

ソフトウェア開発の品質向上で成果が出た実践的な「なぜなぜ分析」の方法 について

About the practical method of "why-why analysis" that has yielded results in improving the quality of software development

株式会社 デュプロ 第2開発部 石黒 洋
Hiroshi Ishiguro

Abstract As part of our efforts to improve the quality of the development department that develops embedded software, we have been analyzing the causes of defects.

As a method of analysis, I adopted a "why-why analysis," which enables deeper and more effective analysis, but it was not easy to actually continue that analysis and preventive measures consistently.

In this paper, I will explain the innovations that made this continuation possible. I will also discuss what I think is the best way to improve the software development scene and the current situation, which I learned from that experience.

1. はじめに

1.1 背景

弊社は産業用機械の中規模メーカーで、ソフトウェア開発部門では主にマイコンによる組込みソフトウェア開発を行ってきた。近年、市場要求の高まりにより、ソフトウェアの高性能化、高機能化や複雑な派生開発の案件が多く発生し、ソフトウェア開発の難易度が急激に高まった。それに伴い不具合によるプログラム変更、すなわち手戻りが多く発生する事になった。

この課題に対して、手戻りを低減させるための改善活動を継続的に行いその活動における柱の1つとして

「手戻り事例についてなぜなぜ分析を用いて原因分析と再発防止を行う」事を実施した。

1.2 改善活動の概要と解決すべき課題について

なぜなぜ分析による品質改善活動は2018年に開始した。リリース後に発生した不具合によるプログラム変更1件1件を、その設計担当者と所属するチームリーダーが分析と再発防止案を作成し、ソフトウェア開発部門のリーダーが毎週定期的に行っているリーダーミーティングでその内容を議論する形を取った。そのような中でなぜなぜ分析の導入当初は、以下のような課題が出ていた。

- ・分析に時間がかかり、発生件数に追いつかず、多くの積み残しが発生した。
- ・件数の消化を急ぐばかりに再発防止が現実的で無かったり、場当たりのになるなどの問題が出た。
- ・再発防止がテスト、レビュー、検証に依存する傾向となり、チェック項目が肥大化した。

2. 「なぜなぜ分析」の実施と課題への取り組みについて

2.2 なぜなぜ分析について

なぜなぜ分析は一般的な手法であるが、まずは前提となる概要を示す。本分析手法は事故、故障などの事象に対して原因分析と再発防止に用いられる。事象発生からの時系列的な巻き戻しによる観点などを用いてその原因を掘り下げて、真因を導き再発防止策を

株式会社デュプロ Duplo CORPORATION

〒252-5280 神奈川県相模原市中央区小山4-1-6

4-1-6 Oyama, Chuo-ward, Sagami-hara-city, Kanagawa, Japan Tel:042-775-3600

e-mail: h-ishiguro@duplonet.co.jp

【キーワード】なぜなぜ分析、再発防止、品質向上、継続性、技術者の育成

策定する。分析にはその発生した事象に対して「なぜ」を繰り返すことで原因を深掘りできる。分析を進めると原因と事象が枝分かれして細分化され真因まで行きつく。これにより表面的な事象からの思い付きによる再発防止を防ぐ事ができる。以下の図は、サンプルの事故に対してなぜなぜ分析のあり/なしの2パターンでの例である。

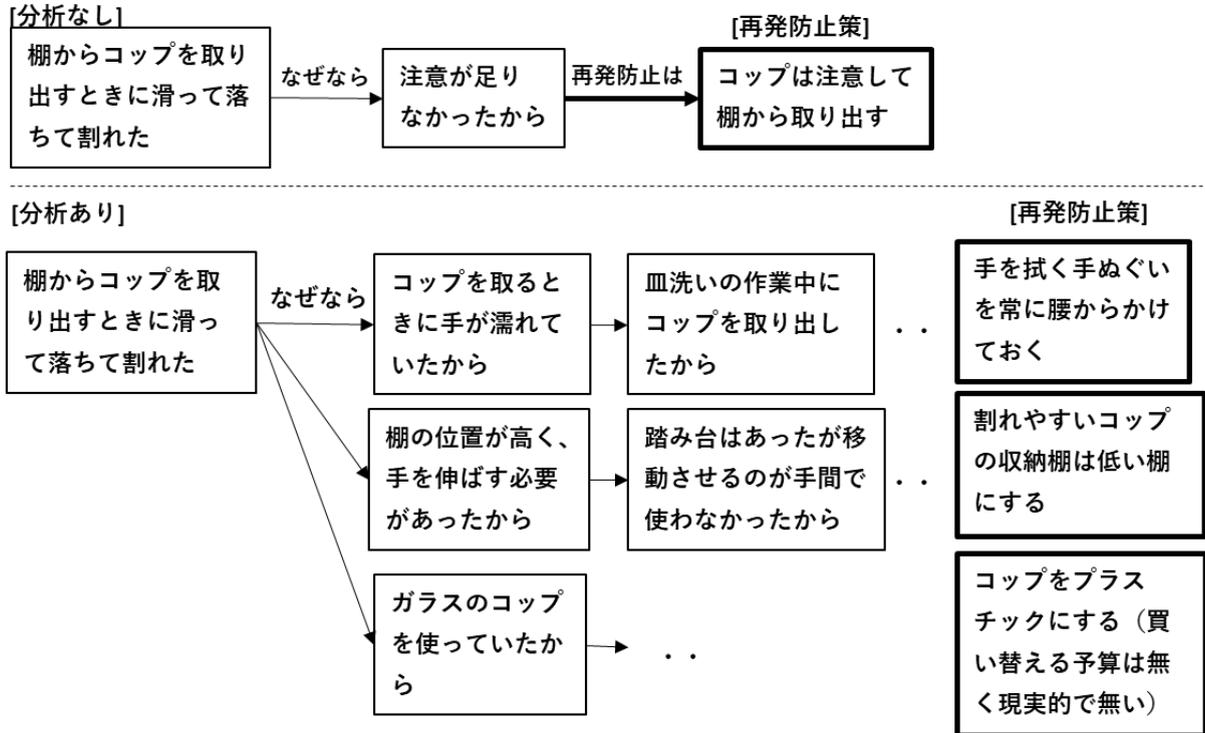


図1. なぜなぜ分析の例と効果について

2.2 継続的に行うためのなぜなぜ分析の工夫について

このような分析と再発防止を進めていく中で1,2で挙げたような課題がでるようになった、これらの課題に対応するため、主に以下の4つの工夫を行った。

(1) 分析を「設計時」に絞る

なぜなぜ分析の一つのパターンとして、設計プロセスにおける『設計時』または『検証時』に枝分かれするが、基本的には設計時 (+デバック) までに着目した。これはV字モデルの左側に位置し、手戻り要因となる不具合自体を作り出さない、すなわち「発生防止」を行う事になる。一般的に、手戻りの発見が後工程になるほど対応に多くの時間がかかるため、できる限り上流(源流)でのプロセス改善に着目した。また検証での再発防止、すなわち「流出防止」に重点を置くとチェック項目が肥大化することが想定され、それを防ぐためでもある。(図2を参照)

(2) 設計担当者の技術スキルの課題を排除しない。

一般的なぜなぜ分析では、問題の原因を個人に委ねることは真の再発防止につながらないとされ、場合によっては個人攻撃に発展する恐れもある。しかしながら、同じ間違いを繰り返さないように設計担当者が自分自身の設計スキル不足、確認不足を認識する再発防止策も、担当者や状況を選別しながら行った。

この取組みは(1)で説明した設計プロセス+デバックの要因分析に比重を置いたこととも関連している。設計担当者は(1)の範囲で開発設計とその計画化の責務を与えられる。ソフトウェア開発のような複雑な作業において、その設計プロセスを詳細に仕組み化、マニュアル化する事は現実的で無く、また逆に柔軟性を失わせる事もあると考えている。もちろん設計レビューにおける第三者チェックも行っており、レビューの課題化も重要であるが、高品質な製品開発は、担当者の創造性や技量にある程度は委ねられるべきであると考えている。それは担当者がソフトウェア開発者としての自律心、責任感、技術の探求心を向上させることにもつながるとも考えてい

る。それらの考えの元、設計者のスキル不足を要因とし、そのスキル獲得を再発防止とする分析も行った。（図2を参照）

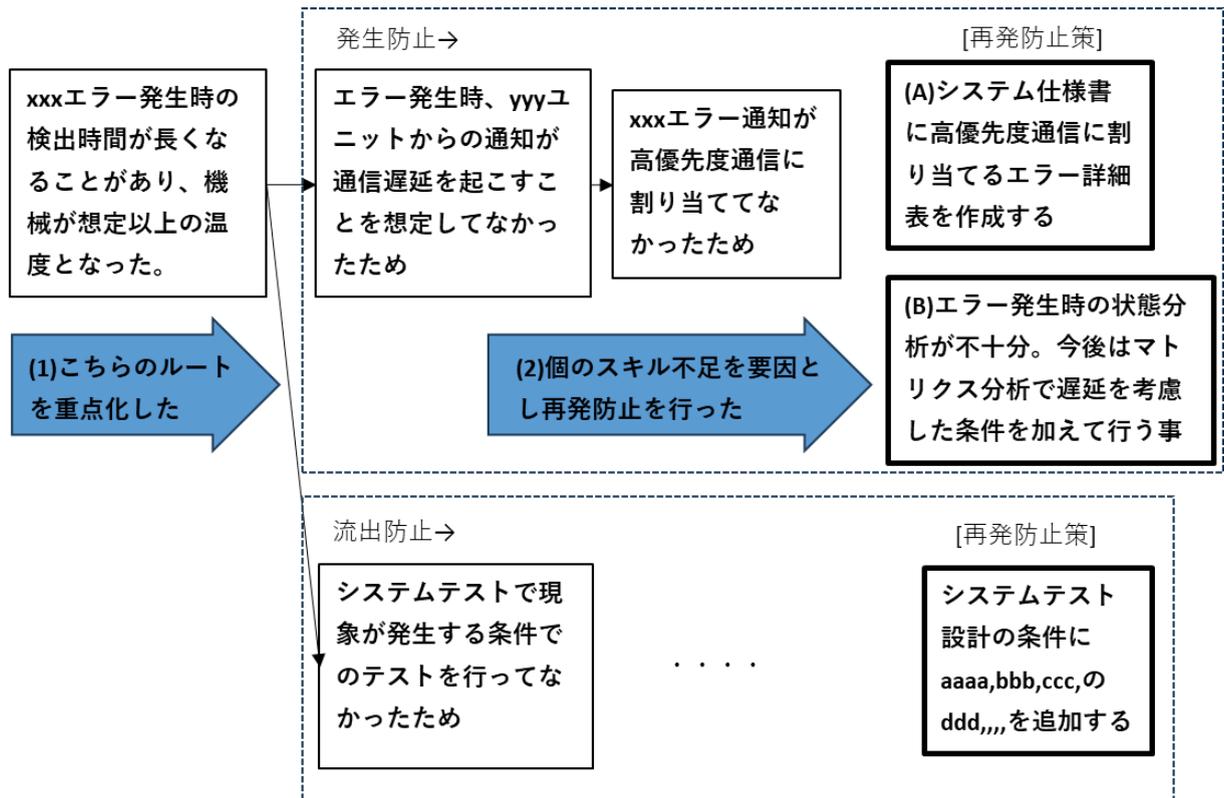


図2. なぜなぜ分析の工夫(1), (2)の例

(3) 時間（工数）の問題にしない。

多くの開発プロジェクトでは開発目標に対して日程や工数がタイトになると考えている。実際に時間不足による設計検討の不足が原因で手戻りが発生することもあるが、分析時にはその設計検討不足に焦点を当てることを行った。

時間（工数）の問題に着目すると、自分たちの設計問題に焦点が向かなくなる場合がある。場合によっては管理側の問題として他責化する。また実際に時間が多くあってもその時に本当に手戻りが発生しないような設計ができたかは分かるものではない。この事より分析する時の考え方としては「時間があれば何ができたかを具体的に抽出」し、次の機会にその行うべき設計事項をしっかりと実施できるように計画することが重要になる。

(4) 無理に再発防止までもっていかない。

分析は真因と思われる内容抽出までとし具体的な再発防止を行わない事もある。これは安易な対策（ルール化など）を無くすためである。分析結果や再発防止案の内容についてリーダー間で合意を取れない場合もある。このように再発防止案の内容について迷ったら、無理に進めず記録として残し、後に関連する手戻りなどがあれば、並べて検討する事にした。あとから類似の内容があると、以前の内容を参考にして、真因をブラッシュアップし、より効果的な再発防止を行う事が可能になる。

明確な再発防止まで持っていけない事例でも、事例紹介としてソフトウェアメンバーに展開し、情報共有を行う事で一定の効果が得られると考えている。

3. 改善取り組みの効果について

3.1 具体的な品質向上の成果について

なぜなぜ分析の方法が固まってから4年の手戻りポイント（プログラム変更件数×その修正内容の影響度、重要度）の推移についてポイントが急激に下がった。またなぜなぜ分析の結果、立案される再発防止策の結果として開発プロセスの整備も進み、各設計プロセスで用いる資料の改訂が進んだ。（図3参照）

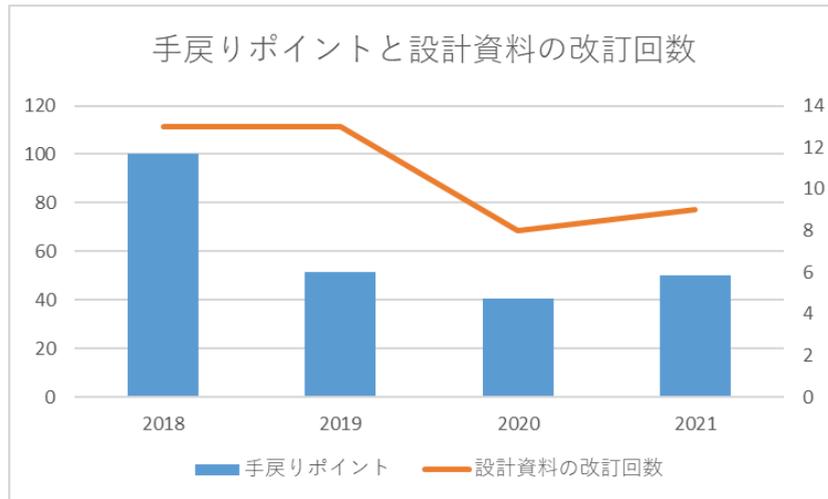


図3. 手戻りポイントと設計資料の改訂回数

上のグラフは4年間の手戻りポイントの推移で、1年目を100としてのそれ以降の手戻りポイントを示している。値としては100→51→40→50となっている。この4年間での開発内容、難易度等に大きな差が無いが手戻りポイント自体は4年間でほぼ半減している。また設計資料の改訂回数も4年間で同様に推移している。

3.2 副次的な成果について

手戻りの減少とともに、以下の効果があったと考えている。

- (1) なぜなぜ分析を行う事で、論理的思考力が付き、それ以降の設計業務に活かされたとの意見も多く、技術者の育成ツールとしても有効であった。
- (2) なぜなぜ分析の内容をチームリーダー間で共有し、また各チームメンバーに落とし込む事で、ソフトウェア部門全体に技術情報を水平展開できるようになった。これにより開発プロセスにおける各プロセスで求める設計内容、粒度や品質について部門で一貫性を持つことにつながった。この一貫性により部門内におけるソフトウェア開発の属人性を減らすことになり、ソフトウェア開発部門全体でのリスク低減にも効果を発揮した。
- (3) プログラム品質の向上（手戻りの低減）に大きな成果があったことはもちろんだが、設計者が原因分析、再発防止の立案を行うことで、失敗から学び、次に活かす姿勢を醸成できたと考えている。
- (4) 実際に手戻りが発生した際、それが単純な個人のミスと認識されることも少なくない（後から振り返ると、何とでも言える場合が多い）。このような短絡的な認識は設計に携わった当事者にとって精神的な負担が大きく意欲を削ぐことも少なからずあった。しかしながら分析によって実際には単純ではなく、複雑な要因が絡んでいることが明らかになったりもする。その要因を浮き彫りにし具体的な技術課題として扱うことで、担当者の心理的負担を軽減することにもつながった。
- (5) なぜなぜ分析が浸透することで、実際には手戻りが発生してない、ひやりはっとレベルの出来事でも洞察を持ってチームで考えることが事ができるようになり、手戻りの未然防止ができるようになってきている。

4 考察およびまとめ

4.1 効果的に継続できた環境要因などについての考察

1.1背景で述べた通り、弊社の装置開発においてソフトウェア品質要求が高まりと、属人的な開発手法の行き詰まりに対し、多くの試行錯誤的な改善活動を行ってきたが、十分な成果を出せない事もあった。それらの課題に対して導入した「なぜなぜ分析」自体はとても汎用的で基本的なツールである。そのツールの活用が陳腐化せず、効果的に継続出来ている事が重要であると考えている。そのような継続性を可能にした環境要因などについて以下の通り考察した。

(1) なぜなぜ分析の仕組み化とリーダーの共通認識について

なぜなぜ分析の実施を標準的な変更管理の手順に組み込み、仕組み化した。不具合修正でのプロ

グラム変更時には原則的になぜなぜ分析を実施検討を行う事でリーダー間で整合をとった。また週に1回の定期的なソフトリーダーミーティングを実施して、その内容を取り上げた。毎週金曜夕方リラクセスした時間帯に実施するミーティングで率直な議論を交わしてきた。それを継続する中で問題とその対応に対する共通認識も徐々にとれるようになり、ある程度パターン化することで効率的でシームレスな分析ができるようになった。

(2) その他の改善活動（ソフトウェア工学の習得）とのリンクについて

手戻りのなぜなぜ分析以外の改善活動として外部コンサルタントの活用によるコンポーネント設計、共通アーキテクチャ設計などのソフトウェア設計技術の導入などがある。また毎年ワーキンググループを立ち上げて、時間のかかる技術課題や新規技術調査、手のかかる開発資料の整備を継続的に行ってきた。これらは一般的なソフトウェア工学に基づいた技術探索・導入であり、それにより技術的な課題解決手法の解決範囲を深め、広げることができた。

これらの複数の平行する改善活動は、なぜなぜ活動とお互いにリンクしている。実際に発生した手戻りを起点になぜなぜ分析を行い、その課題解決として別の改善活動のワーキンググループを立ち上げる事もあった。実際にこのような技術的な解決手法の拡大が無ければ、再発防止で取れる手段は少なくなり、現実的で無く、具体性に欠ける防止策となりえたと考えている。もしそのような状況になっていれば、なぜなぜ分析の活動自体が陳腐化し頓挫していた可能性もあると考えている。

(3) 設計担当者個人のスキル不足の扱いについて

なぜなぜ分析の工夫の(2)で述べた、手戻りの原因を個人のスキル不足としたの件については、「ソフトウェア職人気質」の考え方から影響も受けている。「ソフトウェア職人気質」は「ソフトウェア工学」と対比して扱われる考え方で、「ソフトウェア工学」が一般的な内容に対し、「ソフトウェア職人気質」は属人的なものになる[1]。その考えの元、経験を積み、経験を活かし、開発メンバーとして実戦的な戦力になってもらうには、演習では無く、実際の開発プロジェクトにおけるOJTが最も優れていると考える。なぜなぜ分析を通して自らのミス进行分析し自省する事は、その経験から得られる知見を何倍にも増幅させる効果があると考えている。ただし実際に自らのミスを振り返り、謙虚に力量不足を認めるのは心理的な抵抗があり簡単でない。その時にはリーダーが自らの経験を述べるなどで心理的なハードルを下げるなどの工夫も重要である。また、なぜなぜ分析で見つける様々な要因を発見的に扱う事で前向きに捉える事も重要であると考えている。

4.2 まとめ

今回述べたかった事をまとめると、以下の通りである。

- ・なぜなぜ分析を効果的に継続することで、ソフトウェア品質が飛躍的に向上した。
- ・効果的に継続するため、実情に沿った工夫を重ねた。

ここで再度強調したいのは、「継続性」である

今回、なぜなぜ分析による改善活動が大きな効果を発揮できた要因として、「なぜなぜ分析」という汎用的で効果的なツールを活用したことに加え、その運用がうまく行えたことが挙げられる。リーダーが議論を重ね、ツールの使い方を深掘りしながら、現実即した形で運用方法を柔軟にかつ、粘り強く更新してきた結果である。そのような基盤が今回の成功体験につながったと考えている。

手戻りを起点とした品質向上の取り組みであったが、その結果、現在では手戻り自体が大幅に減少している。それ自体は非常に良いことであるが、難易度の低い手戻りが減ったことで、今は難易度の高い手戻りが残るようになっている。具体的な内容としては、経年劣化や環境変化、部品の個体差やなどで発生する予期せぬ入力に対するロバスト性の向上が挙げられる。また、レガシーコードの保守開発は依然として容易なものではない。さらに新しい付加価値創造のため、新規技術の導入なども課題である。これらの課題は簡単には尽きることがないと考えている。それに対して、なぜなぜ分析を常にブラッシュアップしながら活用し、個人および組織が学習する事が、これらの技術課題に取り組む挑戦権を保ち続ける条件になっていると考えている。

参考文献

- [1] 酒井由夫、「リコールを起こさないソフトウェアの作り方」、P.96、2010年発行