

第3分科会 (Bグループ)

組み込みソフトウェア開発現場のための「過去トラ」の有効活用
～高品質・短納期を目指して～

Method of effective leveraging previous bugs for engineers of embedded software
- Achieve high quality and low cost -

主査	飯泉 紀子	株式会社日立ハイテクノロジーズ
副主査	田所 孝文	株式会社山武
研究員	玖野 真也	富士ゼロックス株式会社
	城 多寿子	日本電気通信システム株式会社
	佐潟 昌邦	矢崎計器株式会社
	高波 元	アルパイン株式会社

研究概要

組み込みソフトウェア開発では、その開発規模が年々増大する中、高品質・低コストかつ短納期の達成を強く求められている。そのため、開発の上流工程での品質作り込みと下流工程での手戻り低減が、ますます重要になってきている。開発現場ではさまざまな改善活動が行われており、そのひとつに過去のトラブル事例(以下、「過去トラ」と呼ぶ)の活用がある。これは、不具合事例を収集し共有することで、手戻りの予防や、再発クレームの削減を目指すものである。しかし、実際に発生している手戻りやクレームの内容を見ると、「過去トラ」の教訓やノウハウを活用していれば防げたものも多く、「過去トラ」が有効活用されているとは言いがたい状況である。そこで組み込みソフトウェアの開発現場で「過去トラ」を有効活用するための施策を検討し、検証を行った。

Abstract

The development scale of embedded software is enlarging year by year, however short period development with high quality and low cost is strongly required. In order to satisfy this requirement, improvement quality level in early stage of development process and reduction of manpower for reworking in next process are very important. Various improvement activities have gone on development sites. Leveraging previous bugs is one way to achieve prevention of reworking manpower and reduction of repeated complaints by collecting and sharing previous bugs. But actually there is large manpower and a lot of complaints, since we can't leverage the lessons and know-how that we experienced in previous bugs. These bugs have not been effectively used. Therefore we researched a way of effective leveraging previous bugs for engineers of embedded software.

1. 背景

組み込みソフトウェア開発を担当する設計開発部門では、QCD(品質、コスト、納期)を確保するためにさまざまな改善活動を実施し、再発防止に取り組んでいる。しかし、どうしても納期優先となってしまう、上流工程での設計品質に問題を残したまま下流工程に進んでしまうことが多々ある。設計品質の問題を残したまま下流工程のテストを実施すると、その工数が当初計画より超過してしまい、十分なテストを実施できなくなる。その結果、開発当初(プロジェクト発足時)に立てた品質目標を達成できずに、不具合を市場に流出させてしまう。

この傾向は、情報処理推進機構(IPA)「2008年版組み込みソフトウェア産業実態調査報告書」[1]にも表れている。付録1に示すデータによれば、ソフトウェア開発工程ごとの工数比率ではソフトウェアテストが最も多くの工数を要するにもかかわらず、製品出荷後の品質問題の原因はソフトウェアの不具合によるものが約8割を占めている。また、品質問題の対策費は、1件あたりでも総額でも、高額化する傾向にある。

過去に発生したものと同一、あるいは類似の不具合が市場に流出するたびに、改善活動の関係者は、「過去トラ」を利用していけば防げたはずと悔しい思いをする。そこで我々は、「過去トラ」の有効活用を研究テーマに設定した。

2. 活動の目標

組み込みソフトウェアの過去の不具合事例やクレーム情報を再発防止に役立てるために、現状の課題を洗い出し、実際の開発現場への適用施策を検討し効果を確認する。

3. 活動内容

「過去トラ」を組み込みソフトウェアのトラブル再発防止に有効活用するために、現状調査、課題の洗い出し、施策検討、そして実施結果の確認の流れで活動を行った。

3.1. 「過去トラ」の現状調査

各研究員の所属する会社で、実際にプロジェクト活動の中でどのように「過去トラ」を活用しているかを調査した。調査するにあたり、「過去トラ」の活用ライフサイクルを図1に示すように定義した。最初に、あるプロジェクトで発生したトラブルはプロジェクト員や品質保証・改善推進者の手で、再利用可能な形で「過去トラ」として登録・管理される。次に、別のプロジェクトの計画時点では、プロジェクトリーダーが、設計ノウハウとして「過去トラ」を参照する。プロジェクトの終結時点では、プロジェクトリーダーや改善推進者が、利用した「過去トラ」が有用であったか効果を確認し、利用価値を高めるためのフィードバックを行う。そして、プロジェクトで発生した新規のトラブルが、また新たな「過去トラ」として登録・管理される。

この「過去トラ」活用ライフサイクルの観点で、「登録・管理」、「利用」、「効果確認」の3点について現状を調査した。そして、その結果を問題点として整理することとした。

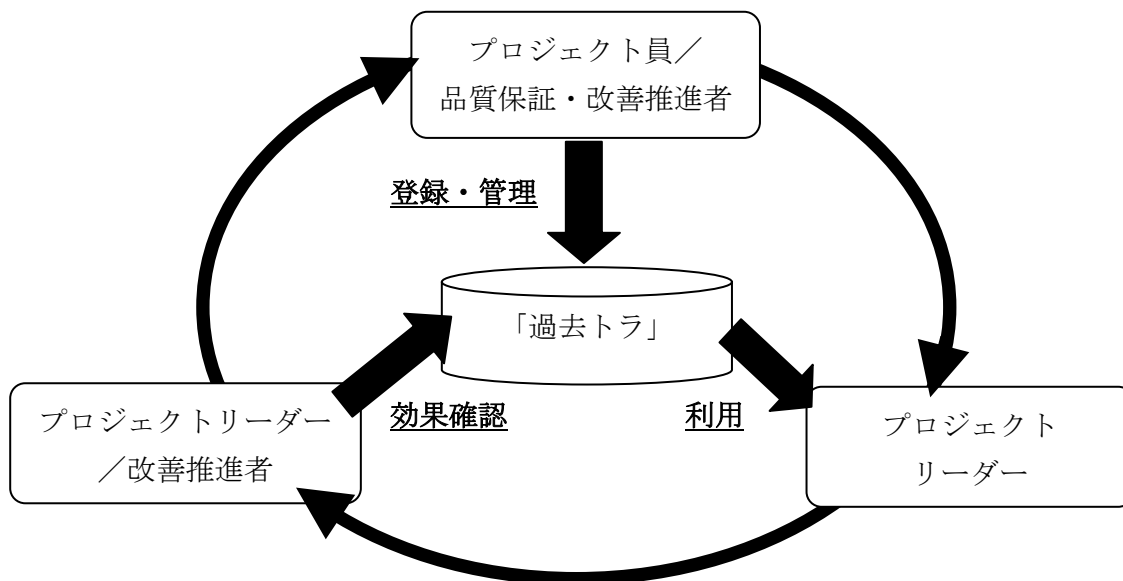


図1 「過去トラ」の活用ライフサイクル

3.1.1. 「過去トラ」の登録・管理の現状

あるプロジェクトで発生したトラブルが、どのように「過去トラ」として登録・管理されるかについて調査した。

【事例1】「過去トラ」はBU(Business Unit)/プロジェクト毎にデータベース化されている。登録・管理されている情報は、市場導入前における不具合情報から市場導入後における不具合情報であり、データベースに登録された情報の管理はそれぞれの担当責任部門が行っている。登録情報の見直しは定期的に行われておらず、登録件数は増え続けている。

【事例2】「過去トラ」は全社共有のデータベースで管理されている。登録・管理されている情報は、市場クレーム情報、工程異常連絡表、再発防止設計事例集、開発段階から市場導入後に発生した不具合事例などで、多種多様な帳票が使われている。各帳票によって記載内容、詳細度は異なる。一度、登録された「過去トラ」は見直しがされないため、参照・利用される機会は少なく、登録件数だけが年々、増加し続けている。

【事例3】「過去トラ」は事業部/部門ごとのデータベースで管理されている。特に重要な「過去トラ」については、全社共通の「過去トラ」リストで管理されている。データベースに登録・管理されている情報は、開発工程から市場導入後に至るまでの、ソフトウェアの実装ミス、仕様書の間違い、改善要求など幅広い。「過去トラ」情報は設計開発部門や品質管理部門であれば自由に閲覧し登録することが可能であるが、実際に事業を跨っての閲覧はほとんどされていない。「過去トラ」の登録内容は登録者に任されている部分が多い。登録件数はデータベースのうち最大のもので10数万件、登録のペースは数10

～100件／稼働日である。

【事例4】「過去トラ」は事業部／部門ごとのデータベースと全社データベースで共有管理されている。事業部／部門ごとのデータベースでは設計工程のレビューで検出した不具合から市場導入後に発生したすべての不具合情報を固有財産(品質記録)として管理している。全社データベースでは市場導入後に発生した重要品質問題を管理している。また、データベースに登録された情報とは別に、顧客リリース後に発生した不具合の原因分析を行った「過去トラ」情報もあるが、記述されている内容はプロジェクトごとに独自で、整理はされていない。

上記から「過去トラ」の管理方法の現状として、

- (1) 「過去トラ」を登録・管理するためのプロセス、仕組みをもっている
- (2) 「過去トラ」の登録件数は増加し続けている
- (3) 「過去トラ」の登録内容はメンテナンスされていない

の3点が存在することがわかった。

3.1.2. 「過去トラ」の利用の現状

プロジェクトの計画時点で、「過去トラ」がどのように利用されているかを調査した。

【事例5】 開発する製品群ごとに集められた「過去トラ」を利用している。異なる製品群の「過去トラ」は利用していない。

【事例6】 各プロジェクトの立ち上げ時には、開発開始の承認を受けるための条件として、「過去トラ」の再発防止方針の報告が義務付けられている。再発防止策の検討、対策実施、結果報告を開発部門と評価部門が行う。どの「過去トラ」を利用するかについてはプロジェクトリーダーの経験、裁量によるところが大きい。

【事例7】 再発防止策の検討と対策実施を行い、その結果報告を関連する開発・品質部門に義務付けている。ただし、事業領域を跨って再発防止されるケースは少ない。短納期などの事情で「過去トラ」の活用が省略されることがある。

【事例8】 「過去トラ」の利用については各プロジェクトリーダーの経験、裁量に委ねられている。外部の会社や海外法人のメンバーを含む流動的で一時的な開発体制の中、「過去トラ」情報の確実な伝達や共有が行われていない。

上記から「過去トラ」の利用の現状として、

- (4) 「過去トラ」の利用方法は開発現場任せになっている
- (5) 利用される「過去トラ」は身近なものに限られる

(6) 「過去トラ」が利用されないこともある
の3点が存在することがわかった。

3.1.3. 「過去トラ」の効果確認の現状

プロジェクトの終結時点で、「過去トラ」を利用した効果がどのように確認されているかを調査した。

【事例9】 プロジェクト終結時には、新たに発生したトラブルの再発防止が重視され、既存の「過去トラ」の利用効果についての振り返りはほとんど行われない。

【事例10】 「過去トラ」の利用で得られる効果を予測・設定した目標に対して、その達成度合いを検証しているが、フィードバックは開発チーム内に限られている。

上記から「過去トラ」の効果確認の現状として、

(7) 「過去トラ」の利用効果を確認していない

(8) 「過去トラ」の利用効果がフィードバックされる範囲が限定されている

の2点が存在することがわかった。

3.2. 「過去トラ」の活用における課題

上記の(1)から(8)の現状をもとに、「過去トラ」活用の課題を整理した(図2)。

まず、(1)と(2)は登録時の課題であるため、今回取り組む課題から除外する。

(3)と(5)については、有用な「過去トラ」が存在することを、利用する側がどのように把握するかという課題があると考えられる。(4)については、活用できる「過去トラ」をどうやって検索するかという課題と、「過去トラ」に含まれる教訓や対策をどのように再発防止策として実プロジェクトに取り込むかという課題があり、後者の課題は(6)についても同様であると考えられる。また、(6)、(7)、(8)については「過去トラ」を利用することで得られる効果をどのように可視化し、さらなる利用を促すかという課題があると考えられる。以上を、課題AからDにまとめた。

【課題 A】 「過去トラ」の存在をどのように把握するか

「過去トラ」を提供する側が技術課題への有用な解決策を登録しても、利用側がそれを見落としてしまう、見落としていることに気づいていない。

【課題 B】 利用者が求めている「過去トラ」をどのように検索・選択するか

「過去トラ」を利用する側が必要としているものを、どうやって検索すればよいのかわからない。あるいは、的外れなものを選択している。

【課題 C】 「過去トラ」の教訓・対策をプロジェクトにどのように取り込むか

「過去トラ」を利用する側が解決策を取り込もうとしても、提供する側の特性に基づいた解決策のために利用する側の特性に合わせた補正ができない。

【課題 D】 「過去トラ」の利用効果をどのように評価し可視化するか
 一度、有効に利用された「過去トラ」は再利用性価値が高まるが、高まった価値を残さないと新たな利用の促進に繋がらない。

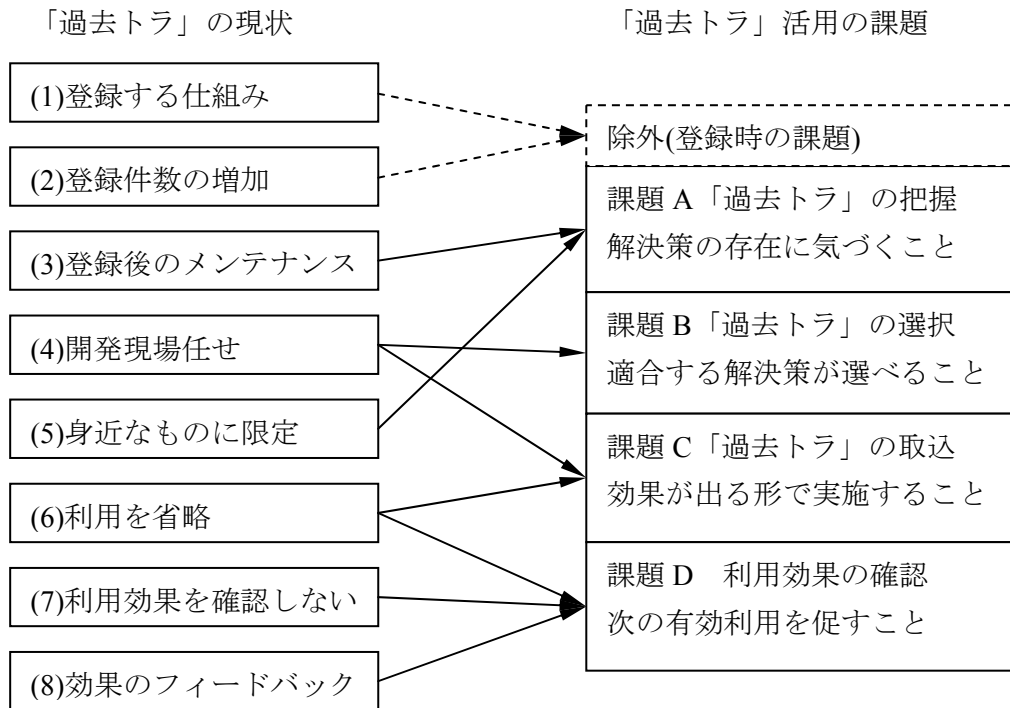


図2 「過去トラ」の現状と課題の関係

3.3. 施策の検討

課題 A から D に共通するのは、「過去トラ」を利用する開発現場が鍵を握るということである。施策を検討するにあたり、開発現場へのヒヤリングを実施した。その結果、

- ・ 既にある資料や開発プロセスを用いて、負担増は抑えてほしい
- ・ 情報提供者から直接説明を受けたい
- ・ より多くの開発チームが成果を共有できるようにしてほしい

という要望が多く寄せられた。つまり、「過去トラ」を提供する側と利用する側(課題 A から C)、利用する側とその次にまた利用する側(課題 D)の間に、情報伝達の課題が存在することがわかった。

そこで、「過去トラ」を提供する側と利用する側とが顔を合わせる会議形式を採用することとした。会議自体は改善推進者が定例などで既に開催している場合が多く、負担を感じさせることはほとんどない。また、会議に参加するプロジェクトリーダーがプロジェクトの代表として、情報提供と情報の持ち帰り展開を行うことで広く共有することが可能となる。

具体的には、以下に示す対策 A から D を施策とした(課題 A から D にそれぞれ対応する)。

【対策 A】 他プロジェクトにも関連すると判断されたトラブルを報告するために、プロジェクトリーダー以上の開発者が参加する会議を設ける。

【対策 B】 報告された「過去トラ」が、どのプロジェクトで利用されるべきかを議論するために、プロジェクトリーダー以上の開発者が参加する会議を設ける。

【対策 C】 「過去トラ」の再発防止策と実施成果を共有し、さらなる改善を検討するために、プロジェクトリーダー以上の開発者が参加する会議を設ける。

【対策 D】 品質の向上に成果があった「過去トラ」の利用成功事例について発表し、プロジェクトリーダークラスの開発者が成果を共有する場を設ける。

3.4. 実施結果

対策 A を実施した結果、トラブル事例を共有するステップを踏むことで「過去トラ」は確実に各プロジェクトに展開されるようになり、「過去トラ」の存在が知られていないという状態が解消した。

対策 B を実施した結果、どのプロジェクトでどの「過去トラ」を利用するかが会議の中で絞り込まれるようになり、開発現場任せの状態が解消した。

対策 C を実施した結果、これまで個別に検討されていたトラブル事例が広い視点で分析された「過去トラ」の教訓と対策になり、各プロジェクトで利用できるようになった。

対策 D を実施した結果、効果のある改善方法が確実に展開され、各プロジェクトで実施できるようになった。

4. 考察

施策を実施した結果、課題を克服し、現状の改善が見られただけでなく、さらに二次的な効果を得ることができた。

- (1) 品質向上の効果があった成功事例を展開することで、情報提供者の「過去トラ」についての理解と意識が深まり、プロジェクトへの取り込みが以前に比べてスムーズになり、その施策もさらに改良された形で各プロジェクトに導入される傾向がでてきた。
- (2) 開発チームの「過去トラ」への関心が高まったことで、データの重要性や情報収集への意識が向上し、あとで分析や効果に用いることを考慮したデータの残し方についても改善するようになってきた。
- (3) 情報展開する場を与えることで「過去トラ」以外についても開発チーム間でのコミュニケーション、情報交換が活発になってきた。

さらに、施策の効果をより確実なものにするためには、成功事例を生み出す原動力となる品質改善の取り組みに対する報奨制度を導入することも効果的である。これにより、開発現場のモチベーション向上が期待できる。

なお、「過去トラ」情報の提供者に対する問い合わせ対応負荷を軽減する措置、会議形式では従来の作業を圧迫してしまうケースへの導入方法の検討、「過去トラ」の活用を進める改善推進者やプロジェクトリーダーの異動などのリスクを回避するための標準化についても検討する必要があると思われる。

5. 結論

組み込みソフトウェアの過去の不具合事例やクレーム情報を再発防止に役立てるという当初の目的に対して、「過去トラ」の情報共有の場を設けることでその目的を達成することができた。これにより、「過去トラ」が確実にプロジェクトへ展開され、開発現場任せの状態が解消かつ有効利用されるようになった。さらに「過去トラ」への理解と意識が深まり改良された形でのプロジェクトへの導入やデータの重要性、コミュニケーション、情報交換が活発になるなどの二次的な効果もあり、課題を克服する以上の効果を得ることができた。数社の事例による施策の検証ではあるが、「過去トラ」が活用できていない組織、企業はほかにもたくさんあると思われる。本分科会で研究した施策で得た効果を1つでも体験してもらい、眠っていた「過去トラ」を有効活用されることを期待する。

6. 今後の展開

大量の「過去トラ」全体を有効活用するためには、蓄積した「過去トラ」を適切に分類し体系化する必要があると思われる。また、効果を客観的に判断するためには、活用状況や品質を定量化し計測する必要がある。これらについては検討を行っていないので今後の課題である。さらに、増え続ける「過去トラ」の改廃や再発したときの既存情報へのフィードバックなど、情報の新鮮さを維持する仕組みなどを取り入れることで、より価値の高い「過去トラ」情報になることが期待できる。

7. 参考文献

[1] 情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター、「2008年版組込みソフトウェア産業実態調査報告書」、

(http://www.meti.go.jp/policy/mono_info_service/joho/2008software_research.html)、

2008年7月

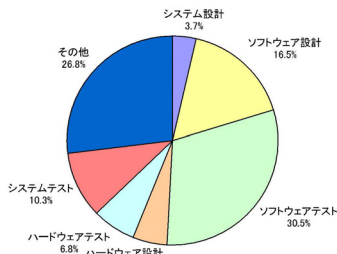
[2] 構造化知識研究所、「知識の構造化によるトラブル未然防止—構造化知識の活用による問題点の早期摘出と対策実施—」、(http://www.ssm.co.jp/download/pdf/ssm_summary.pdf)、

2008年

付録1 情報処理推進機構「2008年版組込みソフトウェア産業実態調査報告書」[1]より抜粋

Q3-1-2d 工程ごとの工数比率:大工程分類

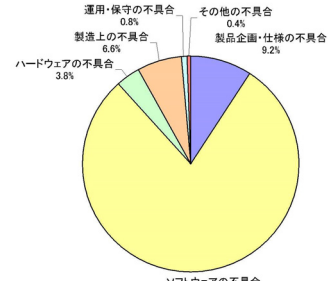
プロジェクト責任者



システム設計:企画・顧客要求獲得、システム要件定義、システムアーキテクチャ設計
 ソフトウェア設計:ソフトウェア要求定義、ソフトウェアアーキテクチャ設計、ソフトウェア詳細設計
 ソフトウェアテスト:ソフトウェア実装・最終テスト、ソフトウェア結合・統合テスト、ソフトウェア総合テスト
 ハードウェア設計:ハードウェア要求定義、ハードウェアアーキテクチャ設計、ハードウェア詳細設計
 ハードウェアテスト:ハードウェア実装・最終テスト、ハードウェア結合・統合テスト
 システムテスト:システム結合・統合テスト、システム総合テスト
 その他:プロシミュレーション、開発プロセス改善シナリオ、技術要素、運用・サポート・保守、その他

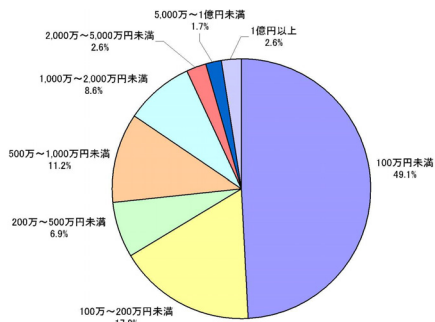
Q6-2 製品出荷後の不具合原因比率

プロジェクト責任者



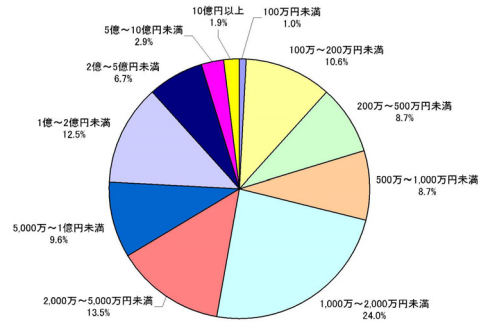
Q8-3-1B 品質問題が発生したことによる1件当たりの対策費

事業責任者



Q8-3-1 品質問題が発生したことによる対策費の総合計(2006年会計年度)

事業責任者



Q4-1 組込みソフトウェア開発の課題

事業責任者

