

特異スペクトル解析を用いた欠陥周期性分析手法の提案

～ メンテナンスフェーズにおける欠陥検知の周期性を導出する方法 ～

Proposal of defect analysis method using Singular Spectrum Analysis

- A method of deriving the periodicity of defect detection in the maintenance phase -

研究員 : 安樂 啓之 (インフォテック株式会社)
 澁谷 将行 (株式会社トーセイシステムズ)
 福田 秀樹 (T I S 株式会社)
主査 : 細川 宣啓 (日本アイ・ビー・エム株式会社)
副主査 : 永田 敦 (株式会社日新システムズ)

研究概要

ソフトウェアの派生開発やメンテナンスフェーズを対象とした欠陥対策は、新規開発と並んで重要である。現状のメンテナンスフェーズに対する品質に関連する研究では、残存欠陥の「数」に着目した研究が多く行われている。一方で、残存欠陥の検知数がどのような周期で変動するか、その変動要因は何かの研究は行われていない。そこで我々は、欠陥検知の周期的な傾向変化の検出を行い、その要因を特定することを目指した。

本研究では、周期的な傾向変化を把握する方法として、時系列データの変化を捉えるのに適した特異スペクトル解析を利用する。特異スペクトル解析の結果から特徴的な傾向変化の変動要因を調査することで、周期的な欠陥検知を引き起こす要因を特定する手法を提案する。

Abstract

Defect countermeasures for software-derived development and maintenance phases are important in line with new developments. In research related to quality in the current maintenance phase, many studies focusing on the "number" of remaining defects are conducted. On the other hand, research on what kind of cycle the detected number of remaining defects fluctuates, and its variation factor has not been done. Therefore, we aimed to detect the periodic trend change of defect detection and identify the factor.

In this research, as a method of grasping periodic trend changes, we use a singular spectrum analysis suitable for capturing changes in time series data. We propose a method of identifying factors causing periodic defect detection by investigating the characteristic fluctuation factors of the trend change from the result of the singular spectrum analysis.

1. 背景と問題

1.1 背景

ソフトウェアプロダクトは初期開発しリリースされるまでの期間より、派生開発やメンテナンスフェーズの期間の方が一般的に長期間である。Shari Lawrence Pfleeger は、一般的なソフトウェア開発においては、開発に 1~2 年、保守には 5~6 年ほどかかるとしている^[1]。また IBM のレポートでは、ソフトウェアの全ライフサイクルコストの 40%~80%は保守が占めており、その 60%は機能拡張/仕様追加・変更などの派生開発が占めているとしている^[2]。これらのことから長期間安定してプロダクトを運用し派生開発やメンテナンスを施していくためには、新規開発時と同様に、メンテナンスフェーズについても障害の発生を抑止することが求められる。また、派生開発やメンテナンスフェーズで

は、システムが実稼働していることから、即時対処が求められることが多い実情がある。

1.2 周期性検知に対する課題

派生開発やメンテナンスフェーズにおいては、期間を区切って顧客と契約し、モジュール改修やデータ修正作業などを繰り返すことが多い。毎回必ず同一作業を実施する訳ではないにせよ、似たような活動が繰り返される。一般的に派生回数が増える、あるいはメンテナンス期間が長くなると欠陥の検知数が繰り返し増減することは、開発者であれば誰でも経験していることである。この欠陥の検知数の増減を、時系列分析の観点で観察した論はあまりない。本稿では、「周期性」を時間軸で推移する中で繰り返し発現する性質と置き、課題をメンテナンスフェーズや派生開発における欠陥検知の周期性に定める。

1.3 問題解決のアプローチ

1.2 で述べたメンテナンスフェーズの欠陥検知の周期性を解き明かすために、欠陥検知の件数（以降、欠陥検知数とする）の推移から特徴的な「変動」、つまり急激な変化が発生するサイクル（=周期）を見つけ、要因分析を行うことを提案する。この提案手法を使って、発見した周期性に基づいて、欠陥検知数の予測を織り込むなど要員配置の最適化をより精緻に行うことを期待して、以下の研究課題(RQ: Research Question)を設定した。

RQ1: 派生開発やメンテナンスフェーズにおける欠陥検知の周期性を検出することは可能か？

RQ2: 周期的な欠陥はなぜ発生するのか？

以降、2章にて提案する手法の内容について述べ、3章にて提案手法を3事例のべ約3,500件のデータに適用した結果を説明する。4章で実験を通じて得られた知見、考察を整理し、5章で本研究の意義と妥当性への脅威、将来への展望を述べる。

2. 提案内容

2.1 先行研究

派生開発やメンテナンスフェーズにおける欠陥に関する研究の多くは、残存欠陥数の予測を対象としている^[3]^[4]。これらの研究ではレイリーモデルといった数式モデルによって、検出された欠陥数をもとに、残存欠陥数を予測するものが多い。ここで予測された残存欠陥数はソフトウェアの品質評価に用いられる。その結果をリリース判定、あるいは品質改善の必要なバージョンの特定などに利用する。

また、欠陥エンジニアリングの先行研究では、ソフトウェア欠陥発生メカニズムに着目したモデルを提案し、再発防止に役立てるもの^[5]や、ソフトウェア欠陥の共起性に着目し、潜在的な欠陥の検出に役立てる研究^[6]などがあったものの、周期性に着目した研究は筆者らが調査した限り見当たらなかった。

2.2 先行研究との違い

本研究は、先行研究が対象としていなかった欠陥検知数の変化、特に周期的な変化に着目し、その要因を解き明かす。もし、周期性の存在と、欠陥検知数の変動要因を知ることができれば、ソフトウェア欠陥の発生傾向の分析や予測につなげることが出来る。これによってソフトウェアの品質向上のための新しいアプローチをとることが可能になると考えた。

2.3 提案内容

単位期間あたりの欠陥検知数の推移から特徴的な「変動」を見つけ、その周期性を発見し、それに基づく要因分析を行うことによって、ソフトウェア欠陥の発生傾向の分析や予測を行う手法、CDAM (Cyclic Defect Analysis Method) を提案する。

CDAM では上記周期性を発見するために、単位期間あたりの欠陥検知数を、バグ管理システム (BTS : Bug Tracking System) に存在する障害チケットから取得する。さらにその周期性を発見するために時系列データの異常検知手法である「特異スペクトル解析 (SSA : Singular Spectrum Analysis)」を用いる。これにより本提案が簡便、かつ広く適用可能であることを示す。

以降、2.3.1にてSSAの概要、2.3.2でSSAの採用理由、2.3.3で欠陥検知の周期性検出方法を説明する。

2.3.1 SSAの概要

SSAは時系列データの非定常な変化点検知に用いられる手法である。変化点検知とは、時系列データ上の傾向や性質が変化した点を検出することを指す。SSAは、各時点における過去の特徴と現在の特徴を比較し、その違いを変化度として0~1の範囲で算出する。変化度は0から1に近づくほど、傾向の変化が大きいことを示す値である。なおSSAのメリットとして以下がある。

- ・ SSAは、一定量の時系列データを特異値に分解し、その比較から特徴的なパターンを抽出する手法である為、ノイズの影響を受けにくい。
- ・ 表1に示すように解析を行う為のパラメータが少ないため、解析を容易に行うことができる。
- ・ モデル学習が不要で、データクレンジングなどの事前準備なしでも解析を行える。

上記を踏まえ、SSAのアルゴリズムを、図1をもとに説明する。なお本稿では、3章以降の実験で用いるために必要な内容に留める。詳細は参考文献^[7]^[8]を参照されたい。

【計算手順】

表1 計算パラメータ

t	時刻
w	スライド窓幅 (履歴行列の行数)
K	履歴行列の列数
M	特異値を求めるパターン数
L	ラグ (履歴行列とテスト行列の相対位置)

- (1) 時刻tに対してパラメータ「w, k, L」を用いて、履歴行列 X_1 (過去側)とテスト行列 X_2 (現在側)の行列を作成する。
- (2) X_1 と X_2 に対して、特異値分解し、左特異ベクトルの行列UとQを求める。
- (3) 左特異ベクトルの行列の最大特異値を算出する。
- (4) 最大特異値から変化度を算出する。
- (5) (1) ~ (4)を行列が作成できる全ての時刻tに対して繰り返す。

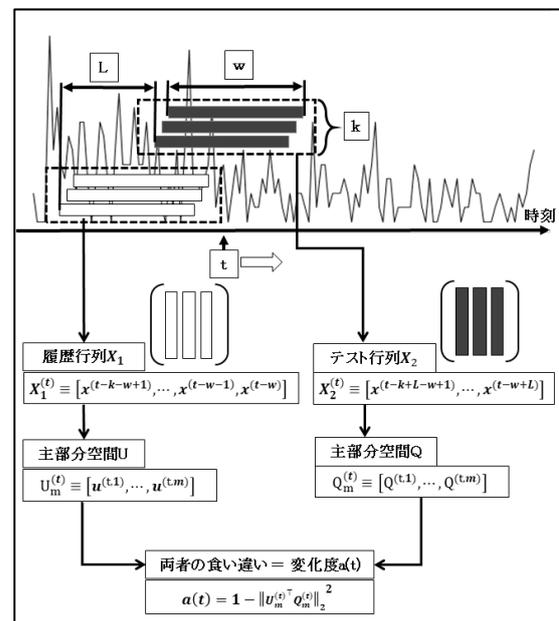


図1 特異スペクトル解析概要

2.3.2 SSAの採用理由

通常は時系列分析を行う際は、バイズ推定などの時系列モデル、あるいは自己回帰モデル (ARモデル)、自己回帰和分移動平均モデル (ARIMAモデル) などを用いるのが一般的である。しかし、これらの分析を用いる場合には、モデル学習が必要となり、導入負荷が高くなる。

一方で SSA は、2.3.1 で示した通り、モデル学習が不要であり、クレンジングなどの事前準備をしなくても導入可能でありながら、ノイズに強く、時系列上の変化を捉えることが可能な分析方法である。また、計算パラメータの推定を力まかせ探索 (Brute-force search) で行えるため、他の時系列分析よりも導入が容易な分析方法である。

欠陥検知の周期性を検出する「効果」があり「簡便」な方法を採用することで、今後の知見の蓄積や研究のハードルを下げられると考えたため、CDAM では導入が容易な SSA を採用することにした。付録 3 にその他の代表的な異常検知の方法論との比較を示す。

2.3.3 欠陥検知の周期性検出方法

CDAM による欠陥検知の周期性を検出する手順を以下に説明する。

(1) 解析対象データの準備

単位期間 (日次, 週次, 月次, 年次など) あたりの欠陥検知数の推移が分かるデータを準備する。CDAM では、BTS に本番リリース後に登録された障害チケットの登録日をもとに単位期間ごとの登録チケット数を集計する。以降、集計したデータを時系列データと呼ぶ。

(2) 欠陥検知数の変動傾向を変化度として可視化

(1) で準備した時系列データに対して SSA を適用し、欠陥検知数の変動傾向を変化度として可視化する。具体的には SSA で得た結果をグラフ化することで、目視で類似の変動傾向を捉えることを可能にする。

また特徴的な変動傾向を明瞭に検出できるようにするために、計算パラメータ w (スライド窓幅) を調整する。これは、変化度を明瞭に検出するために一番寄与するパラメータが w だからである。 w は時刻 t における特徴を求める際に、どれだけの期間のデータを用いるかを意味するパラメータである。変化度の発生間隔が w よりも小さいと変化が探知できなくなるため^[8]、変化度の山とそうでないところが明瞭に判別できるように、 w を総当たりで変化させながら設定する。

(3) 変化度の傾向から周期性を検出

(2) で可視化した変化度の変動傾向を下記の手順で確認し、欠陥検知に対する周期性を検出する。

① 変化が特徴的な箇所を探す

変化が特徴的な箇所は、何らかの事象がプロジェクトに発生している可能性が高い。そのため、優先的に確認を行う。代表的な変動傾向は付録 2 を参照。

② 類似した変化がある箇所を探す

① で検出した特徴的な変化と同様の変化が、他にもないかを確認する。同様の変化であれば周期性が存在する可能性が高い。

③ 傾向が変動した要因を調査する

①, ② で特定した個所の変動要因をチケット情報やプロジェクト記録、あるいはインタビューなどにより調査する。この調査によって、①と②で特定した箇所で、同様の変動要因が確認できれば、周期性の検出と、周期性をもたらす要因の特定ができたことになる。

なお手順 (2) の部分は、R 言語を用いて行った。これは、SSA のアルゴリズムを実行するのに必要なライブラリが R 言語で公開されており、計算パラメータ w の探索的な推定ロジックを容易に組み込めたからである。

3. 実験と評価

3.1 実験手順

2 章で提案した内容についての有効性を確認するため、以下の手順で実験を行った。

(1) 解析対象データの準備

実験対象プロジェクトに対して、BTS から抽出した本番リリース後の障害チケットの登録数を集計し、時系列データを作成する。本研究で用いた対象プロジェクトのプロファイルとチケット件数を表 2 に示す。

表 2 対象プロジェクトプロファイルとチケット数

プロジェクト	ドメイン	データ期間	チケット数	集計単位
A	金融系基幹システム	約 3 年半	約 2,000 件	週次
B	組み込みシステム	約 7 年半	約 1,500 件	週次
C	金融系基幹システム	約 5 年	約 30 件	月次

(2) 欠陥検知数の変動傾向を変化度として可視化

(1) に対して、 w を 4~80 まで 1 刻みで変化させながら SSA を適用し、特徴的な変動傾向が明瞭になる w を探し出す。

(3) 変化度の傾向から周期性を検出

2.3.3 の (3) に従い、(2) で可視化した変化度の変動傾向を時系列で確認し、変化度の周期性を検出する。

3.2 実験結果

3.2.1 欠陥検知数の変動傾向を変化度として可視化

各プロジェクトに対して SSA を適用した結果、変動傾向を捉えることができた。紙面の都合上、プロジェクト A 全 76 枚の図のうち、特徴的な差が現れた、 w のパターンの図のみを図 2 に示す。なお、図中の棒グラフは欠陥検知数、折れ線は変化度を表す。(拡大図は付録 1 を参照されたい)

図 2 を目視で確認した結果、以下の理由から $w=12$ は変化が明瞭であり、変動傾向を捉えることが可能であると判断した。

- ① $w=4$ の場合は、変化度の変動が激しく類似を検出するのが困難である。
- ② $w=8$ を超えると変動は多少落ち着くものの、全体の半分以上で変化度の変動が観察される。
- ③ $w=12$ (およそ四半期) の近傍を境に変化度の変動箇所が顕著に少なくなる。
- ④ $w=12$ を超えて増やしていくと、四半期の倍数 (半期, 年間など) においては変化度の山とそうでない値が明瞭になる。
- ⑤ $w=26$ を超えると、データ取得後 w に比例したタイミングで一度高い変化度を示す。それ以降についてはだんだんと変化度のメリハリが失われる傾向になる。

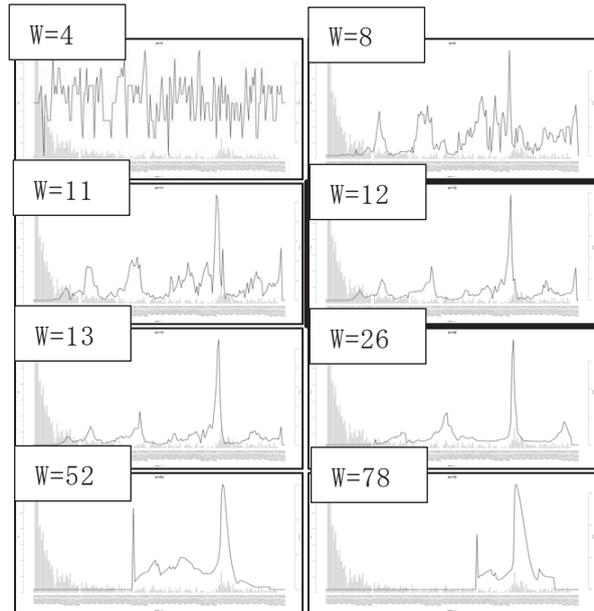


図 2 特異スペクトル解析結果

3.2.2 変化度の傾向から周期性を検出

3.2.1 の結果に対して、変化度の周期性の存在確認と要因調査を行った。その結果、周期性のある変動と、周期性がない変動を検出した。要因を調査した結果を図 3~図 5 および表 3 に示す。

本実験から、欠陥検知の変化度に周期性が存在することが明らかになった。また、その周期性に影響を与える要因として、企業やプロジェクトにおける活動が大きく関係して

いることが確認できた。（表3の①④⑦が該当）一方で、変化度の周期性の分析に影響を与えるノイズが存在することも確認できた。（表3の②③⑤⑥⑧が該当）

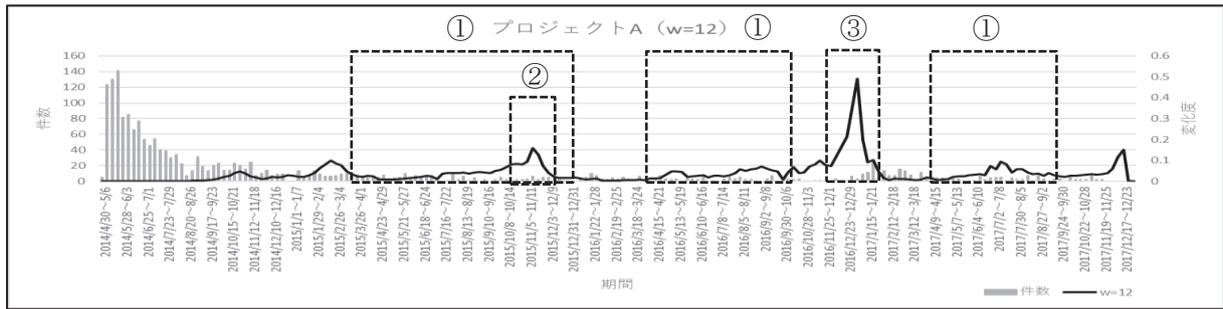


図3 プロジェクトA (W=12) のグラフ

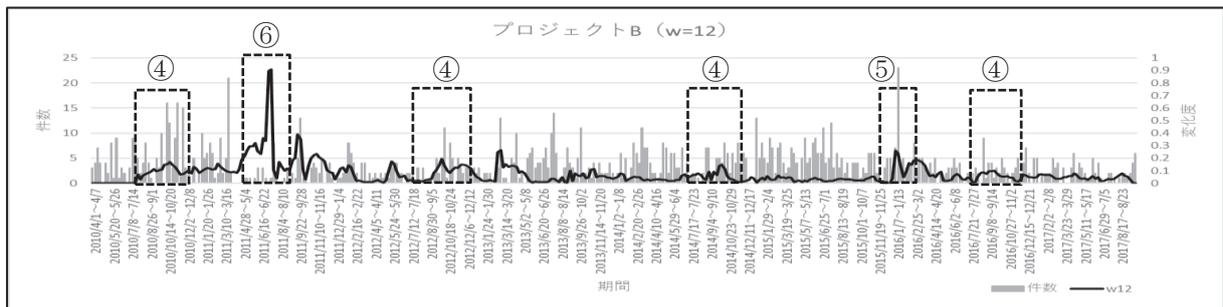


図4 プロジェクトB (W=12) のグラフ

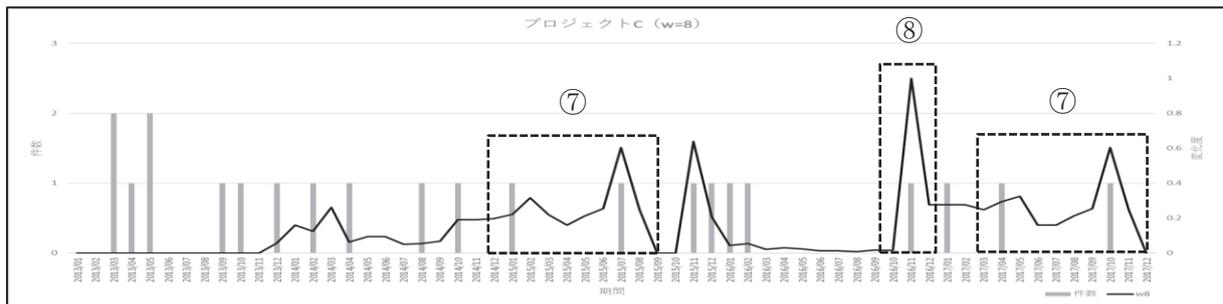


図5 プロジェクトC (W=8) のグラフ

表3 周期性の有無と変動要因の一覧

No	周期性	要因	詳細
①	有	顧客との契約	顧客との契約に合わせて、変化度の上昇と下降がセットになっていた。
②	無	リファクタリング実施	構造改善を目的とした計画的なリファクタリングに伴う調査によって欠陥検知が急増した。
③	無	見積より短い期間での開発	大規模な派生開発を事情により当初見積より短い期間で実施。開発時に除去できなかった欠陥を多く検知した。
④	有	外部展示会対応	外部展示会に出展するために広範囲のテストを実施。普段テストしない機能のテストにて欠陥検知が急増した。
⑤	無	チケットの一括登録	外部委託作業により検知された欠陥を、納品時にBTSへ一括登録したことで変化度が大きくなった。
⑥	無	検知数の急激な減少	直前(2011/3/10付近)にチケットの大量登録があったが、その後の登録数激減により、変化度が大きくなった。
⑦	有	新規案件の立ち上げ期	前案件の終了後一旦体制を縮小し、新たな案件立ち上げに合わせて新規要員をアサインした。

No	周期性	要因	詳細
⑧	無	久しぶりの欠陥登録	しばらく欠陥が検知されていない状況で欠陥を検知したことで、変化度が大きくなった。

4. 考察

冒頭に提示した RQ に対し、実験結果を踏まえて考察を示す。

(1) RQ1: 派生開発やメンテナンスフェーズにおける欠陥検知の周期性を検出することは可能か？

本研究で提案する CDAM を派生開発やメンテナンスフェーズにあるプロジェクトの時系列データに適用することにより、欠陥検知の周期性について数理モデルを用いて定量的に説明できた。CDAM にて採用した数理モデルは、以下の特長をもつ簡便なモデルである。

- ① R 言語などの汎用的な分析ツール、および BTS などから取得した時系列データがあれば分析が可能。
- ② SSA のロジックは公開されており適用自体は容易。
- ③ データクレンジング等の事前準備が不要であるため、最終的な目視分析を除いた一連の手順は自動化が可能。

また、CDAM の適用によって、対象とした全てのプロジェクトで変化度の周期性を検出でき、あわせて、その要因を追究することができた。これらのことから、メンテナンスフェーズにある他のプロジェクトにおいても CDAM を適用すれば、周期性の検出、および要因の分析が可能になると考える。

(2) RQ2: 周期的な欠陥はなぜ発生するのか？

検出した変化度の周期性をもたらす要因は、3.2.2 に示す調査の結果から企業やプロジェクトにおける周期的な活動、及び外部イベントなどが考えられる。特に表 3 で示した展示会の周期と連動した変化度が観察されたことは、その代表的な例である。

3.2.1 の中で、CDAM による分析から検出された変化度の周期は、四半期、半年、1 年であった。この周期で行っている活動を調べたところ、企業活動の周期性と強い関連性が認められた。例えば、四半期や半期単位で行われる品質活動などの短期計画、あるいは年度と連動した計画などは、変化度の周期と同じ四半期、半年、あるいは 1 年単位である。さらに、3.2.2 の表 3 で示した①契約の周期、④の展示会周期、あるいは⑦の要員サイクルとの関連性からも読み取ることができる。なお、結果的に周期的な欠陥の検知は企業活動の周期と関連づいていることが分かったが、この変化の兆しは欠陥数の多寡だけでなく SSA を利用して変化度を可視化することによって初めて分かりやすく検知できるものである。

企業活動の周期と変化度の周期が関連していることを踏まえ、企業の活動計画を立案することによって、品質活動の作業量を平準化することが出来ると考える。

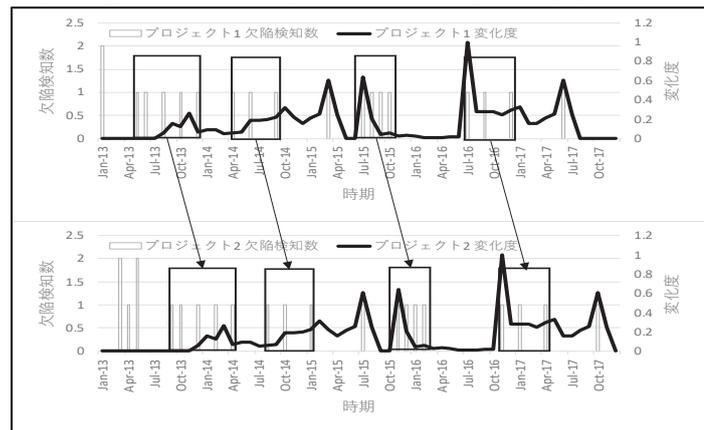


図 6 複数プロジェクトのピーク調整イメージ

プロジェクト C を例として説明する。図 6 は、プロジェクト C に CDAM を適用し $w=8$ を指定して出力したグラフを上下に並べて、4 か月ずらしたものである。同様のプロジェクトを別に立ち上げると仮定した場合、変化度の周期性を考慮することができれば、図の下端に示すように、開始時期をずらし、欠陥検知のピークが重ならないように調整した案件計画を立てることも可能となる。これにより、欠陥検知のピーク（図中の枠囲みで示した

箇所)が重ならないため、これら両プロジェクトの品質活動を平準化することができる。

なお、SSAの結果より、変化度の周期性を確認した副次効果として、周期性の無い変化(表3の②③⑤⑥⑧)も捉えることができた。変化度の周期性を持たない変化は、本来は想定外の欠陥検知の変動である場合が多く、プロジェクトに影響を与える可能性が高い。この想定外の要因を知ることで、要因の除去や、発生予測を事前に織り込むことが可能になると考える。

5. まとめ

5.1 本研究の意義

今まで派生開発やメンテナンスフェーズでは残存欠陥数が多く注目されていたが、その残存欠陥の検知傾向を変動させる要因があることを簡便に特定する手法を示せた。これによりメンテナンスフェーズにおける欠陥に対する新たな管理指針を示すことができたと言える。

本研究において観察された周期性の捕捉は、本質的には品質を見積もることに他ならない。従来品質活動は、コストと期間、要員数を過去のプロジェクトの類推などから見積もっていた。これに加え、欠陥の発生数、及び検出タイミングを周期性から予測することで、品質活動の見積りをより精緻に行うことが期待できる。

5.2 妥当性への脅威、今後の展望

本稿での研究では3プロジェクトのデータに基づいて解析を行っているが、本研究で扱っていない、異なるメンテナンスフェーズのプロジェクトについても当てはまるかについては、検証を行う必要があると考える。

4章の考察で示した欠陥検知の周期性と、企業やプロジェクトにおける周期的な活動との関連性、およびその活動がプロジェクト固有要因によるものか、共通的な要因によるものかといったより詳細な分析には、幅広いデータによる裏付け強化が必要である。また、本稿で発見した周期性を発生させる要因以外にも、要因はあると考える。そのため、今回の手法での追実験や他案件への適用に期待したい。この研究が、より広範囲・より深い欠陥研究・洞察の礎となれば幸いである。

参考文献

- [1] Shari Lawrence Pfleeger 著, 堀内 泰輔訳, ソフトウェア工学 理論と実践, P.473, ピアソン・エディケーション, 2001
- [2] Rational Software IBM Japan Ltd., 品質マネジメント技術 COQ: Cost of Quality 概説, 2011
- [3] 安藝優子, 野中誠, 摘出済み欠陥数を考慮したレイリーモデルに基づくソフトウェア欠陥予測手法, 経営情報学会 全国研究発表大会要旨集 2010, 2010
- [4] 吉村博昭, 森岳志, 倉下亮, 野中誠, プロセスおよび出荷後バグメトリクスによる残存バグ予測, SQiP2010, 2010
- [5] Project Fabre (細川宣啓ら), “過失に着目した欠陥のモデリング”, JaSST2013, 2013
- [6] 細川宣啓, 永田敦, 澁谷 将行, 山崎 真一, 「ソフトウェア欠陥の共起性を利用した欠陥推定手法の提案 ~ 共起欠陥推定アプローチによる潜在欠陥の捕捉 ~」, 日本科学技術連盟 SQiP 研究会, 2016
- [7] 井出剛, 入門 機械学習による異常検知 -Rによる実践ガイド-, P.199, 2015
- [8] 井出剛, 井上恵介, 非線形変換を利用した時系列データからの知識発見, 日本ソフトウェア科学会データマイニング研究会, 研究会資料シリーズ ISSN 1341-870X, No. 29, pp. 1-8., 2004

付録 1. 変化度の山を明瞭に示す w 値の観察データ

プロジェクト A, B に対して, CDAM による分析を行うため, SSA のパラメータ w について, 4~80 に変化させながら各グラフを観察した. 以下に観察した結果, 特徴的であることを示す. なお, グラフの棒グラフが検知した欠陥数の実数, 折れ線グラフが変化度を示す.

- ① $w=4, 5$ の場合は, 変化度が $10^{-16} \sim 10^{-15}$ などほとんど 0 に近い値をとる. 振幅は小さいが常に激しく振動している状態にあり, 変化度の山を明瞭には判断することができない. そのため, 不適であると判断する.
- ② $w=5$ を超えると変化度の振幅が 0.1 を超え, 1~0 のスコープで観察可能な範囲になるが, 変化の激しい傾向は変わらず, かなりの範囲 (全体の半分以上) を変化度の山として捉えてしまうため, 不適と判断する.
- ③ $w=12$ (およそ四半期) を境に変化度の山を示す範囲が顕著に少なくなる. プロジェクト A の場合, 図 3, および図 4 を見ると変化度の山を示す箇所が急減していることが分かる. プロジェクト B においては, 図 17, 図 18 が示すように $w=10$ を境に変化している. 両方のプロジェクトを考慮すると $w=12$ の近傍から傾向が変化し, 変化度の山を示す箇所が明瞭に示されるようになる.
- ④ $w=12$ を超えて増やしていくと, 四半期の倍数 (半期, 年間など) においては変化度の山とそうでない値が明瞭になる.
- ⑤ $w=26$ を超えると, データ取得後 w に比例したタイミングで一度高い変化度を示す. それ以降についてはだんだんと変化度のメリハリが失われる傾向になる.
- ⑥ 上記から, 変化度の山を明確に, かつ最初の山をできるだけ早く捉えることのできる, $w=12$ を分析に採用した.

上記①~⑤を読み取れるように $w=4$ (1 か月, 最小値), $w=8$ (2 か月), $w=11, 12, 13$ (3 か月, およびその近傍値), $w=26$ (6 か月), $w=52$ (1 年), $w=78$ (1 年半) を指定し, 特異スペクトル解析を行ったグラフを図 1~図 16 に図示する. また, プロジェクト B の $w=9, w=10$ のグラフを図 17, 図 18 にそれぞれ示す.

プロジェクト A に関するグラフ

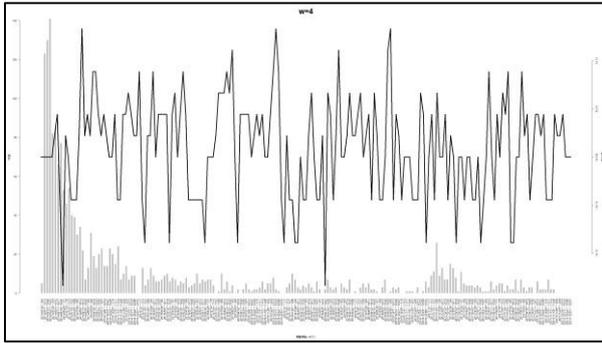


図 1 プロジェクト A (w=4)

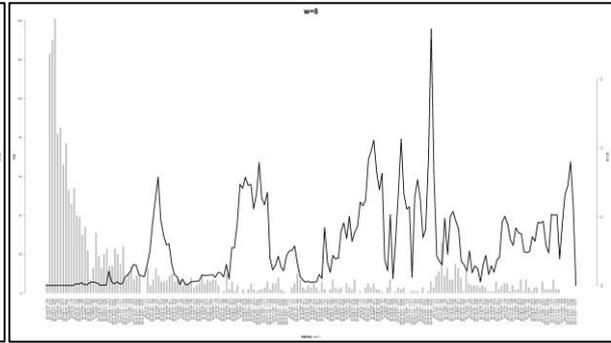


図 2 プロジェクト A (w=8)

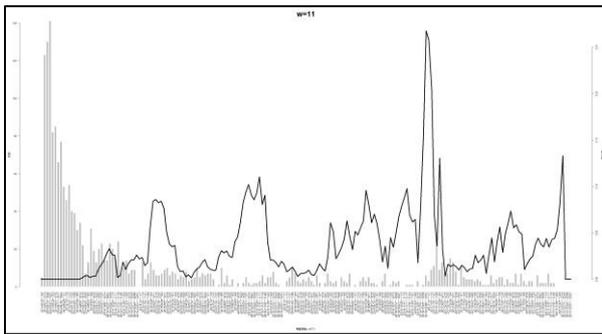


図 3 プロジェクト A (w=11)

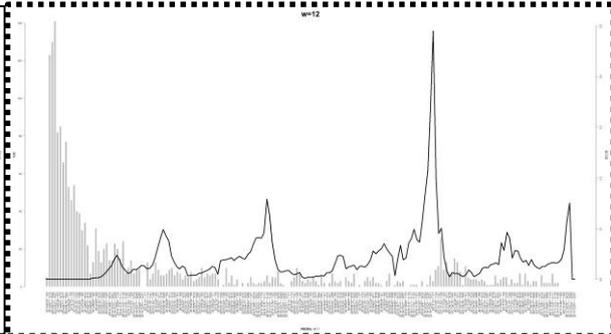


図 4 プロジェクト A (w=12)

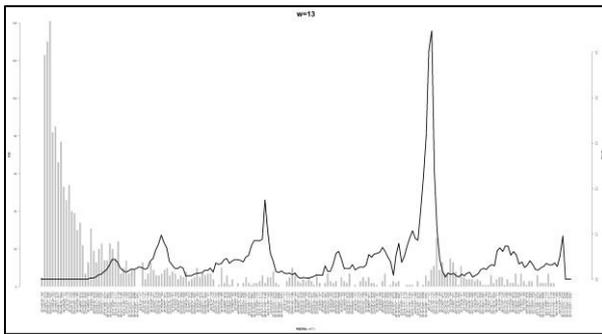


図 5 プロジェクト A (w=13)

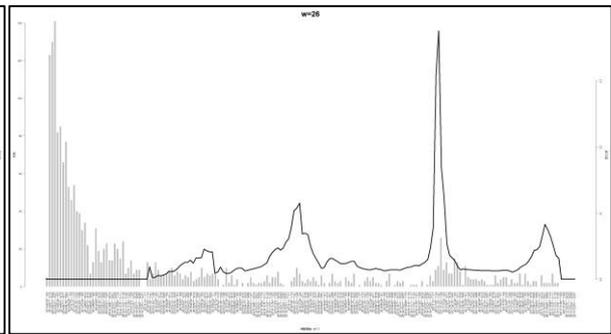


図 6 プロジェクト A (w=26)

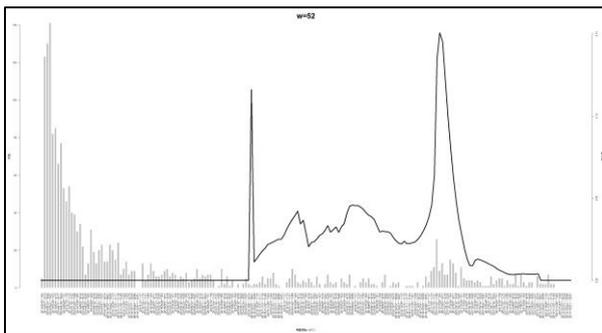


図 7 プロジェクト A (w=52)

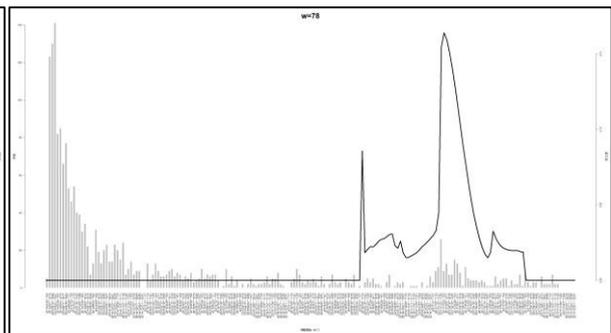


図 8 プロジェクト A (w=78)

プロジェクト B に関するグラフ

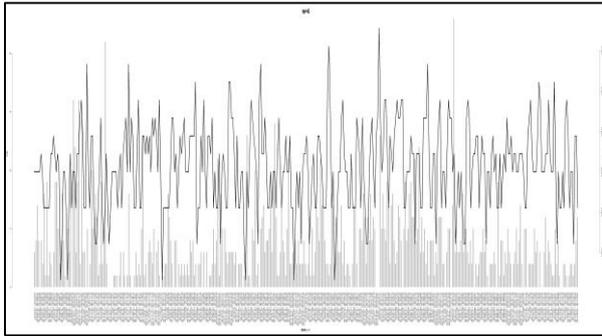


図 9 プロジェクト B (w=4)

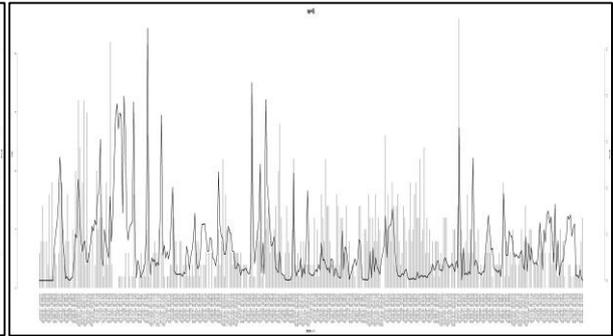


図 10 プロジェクト B (w=8)

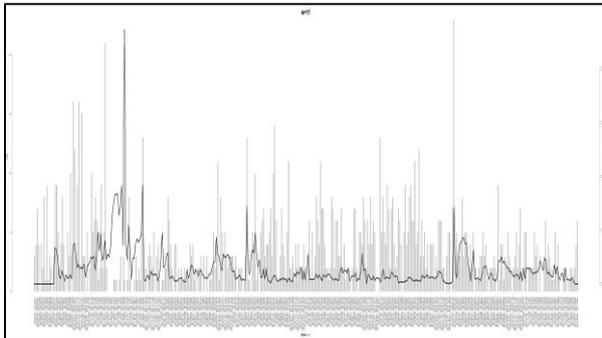


図 11 プロジェクト B (w=11)

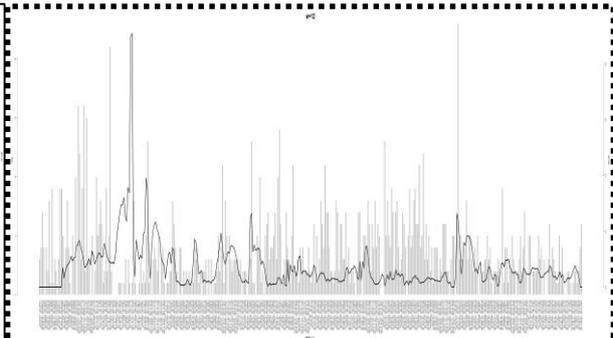


図 12 プロジェクト B (w=12)

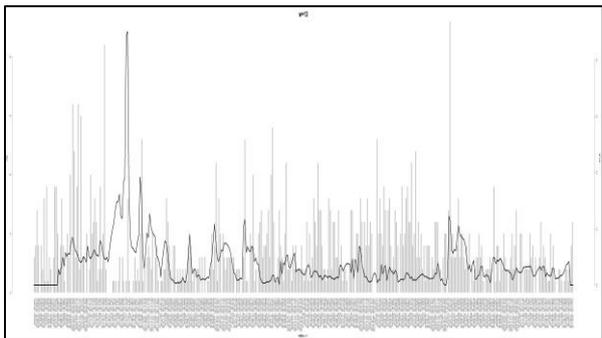


図 13 プロジェクト B (w=13)

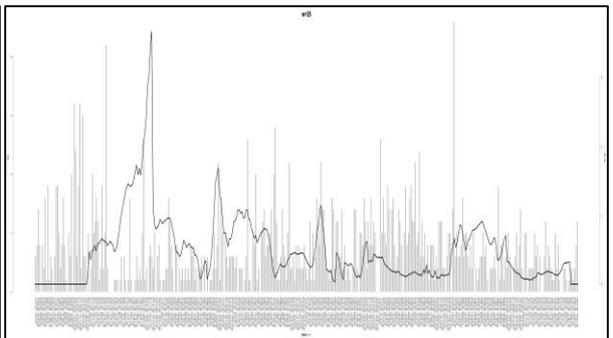


図 14 プロジェクト B (w=26)

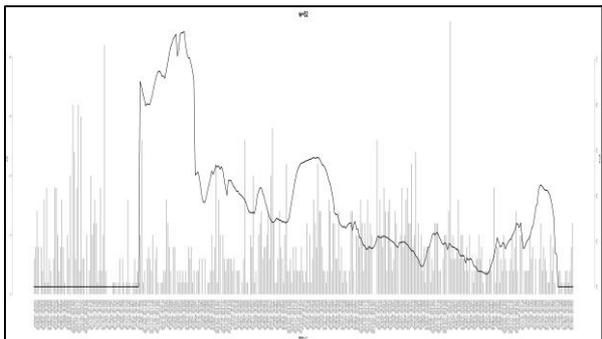


図 15 プロジェクト B (w=52)

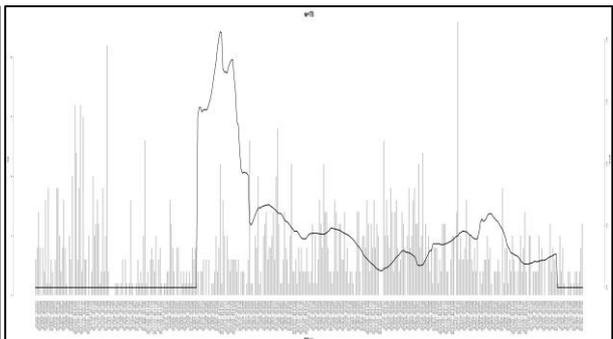


図 16 プロジェクト B (w=78)

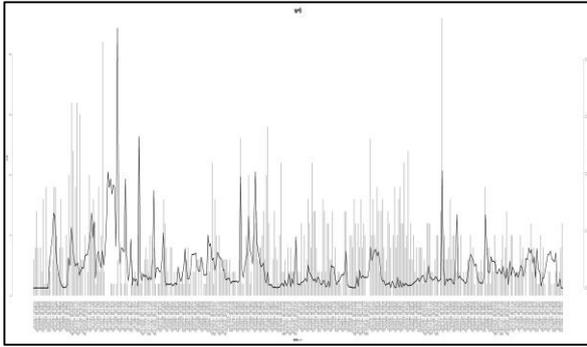


図 17 プロジェクト B (w=9)

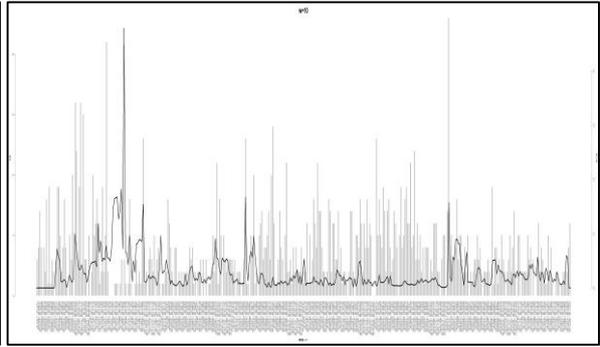


図 18 プロジェクト B (w=10)

付録 2. 変化度に基づく欠陥検知傾向の変化とプロジェクト状況

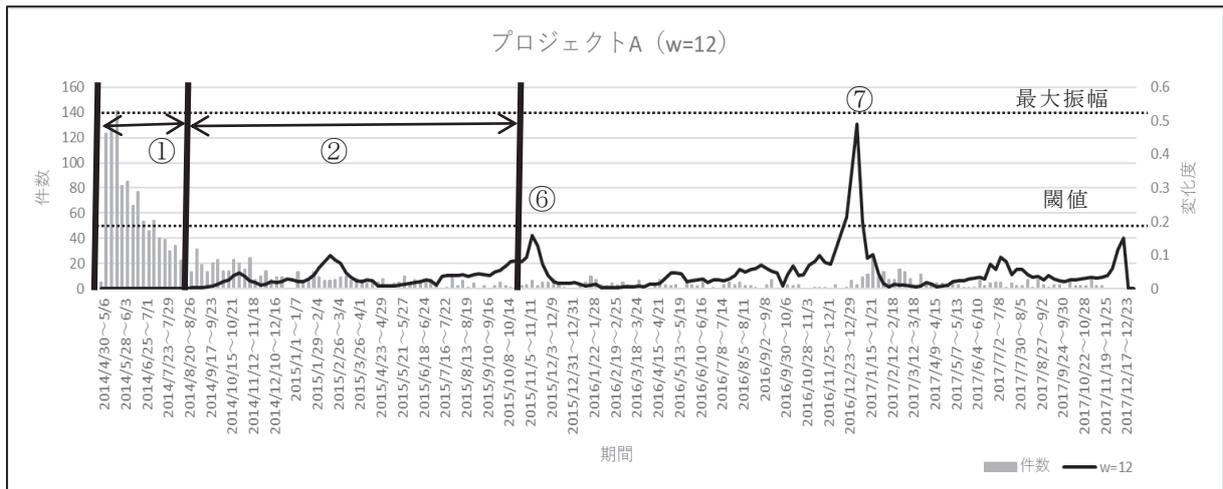


図 19 プロジェクト A, w=12 のグラフ

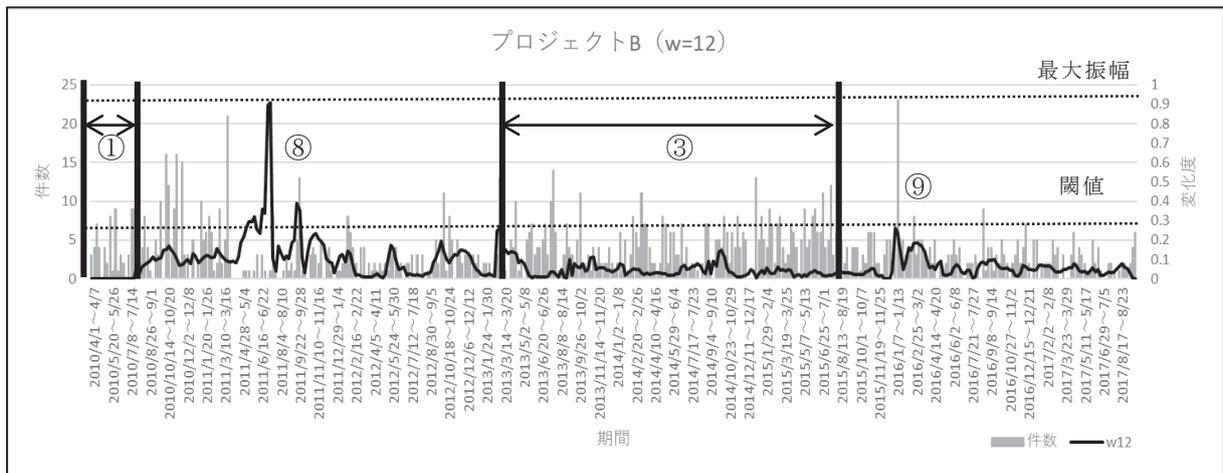


図 20 プロジェクト B, w=12 のグラフ

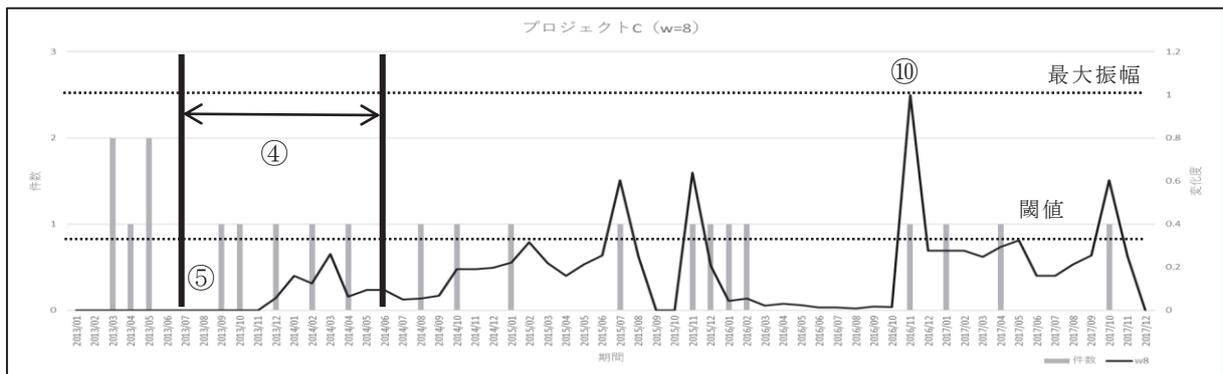


図 21 プロジェクト C. w=8 のグラフ

元データは日次、週次、月次など様々な括り方ができる。できる限り詳細な単位で見た方が変化に早く気づきやすいが、日次データの場合は土日や祝日など、およそ欠陥が検知されにくい日が挟まるため、プロジェクト A, B では週次データを分析対象とした。プロジェクト C は元々の欠陥件数が対象期間に対して少なすぎ、週次だと殆どが 0 件になってしまうことから、月次データを分析対象とした。

(1) 欠陥検知傾向が安定している状況

- ・ プロジェクト A, B の①の区間は、データ取得開始後 12 週間以内であるため、チケットの発行頻度の変動傾向を検知していない。
- ・ 同プロジェクト②の区間では、チケットの発行は低いレベルを保ち、変化度も同じ傾向を示している。
- ・ プロジェクト B の③の区間では、四半期の切れ目にチケットが減少する傾向、および期間内は数回程度上下することを繰り返している。
- ・ プロジェクト C については、他の 2 プロジェクトと異なり、月次でのデータの提供であったため、 $w=12$ 相当での確認ができなかった。一方、 $w=8$ において明瞭な傾向が出ていたので、本内容での確認を行うこととした。
- ・ プロジェクト C の④の区間では、1 か月おきにチケットの検知を繰り返している。
- ・ プロジェクト C の⑤以前の箇所（開始 8 か月以前）は $w=8$ であるため比較対象外となるので、変化度の検出がない。

グラフ上の上記に示した内容、およびそれに類似した変化度が低い状況を維持している箇所を観察した結果、以下のいずれかを示すものと考えられる。

- ① 障害チケット検知数の変化がない。
- ② w 値と近いサイクルで障害チケット検知数の周期的な変動がある。
- ③ 開始後 w に相当する期間が経過していない。

(2) 欠陥検知傾向に顕著な変化が発生した状況

それぞれのグラフの変化度が 0.2~0.3 以上を示している箇所を観察することにより、以下が分かった。

- ・ プロジェクト A の⑥については、グラフ上判別しにくいですが、変化度が 0.2 を超えている箇所を境に月当りの欠陥検知数がそれまでに比べて 4 倍程度となっていることが分かった。また当時の担当からヒアリングしたところによると、この時期に大規模なアプリケーションの構造改善を目的としたリファクタリングを実施しており、欠陥の検知量が増えているとの事であった。
- ・ プロジェクト A の⑦のタイミングでは、欠陥検知数が上昇している。当時のプロジェクトの記録を確認したところ、この時期には大規模な派生開発案件があったが、年度内に開発を完了する必要があるという事情から短期開発となったため、障害が増えていたことを確認した。
- ・ プロジェクト B の⑧のタイミングでは、およそ 12 週間前に欠陥検知のピークがあったが、検知時タイミングの欠陥検知数は落ち着いており、それによる変化を検知したのと考えられる。
- ・ プロジェクト B の⑨のタイミングでは、欠陥検知数の急増に伴い、変化度の上昇を確認した。これについて当時の担当へ確認したところ、外部委託分の障害チケットの一括入力があったとのことで、傾向の変化が表れているということが分かった。
- ・ プロジェクト C の⑩もしばらく欠陥の検知がなく、9 ヶ月ぶりに検知されたことによつて変化度の上昇を示していた。このプロジェクトはパッケージベースのプロジェクトでリリース後 10 年ほど経過していることから、欠陥はほとんど検知されない傾向にある。

グラフ上の上記に示した箇所から、変化度が急増しているケースは以下が共に成立していることを示しているものと考えられる。

- ① 欠陥検知の急増/急減がある。
- ② w 値と近いサイクルで障害チケット検知数の周期的な変動がある。

なお、顕著な変化を捉えるのに、今回の 3 プロジェクトでは変化度の最大振幅の 30%程度を目安にすると良いようである、ということが分かった。

本分析で得た知見のまとめを付録 4 に示す。

付録3. 各種機械学習モデルの比較（井出 剛著「入門 機械学習による異常検知」より筆者らにて整理）

表1 各種機械学習モデルの比較

機能	入力対象	確率モデル	検出対象	応用	特徴
外れ値検出	多次元ベクトル	独立モデル (単純な閾値) (クラスター外れ値)	外れ値	不正検知 侵入検知	急激な変化に弱い
変化点検出	多次元時系列	近傍法による異常部位検出	時系列上の急激な変化 バースト的異常	ネット攻撃検出 ワーム検出 音声認識	リアルタイム分析
		特異スペクトル解析/変換	時系列上の急激な変化 バースト的異常	心電図による異常検出	ノイズに強い リアルタイム分析
		自己回帰モデル (赤池情報量基準 AIC)	時系列上の急激な変化 バースト的異常	過去データを用いた売上予測	自分自身の過去データからの予測
		線形状態空間モデル (部分空間同定) カルマン・フィルタ (逐次推定法)	測定困難な時系列データの異常・急激な変化	脳圧推定	過去と未来の共通点を探す手法 状態系列推定

付録 4. SSA による可視化に際して、知り得たこと

3つの実験データの分析において、SSAを行う上で得たノウハウを下記に記す。

① w 値の設定値に関して

- ・ パラメータで唯一設定したのは w であるが、最小値は m の2倍の値である。また、最大値は元となるデータ期間の長さの $2/3$ までである。これらは履歴行列、テスト行列の抽出の仕組み上、その範囲を超えられないためである。（最小値より小さい値、最大値より大きい値を指定するとエラーとなる）
- ・ w で指定した周期に達するまでは、変化度は0のままである。

② 変化度を見極めるための工夫

- ・ 変化度を見極める際に、グラフの平滑化を行うと、細かな変化を滑らかにならすことができる（これをスプライン曲線による補間という）。グラフのノイズが除去できるため、似たような変化をより発見しやすくなる。

③ データ量の多寡による変化度の現れ方の違い

- ・ データ量が少ないほど、変化量の振幅が大きくなりやすい。

④ 周期性の読み取り方

- ・ 周期性には2つある。1つは、周期的に同じような欠陥検知の傾向があるケース。この場合、変化度のグラフは振幅が小さくなだらかになる。もう1つは、大きな変化が周期的に発生するケース。この場合は似た振幅形状のグラフが複数箇所に現れる。
- ・ 今回の解析で使ったソースコードを用いると、ループ処理で w の値を指定した間隔でしらみつぶしに変化させていくため、一度に複数の変化度のデータが得られる。これを比較しながら、最も変化度の山が際立つものを選び出す。
- ・ ある w 値で周期性が得られた場合、その倍数の w 値でも同様の波形が得られるが、最初の変化度の山をより早く検知できるように、それらのうち最も小さい w 値を選択するのがよい。
- ・ 初期リリースで欠陥が多発し、その後落ち着いてくるプロジェクトでは、最初の変化量の山を過ぎてから、 w 値の期間が経過したところで履歴行列の欠陥数よりテスト行列の欠陥数が減ることによって、変化量の山が発生する。変化量としては欠陥数が増えても減っても山が立つため、読み取る際に注意が必要。