ＳＴＭＥｐｊによるソフトウェア品質改善の推進（２）
１Ｃ－７　― プログラム静的解析センターの運用 ―

原田明宏† 杉山昭洋† 吉崎浩二† 小笠原秀人‡
†株式会社トプコン　技術本部　S&S 推進部
‡株式会社東芝　研究開発センター　S&S 研究所

１．はじめに

ソフトウェアの欠陥を開発工程のより上流で発見･修正する方法として「コード･レビュー(以後、レビュー)」の効果が重要視されてきている。

当社では、プログラム静的解析ツールを導入し、'97 年 7 月から約一年間に社内ソースコード(C､C++)約 160 万 step の解析を行った。

解析によって得られた警告メッセージやメトリクスをもとに､開発担当者を交えてレビューを行うことにより、約 3600 箇所のソースコード修正に寄与した。その運用システムと活動内容について報告する。

２．レビューの効率化

２．１　レビューの自動化

近年、レビューによるソフトウェアの欠陥の発見効率が注目されている。しかし、手作業による最も効率的なレビュー速度は､150 行/時と言う報告[1]があり、ここ数年のソフトウェア容量の急激な増加に伴い、人手で行うレビューは事実上困難な状況になってきている。

そこで、静的解析ツールを利用してレビュー箇所の抽出を自動化し、効率的なレビューの実施を目指した。

２．２　静的解析ツールの導入

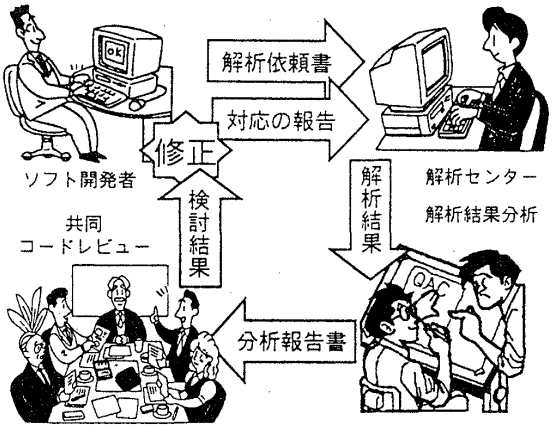
当社では'97 年 6 月より STMEpj によるソフトウェア品質改善を推進しており、その活動の一環として静的解析ツール QAC/C++[1]を導入した。同時に QAC 解析センター（以後、センター）を開設、社内製品ソフトウェアの解析を行っている。QAC/C++は C/C++言語プログラムを解析し、ソースコード上の問題点を指摘する警告メッセージをソースコード中に埋め込んだファイルを出力する。

すでに東芝 S&S 研究所において運用実績があり、運用に関する技術、及びツール利用に関して同研究所の指導の下に、運用を開始した。

３．解析センターの運用

次にセンターで行っている活動について述べる。(図１)

図１ センターの運用



１）警告メッセージの重要度分類

QAC/C++には多くの警告(C:800 以上、C++:300 以上)が用意されており、その種類も危険な記述を指摘するものからスタイルに関するものまで様々な警告がある。それらすべてを出力すると膨大な数の警告メッセージが埋め込まれるため、あらかじめ重要度別に分類して埋め込む警告メッセージの数を絞り込むことが必要である。

センター運用開始時に、東芝 S&S 研究所より警告を選択するための重要度別に分類した警告データを提供していただき解析を行った。

その後、当社における解析事例より、当社の警告出現傾向を分析して警告の選択を行った。その際に分類した警告に対してセンターガイドとしてA、B、C のランクを設け、レビュー時に開発担当者に注意を促すようにしている。表１に各ランクの内容を示す。

表１ センターガイド

Aランク	危険度の高い警告	比較と代入の記述ミス等
Bランク	品質向上のため修正すべき警告	条件分岐におけるその他の場合抜け等
Cランク	保守性､移植性に関する警告	制御範囲の明示等

２）メトリクス基準値の設定

QAC/C++の機能としてメトリクス計測機能がある。この機能をソフトウェアの定量的な評価に利用するため、警告と同様にメトリクス計測値について分析し、当社における基準値を設定した。表２に主に使用しているメトリクスを挙げる。

Promoting the software quality improvement activites by STMEpj (2)
Akihiro Harada†, Akihiro Sugiyama†, Kouji Yoshizaki†,
Hideto Ogasawara‡
†{a-harada, akihiro_sugiyama, kouji_yoshizaki}@topcon.co.jp
‡hideto@ssel.toshiba.co.jp
†S&S Group, S&S Promoting Dept., TOPCON Corporation
‡System&Software Research Laboratories, TOSHIBA Corporation

3）共同レビューの実施

解析結果は、解析結果レポート(表3)としてまとめて依頼部門に渡される。

しかし、そのままでは警告の意味や修正方法などが分かりにくいものもあるため、解析担当者と依頼部門の開発担当者で共同レビューを行っている。レビュー時には、指摘された警告の具体例とその意味、及びその危険性について説明し、警告の対応方法などを検討している。

また、共同レビュー後はさらに開発担当者間でより詳細なレビューの実施を依頼し、プログラムへフィードバックしている。そこで得られた対応データは、センターに戻され、当社ソフトウェアの品質改善データとして蓄積される。

4）コーディング･ガイドラインの作成

センターでは蓄積された品質改善データをもとに、誤り易いコーディング上の傾向を把握し、それらを「コーディング・ガイドライン」としてまとめ、説明会等を通して技術者に教育を行っている。

「コーディング・ガイドライン」は、表4に示すように三つの章に分かれており、警告メッセージの重要度分類に対応する形で構成されている。

5）利用環境の整備

センターでは、過去の解析事例及び依頼部門から戻される品質改善データをデータベース化し、警告の重要度分類やレビュー時の統計資料として利用している。

また､ユーティリティツールの拡張や統計データ(図 2)のグラフ化ツールなどを整備し、解析作業の効率化を図っている。

表2 主な使用メトリクス

メトリクス	内容
コメント密度	ファイル内のコメントの占める割合
Cyclomatic数	モジュールの複雑度
最大ネスト数	モジュール内におけるネストの最大数
命令行数	モジュールの命令行数

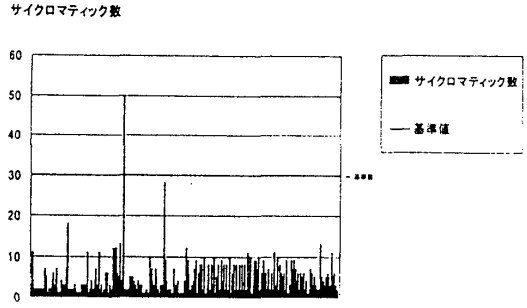
表3 解析結果レポートの内容

メトリクス計測結果	設定された基準値からの外れ率
警告メッセージ一覧	警告メッセージの出現頻度一覧
レビュー資料	各警告メッセージに対する具体例及びその説明

表4 コーディング・ガイドラインの構成

プログラミング・標準	プログラマが必ず従うべき項目
プログラミング・ガイドライン	安全なプログラムを記述するために守るべき項目
スタイル・ガイドライン	保守性･移植性を向上させるための項目

図 2 統計データ例 (Cyclomatic数)



4．センター運用による効果

'97年7月から一年間に約160万stepのソースコードを解析し、共同レビューを実施してきた。その活動を通して、約 3600 箇所のソースコードの修正に寄与した。その間に、解析したソフトウェアは制御系のプログラムから、Windows アプリケーションに至るまで多岐にわたる。

特に制御系ソフトウェアの開発に使用されるコンパイラは、市販のコンパイラに比べソースコードのチェック機能が弱いものが多く、解析を行うことによりソースコードを詳細にチェックし、欠陥となる可能性の高い部分を指摘できるメリットは大きい。

また、開発担当者間にレビューの効果が認知されはじめ、積極的な解析の依頼件数も増加してきた。

5．まとめ

今後はレビュー・サイクルをより短くするために、依頼形式から開発担当者が各自で解析を行えるような環境を構築していく予定である。また、継続して警告の出現傾向と修正件数から、随時ガイドラインの改訂を進めていかなければならない。

最後にセンターでの活動を通して､焦点を絞ったレビューの実施により、ソースコード中の欠陥を早期に発見可能となった。また､共同レビュー、ガイドライン教育の実施によりプログラミング記述の曖昧さを減らし、開発担当者とのコミュニケーションを通じて、当社のプログラミング技術向上に寄与することができた。

参考文献

[1]Thomas Liedtke他：On the Benefits of Reinforcing Code Inspection Activities Process Improvements in the Telecomms Environment, EuroSTAR'95,1995

[2]小笠原他："プログラム静的解析の適用事例と効果分析",ソフトウェア・シンポジウム'98,1998.

[i] 英国PRL社の開発した静的解析ツール。日本では(株)東陽テクニカが販売している。