「認知プロセスの誤り」がソフトウェア欠陥の埋め込みに与える影響

Influence of “The error of cognitive process” on embedding of software defect

主査　　：細川　宣啓（日本アイ・ビー・エム株式会社）
副主査　：永田　敦（ソニー株式会社）
研究員（五十音順）：
安楽　啓之（インフォテック株式会社）
石黒　照敏（株式会社デンソークリエイト）
植田　好美（サントリーシステムテクノロジー株式会社）
福田　秀樹（ＴＩＳ株式会社）

研究概要

ソフトウェア開発の現場では，ソフトウェアに欠陥を残さないためにさまざまな取り組みが行われてきた.

しかし,人がソフトウェア開発をするにあたっては,組織の文化,顧客との関係,経験などその人を取り巻く環境の影響を常に受け続けている.

そうした中で人が誤り,欠陥を埋め込むプロセスを分析し,ソフトウェア開発に応用することで欠陥のないソフトウェア開発をするための研究が行われている.

本研究では,過失が発生するプロセスを認知プロセスの誤りの関与によって説明し,実験による検証を試みる.この検証によって,人を取り巻く状況などによる認知阻害が誤りを引き起こし,さらにその認知プロセスの誤りによって過失が発生することを示す.

認知プロセスの誤りに対抗できれば,過失の発生を防ぐことが可能になり,ソフトウェア欠陥の埋め込みを防止する新たな選択肢を示すことが期待できる.

Abstract

In order not to leave a defect to software, the measure is taken at the field of software development.

In this research, participation of "the error of a cognitive process" explains the process which negligence generates, and we try verification. It is shown that negligence occurs by this by "the error of a cognitive process" by the cognitive prevention resulting from environment etc.

Since negligence will not occur if "the error of a cognitive process" can be prevented, it becomes possible to prevent the embedding of a software defect.

1. 研究の背景

1.1　研究の背景

ソフトウェア開発の現場では,開発工程において埋め込まれるソフトウェア欠陥（以下,「欠陥」）を除去,予防するために,日頃から欠陥分析が行われている.

この欠陥分析による品質向上方法には２つのアプローチがある.１つ目は発生した不具合事象から欠陥を検出し,除去するという方法である.これをボトムアップアプローチと呼ぶ.２つ目は,過失による欠陥の埋め込みを想定し,それに対し予防や早期検知を行うという方法である.これは,開発者の誤りを誘発しやすい状況下では,過失の発生と欠陥の混入が起きるというメカニズムに着目したもので,これをトップダウンアプローチと呼ぶ.

ボトムアップアプローチは個々の欠陥に対する直接的な対策になりがちで,状況の異な

る他の欠陥への横展開が難しい.それに対し,トップダウンアプローチは過失を起こす原因を分析し,それに対して対策をするため,前者よりも広範囲で的確な対策が期待できる.よって,本研究ではこのトップダウンアプローチに着目することとした.

1.2　先行研究

本研究では,「欠陥モデリング」[1]の考え方をベースとして,欠陥が混入するメカニズムについて検討していく.「欠陥モデリング」は Project Fabre（2013）[1]により提案された手法である.この手法により,欠陥を誘発するような状況下において,人が過失を犯し,ソフトウェアなどの開発成果物に対して欠陥を埋め込んでしまう仕組みが可視化できる.

欠陥モデリングで示されている誘発因子から表出現象に至る仕組みと,1.1 で述べた「ボトムアップアプローチ」と「トップダウンアプローチ」を図 1 に表現する. 欠陥対策は,Gate1～Gate3（以下 G1,G2,G3）の各段階で実施できる.

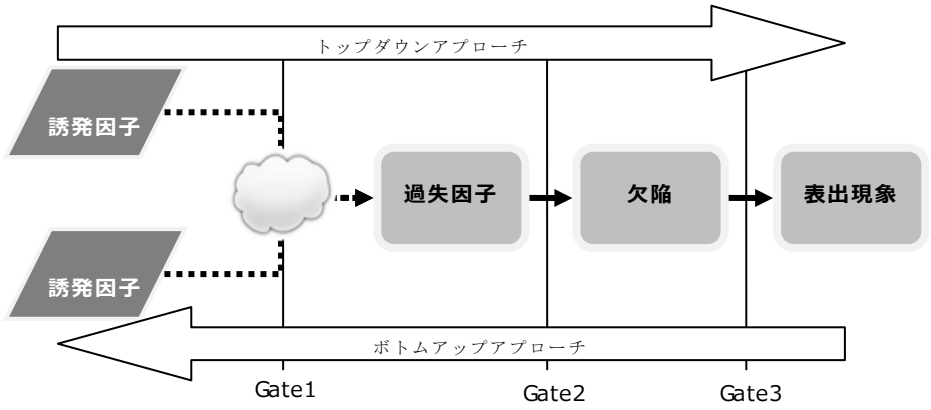


図 1.欠陥が不具合として表出するまでの仕組み

また,それぞれの用語については,Project Fabre（2013）[1]で表 1 の通り定義されている.（付録１参照）

表 1.欠陥モデリングの用語定義

用語	定義内容（抜粋）
誘発因子	成果物の中に含まれる,人間の思考の誤りを誘発する“トリガー”となる要素.
過失因子	人間の思考や判断の誤りそのもののこと.欠陥は過失因子の集合（=連続）として生み出される.
欠陥	成果物に含まれた,人間の思考の過ちが具現,表出化したもの.
表出現象	欠陥によって引き起こされる不具合,障害.

なお, 本研究におけるデータの蓄積方法は,欠陥予測データベースの考え方を採り入れている.欠陥予測データベースは,「ソフトウェア欠陥予測アルゴリズム　～欠陥混入メカニズムのモデリング手法を利用した欠陥予測方法の提案～」(2014)[2]で論じられており,欠陥モデリングを拡張して欠陥予測データベースを構築し,欠陥の発生予測を行った.（付録２参照）

2. 解決すべき課題

より根本的な欠陥対策には,欠陥モデリングおよび欠陥予測データベースを活用し,トップダウンアプローチで臨むのが良い.しかし,研究者にて欠陥モデリングを用い,現実の対策を検討してみたところ,以下のような課題が見つかった.

(1)困難な誘発因子への対策立案

欠陥を埋め込む元になる誘発因子に直接手を打とうとした場合,プロジェクト開始時点で与件であるため受け入れざるを得ず,手が打てないものや対策が非現実的になるものが存在する.（例:「現行システムの改修などで,ハードウェアの保守期限によりカットオーバー時期をずらせず短納期開発を受け入れざるを得ない」,「プロジェクトの所属

する組織はそう簡単に改組できないので,組織の体質には手が打ちづらい」など）

(2)限定的な過失因子への対策立案

過失因子に手を打とうとすると,対策が個々の過失と対になるため,近視眼的になりがちで,真の原因になかなか近づけない.（例：「パラメータ設定を誤る,といった過失に対して“パラメータの状態をチェックする”というチェックリストを作る」など）

(3)誘発因子と過失因子の不確実な因果関係

誘発因子の状況が生じたからと言って,必ずしも過失因子が引き起こされるわけではない.（例:「同じプロジェクトにアサインされた複数の新入社員が,同じ過失を犯すとは限らない」など）

ここで課題(1)(2)は,必ずしも誘発因子,過失因子そのものに対策することが最善ではないことを示唆している.また,課題(3)は誘発因子が直接的に過失因子を引き起こすわけではないことを示している.

この課題の解決策を考えてみる.誘発因子から直接的に過失因子が生み出されるのではなく,その間に何らかの仕組み（図 1 中の雲）が存在し,これが開発者に過失を犯させていると仮定する.そうすると,その仕組みが働かないように手を打つことができれば,過失の発生を予防することができる.つまり誘発因子,過失因子そのものに手を打たなくてよくなり,課題(1)(2)は考慮しなくてよくなる.課題(3)の原因もこの仕組みでうまく説明できる.

欠陥対策は原因に対して行うことが望まれるという一般的な開発モデルの考え方からも,図 1 の G3,G2 でなく G1 の部分で対策を打つことに有効性があると考えられる.

ではその仕組みとは何か.本研究はこの命題を明らかにするために,以下の Research Question を設定した.

RQ：なぜ,人は誘発因子の影響を受け,過失を犯すのか？

3.課題へのアプローチ

3.1 誘発因子・過失因子と認知モデル

過失を犯した人と誘発因子,過失因子の関係から過失発生の仕組みについて考察する.

まず,誘発因子は過失を犯させるトリガーとなる要素のことで,環境や状況が該当する場合が多い.たとえば,「大量のドキュメントの存在」「縦割りの組織の中での開発」「既存システムは安定しているという認識」「未経験」などが例として挙げられ,これらが過失を犯した人の周囲を取り巻き,常に干渉し続けている.

一方,過失因子は人の思考,判断の誤りであり,その主体は過失を犯した人である.（図 2）

この関係を利用して過失因子の原因の分析をしてみたが,それぞれの誘発因子が人に与えている影響が分からないと,過失因子に対する誘発因子の影響を説明することが出来ない.その為,過失因子から因果関係を元に誘発因子を辿ることもできず,結果として,誘発因子が過失因子の原因分析に直接用いることが出来ないという問題が発生した.

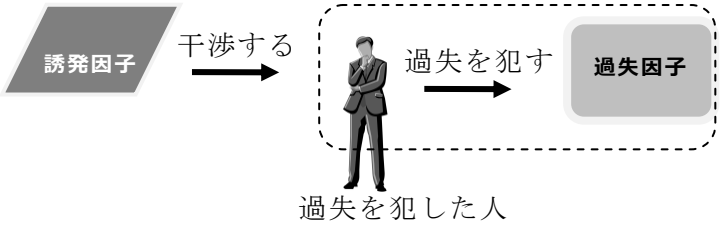
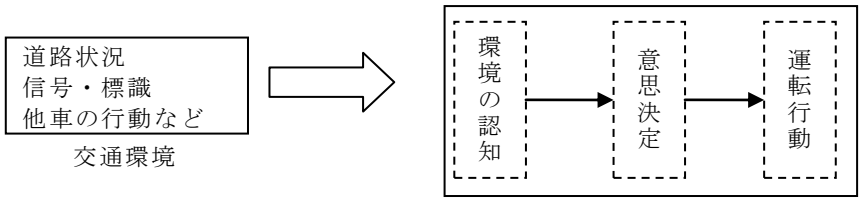


図 2.誘発因子,過失因子,および過失を犯した人の関係

3.2 ヒューマンエラーと認知モデル

3.1 で述べた問題に対して,一般的なヒューマンエラーの仕組みから誘発因子が過失を犯した人に与える影響を考えた. 芳賀繁が「事故が無くならない理由 安全対策の落とし穴」[3]の中で認知モデルを用いて「認知モデルでは交通環境の困難度を正しく認知するこ

と,その認知に基づいて行動を決定すること,そしてその行動を意図したとおりに実行することが重要と考えている.この,認知,判断,実行がうまく出来ないと事故が起きると考える.」と説明している（図 3）.以降,認知,判断,実行の一連のプロセスを「認知プロセス」と呼ぶ.



（「図 5-4 交通事故発生に関するスキルモデルと認知モデル」[3]を元に作成）

図 3.交通事故発生に関する認知モデル

この認知モデルをソフトウェア開発に用いることで,3.1 で述べた誘発因子が人に与えている影響を説明できると考えた.

3.3 誘発因子と認知プロセスの誤り,過失の関係についての仮説

3.2 で述べたこの認知モデルを,ソフトウェア開発における欠陥モデリングに応用し,誘発因子と過失因子の関係に当てはめて説明すると以下のようになる.

・交通環境は,誘発因子に置き換えることができる.

・人間は誘発因子の影響下において認知を行い,判断し,実行する.

・上記の認知プロセス（認知・判断・実行）を誤った結果,過失因子が発現する.

以下にソフトウェア開発に認知モデルを適用した場合の,認知,判断,実行の誤りを具体例として示す.

・認知の誤り：設計書を読んだが,内容を誤解した,仕様で決まってない等の認知の誤謬

・判断の誤り：仕様はあるものの,アルゴリズムを誤った,処理内の条件設定を誤ったなどの作業の判断ミス,問題解決方法の誤り

・実行の誤り：仕様もあって,問題も正しく解決出来ていたが,実際にコーディング時にタイプミスするなどの実行時の誤り

上記をもとに,誘発因子,認知プロセス,過失の因果関係を整理したものを,以下に示す.

(1)誘発因子と認知プロセス

誘発因子がある状況では,開発者の認知プロセス（認知・判断・実行）が阻害され,誤った方向に誘導されるため,「認知プロセスの誤り」が生まれる.

(2)認知プロセスの誤りと過失

(1)により,「認知プロセスの誤り」が生まれ,これら認知プロセス（認知・判断・実行）の誤りが人に過失を犯させる（=過失因子を引き起こす）

上記より,誘発因子が引き金となって認知プロセスの誤りを起こし,その認知プロセスの誤りによって人は過失を起こしているという仮説が立てられた.その場合,たとえ誘発因子を取り除くことが困難な状況であったとしても,認知プロセスそれぞれの誤りに対応することが出来れば,最終的に過失を防ぐ,あるいは緩和することが可能になるはずである.

3.4 認知プロセスの誤りの分類

認知プロセスの誤りの分析については,行動分析学を参考にした.行動分析学は,杉山尚子が「「行動の原因を解明し行動の法則を発見する基礎科学」と「現実社会における人々の問題を基礎科学で発見された法則に基づいて解決していく応用科学」の２つの側面を同時にあわせもつ心理学」[4]と説明しており,石田淳は「教える技術」[5]の中でこれに基づいた教育法を示している.

具体的には,教育の対象を「知識」,「技術」に分類している.知識は情報として知っているもの.一方,技術はそれを実行する為に必要な事項になる.認知プロセスに当てはめる

と,知識は認知に対応し,技術は判断,および実行が該当する.教育手法である当分類に従い整理すれば,効果的な対策を立案できると考えた.

4.実験と実験結果

3.3(1)(2)で立てた仮説を実証するために,実験を実施した.まず,提案する手法によって「認知プロセスの誤り」を導き出す.さらに,それらの「認知プロセスの誤り」が実際の現場において発生していたかどうか,また過失を引き起こす要因となっていたかどうか確認する.

4.1 認知プロセスの誤りの分析方法

認知プロセスの誤りは,欠陥モデルの誘発因子と過失因子の因果関係を元に,その誤りの内容と対象を分析することで導き出した.以下にその分析プロセスを示す.

(1) 認知プロセスの誤りの推定

まず,欠陥モデリングにおける誘発因子と過失因子の因果関係をもとに,開発者の認知プロセスの中で発生が想定される誤りを導出する.①認知に問題はなかったか,②判断に問題はなかったか,③実行に誤りはなかったか,という 3 つの観点から考える.

その際に,誘発因子と過失因子に因果関係が希薄な場合,その誘発因子は分析の対象から外した.過失因子との関係が希薄な誘発因子は,原因を分析してもその結果の納得感がなく,対策を検討しても狙いがぼやけ,問題の解決につながらない可能性が高いためである.

(2) 認知プロセスの誤りのカテゴリ分類

(1)で検討した内容を元に,誘発因子から発生する認知プロセスの誤りを「知識の誤り」と「技術の誤り」のいずれかに分類する. 表 2 の判断基準に従う.

表 2.知識の誤りと技術の誤りの説明,および判断基準

カテゴリ	認知プロセスとの対応	判断基準
知識の誤り	認知に問題がある	誘発因子が状況の認知（入力となる情報,あるいはその認識を誤っている）を阻害している場合
技術の誤り	判断,実行に問題がある	誘発因子が認知による状況判断,あるいは認知に基づく実行を誤らせている場合

(3) 誤りの内容の選択

作業者の過失について最も近いと思われる理由を表 3 から選択する. 誘発因子の影響をうけ,過失を犯した理由はどれか,という観点で考える.

※認知プロセスの誤りの内容についての説明は付録 3 参照

表 3.認知プロセスの誤りの内容

カテゴリ	誤りの内容
知識	未知,誤認,漏れ,誤り
技術	概念の誤解,評価誤り,基準なし,判断ミス,問題の構造化誤り,手順漏れ,作業誤り,ルール不徹底,優先順位誤り

(4)対象の選択

表 4 から誤りの対象として最も適切なものを選択する.

※認知プロセスの誤りの対象についての説明は,付録 3 参照

表 4.認知プロセスの誤りの対象

カテゴリ	対象
知識	要求・仕様,要求機能HWインタフェース,SWインタフェース,ユーザインタフェース,機能記述,プロセス間通信,データ定義,モジュール設計,論理記述,エラーチェック,設計標準,ロジック,計算,データハンドリング,モジュール実装,モジュール関連系,プログラム標準,テストハードウェア,テストソフトウェア,インテグレーションテスト,開発環境
技術	要求獲得,設計,実装,テスト,レビュー,ドキュメント作成,検討,利用,プロジェクト運営

なお,この誤り・対象の選定は作業者の判断に委ねられ,個人の主観の影響を受けることが懸念されるため,複数名によるレビューやすり合わせを行い,合意しつつ進めた.

4.2 実験内容

研究員が作成した欠陥モデルの情報を入力として,以下の手順で実施した.

手順 1：実際に 4.1 で述べた手法を利用して過失の引き金となる認知プロセスの誤りを導き出した.

手順 2:手順 1 で導出した認知プロセスの誤りに対し,欠陥モデルを作成した研究員に,ヒアリングして「導出した認知プロセスの誤りは過失を引き起こす直接的な要因となっているかどうか」を確認した.

ヒアリングした結果を,以下の 2 段階で評価した.

＜導出した認知プロセスの誤りの妥当性＞

○：現実的に洗い出した認知プロセスの誤りにより過失が発生していた.

×：導出した認知プロセスの誤りと過失は無関係である.

4.3 実験結果

4.3.1 導出した認知プロセスの誤り（手順 1 の結果）

今回の実験で導出した認知プロセスの誤りは,知識関連では,未知:6 件,誤認:6 件,漏れ:5 件,誤り:4 件,技術関連では,概念の誤解:8 件,基準なし:3 件,判断ミス:5 件,手順漏れ:1 件,作業誤り:1 件,ルール不徹底:1 件（全 40 件）となった.

表 5 に誘発因子/過失因子と認知プロセスの誤りの例を示す.

表 5. 誘発因子/過失因子と認知プロセスの誤り

事例	誘発因子	過失因子	導出した認知プロセスの誤り
1	他社作成のプログラムがベースとなっている.	プログラムを誤って修正した.	知識の誤り（未知）
2	巨大な関数.	プログラムの修正を一部漏らした.	知識の誤り（漏れ）
3	担当者の経験が浅い.	異常状態を想定できず,回復不能な設計とした.	技術の誤り（概念の誤解）
4	熟練者が担当している.	境界値の算出を誤った.	技術の誤り（判断ミス）

4.3.2 手順 2 の結果

手順1で導出した40件の認知プロセスの誤りに対して評価した結果を図 4 に示す.「洗い出した認知の誤りにより過失が発生していた」割合は 87.5%であった.本手法によって導き出した認知プロセスの誤りによって過失は発生していたと言える.(ヒアリングで確認した内容については表 6 を参照.)

過失とは無関係（×）と判定された認知プロセスの誤りのうち,1 件は元々想定しており受容していた誤りで,残り 4 件は他の認知プロセスの誤りと誤って導出されたものである（例えば,「未知」が生じていたが,「誤認」を選定した）.

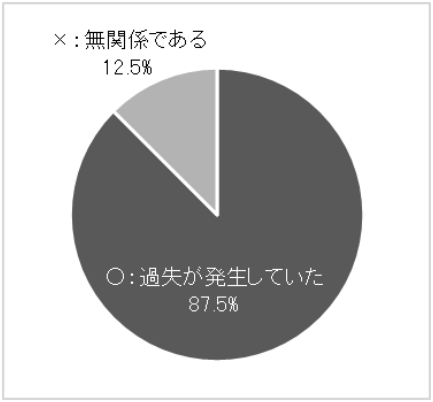


図 4　導出した認知の誤りによる過失が発生していたかどうか

表 6.導出した認知プロセスの誤りが要因となっていたかどうか

事例	導出した認知プロセスの誤り	ヒアリングで確認した内容
1	知識の誤り（未知）	他社が開発したプログラムで,事前の調査も不足しており,製作担当者はベースの構造を熟知していなかった.実際に知識の誤り（未知）が発生していたと言える.
2	知識の誤り（漏れ）	変更対象の関数は分岐,コメントが多く,保守が困難な状態であった.担当者は変更内容を知っていたが,上記状況に伴って,一部の修正対象箇所に気付かず,漏らしてしまった.実際に知識の誤り（漏れ）が発生していたと言える.
3	技術の誤り（概念の誤解）	対象の概念を熟知しておらず,誤った設計を行ってしまった.従って,技術の誤り（概念の誤解）が発生していたと言える.
4	技術の誤り（判断ミス）	今回の担当者はプロジェクトに長く所属しており,ドメイン知識も豊富な熟練者のため,問題なく開発できるという誤った判断により,レビュー等が疎かになった.実際にマネジメントで技術の誤り（判断ミス）が発生していたと言える.

5.考察

4.の実験の結果を元に,以下の通り考察した.

5.1　本研究の成果

本研究により導かれた「認知プロセスの誤り」の概念は,2 に挙げた(1)～(3)に対して有効な,原因分析の対象となることを,表 5 の事例１を例に説明する.

誘発因子の「他社作成のプログラム」は,前提条件となっており,除去することは困難である.これが 2.(1)「困難な誘発因子への対策立案」に該当する.また,過失因子「プログラムを誤って修正した」に着目すると,「チェックリストの用意」,あるいは「変更点を網羅するテストの実施」などの対症療法的,限定的な対策になりがちである.これが 2.(2)「限定的な過失因子への対策立案」に該当する.これらに対し,認知プロセスに着目することで,「未知のプログラム」への対策,たとえば「そのプログラムの開発者の支援を受ける」「サンプルプログラムを作り,メンバーで共有」等の有効な対策を検討することができる.また,4.の実験の結果により,研究員の属する過去プロジェクトで誘発因子・過失因子を元に導き出した「認知プロセスの誤り」が起きており,「認知プロセスの誤り」が過失の発生に影響を及ぼしていることが分かった.このことから,誘発因子が発生する環境下では「認知プロセスの誤り」が生まれ,その結果,人は過失を犯すと言える.これにより 2.(3)で述べた「誘発因子と過失の不確実な因果関係」は「認知プロセスの誤り」により説明できる.

また,本実験にて認知プロセスを導出する際，入力した欠陥モデルにおける誘発因子と過失因子に因果関係がないものが見つかった.今回は複数の研究員で誘発因子と過失因子の妥当性を話し合い,不要な誘発因子を削除した上で分析を行った.この削除を行うことで,導き出した誘発因子に納得感が生まれ,的確な誤りを導出できたと考える.そのため,4.1 に示す分析方法に誘発因子の妥当性判断を導入している.本分析の質を高めるために,洗練された欠陥モデルを使用することや誘発因子と過失因子の関係に対するレビューを実施することを推奨する.

実験の結果からある誘発因子が発生している環境下では,特定の「認知プロセスの誤り」が生まれやすいことが分かった.例えば,誘発因子「動作実績のあるソフト」の元では，知識の誤りの「誤認」が多く生じており,また，誘発因子「短納期の開発」や「経験が不足している」では,技術の誤り「概念の誤解」が生じる可能性が高くなっていた.すなわち,誘発因子が引き金となって発生する「認知プロセスの誤り」には関連があると想定できる．これに着目し,誘発因子から発生が推定される「認知プロセスの誤り」を導き出すこと.「認知プロセスの誤り」に対して予防,再発防止につながる対策を立案すること.これらを行うことができれば,最終的に過失防止につながることが期待できる.

5.2　妥当性への脅威

今回の実験では,研究員の作成した欠陥モデルを元に実験を行っており,知識の誤り「未知」はある研究員の属するプロジェクトのみに存在するなど,認知プロセスの誤りの発生に偏りが生じた.つまり,データ件数の少なさやサンプルデータの偏り（所属する業界・業種が限定的）があり,有効性についてさらに確証を得るためには,さらなる精度向上のためのデータ量の拡充や幅広い業界･業種のデータの収集,カバレッジ率の向上などが必要と考える.

また,実験の中では「設計書の内容が曖昧になっている」という誘発因子から内容が明確でないために誤って伝わったと分析し,知識の誤り「誤認」を導出したが,実際には担当者の認識は正しく,単に作業を誤って欠陥を混入させたことが事実であり,間違った結果に誘導されたものが見られた（実験結果の×が該当）.欠陥モデルの誘発因子の記載に情報が不足していることが理由の一つと考えられる.本手法の入力となる欠陥モデルの作り方の改善が今後の課題となる.

残課題はあるものの,上記で述べたように認知プロセスの誤りに焦点を当てて対策を講じることで,過失の発生を予防できると考える.認知プロセスの誤りを活用し,ソフトウェ

ア開発における欠陥の未然防止に貢献していく.

6.まとめ

本研究により,誘発因子により発生する認知プロセスの誤りが,人の過失を引き起こすことを先行研究結果,および実験により示した.

(1)本論文にて用いた手順により認知プロセスにおける誤りの分析を行った.

(2)認知プロセスの誤りについての分析結果を当事者へ確認した結果,実際にその誤りが起きていたことを確認した.

今後,ヒューマンエラーに関する知見をソフトウェア開発に応用し,「認知プロセスの誤り」に着目した分析を行うことにより,新たな原因分析,および対策の立案につながることを期待している.そのためには,以下に挙げる内容を明らかにする必要があると考える.

・「認知プロセスの誤り」についての分析手法の確立
・分析された「認知プロセスの誤り」の影響度評価
・「認知プロセスの誤り」への対策立案方法

これらの有効性を示し,実際の運用に耐えうる事例を積み上げていくことにより,ソフトウェア欠陥を埋め込まないような開発の実現へと寄与できることを目指していきたい.

参考文献

[1]Project Fabre(細川宣啓ら),"過失に着目した欠陥のモデリング",JaSST2013,2013
[2]細川宣啓,永田敦,柏原一雄,岡本晃,鈴木裕一郎,田村光義,東久保理江子,保栖真輝,「ソフトウェア欠陥予測アルゴリズム」,日本科学技術連盟 SQiP 研究会,2014
[3]芳賀繁,事故がなくならない理由 安全対策の落とし穴,PHP 研究所,2012
[4]石田淳,「行動科学を使って出来る人が育つ！教える技術」,p54-56,かんき出版,2011
[5]杉山尚子,「行動分析学入門－人の行動の思いがけない理由」,集英社 e 新書,2014
[6]細川宣啓,永田敦,太田範,奥山剛,松本達平,渡邊孝志,「自律した品質改善活動に寄与する欠陥特性の提案」,日本科学技術連盟,SQiP 研究会,2014