



ソフトウェアバグ発生時の 真因分析手法の策定

片岡 智美*・古戸 健・松本 達治

The Analyzing Method of Root Causes for Software Problems — by Tomomi Kataoka, Ken Furuto and Tatsuji Matsumoto — In this technical paper, the authors propose an analyzing method of the root causes for software problems. To prevent the recurrence of the same problems, it is necessary to logically identify the root causes and take appropriate measures. Therefore, the authors applied “the 5 whys analysis,” a technique that has been used mainly in the manufacturing process, to the trouble shooting of software. First, the authors visualized the procedures of software design by arranging documents illustrating work pieces made in each process to find out the procedures that involve errors. Next, they prepared many “5 whys” samples related to software development, so that even inexperienced users can be guided by referring to them. Consequently, the method effectively worked on the software development and successfully shortened the lead time required for identifying errors and analyzing their causes.

Keywords: software, trouble, analyze method, 5 whys

1. 緒 言

ソフトウェアの品質は適切なプロセスで確保する必要がある、その開発プロセスは、実際の運用で生じた問題点をフィードバックして、日々改善することが求められる。特にソフトウェアのバグ発生時には、適切な再発防止策を立案するため、プロセス上の真の要因を究明する必要がある^{(1),(2)}。

一方、トラブルの要因を系統的にモレなく出し切るための分析手法に「なぜなぜ分析」がある⁽³⁾。この手法は、TQM^{※1}の一環として、製造業を中心に用いられている改善の方法論として知られている⁽⁴⁾。

当社のソフトウェア開発現場でも、バグ発生時の要因分析になぜなぜ分析手法を適用してきた。しかし、ソフトウェア開発に適用する際の具体的な手順がなく、分析作業は分析者個人の力量に委ねられていたため、十分な分析品質が得られないことによる手戻りも多く発生していた。その結果、分析に時間がかかり、バグ発生からプロセス改善までのリードタイムが長くなる傾向にあった。

そこで、ソフトウェア開発になぜなぜ分析を適用する際の課題を洗い出し、高品質な分析を効率的に行うための手順を策定したので、報告する。

2. なぜなぜ分析適用時の課題

真因分析の手順を策定するにあたり、当社のこれまでのなぜなぜ分析の適用事例からその傾向を調査し、ソフトウェア開発になぜなぜ分析手法を適用する際に分析担当者が陥りやすい問題を洗い出した。その結果、ソフトウェア

開発特有の問題は大きく三つあることがわかった。

第一の問題点は、なぜなぜ分析結果がプログラムのバグを引き起こすメカニズムの追求に終始するケースである(図1(a))。この例では、「特定フレームを送信しない」というバグ事象に対し、「通信ドライバの初期化処理の誤り」という要因を導出している。しかし、これはあくまでバグ事象を顕在化させている原因に過ぎず、当該バグの修正方法を導くことはできても、バグを作り込んだプロセス上の要因を明らかにすることはできず、適切な再発防止策の立案に結びつけることは困難である。その点で、図1(a)のケースは分析が不十分である。

第二の問題点は、分析者の思い込みや印象に頼った分析が行われるケースである。なぜなぜ分析では、分析をする前に、その問題とすべき対象物やことがらを整理して、事実のみを把握することが重要である⁽³⁾。しかし、図1(b)の例では、事実と判断できる確証がないにも関わらず、設計者の心情やレビュー時の配慮に言及するなど、この事実認識が主観的で漠然としたものとなっている。工場の製造設備などで問題が発生した場合は、製造設備や材料、設備の操作履歴などから原因となる客観的事実を収集できる。しかし、ソフトウェア開発はほとんど全工程が人手による作業なので、誤りが入り込んだ工程も人為的なミスであることが多く、事実の認識が担当者の主観に大きく左右されやすい。従って、設計書などの客観的な確証をもって事実確認するべきである。

第三の問題点は、上流工程からの入力情報の誤りに順々

に着目してしまったが故に、なぜの繰り返し回数が十分であるにも関わらず、深度が不足するケースである(図1(c))。このケースでは、確かに担当者の「作業工程」に着目し、かつ、詳細設計書から基本設計書へと客観的な確認である「設計書」をたどって誤りの発生した工程を特定している。しかし、誤りの入り込んだ工程の特定に「なぜ」の分析ステップを費やした結果、誤った作業が「なぜ」行われたのかという本来必要な真因分析がなされていない。このような作業誤りが発生した工程の特定までの作業は、本格的な「なぜなぜ分析」作業の前工程で完了させておくことが、より深い真因分析のためには必要である。

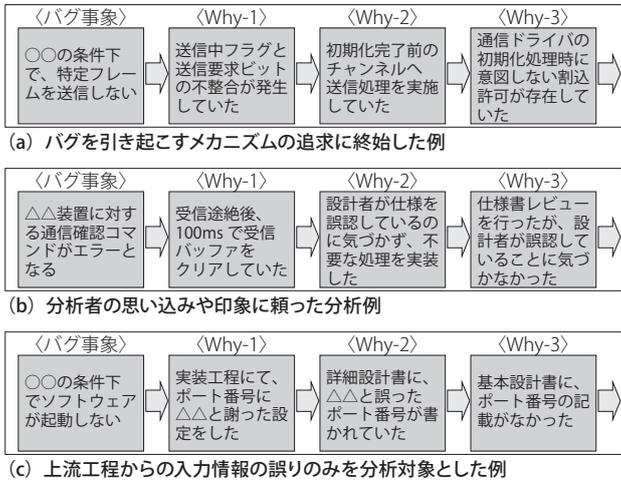


図1 従来のなぜなぜ分析適用事例

3. 課題解決方針の策定

目的とするソフトウェアバグ発生時の真因分析手順を策定するにあたり、2章で示した課題の解決方針を検討した。当社の開発プロセスはV字モデルであり、解決方針を検討するにあたっては、この開発プロセスに即したものとなるよう留意した。なお、V字モデルとは、ソフトウェア開発プロセス手法の一つで、図2に示すように、V字の左側で

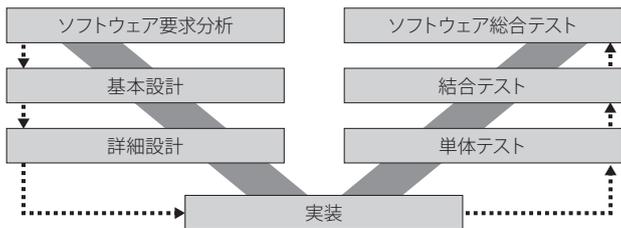


図2 V字モデルと開発プロセス

上流から下流に向けて、要求を詳細化し実装するプロセスと、V字の右側で下流から上流に向けて、実装した結果が要求のとおり正しく機能していることを検証するプロセスで構成される開発手法である。

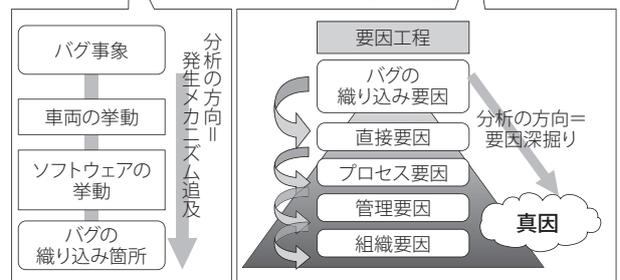
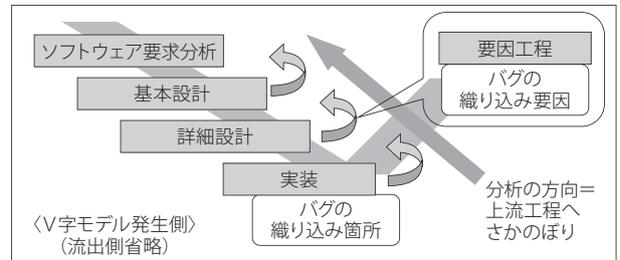
【解決策1】 第一の課題「プロセス上の要因分析が不十分」については、「プログラム上のメカニズム」調査と「作業プロセス上の誤り」調査を分けて行うことで、前者の調査に偏らない事実調査を行う手順を策定することとした(図3【手順1】)。

【解決策2】 第二の課題「客観的な事実収集が不十分」については、ソフトウェア開発プロセスの初期段階からプログラム製作段階に至るまでの間に作成されるすべての設計書をプロジェクトの登録文書からリストアップし、どの工程の設計書が誤りを含んでいるかを調査することで、主観に左右されない確認収集を行うこととした。

【解決策3】 第三の課題「工程内の真因追及が不十分」については、[解決策2]で述べた方法で収集した設計書を工程の順序に並べ、どの工程までの設計書が正しく、どの工程から設計書に誤りが含まれるかを調べることで、V字モデルのソフトウェア開発プロセス上で誤りが織り込まれた工程を特定することとした(図3【手順2】)。

ここまでの分析作業を機械的な手順で行うことにより、

【手順2】プロセス上の要因分析



【手順1】発生メカニズム調査 【手順3】なぜなぜ分析

図3 あるべき分析の流れイメージ図

図3の【手順3】で示すような、プロセスルール上の要因、管理上の要因、組織をとりまく風土など、誤り発生の背景となるより深い要因まで分析する作業により多くの時間を割ける手法を策定することにした。

4. ソフトウェアバグ発生時の真因分析手順の策定

4-1 バグ発生時の真因分析の流れ

3章で示した方針に従い、バグ発生時の真因分析手順を策定した。真因分析は、図3で示した、「あるべき分析の流れ」に沿って行うものとし、次の(1)～(3)の手順で実施する。

(1) 【手順1】発生メカニズム分析

プログラム上で、対象のバグ事象を引き起こしているメカニズムを追求し、バグの織り込み箇所を特定する。

(2) 【手順2】プロセス上の要因分析

V字モデル開発プロセスに沿って、設計書類を調査し、誤りが織り込まれた工程（要因工程）とその誤り内容（バグの織り込み要因）を抽出する。

(3) 【手順3】なぜなぜ分析

要因を深掘りし、バグが織り込まれた真因を導出する。

以上の手順に、3章で示した解決策織り込むため、図4に示すツール①～③を用意した。以降に、各々のツールの詳細を説明する。

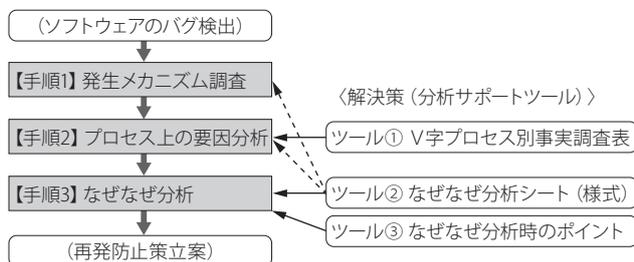


図4 バグ発生時の真因分析手順と解決策

まず、一連の手順通りに、順を追って確実に作業できるよう、図5に示すツール②「なぜなぜ分析シート」を用意した。このシートには、個々手順の出力となる一連の項目に対応した記載欄を設けた。この欄を順に埋めることで手

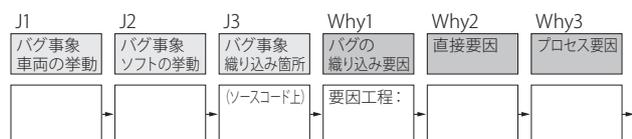


図5 なぜなぜ分析シート（一部抜粋）

順のモレを防ぐことができるとともに、分析者以外のメンバーが分析結果を客観的に確認できるよう工夫した。

4-2 V字モデルのプロセスに沿った事実調査 続いて、バグを織り込んだ時期の事実の把握を行い、「要因工程」の絞り込みと「バグの織り込み要因」の特定を行う手順について説明する。

この手順では、事実を正しくモレなく調査する事が重要であり、この作業を進めるために、図6に示すツール①「V字プロセス別事実調査票」を用意した。この帳票では、開発プロセス名と調査項目をマトリクス上に配置しており、分析に不慣れな作業者であっても、順を追って欄内に記載していくことで無駄やモレなく調査を行えるよう工夫した。

No.	手順(2): 構成管理情報		手順(3): 事実・問題点		要因工程
	成果物構成情報	レビュー時記録	変化点抽出 ・事実の把握	あるべき姿に 対する問題点	
1	フォルダ名 ・ファイル名 ・文書番号 ・作成日 ・更新日付 ・担当者 など	フォルダ名 ・ファイル名 ・文書番号 ・実施日 ・レビュー ・実施形態 ・レビュー記録 など	基本設計書 「条件a=0のとき、負荷 B=OFF→ON」 詳細設計書 「条件a=1のとき、ポート Bに1をセット」 ソースコード上 「条件a=1のとき、ポート Bに1をセット」	手順(4): 発生側要因工程 (正) 「条件a=1とすべき ところが、a=0となっ ていた。」 「条件a=1とすべき ところが、a=0となっ ていた。」	発生
2	基本設計				
3	詳細設計				流出
4	実装				
5	単体テスト	手順(1): 発生メカニズム分析 結果より転記	結合テスト仕様書 条件aに関するテスト項目 がなかった	「条件a=0のとき負荷 B=OFF→ONのテスト項目 がなかった。」	
6	結合テスト				
7	ソフトウェア 総合テスト				

図6 V字プロセス別事実調査票

V字プロセス別事実調査票を使った具体的な調査手順を以下に示す。

- 図4の【手順1】発生メカニズム調査の結果を元に、実装プロセスの事実・問題点欄にバグ織り込み箇所とその内容を記載する。
- プロジェクトの登録文書を参照し、当該バグが織り込まれた時期の各プロセス成果物の作成およびレビュー時の記録情報を記入する。
- (2)で収集した成果物を参照し、(1)に記載したバグ織り込み内容と関連する事実や誤りの有無を確認に基づいて洗い出し、事実・問題点欄に記載する。
- (1)に記載したバグ織り込み工程から順に、事実・問題点欄の記載を参照しながら上流側プロセスにたどり、一つ上流の工程までは正しく、当該工程に誤りがある場合に、当該工程を発生側要因工程と判断することで、発生側の要因工程を特定する。図6に示す例では、事実・問題点欄の記載内容から、「詳細設計」工程に「条件a=0」の誤りが含まれるが、一つ上流の「基本設計」工程は正しいことから、「詳細設計」工程を発生側要因工程としている。

(5) 基本的には、発生側要因工程に対し、V字モデル上で相対する（V字の右に当たる）検証工程の、一つ上位の工程が流出側要因工程であると考えて、流出側の要因工程を導く。図6に示す例では、(4)で発生側要因工程と特定した「詳細設計」工程に相対する検証工程である「単体テスト」工程の一つ上位にあたる「結合テスト」工程を、流出側要因工程としている。この考え方は、V字モデルでは、そのバグが織り込まれた発生側（詳細化・実装）工程に相対する流出側（検証）工程の一つ上位の工程で検出すべきという原則に基づいている。なお、分析者が、この考え方に不慣れな場合でも手順通りに特定できるよう、図7に示すプロセスおよび成果物の流れを用意し、発生側要因工程から流出側要因工程を導出する手助けとした。

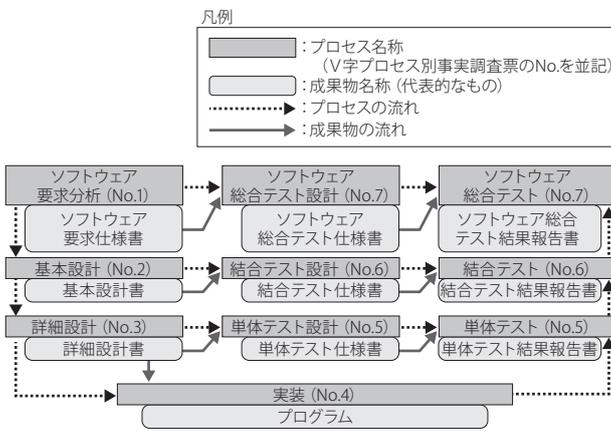


図7 V字モデルとプロセス成果物の流れ

(6) 以上の手順で特定した、発生側要因工程および流出側要因工程に対する、「あるべき姿に対する問題点」欄の記載内容を、次項で説明するなぜなぜ分析のWhy1に設定する。

これまでに説明した手順を用いることで、従来は、事実調査が十分でなく、しかも分析担当者の直感に頼っていたバグの織り込み工程の特定作業を定型化でき、また、確認を用いて客観的に分析結果を判断できる仕組みが整った。

4-3 なぜなぜ分析の実施 次に4-2で導き出した発生側および流出側のWhy1をスタートとして、図5のなぜなぜ分析シートを使用してなぜなぜ分析を行う。

真因を導くためには、要因を規則的に順序よく深く掘り下げる必要があるが、これまでは適用事例も少なく十分なノウハウがないという問題があった。なぜなぜ分析のノウハウが示された一般的な文献もあるが、製造設備上の問題を対象とするサンプルが多く、ソフトウェアのバグを対象

に分析する際には参考になりにくかった。

そこで、不慣れな分析者でも品質の良い分析を行うことができるように、参考文献(5)で提案されている「なぜなぜ分析実施の10則」などを参考に、ソフトウェア開発で分析時に陥りやすい誤りを整理し、ソフト経験に即したサンプルを付記して、表1に示すようなツール③「なぜなぜ分析時のポイント」として整備した。

表1 なぜなぜ分析時のポイント(例)

No.	ポイント	ソフトウェア開発における例(×:悪い例、○:良い例)
B4	流用元PJの潜在バグも自PJ視点で分析する	(×) 流用元PJのシステムデザインにバグが含まれていた。 →流用元PJ開発時に作り込まれたバグであるため、発生要因は不明。 (○) 流用元PJのシステムデザインにバグが含まれていた。 →流用時に、必要な成果物が揃っていないにも関わらず、当該PJへ適用する際の差分検証を実施しなかった。
B8	対個人ではなく仕組みとしての視点で分析する	(×) 担当者のスキルが低く、分析が不十分だった。 (○) 車両の振る舞いという観点から、エラー時の挙動の分析が不十分だった。 (○) 担当者は必要な教育を受けていないにも関わらず、デザインリーダとしてアサインされた。

5. 開発プロセスへの適用と手順導入の効果

4章で述べた一連の分析手順は、試行運用を経て2010年7月に量産開発部門の標準ルールとして採用された。SQA*2部門の指導・監督の下、各開発プロジェクトの担当者が、バグ発生時に本手順に基づいて分析を行うという体制で運用を開始した。

SQA部門が2010年度末にとりまとめたソフトウェア品質活動実績によると、バグ発生から真因分析完了までのリードタイムは2009年度と比較して約40%短縮でき、バグ発生からプロセス改善までのトータルリードタイムは約1/3になり、問題の再発防止策を策定するまでのスピードが向上できた。

本稿で述べた各手順のソフトウェア品質向上への寄与を分析するため、SQA部門にアンケート調査を行い、今回定めた方策のツール①「V字プロセス別事実調査票」ツール②「なぜなぜ分析シート」ツール③「なぜなぜ分析時のポイント」のそれぞれについて、「リードタイム短縮」と「分析品質向上」の観点で評点を算出した。その結果を図8に示す。別途実施したSQA部門へのヒアリング結果と合わせて手順導入の効果を考察した。以下にその考察結果を述べる。

4-1【手順2】「プロセス上の要因分析」で使用するツール①「V字プロセス別事実調査票」は、客観的な事実確認に基づいてバグ織り込み工程の特定ができることから、事実関係の把握が正確にでき、分析品質の向上と共にリードタイム短縮にも有効であることが分かった（共に評点3.4）。更に、「調査票作成の手間は増えるが、事実確認が十分でないことが理由の分析の手戻りが減り、結果的にリードタイム短縮に貢献している」とのコメントも得られた。このことから、事実調査の品質向上が、源流対策推進活動に寄与できていることがわかった。

しかし、4-1【手順3】「なぜなぜ分析」で用いるツール②「なぜなぜ分析シート」およびツール③「なぜなぜ分析時のポイント」については、前者はリードタイム短縮には一定の効果があるものの、何れの手順も現時点での分析の品質向上効果への寄与は判断できかねた。その理由として、なぜなぜ分析の品質には、分析者の熟練や慣れが大きく影響するためと考えられた。そのため、ツール③を教育用資料として活用するとともに、過去事例を整理し、当社ソフトウェア開発に特化した分析ノウハウをさらに蓄積し活用していくことが重要と考えている。

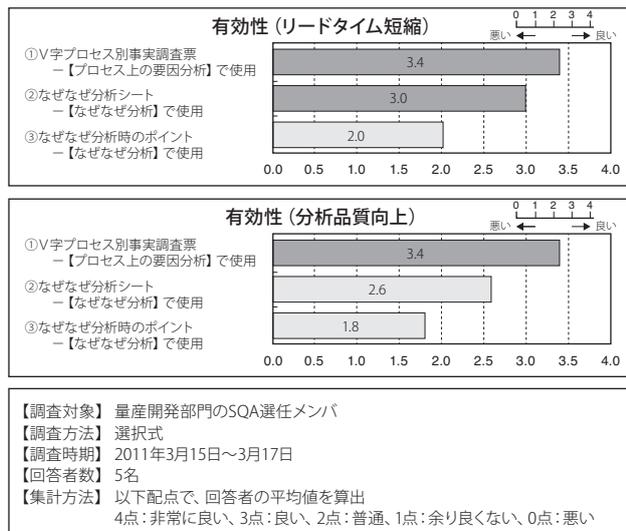


図8 SQA部門へのアンケート結果

6. 結 言

本稿では、高い品質が求められる車載ソフトウェアのプロセス改善活動をより効率的に行うために策定したなぜなぜ分析運用手順と、その導入効果について述べた。

今回策定した手順は、V字モデルを採用する車載ソフトウェア開発に限らず、開発プロセス上で発生した問題の真因分析に広く応用できる手法であると考えている。

分析品質の点ではまだまだ改善の余地があり、今後は、さらに分析品質を高めるため、教育面でのツールの活用を促進するとともに、当社開発プロセスや対象製品に特有の分析事例を蓄積し、なぜなぜ分析時のノウハウとして活用を進めていく。

用語集

※1 TQM

Total Quality Management：総合的品質管理。

※2 SQA

Software quality assurance：ソフトウェア品質保証。

参 考 文 献

- (1) ソフトウェアエンジニアリング研究所、「能力成熟度モデルのキープラクティス1.1版」、CMU/SEI-93-TR-25、カーネギーメロン大学(1993)
- (2) 寺久保敏 他、「CMMレベル3に準拠した車載向けソフトウェア開発プロセスの構築」、SEIテクニカルレビュー第166号、pp45-50
- (3) 小倉仁志、「なぜなぜ分析徹底活用術－「なぜ？」から始まる職場の改善」、JIPMソリューション(1997)
- (4) 早川勲 他、「ソフトウェア品質シンポジウム2008、「ソフトウェア開発へのなぜなぜ5回の適用」、pp185-194
- (5) 小倉仁志、「なぜなぜ分析を使いこなそう！なぜなぜ分析徹底攻略ドリル」、JIPMソリューション(2002)

・CMMは、カーネギーメロン大学の登録商標です。

執 筆 者

片岡 智美*：(株)オートネットワーク技術研究所
ソフト開発センター
車載ソフトウェアの開発プロセス、開発環境構築業務に従事



古戸 健：(株)オートネットワーク技術研究所 ソフト開発センター
主任研究員

松本 達治：(株)オートネットワーク技術研究所 ソフト開発センター
センター長

*主執筆