドエラーを発生させる低リスクを共有する変更リストをグループ化することによってソフトウェアをビルドするよう命令することができる。トリガコンポーネント３０８は、各変更リストがソフトウェアビルドエラー引き起こす確率に基づいて、ソフトウェアビルドエラーを引き起こし得る低リスクを有する変更リストを識別することができる。いくつかの実施形態では、トリガコンポーネント３０８は、ビルドコンポーネント３１０に、低リスク変更リストをビルドする前に高リスク変更リストをビルドするよう命令することもできる。また、トリガコンポーネント３０８は、変更リストがソフトウェアビルドエラーを引き起こしそうであることを示す確率に基づいて、ソフトウェアビルドエラーを引き起こす高リスクを有する変更リストを識別することができる。たとえば、トリガコンポーネント３０８は高リスク変更リストを複数のコンピューティングシステムに送信することができ、高リスクの変更セットが並列にビルドされることを可能にする。トリガコンポーネント３０８は、高リスク変更リストが並列にビルドされると、高リスク変更リストに関するより高速なフィードバックを報告コンポーネント３０６に提供することができる。

【００３９】

[0044]いくつかの実施形態では、ビルドコンポーネント３１０がソフトウェアビルドエラーを返さない場合、トリガコンポーネント３０８は変更リストをアップデートモジュール３１２に送信することもできる。アップデートモジュール１２２は、変更リストに関する履歴データ、および変更リストに基づくソフトウェアビルドの成功または失敗を格納することができる。いくつかの実施形態では、トリガコンポーネント３０８は、単一の変更リストがソフトウェアコードに行われた変更を組み込むことを示す単一の変更リストを受信することができる。トリガコンポーネント３０８は、ビルドが成功または失敗すると、ビルドの結果をアップデートモジュール１２２に送信することができる。次いで、アップデートモジュール１２２は、変更リストがソフトウェアビルドエラーを引き起こしたかどうかを反映するために、履歴データをアップデートすることができる。他の実施形態では、ビルドコンポーネント３１０がソフトウェアビルドエラーを返すと、トリガコンポーネント３０８は、変更リストをビルドブレークモジュール１２２に送信することができる。ビルドブレークモジュール１２２は、ソフトウェアビルドエラー引き起こした変更リストを識別して、開発者にフィードバックを提供することができる。ビルドブレークモジュール１２２は、以下で図４に関連してより詳細に説明する。

【００４０】

[0045]図３のブロック図は、予測分析モジュール１２０は図３に示されるすべてのコンポーネントを含むべきであることを示すことを意図するものではないことが理解されるべきである。むしろ、予測分析モジュール１２０は、より少数の、または図３に示されていないさらなるコンポーネントを含むことができる。たとえば、予測分析モジュール１２０は、ビルドコンポーネント３１０を含まなくてもよい。むしろ、ビルドコンポーネント３１０の機能は、プロセッサ、または他の任意の適切なハードウェアデバイスによって実装され得る。

【００４１】

[0046]図４は、ソフトウェアビルドエラーを分析するために使用されるビルドブレークモジュールの例を示すブロック図である。ビルドブレークモジュール１２２は、図１のコンピューティングシステム１００などのコンピューティングシステムに実装され得る。いくつかの実施形態では、ビルドブレークモジュール１２２は、フィルタコンポーネント４０２、予測コンポーネント４０４、検証コンポーネント４０６、報告コンポーネント４０８、およびバージョン制御コンポーネント４１０を含むことができる。ビルドブレークモジュール１２２のコンポーネントは、ソフトウェアビルドエラーを分析することができる。

【００４２】

[0047]ビルドブレークモジュール１２２は、ソフトウェアコードへの任意の適切な数の変更、およびビルドコンポーネントによって生成されるビルドログを受け入れることができる。ビルドログは、ビルドプロセス中に生成された情報を含むことができる。たとえば、ビルドログは、ソフトウェアビルドエラーを生成しなかったソフトウェアコードの特定の部分を示す情報を含むことができる。いくつかの実施形態では、フィルタコンポーネント４０２は、ソフトウェアビルドエラーの原因ではないと知られているソフトウェアコードへのあらゆる変更を除外することができる。たとえば、フィルタコンポーネント４０２は、ソフトウェアビルドエラーをもたらさなかった以前のビルドに含まれるあらゆる変更を識別することができる。他の実施形態では、フィルタコンポーネント４０２は、アップデートモジュール１２２に格納された履歴データに基づいて、動的な分析を実行することができる。フィルタコンポーネント４０２は、予測コンポーネント４０４にソフトウェアビルドエラーを引き起こした可能性がある候補変更リストのセットを送信することができる。

【００４３】

[0048]予測コンポーネント４０４は、変更リストがソフトウェアビルドエラーを引き起こした可能性を検出することができる。図２に関連して上述したように、ソフトウェアビルドエラーの確率は、ロジスティック回帰または線形回帰などの任意の適切なタイプの回帰を使用して計算することができる。いくつかの実施形態では、予測コンポーネント４０４は、各変更リストがソフトウェアビルドエラーを引き起こした確率を計算することができる。

【００４４】

[0049]いくつかの実施形態では、予測コンポーネント４０４は、変更リスト、および各変更リストがソフトウェアビルドエラーを引き起こした確率を、検証コンポーネント４０６に送信することができる。検証コンポーネント４０６は、各変更リストでビルドを再発生させることができる。いくつかの実施形態では、検証コンポーネント４０６は、まずソフトウェアビルドエラーを引き起こす確率が最も高い変更リストを選択することができる。検証コンポーネントは、ソフトウェアビルドを再作成して、変更リストがソフトウェアビルドエラーを引き起こすかどうかを決定することができる。次いで、検証コンポーネント４０６は、ソフトウェアビルドエラーを引き起こす確率がより低い変更リストを選択することができる。他の実施形態では、検証コンポーネント４０６は、バイナリサーチまたはデルタデバッギングを使用して、ソフトウェアビルドエラーを引き起こす変更リストを決定することができる。検証コンポーネント４０６は、ソフトウェアビルドエラーを引き起こす変更リストを決定して、変更リストを報告コンポーネント４０８に送信することができる。報告コンポーネント４０８は、変更リストおよびソフトウェアビルドエラーを、特にディスプレイデバイス、ＩＤＥで生成されたダイアログボックス、電子メール通知、またはニュースフィードに送信することができる。

【００４５】

[0050]また、報告コンポーネント４０８は、変更リストおよびソフトウェアビルドエラーを、バージョン制御コンポーネント４１０に送信することができる。バージョン制御コンポーネント４１０は、ソフトウェアビルドエラーをもたらすソフトウェアコードへのあらゆる修正を除去することができる。たとえば、バージョン制御コンポーネント４１０は、ソフトウェアコードがコンパイルされて、機械実行可能アプリケーションにリンクされることを防止するソフトウェアコードへのあらゆる変更を除去することができる。また、報告コンポーネント４０８は、変更リストおよびソフトウェアビルドエラーをアップデートモジュール１２２に送信することができ、それは、以下で図５に関連してより詳細に説明する。

【００４６】

[0051]図４のブロック図は、ビルドブレークモジュール１２２は図４に示されるすべてのコンポーネントを含むべきであることを示すことを意図するものではないことが理解されるべきである。むしろ、ビルドブレークモジュール１２２は、より少数の、または図４

に示されていないさらなるコンポーネントを含むことができる。たとえば、ビルドブレークモジュール１２２は、検証コンポーネント４０６を含まなくてもよい。むしろ、検証コンポーネント４０６の機能は、プロセッサ、または他の任意の適切なハードウェアデバイスによって実装され得る。

【００４７】

[0052]図５は、予測分析モジュールをアップデートするために使用されるアップデートモジュールの例を示すブロック図である。アップデートモジュール１２４は、図１のコンピューティングシステム１００などのコンピューティングシステムに実装され得る。いくつかの実施形態では、アップデートモジュール１２４は、特徴抽出コンポーネント５０２、履歴データベース５０４、およびアップデート予測モジュール５０６を含むことができる。

【００４８】

[0053]いくつかの実施形態では、アップデートモジュール１２４は、変更リスト、および変更リストに関連付けられるビルド結果を検出することができる。次いで、アップデートモジュール１２４は、特徴抽出コンポーネント５０２を使用して、変更リストに関連する特性を抽出することができる。たとえば、特徴抽出コンポーネント５０２は、特に、ソフトウェアコードに変更を加えた開発者、変更によって影響を受けたプロジェクト、ソフトウェアコードをコンパイルするために開発者が使用したコンピューティングシステム、ソフトウェアコードへの変更をレビューした個人の数、または変更が提出された時間などの特性を抽出することができる。

【００４９】

[0054]特徴抽出コンポーネント５０２は、変更リスト、ビルド結果、および特性を履歴データベース５０４に送信することができる。履歴データベース５０４は、変更リスト、および変更リストの特性を、ビルド結果とともに表に格納することができる。いくつかの実施形態では、履歴データベース５０４は、予測分析モジュール１２０にデータを送信することができ、予測分析モジュール１２０が、変更リストがソフトウェアビルドエラーを引き起こし得る可能性の正確な確率を計算することを可能にする。他の実施形態では、履歴データベース５０４は、変更リスト、ビルド結果、および変更リストの特性をアップデート予測モジュール５０６に送信することもできる。アップデート予測モジュール５０６は、履歴データに基づいて予測モデルをトレーニングして、予測モデルがしきい値を上回る精度を有する場合、予測モデルを予測分析モジュール１２０に送信することができる。

【００５０】

[0055]図５のブロック図は、アップデートモジュール１２４が図５に示されるすべてのコンポーネントを含むべきであることを示すことを意図するものではないことが理解されるべきである。むしろ、アップデートモジュール１２４は、より少数の、または図５に示されていないさらなるコンポーネントを含むことができる。たとえば、アップデートモジュール１２４は、アップデート予測モジュール５０６を含まなくてもよい。むしろ、アップデート予測モジュール５０６の機能は、プロセッサ、または他の任意の適切なハードウェアデバイスによって実装され得る。

【００５１】

[0056]図６は、ソフトウェアビルドエラーを予測する、有形のコンピュータ可読記憶媒体６００を示すブロック図である。有形のコンピュータ可読記憶媒体６００は、コンピュータバス６０４を介してプロセッサ６０２によってアクセスされ得る。さらに、有形のコンピュータ可読記憶媒体６００は、現在の方法のステップを実行するようプロセッサ６０２に指示するためのコードを含み得る。

【００５２】

[0057]本明細書に記載の様々なソフトウェアコンポーネントは、図６に示されるように、有形のコンピュータ可読記憶媒体６００に格納され得る。たとえば、有形のコンピュータ可読記憶媒体６００は、予測分析モジュール６０６、ビルドブレークモジュール６０８、アップデートモジュール６１０を含むことができる。予測分析モジュール６０６は、ソ

フトウェアコードへの任意の数の変更を検出して、ソフトウェアコードがソフトウェアビルドエラーを含む可能性を予測することができる。ビルドブレークモジュール６０８は、ソフトウェアビルドエラーを分析して、ソフトウェアコードへの変更がビルドブレークを引き起こした可能性を決定することができる。アップデートモジュール６１０は、ソフトウェアコード変更、および対応するソフトウェアビルドエラーの履歴情報を格納することができる。アップデートモジュール６１０は、履歴情報を予測分析モジュール１２０に提供することができ、予測分析モジュール１２０が、ソフトウェアビルドエラーに関する正確な予測を計算することを可能にする。

【００５３】

[0058]特定の用途に応じて、図６に示されていない任意の数のさらなるソフトウェアコンポーネントが、有形のコンピュータ可読記憶媒体６００内に含まれ得ることが理解されるべきである。主題を、構造的な特徴および／または方法に特有の言語で説明してきたが、添付の特許請求の範囲において定義される主題は、上述の特定の構造的な特徴または方法に必ずしも限定されないことが理解されるべきである。むしろ、上述の特定の構造的な特徴および方法は、特許請求の範囲を実装する例示的な形態で開示される。

【図１】

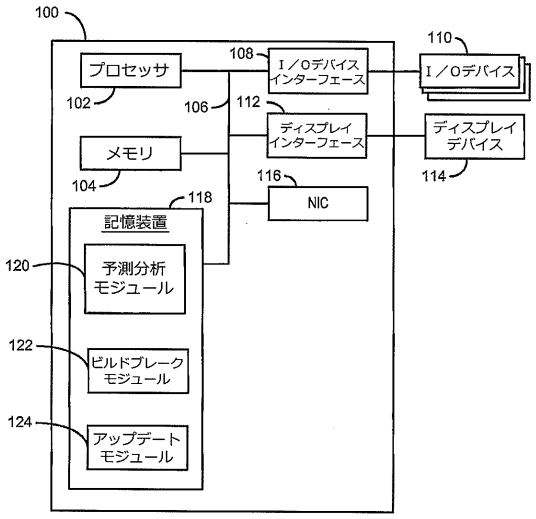

FIG. 1

【図２】

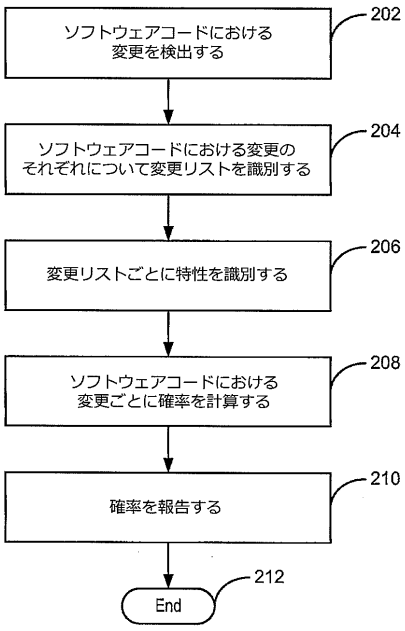

FIG. 2

【図３】

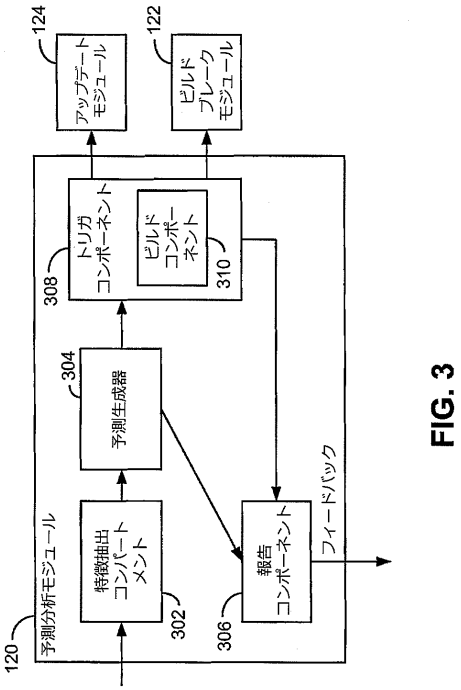

FIG. 3

【図４】

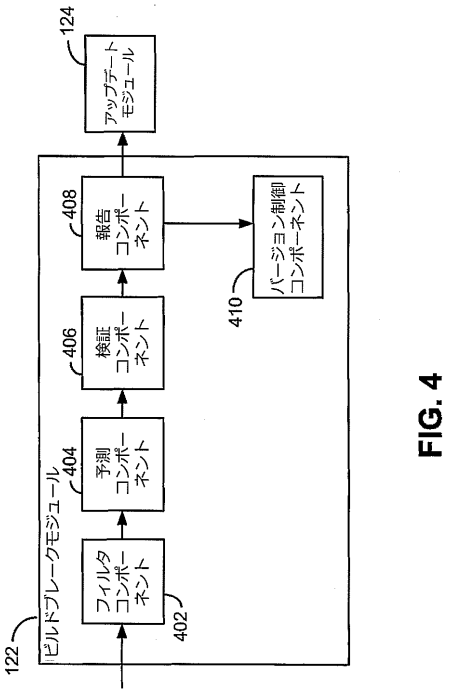

FIG. 4

【図５】

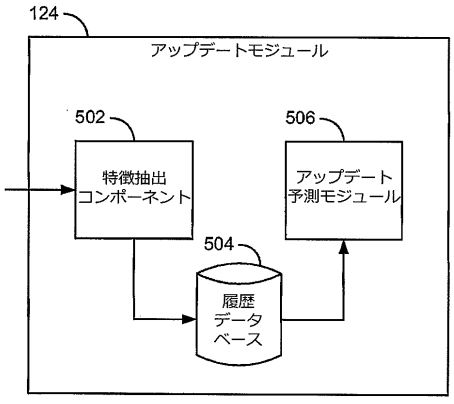

FIG. 5

【図６】

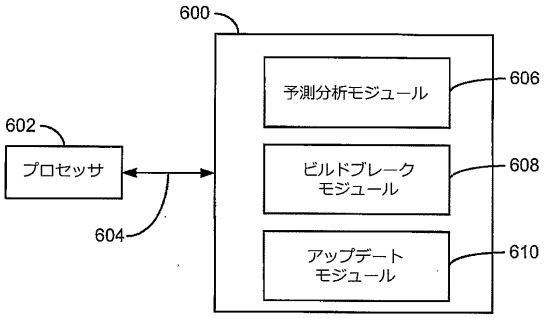

FIG. 6

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055226

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44 G06F11/36
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	T.L. GRAVES ET AL: "Predicting fault incidence using software change history", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 26, no. 7, 1 July 2000 (2000-07-01), pages 653-661, XP055091237, ISSN: 0098-5589, DOI: 10.1109/32.859533 abstract 1. Introduction; page 1 1.1 Software Analysis fault; page 1 - page 2 1.2 Product Measures 1.3 Process Measures; page 2 - page 3 2 CHANGE MANAGEMENT DATA ; page 2 - page 3 3.1 Generalized Linear Models 3.2 Simulations to Assess Significance; -/--	1-10

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 December 2013	07/01/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Eftimescu, Nicolae

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055226

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	page 3 - page 4 4 RESULTS; page 4 - page 7 5 SUMMARY; page 7 -----	
X	ERIK ARISHOLM ET AL: "Predicting fault-prone components in a java legacy system", PROCEEDINGS OF THE 2006 ACM/IEEE INTERNATIONAL SYMPOSIUM ON INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING , ISESE '06, 1 January 2006 (2006-01-01), page 8, XP055091193, New York, New York, USA DOI: 10.1145/1159733.1159738 ISBN: 978-1-59-593218-1 abstract 1. Introduction; page 8 2. Methodology 2.1 Goal 2.2 Fault-proneness factors 2.3 Dependent and independent variables 2.4 Assumptions and caveats 2.5 Design of the study 2.6 Data analysis; page 9 - page 11 3. RESULTS 3.1 Principal Component Analysis 3.2 Univariate Analysis 3.3 Multivariate Analysis 3.4 Cost-Benefit Analysis; page 11 - page 14 4. THREATS TO VALIDITY ; page 14 5. RELATED WORK; page 14 - page 15 6. CONCLUSIONS; page 15 - page 16 ----- -/--	1-10

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055226

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MOSER R ET AL: "A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction", SOFTWARE ENGINEERING, 2008. ICSE '08. ACM/IEEE 30TH INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 10 May 2008 (2008-05-10), pages 181-190, XP031449073, ISBN: 978-1-4244-4486-1 abstract 1. INTRODUCTION; page 181 - page 182 2. RELATED WORK ; page 182 3. CLASSIFICATION ACCURACY AND COST-SENSITIVE CLASSIFICATION 3.1 Assessing Classification Accuracy 3.2 Cost-Sensitive Classification; page 183 - page 184 4. DATA AND EXPERIMENTAL SET-UP; page 184 - page 186 5. EXPERIMENTS 5.1 Standard Defect Prediction 5.2 Cost-Sensitive Defect Prediction ; page 186 - page 188 6. LIMITATIONS 7. DISCUSSION AND CONCLUSIONS; page 188 - page 189	1-10
A	US 2009/089755 A1 (JOHNSON DARRIN P [US] ET AL) 2 April 2009 (2009-04-02) abstract paragraph [0002] - paragraph [0017] paragraph [0024] - paragraph [0040] paragraph [0043] - paragraph [0066] paragraph [0068] - paragraph [0094] paragraph [0096] - paragraph [0112]	1-10
A	LUCAS LAYMAN ET AL: "Iterative identification of fault-prone binaries using in-process metrics", PROCEEDINGS OF THE SECOND ACM-IEEE INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASUREMENT, ESEM '08, 1 January 2008 (2008-01-01), page 206, XP055092376, New York, New York, USA DOI: 10.1145/1414004.1414038 ISBN: 978-1-59-593971-5 abstract the whole document	1-10

-/--

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055226

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	HAMDI A AL-JAMIMI ET AL: "Efficient prediction of software fault proneness modules using support vector machines and probabilistic neural networks", SOFTWARE ENGINEERING (MYSEC), 2011 5TH MALAYSIAN CONFERENCE IN, IEEE, 13 December 2011 (2011-12-13), pages 251-256, XP032102565, DOI: 10.1109/MYSEC.2011.6140679 ISBN: 978-1-4577-1530-3 the whole document -----	1-10

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/055226

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009089755 A1	02-04-2009	NONE	

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ

(72)発明者 バード, クリスチャン
アメリカ合衆国ワシントン州９８０５２－６３９９, レッドモンド, ワン・マイクロソフト・ウェイ, マイクロソフト コーポレーション, エルシーエイ－インターナショナル・パテンツ
(72)発明者 ジマーマン, トーマス
アメリカ合衆国ワシントン州９８０５２－６３９９, レッドモンド, ワン・マイクロソフト・ウェイ, マイクロソフト コーポレーション, エルシーエイ－インターナショナル・パテンツ

Ｆターム(参考) 5B376 BC69