

組み込みソフトウェア製品ファミリにおける 是正保守の予備的分析

野中 誠^{†1} 桜庭 恒一郎^{†2} 舟越 和己^{†2}

同一製品ファミリを構成する組み込みソフトウェア 8 製品の是正保守データを分析し、主要な結果として次を得た。リアクティブな欠陥修正（他製品の欠陥修正に基づいて行われた欠陥修正）は段階的に増加し、最終的にはすべての欠陥修正の 40%以上に至った。原因となる欠陥修正からリアクティブな欠陥修正が完了するまでの期間および改版数には、大きなばらつきがあった。これらの結果と、分析を通じて得た知見は、プロダクトライン開発の導入検討や運用評価を行う上で有益な示唆を与えらる。

A Preliminary Analysis on Corrective Maintenance for an Embedded Software Product Family

MAKOTO NONAKA,^{†1} KOICHIRO SAKURABA^{†2}
and KAZUMI FUNAKOSHI^{†2}

In this paper, the authors describe an analysis result of corrective maintenance data obtained from eight industrial embedded software products composing a single product family. The major findings include: the percentage of reactive defect correction, which is caused by the occurrence of a prior defect correction in other products, reached over 40%, and the duration of these correction as well as the number of versions between the correction and its causal defect correction had a wide range of distribution. These findings will provide meaningful implication for both practitioners and researchers who consider effective introduction and application of product line software engineering.

^{†1} 東洋大学 経営学部

Faculty of Business Administration, Toyo University

^{†2} 株式会社日立情報制御ソリューションズ

Hitachi Information & Control Solutions, Ltd.

1. はじめに

1.1 背景

ソフトウェアプロダクトライン開発 (SPLE: Software Product Line Engineering)¹⁾ に期待される利点の一部として、ソフトウェア品質の向上と、是正保守^{*1}の効率向上が挙げられる。すなわち、ある製品で発生した故障 (failure) とそれに伴う欠陥修正に関する情報を、製品ファミリ全体で共有し、適切な処置を製品横断的に実施することにより、他製品での類似故障の発生を未然に防ぐことが期待される。その際、製品ファミリ全体の適切な構成管理情報³⁾に基づけば、系統的で効率的な欠陥修正が行える。特に、発見された欠陥がコア資産の中にあつた場合は、原理的には修正箇所が当該コア資産に限定されるため、欠陥修正作業を局所化できると考えられる^{*2}。

1.2 問題意識と研究目的

以上に述べた期待効果は、論理的には妥当であると考えられるが、実際の製品ファミリで是正保守の効率向上に結びつく欠陥修正がどの程度あるのか、先行研究では十分に示されていない。SPLE 適用によるソフトウェア品質の向上や、是正保守の工数削減などの効果はこれまでに報告されているが（例えば文献 4)5)）、その多くはビジネス観点による総合的な評価であり、保守における欠陥修正という詳細レベルでの分析事例は十分に示されていない。

一方、欠陥修正に関する事例分析は、保守分野で過去に示されている（例えば文献 6)）。しかし、その多くは単一製品を対象としたものであり、製品ファミリを対象とした研究は筆者の知る限り十分に示されていない。製品ファミリを対象にした欠陥修正データは、SPLE の導入効果、特に品質面での効果を予測するのに役立つ情報であり、SPLE の実践または導入を検討している組織にとって有益であると考えられる。

他方、製品ファミリの是正保守分析を行うことによって得られる実務的な効果として、SPLE 運用マネジメントの適切性を分析・評価する枠組みが提供できると筆者らは考えている。SPLE の導入の良し悪しを総合的に評価するだけではなく、製品ファミリ開発計画の妥当性の事後評価や、保守における意思決定の結果を可視化することで、SPLE 運用の適切性を評価し、SPLE 導入効果の最大化を図れるものと考えている。本研究の取り組みは、そのような SPLE 運用マネジメントの高度化も視野に入れている。

^{*1} 欠陥修正のために出荷後に行われるソフトウェア製品のリアクティブな変更を、是正保守と呼ぶ²⁾。

^{*2} ただし、回帰テストは関連製品ごとに実施する必要があるため、テスト工数の削減は限定的である。

1.3 研究アプローチと課題設定

以上の問題認識に基づき、本稿では、実際の組込みソフトウェア製品ファミリを分析対象にして、是正保守における欠陥修正データを調査・分析した結果を報告する。特に、製品ファミリにおいて特有なリアクティブな欠陥修正（ある製品の欠陥修正に基づいて他製品で行われた欠陥修正、詳しくは 2.2 節を参照）に焦点を当てた分析を中心に行う。

本調査において設定したリサーチクエスション (RQ) は、次の通りである。

RQ1: リアクティブな欠陥修正およびその原因となった欠陥修正は、すべての欠陥修正のうちどの程度を占めたのか、またその時系列推移はどのようなものであったか。

RQ2: リアクティブな欠陥修正は、その原因となった欠陥修正が完了してから、どのくらいの期間をかけて対応が完了したのか。

RQ3: リアクティブな欠陥修正による改版から、その原因となった欠陥修正による改版に遡るまでの間に、どのくらいの改版数が存在していたのか。

本稿の構成は次の通りである。2 章では、用語の説明を行う。3 章では、分析対象の組込みソフトウェア製品と開発組織の概要について説明し、本調査の文脈を明らかにする。4 章では、RQ に対応する回答を提示し、また、それ以外に得られた知見についても述べる。5 章では本調査に関する考察を、6 章では分析結果の妥当性について議論する。7 章で関連研究について述べ、最後に 8 章で本稿のまとめを述べる。

2. 用語の説明

ここでは、本稿で用いる用語の意味について説明する。特に、欠陥修正の分類については、SPLE 導入と関連づけて説明する。

2.1 製品ファミリ

本稿において、製品とは、顧客にリリースされた組込みソフトウェアを意味する。製品には、機能追加や性能向上、欠陥修正などの対応によって作成された、異なる版が存在する。ある製品の特定の版からのブランチにより、異なる製品（あるいは、同一製品内の別ブランチの版）が派生される。製品と製品の境界は、開発目的や顧客の違いなどにより、開発組織によって主観的に区別される。このようにして構成された複数製品および複数版の全体を、本稿では製品ファミリと呼ぶ。図 1 にその概念図を示す。

2.2 欠陥修正

本稿では、欠陥修正を 3 つに分類しており、便宜上、A、B、C 修正と呼ぶ。

A 修正は、当該版における欠陥が他の製品（または同一製品内の別ブランチの版）に影

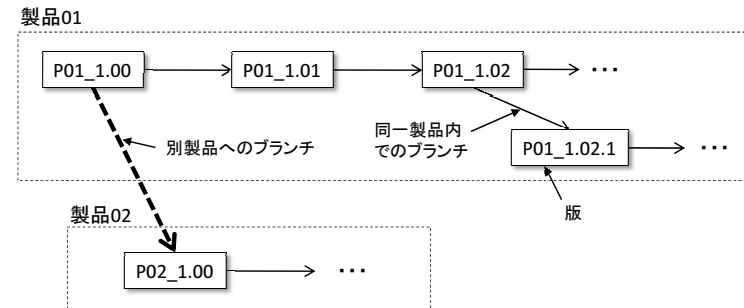


図 1 製品ファミリの概念図

響しておらず、かつ、他の版の欠陥に起因したものではない欠陥修正である。すなわち、局所的で独立した欠陥修正である。ただし、構成管理や根本原因分析が不十分であった場合など、A 修正を実施してから時間が経過した後に、他の製品でも同一原因による故障が発生または発生の恐れが発覚する場合がある。その場合には、当初は A 修正と認識していた欠陥修正が、後になって C 修正（後述）に分類し直されることがある。本稿では、調査期間の範囲で他製品への影響が見出されていない場合は、A 修正に分類して測定する。

B 修正は、ある製品の版で欠陥修正が行われたとき、これと同一の欠陥が原因の故障を回避するために、他の製品の版（または同一製品内の別ブランチの版）でも行われた欠陥修正である。本稿では、特に B 修正に着目しており、より明示的に示すためにリアクティブな欠陥修正と呼ぶ場合がある。

C 修正は、B 修正のトリガーとなった欠陥修正である。

2.3 欠陥修正分類と SPLE 導入の関係

SPLE を導入した場合、複数製品に利用されているコア資産に関しては、A 修正が発生することはない。C 修正がコア資産に対して行われた場合、SPLE の運用が適切であれば（すなわち、製品ファミリ全体の適切な構成管理がなされていれば）、B 修正に相当する作業は各製品のテストのみになる。このように、SPLE の導入により、B 修正の発生を抑制できたり、その対応工数を削減できる。

3. 分析対象の概要

ここでは、本調査の文脈を明らかにするため、製品と開発組織の概要を説明する。また、実際に調査した製品は製品ファミリ全体の一部分であるため、調査範囲についても説明する。

3.1 製品

本調査では、カスタマイズ型の組込みソフトウェア製品を分析対象としている。ここで「カスタマイズ型」と呼んでいるのは、消費財のように不特定の顧客に販売される製品ではなく、特定顧客向け総合システムの一部として、運用環境に合わせて機能や性能を適応させて導入される製品であることを意味している。

製品ファミリに含まれる製品数は、全部で 25 製品である。初版製品のリリース時期は 2003 年 9 月であり、現在も多くの製品がアクティブな保守対象となっている。製品と製品の境界は、システムの入力方式、対応デバイス、外部連携システム、特定顧客へのカスタマイズ要求などの違いにより、開発組織の主観的判断に基づいて区別されている。

本稿の執筆時点において、調査対象の製品開発には SPLE を導入しておらず、先行版の再利用に基づく派生開発を行っている。

3.2 開発組織

この製品は、株式会社日立情報制御ソリューションズ（従業員約 2,500 名）によって開発・保守されている組込みソフトウェアである。株式会社日立情報制御ソリューションズにおける品質保証の取り組みには長年にわたる経験の蓄積があり、2006 年 2 月には別製品を開発する部門にて CMMI（能力成熟度モデル統合）レベル 5 を達成している⁷⁾。このようなことから、品質保証体制は業界内で比較的高い水準にあると考えている。品質マネジメントシステムは製品横断で展開されており、分析対象製品の品質保証も相応の水準は確保されている。

分析対象製品の開発チームの規模は、時期によって変動するが、定常時は 6 名、ピーク時は 28 名であった。なお、この人数にはソフトウェア開発専門の要員だけでなく、ハードウェア開発を兼務する要員が含まれている。さらに、品質保証部門の要員のうち、当該製品の直接的な検証・評価作業を行う要員も含まれている。

3.3 調査対象の範囲

製品ファミリ全体では 25 製品だが、本稿の執筆時点におけるデータの可用性の制約から、P01 から P08 までの 8 製品を分析対象とした。この 8 製品に対して、2003 年 9 月から 2006 年 10 月の間に行われた改版が、本稿における調査対象の範囲である。8 製品の各最新版のソースコード行数（母体ソフトウェア規模、外部調達によるソフトウェア規模を含む）は合計で約 640KLOC であり、1 製品あたり平均 80KLOC 程度である。8 製品は製品ファミリをなしているため、このソースコード行数には製品間で重複する部分も当然ながら含まれている。したがって、例えば出荷後の欠陥密度を求めるにあたって、ここで得られた合計ソ-

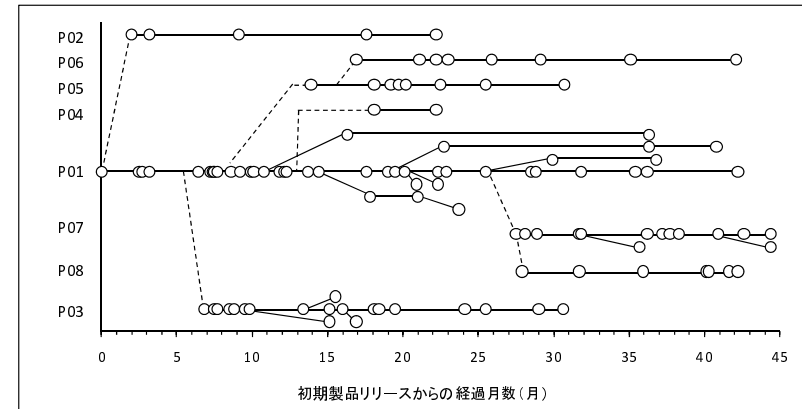


図 2 各製品の改版履歴（ツリー表現）

スコード行数で欠陥修正件数を割っても、妥当なメトリクスではない点に留意されたい。

図 2 に、時系列のツリー表現で示した各製品の改版履歴を示す。図 3 は、改版履歴の累積値を時系列で示したグラフである。なお、本稿で示す改版の時点は、当該版に対して品質保証部が承認した日を意味しており、顧客にリリースした日ではない。また、それぞれの改版には、欠陥修正だけでなく、機能追加（性能向上を含む）の対応も同時に含んでいる場合がある。また、それらの件数は 1 件とは限らず、0 件から複数件までばらつきがある。例えば、機能追加のみで欠陥修正のない改版もある。1 件の欠陥修正の都度に改版しているわけではないことに留意されたい。

図 2 から、P01 が他の製品の派生元であり、中核的な製品であることが分かる。また、図 3 から、P01 は継続的に改版され、8 製品の全改版数のうち 42% を P01 の改版が占めていることが分かる。P03 と P07 は、P01 から派生した以降も多くの改版がなされており、独自の進化を遂げている。一方、P02 と P04 は、改版数が少なく他の製品に比べて安定している。

4. 分析結果

ここでは、調査対象期間に観測された欠陥修正を分析し、1 章で挙げた RQ への回答を示す。特に、リアクティブな欠陥修正（B 修正）に焦点を当てて分析する。

4.1 欠陥修正の比率と推移（RQ1）

図 4 に、初期製品リリース以降に観測された欠陥修正件数の累積値比率の時系列推移を

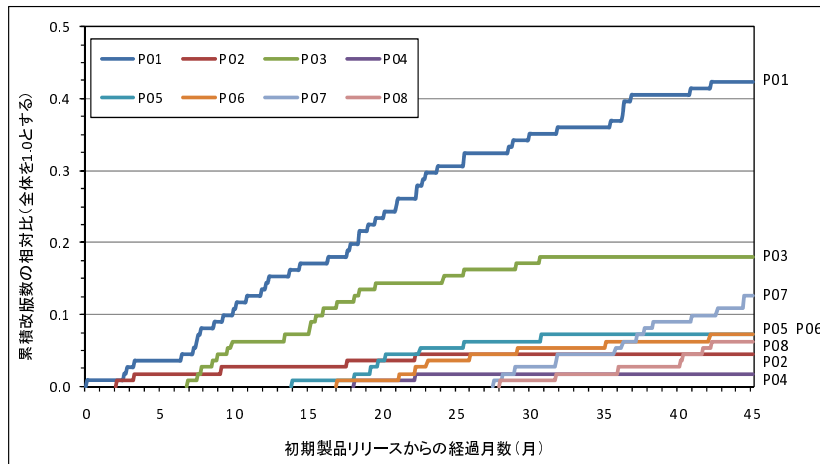


図 3 各製品の改版履歴（累積表現）

示す．図中では，A，B，C 修正を分けて示している．また，横軸上に，各製品の初版がリリースされた時期を印で示している．なお，欠陥修正の時点は，その対応を行った改版の時点と同一である．

B 修正の件数は，最終的には A 修正の件数を上回っている．これは製品ファミリにおける欠陥修正の特徴的な傾向であると考えられる．また，B 修正は直線的に増加するのではなく，C 修正の発生に連動しながら段階的に増加している．B 修正の増加が C 修正に比べて大きいのは，1 件の C 修正に対して複数件の B 修正が実施されているためである．B 修正の推移は，印で示した製品リリースとの直接的な連動性は読み取れないが，製品リリース数の増加に伴って B 修正リスクが高まっていることは間接的に読み取れる．

C 修正の最終的な件数は，全体の約 20%であった．B 修正は C 修正に起因した欠陥修正であることから，B 修正を除外して考えると，A 修正と C 修正の合計に対する C 修正の比率は約 1/3 である．A 修正と C 修正の抑制は容易ではない．しかし，SPLE の適切な運用により，B 修正の抑制，または B 修正の対応期間を 0 にすることが可能と考えられる．C 修正というリスク要因が，B 修正を除いた欠陥修正の 1/3 を占めたという情報は，1 事例ながらも，SPLE の導入効果を試算するうえで価値ある情報といえる．

なお，図 4 の丸印の部分で，B 修正の発生が C 修正よりも遅れている．このタイムラグ

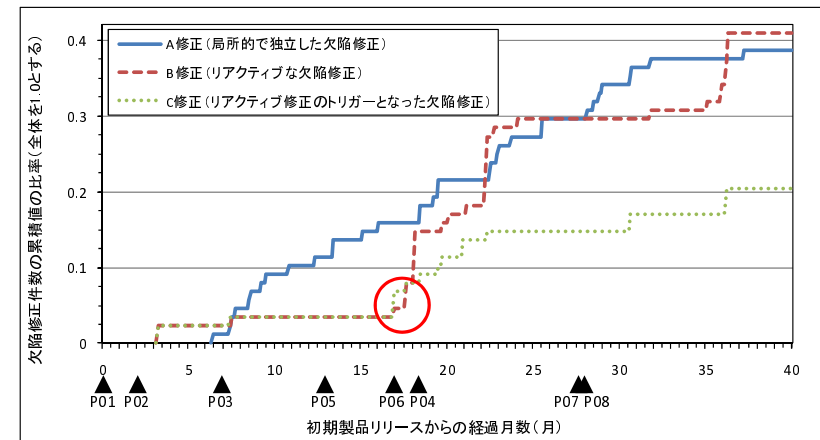


図 4 欠陥修正の推移

が生じている理由は，C 修正が実施された際に，それに関連するすべての B 修正が直ちに実施されるわけではないためである．タイムラグの詳細については次の 4.2 節で述べる．

4.2 リアクティブな欠陥修正の対応期間 (RQ2)

ここでは，B 修正に着目し，リアクティブな欠陥修正の対応期間，すなわち，C 修正が完了してから関連する B 修正が完了するまでの期間を示す．この期間は，C 修正に対する品質保証部の承認日から，関連する B 修正に対する品質保証部の承認日までの差分により求めている．この期間のすべての工数を B 修正に費やしたのではないことに留意されたい．

図 5 に，リアクティブな欠陥修正の対応期間 (週数) の分布を示す．図 5 より，期間が大きくばらついていることが分かる．大別すると，2 週間以内 / 3~6 週間以内 / 7 週間以上の 3 つのグループに分けられる．2 週間以内のグループは B 修正全体の 40%であり，その中でも，対応期間が 0 日という B 修正が約 14%もあった．3~6 週間以内のグループは全体の 40%，7 週間以上のグループは 20%であった．

4.3 リアクティブな欠陥修正までの改版数 (RQ3)

B 修正の対応がなされた改版から，その原因となった C 修正がなされた改版までの間に存在した改版数を，図 2 の改版ツリーを遡って測定した．その結果の分布を図 6 に示す．平均値は 18.4 件であった．改版数が 5 件以下の B 修正は約 30%であったが，残りは改版数が 10 件以上であり，特に，40 件を上回る B 修正が 10%近く存在した．本調査におけるリア

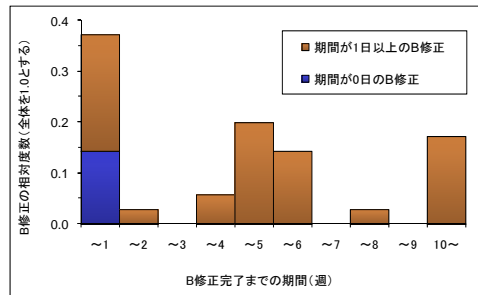


図5 C修正完了から関連するB修正が完了するまでの期間分布

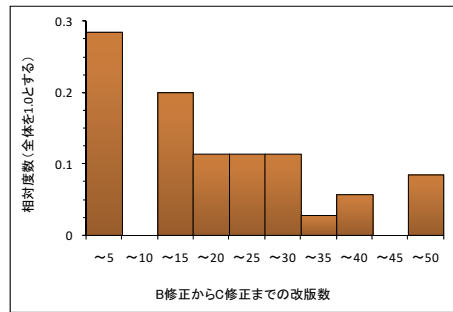


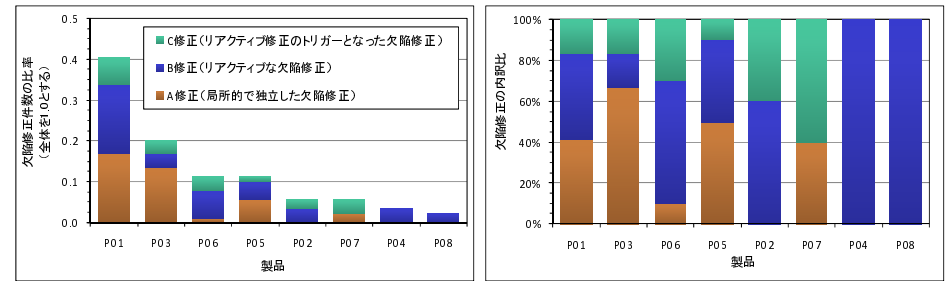
図6 B修正からその原因のC修正までの改版数の分布

クティブな欠陥修正までの改版数は、決して少ない件数であった。

4.4 製品別の欠陥修正の内訳

これ以降は、RQに掲げていないが、製品別の欠陥修正の傾向を分析した結果を示す。図7(a)に製品別の欠陥修正件数の比率を、図7(b)に製品別の欠陥修正の内訳比を示す。横軸には、製品を欠陥修正件数の降順で並べている。

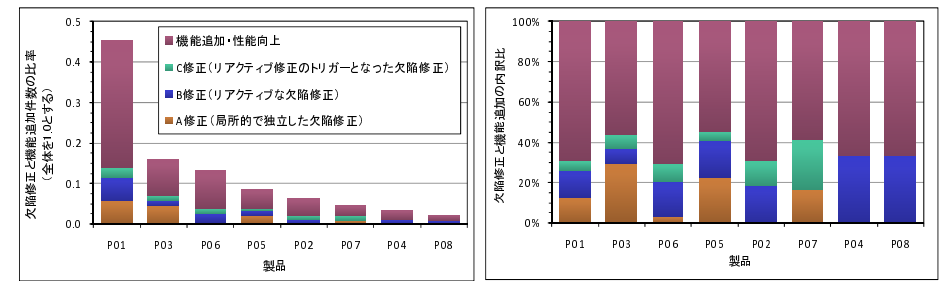
図7(b)から、P07にB修正がなく、P03もB修正の比率が低いことが読み取れる。3.3節で述べた通り、これらの製品はP01からの派生後に独自の進化を遂げた製品である。その独自性のために、B修正の比率が低くなっている可能性が考えられる。逆に、B修正の比率が高い製品は、P02、P04、P06、P08である。これらの製品は、図2のツリー表現で示した通り、製品内でのブランチが存在しなかった製品であり、これらの4製品は派生元製品に対する依存度が高く、独自の進化を遂げていない製品である。



(a) 欠陥修正件数の比率

(b) 欠陥修正の内訳比

図7 製品別の欠陥修正の内訳



(a) 欠陥修正と機能追加件数の比率

(b) 欠陥修正と機能追加の内訳比

図8 製品別の欠陥修正および機能追加の内訳

4.5 機能追加と欠陥修正の内訳

製品が改版される理由には、欠陥修正だけでなく、機能追加も含まれている。調査対象の期間における機能追加の件数は、欠陥修正件数(A, B, C修正の合計)の約1.9倍であった。図7のデータに、機能追加の件数を積み重ねたグラフを図8に示す。図8(b)を見ると、機能追加の比率は55~70%の間に分布しており、製品によって大きな傾向の違いは見られなかった。ただ、4.3節で述べた通り、P03およびP07におけるB修正の比率が低く、これらの製品が独自の進化を遂げていったことを示している。

5. 考 察

ここでは、本調査の分析方法および結果をSPLE運用マネジメントの評価に用いる場合、

どのような検討が行えるか、また検討のために必要な事項が何であるのかについて考察する。

5.1 欠陥修正の比率と推移

まず、欠陥修正の比率と推移(4.1節)について考察する。2.2節で述べた通り、複数製品に利用されているコア資産に関しては、A修正が発生することはない。したがって、SPLE導入後において(A+C)修正に対するC修正の比率が低い場合は、コア資産が複数製品に再利用されていない、製品に占めるコア資産コードの比率が低い、あるいは、コア資産の信頼性が高く、欠陥修正がコア資産以外の部分で多くなされている状況を表す。逆に、C修正の比率が高く、その推移が継続的に増加している場合は、コア資産の信頼性に問題がある状況を表している。この場合、コア資産に対する重点的な品質改善が求められる。

図4を見ると、B修正の件数が大きく増えた時点が3回ある。SPLE導入後にこのような観測が得られた場合、製品ファミリー内での共有コードが多く、大規模再利用を指向するSPLEの製品アーキテクチャとしては適切な状態であることを表すと考えられる。さらに、それらの欠陥修正の対応期間が0日であれば、効率的なSPLE運用が行えている状況を表すと考えられる。一方、B修正が多く発生しているのに、その対応期間が長い場合には、製品アーキテクチャの見直しを含めたSPLE運用の改善が求められる。

5.2 リアクティブな欠陥修正の対応期間

次に、リアクティブな欠陥修正の対応期間(4.2節)について考察する。対応期間の短いB修正は、当該製品での緊急対応が必要であったか、あるいは対応が容易であった欠陥修正であると考えられる。特に、本調査では対応期間が0日のB修正が約14%あったが、これは製品共通の再利用コードに対する欠陥修正であり、他のコードへの波及がほぼなかった可能性が高い。これは、緊急対応を要する場合は性質が大きく異なるため、欠陥修正の内容や対応期間の理由を含めた運用評価を行う必要がある。

一方、対応期間が長いB修正は、C修正の発生に伴って別製品でも対応しなければならなかったにも関わらず、長期間にわたって対応が放置された欠陥修正である。あるいは、故障リスクが低いと判断されたために、他の機能追加を行うときに併せて対応した欠陥修正である。前者の場合は、是正保守マネジメントに問題があった可能性がある。後者の場合は、リスク管理の観点からは妥当な対応といえる。先の段落で述べたのと同様、欠陥修正の内容や対応期間の理由を把握した上で運用評価を行わなければならない。

今回の調査では、欠陥修正の重大度、作業の容易さ、対応時期が遅れた理由などのデータは使用していない。しかし、製品開発部門責任者へのヒアリングなどの結果から、想定できる状況はおおむね上述の範囲であると考えられる。SPLE導入時に適切な評価が行えるよ

う、これらの情報を把握しておく必要がある。

5.3 リアクティブな欠陥修正までの改版数

最後に、リアクティブな欠陥修正までの改版数(4.3節)について考察する。今回の調査では、リアクティブな欠陥修正までの改版数は少なくなかった。この改版数が多いと、必要なB修正対応を見逃すリスクが高まり、B修正完了までの期間がさらに長期化するリスクが高くなる。したがって、欠陥修正に伴う影響波及解析を、製品の版管理および内部モジュールの構成管理が適切に実施された状況下で行わなければならない。SPLEの効果的な運用には、適正な構成管理が重要³⁾であることがうかがえる。

6. 議 論

ここでは、本調査によって得られた結果の妥当性について述べる。

6.1 内的妥当性

本稿は、本質的には調査研究である。このような調査研究における内的妥当性(因果関係の正当性)は、調査した項目が、把握したい事柄を適切に表しているかどうかに関係すると筆者らは考える。本稿における主要な調査変数は、欠陥修正の種類と対応期間である。一方、把握したい事柄は、製品ファミリーにおける是正保守の状況であり、その適切性の評価である。以降、これらの測定方法に関する内的妥当性について議論する。

まず、欠陥修正の種類について議論する。本稿では、欠陥修正をA、B、C修正の3つに分け、その比率と時系列推移を示した。B修正の段階的増加や、C修正の(A+C)に対する比率を見ることで、是正保守の状況やB修正発生リスクの度合いを可視化できた。欠陥分類の判断も客観性が高く、この点においては妥当であると考えられる。ただし、2.2節でも述べた通り、時間の経過とともにA修正がC修正に変わる場合があり、欠陥分類が時間依存である点は否めない。とはいえ、産業界では出荷後の一定期間(通常は半年から一年程度)で欠陥追跡を終えることが多く、このように時間を区切れば分類は安定する。総じて、欠陥分類に関する内的妥当性は適正レベルにあると考えられる。

関連して、欠陥修正と機能追加の区別については、議論の余地がある。この問題は先行研究でも議論されている。Ayewahらは、欠陥追跡システムに記録されたコード変更のすべてが必ずしも欠陥修正ではなく、機能追加も含まれていたことを示している⁸⁾。逆に、機能追加に分類されている対応にも、実際には欠陥修正に分類すべきものが含まれているかもしれない。欠陥修正と機能追加の区別に関して、本稿の範囲では妥当性を評価できていない。

次に、欠陥修正の対応期間について議論する。本稿では、B修正の直接工数ではなく、C

修正完了から関連する B 修正が完了するまでの時間軸上の期間を示した。緊急性の高い欠陥修正であれば期間と直接工数の相関が高いと考えられるが、緊急性の低い欠陥の場合はおそらく無相関になるであろう。したがって、欠陥修正の対応期間という変数は、保守工数の予測というニーズからすれば妥当性は低い。しかし、是正保守の状況を把握するという点では、5.2 節でも述べたとおり、是正保守マネジメントの状態を容易に可視化できる。この点については、一定の内的妥当性レベルに達していると考えられる。

6.2 外的妥当性

外的妥当性とは、評価対象の一般化可能性を表す概念である。筆者らが知る限り、製品ファミリにおける是正保守の分析事例は十分に報告されていない。したがって、本調査で示した内容が、他の製品ファミリにおいても同様の傾向を示すかどうかは結論づけられない。傾向は、製品分野や開発組織など、文脈によっても変わるであろう。この点について、調査を積み重ねた比較が必要である。

調査の文脈である製品分野と開発組織について、組織成熟度は比較的高い部類に属すると思われるものの、業界における逸脱事例というほどではない。これを客観的に示すには定量的なベンチマーキングが必要であるが、筆者の主観的判断では、産業界における一般的な状況であると考えられる。したがって、調査の文脈においては、一定の外的妥当性レベルは確保できていると考えている。

7. 関連研究

本稿で示した欠陥修正の分類について、著者が知る限り、必ずしも十分に研究されていない。Kajko-Mattsson は、IEEE 標準や既存研究を元に、ソフトウェア是正保守の概念モデルを提示している⁹⁾¹⁰⁾。しかし、この概念モデルには、本稿で示した B 修正や C 修正の概念はモデル化されていない。von Mayrhauser らは、欠陥レポートに基づいて、欠陥とコンポーネントの関係を表す欠陥アーキテクチャを作成している¹¹⁾、同様に、本稿で示した欠陥分類は明確にしていない。欠陥分類に着目した研究はこれまでに数多く示されており、典型的には、直交欠陥分類 (ODC) が用いられている (例えば文献 12))。しかし、ODC による研究の多くは欠陥の静的な属性に基づく分類であり、本稿のように欠陥同士の動的な関係性に関する分析は十分ではない。

ソフトウェア是正保守の事例分析はこれまでに報告されているが、製品ファミリを対象としたものは多くない。Defamie らは、ソフトウェア保守時における欠陥対応の期間など様々な分析データを示している⁶⁾。しかし、これは単一製品の分析であり、また、B 修正に関する

分析データを示してはいない。

Clements らは、宇宙船の地上コマンド・アンド・コントロールシステムに SPLE を導入した結果、欠陥数が導入前に比べて 1/10 に減少したことを報告している⁴⁾。また、発見された欠陥はいずれも局所的なもので、他への波及効果のないものであったと述べている。ここでの「波及効果のない」とは、ある欠陥修正が他のプログラム箇所での変更にも波及するものを意味しており、本研究の B 修正や C 修正とは意味が異なる。本研究が対象としたのは、プログラム上の修正は局所的だが、同一欠陥に対応しなければならないプログラム箇所が複数製品に存在するという状況である。本研究は、Clements らが述べる「局所的な欠陥」について、その分類と対応期間が SPLE 導入前後でどのように変化するかを可視化する手法である。

また、近年は、データ利用の障壁が比較的低いことから、オープンソースソフトウェア (OSS) の変更管理データや欠陥記録データを対象とした研究が急速に増えている (例えば文献 13)14))。しかし、本稿で扱った B 修正を行う / 行わないの判断は、開発組織の内外におけるさまざまなビジネス的要因に影響を受けるものであり、OSS 開発における欠陥修正やコード変更とはモチベーションが大きく異なると考えられる。実務的なソフトウェア開発に対して知見をフィードバックすることを考えた場合、本稿のように、産業界の実データの分析結果を示す努力が求められる。この点において、本稿は OSS を対象とした研究とは一線を画していると考えている。

8. おわりに

本稿では、同一製品ファミリを構成する組込みソフトウェア 8 製品の是正保守データを分析し、主要な結果として次を得た。リアクティブな欠陥修正 (B 修正) は初版リリース後に段階的に増加し、最終的には、すべての欠陥修正の 40% 以上を占め、局所的な欠陥修正 (A 修正) を上回る件数に至った。原因となる欠陥修正 (C 修正) からリアクティブな欠陥修正 (B 修正) が完了するまでの期間には大きなばらつきがあり、2 週間以内 / 3~6 週間以内 / 7 週間以上の 3 つのグループに分けられた。特に、対応期間が 0 日という B 修正が約 14% 観測された。また、B 修正から C 修正までの改版数にも大きなばらつきがあり、平均値は 18.4 件であった。これらの結果と本調査で用いた分析手法について、ソフトウェアプロダクトライン開発 (SPLE) の運用評価への適用可能性と妥当性評価を行い、それぞれ妥当な水準と考えられることを述べた。本稿が示した内容は、SPLE の導入検討や効果的運用を行う上で有益な示唆になると考える。

今後の課題として、製品ファミリ全体に対して、欠陥修正の属性（重大度や難易度など）を考慮した上で分析を行い、組込みソフトウェア製品ファミリ開発における是正保守の全体的傾向を明らかにし、SPLEの導入効果について更なる評価を継続したいと考えている。

謝辞 本研究を行うにあたり、株式会社日立情報制御ソリューションズの大島啓二常務取締役、加藤裕昭氏、鈴木康夫氏、渡辺滋氏より貴重な意見を多数頂戴した。ここに謝意を表する。

参 考 文 献

- 1) Paul, K., Böckle, G. and Linden, F.: *Software Product Line Engineering*, Springer-Verlag (2005). 林好一, 吉村健太郎, 今関剛 (訳): ソフトウェアプロダクトラインエンジニアリング, エスアイビー・アクセス (2009).
- 2) IEEE Std 1219-1998: IEEE Standard for Software Maintenance (1998).
- 3) 野中誠: ソフトウェアプロダクトライン開発のマネジメント: 課題と技法, 情報処理, Vol.50, No.4, pp.289-294 (2009).
- 4) Clements, P., Cohen, S., Donohoe, P. and Northrop, L.: *Control Channel Toolkit: A Software Product Line Case Study*, CMU/SEI-2001-TR-030, Software Engineering Institute, Carnegie Mellon University (2001).
- 5) Linden, F., Schmid, K. and Rommes, E.: *Software Product Lines in Action*, Springer (2007).
- 6) Defamie, M., Jacobs, P. and Thollembeck, J.: Software reliability: assumptions, realities and data, *Proc. Int'l. Conf. Software Maintenance (ICSM'99)*, IEEE Computer Society, pp.337-345 (1999).
- 7) 新海良一, 臼井孝雄, 桜庭恒一郎, 飯島三朗: 組込みソフトウェア開発プロセス改善の取り組みと支援サービス, 日立評論, Vol.91, No.5, pp.38-43 (2009).
- 8) Ayewah, N. and Pugh, W.: Learning from defect removals, *Proc. 6th Int'l Working Conf. Mining Software Repositories (MSR '09)*, IEEE Computer Society, pp.179-182 (2009).
- 9) Kajko-Mattsson, M.: A conceptual model of software maintenance, *Proc. Int'l. Conf. Software Eng. (ICSE'98)*, pp.422-425 (1998).
- 10) Kajko-Mattsson, M.: Common concept apparatus within corrective software maintenance, *Proc. Int'l. Conf. Software Maintenance (ICSM '99)*, IEEE Computer Society, pp.287-296 (1999).
- 11) von Mayrhauser, A., Wang, J., Ohlsson, M.C. and Wohlin, C.: Deriving a fault architecture from defect history, *Proc. 10th Int'l Symp. Software Reliability Engineering*, pp.295-303 (1999).
- 12) Bassin, K. and Santhanam, P.: Managing the maintenance of ported, outsourced, and legacy software via orthogonal defect classification, *Proc. Int'l. Conf. Software Maintenance (ICSM '01)*, IEEE Computer Society, pp.726-734 (2001).
- 13) Lintula, H., Koponen, T. and Hotti, V.: Exploring the Maintenance Process through the Defect Management in the Open Source Projects - Four Case Studies, *Proc. Int'l Conf. Software Engineering Advances (ICSEA '06)* (2006).
- 14) Yu, L., Schach, S.R. and Chen, K.: Measuring the maintainability of open-source software, *Proc. Int'l. Symp. Empirical Software Engineering*, 7 pages (2005).