

ソフトウェア保守の健全性診断 あなたの保守プロジェクトは大丈夫ですか？

ソフトウェア品質シンポジウム2014(SQiP2014) ポスターセッション

小田 雅一 ITホールディングス(株)
野中 誠 東洋大学経営学部

SQiPの研究開発活動の一環として、ソフトウェア保守の品質管理に関わる課題と対策について2012年度より延べ11社のメンバーが集まって議論を重ねてきた成果です。

活動開始の原点は？

保守プロジェクトでの障害が絶えない

現行システムの延命などで無理な改修を重ね、
システムが複雑化して品質低下を招いている現状がある

保守の障害は新規開発よりもインパクトが大きい

しかし、保守では品質の定量的な見極めが難しい

新規開発の場合

- ・100行毎に1個のバグが作り込まれとすれば
- ・1万行なら100個のバグがあると推察できる

保守(改修)の場合

- ・100行修正しても他に全く影響しなかったり
- ・1行の修正が100箇所に影響したりする



我々の疑問：新規開発と同じ指標で保守も品質評価できるのか？

本活動は、この疑問からスタートした

何をしたいのか？

活動目的

保守プロジェクトの特性を踏まえた品質確保、
そのための管理の方法について、
企業を越えて知見を共有し、成果を産業界に公開する

活動内容

- ・保守において品質を低下させる要因は何なのか、
 - ・どうすればその要因を排除できるのか、
 - ・保守における品質状況をどのように管理すればよいのか
- について深掘りし、品質見極めの方法を考える



2013年4月、目的に賛同する会社が集まり活動を開始

参加企業
(50音順)

ITホールディングス株式会社

株式会社アグレックス

株式会社インテック

AJS株式会社

MRTコンサルティング

スミセイ情報システム株式会社

東洋大学

TIS株式会社

日本ユニシス株式会社

富士通株式会社

三菱電機インフォメーションシステムズ株式会社

USOL北海道株式会社

どんな活動をやってきたのか？

保守の作業プロセスの分析から着手

「影響範囲見極め」および「テスト項目抽出」が最も品質に影響を与える要因だと結論付けこれらの作業品質を定量的に測る方法を模索



保守には「しがらみ」がある？

新規開発と違い、保守では過去の経緯なども大きく影響する

ドキュメント化されていない、知る人がもう居ないなどの要因が考慮漏れとなる
(例) ・なぜこんなロジックが組み込まれているのか意味不明
・今でもこのパスを通る可能性があるのかどうか判断できない

環境品質の重要性を再認識

「過去の経緯」に加え、「改修要件の正しい理解」や「リリース判定」なども重要な要因と認識、**環境品質**のチェックリスト作成に着手

保守品質関連図

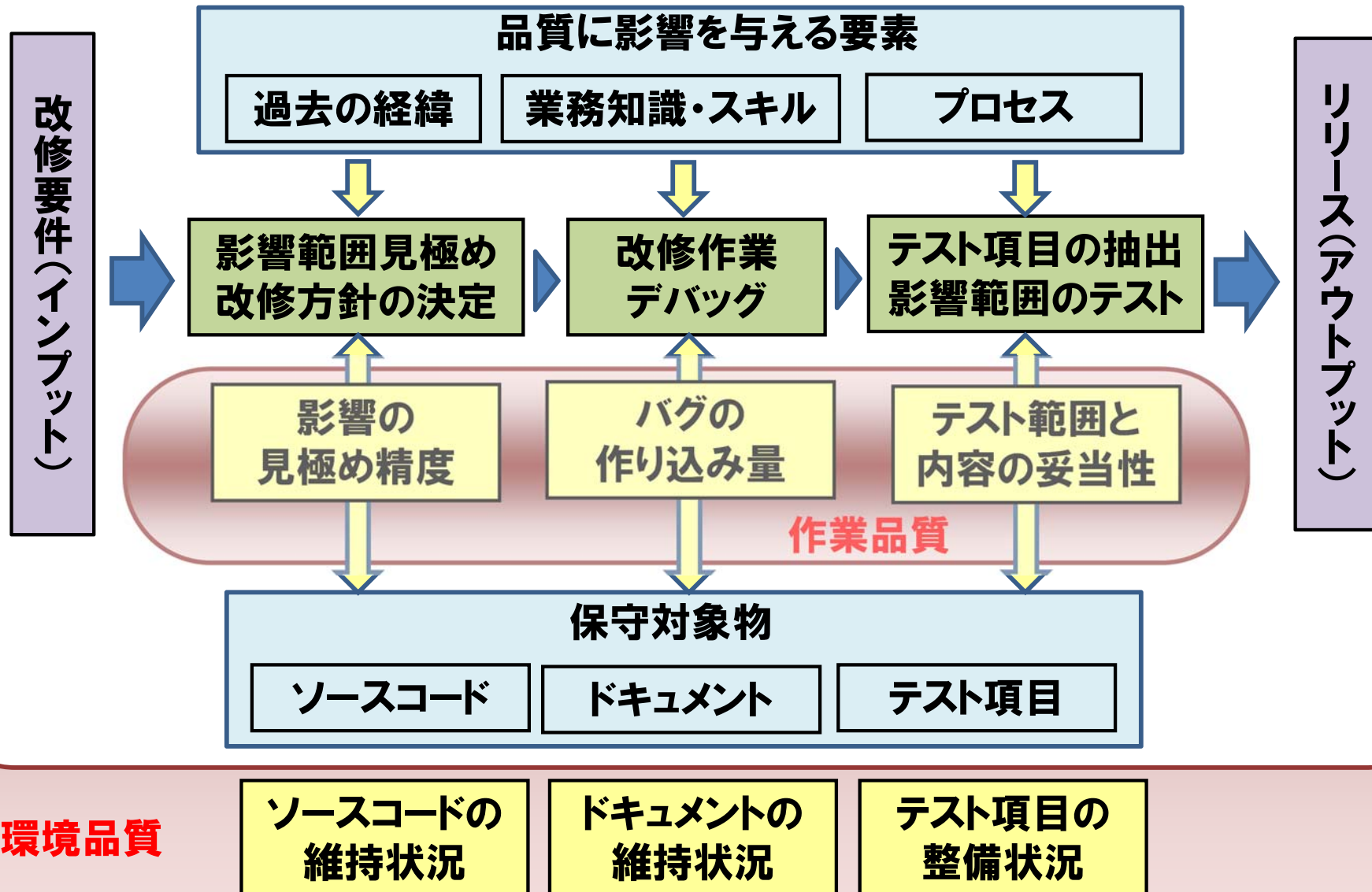
この部分の評価をチェックリスト化した

要件確認プロセスの
整備・活用状況

過去経緯の
整備・活用状況

知識・スキルの
保有状況

出荷プロセスの
整備・活用状況



チェックリストの目的

- ・ソフトウェア保守の環境品質を測ることで弱い箇所を把握し、改善することでソフトウェア保守の品質を向上させる

チェックリスト作成の観点

- ・「保守品質関連図」で示した7つの環境品質要素に着目し、その7つの要素を網羅的にチェックできるように作成した

チェックリストの特徴

- ・全体は「チェック項目・解説」「チェック結果」「事例」で構成
- ・チェック項目は9項目で構成し、各項目5段階での状況チェック
→保守品質関連図の環境品質7要素を9項目でチェックする
- ・チェック項目の意図を理解してもらうため、項目毎に解説を記載
- ・チェック結果はレーダーチャートで表示し、強み・弱みを分かりやすく表示
→基準値(目標値)と評価値(チェック結果)の比較も可能
- ・各プロジェクトで改善の際に参考としてもらうべく、項目毎に対応事例を提示

チェック項目の観点

(1) 影響分析対象物の品質

- ①ソース解析の実施しやすさ
- ②プログラム内の不要ロジックの判別
- ③影響範囲分析に活用するドキュメントの有無及び状態

(2) 影響分析やテスト項目抽出のし易さ

- ④過去の改修記録の有無
- ⑤ソース上の関連が無い場合の影響範囲調査
- ⑥プロジェクトメンバーのスキル
- ⑦ソースの修正箇所と対応したテスト項目の明確化

(3) プロセスの整備と活用状況

- ⑧改修要件の確認プロセスの明確化
- ⑨レビュープロセスの明確化

※チェックリストのサンプルを配布しています。
是非お手に取ってご覧下さい。



チェックリストの活用イメージ

チェックシート
の活用方法の
決定



プロジェクトで
チェックを実施



改善計画立案
改善実施

想定している活用方法は以下の2通り

・組織全体(トップダウン)での活用

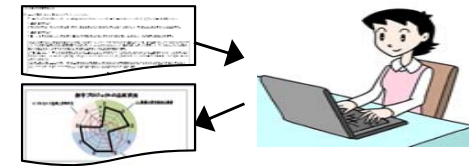
最初に組織全体で目標となるレベル(基準値)を決める。

各プロジェクトでチェック後、目標レベルに達するための改善を実施する。

・プロジェクト単体(ボトムアップ)での活用

各プロジェクトにてチェック結果を元に課題を洗い出し、それぞれで改善を実施する。

チェックリストに基づき、
プロジェクトでチェックを実施



評価が低い項目については、付属している事例なども参考にしながら、改善計画を立案し、改善を実施する。

改善実施にあたっては、必ずしもレベル5を目指す必要はない。
プロジェクト特性、費用対効果、優先順位などを鑑み、自分たちがまず
「いつまでにどのレベルまで達するのか」を検討した上で改善計画を立案し、実施する必要がある。

「チェック項目・解説」 サンプル

(1) 影響分析対象物の品質

チェック項目

① ソースは解析しやすい(見やすい)ようになっていますか？

- ☐ コーディング形態が揃っている、コメントも分かりやすく書かれている、モジュール化もできているなど、新規メンバーでも理解しやすい
- ☐ 上と下との間くらい
- ☐ 部分的にでもコーディング形態は揃っており、最低限のコメントも記載されているなど、新規メンバーでもある程度説明を受ければ理解できる
- ☒ 上と下との間くらい
- ☐ コーディング形態はバラバラでSE依存、コメントも少ないあるいは記述が分かりにくいなど、新規メンバーが理解するには時間を要する

(解説) 言うまでもなく、影響分析を漏れなく的確に行うためには、ソースコードが解析しやすく整備されていることが重要です。ソース検索で引っ掛かって、その箇所が今回の改修の影響範囲だと判断するにはコメントが分かりやすいこと、最新情報であることも重要です。常に前後のロジックを解析しないと影響範囲かどうか見当が付きにくいようでは非効率であり、判断ミスにもつながります。

プロセスとしては、コーディング規約が整備されていることはもちろん、それがメンバーに周知され、順守されていることが見やすいソースを維持管理する基本となります。コメントの書き方も規約に明記されていればなお良いでしょう。順守度を高めるには、QAグループなど第三者により定期的にチェックすることも有効です。

チェックに際しては、ここに挙げた一通りのことが出来ていれば最高、全く出来ていなければ最低というレベル感で自己評価してください。目安として、新しくプロジェクトにきたメンバーが理解し易い状態かどうかを考えてもらおうと評価しやすいと思います。

解説

「チェック結果」 サンプル

レーダチャートにてチェック結果を表示

【基準値】
目標値があれば
それをここに記載

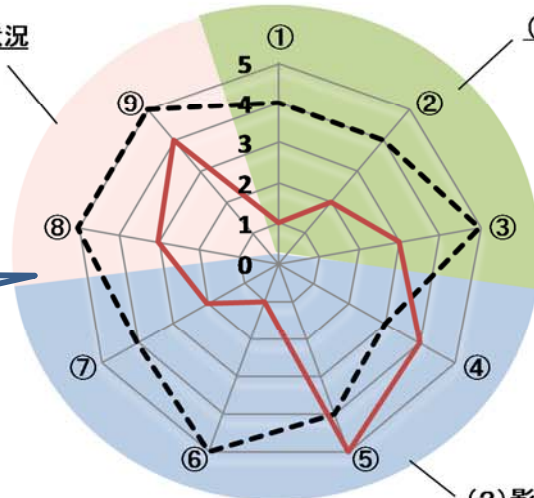
基準値と評価値
の比較も可能

【評価値】
チェック結果を
自動表示

保守プロジェクトの品質状況

(3) プロセスの整備と活用状況

(1) 影響分析対象物の品質



【凡例】

--- 基準値
— 評価値

チェック分類	チェック項目	基準値	評価値
(1) 影響分析対象物の品質	① ソースは解析しやすい(見やすい)ようになっていますか？	4	1
	② プログラム内の使われないロジックが判別できるようになっていますか？	4	2
	③ 影響範囲分析に活用するドキュメント、その存在と状態はどのようになっていますか？	5	3
(2) 影響分析やテスト項目抽出のし易さ	④ 過去の改修について、依頼内容から最終的な改修内容までの記録が参照できますか？	3	4
	⑤ ソース上の関連が無い影響範囲についても、業務フローや関連する法規制などから洗い出せますか？	4	5
	⑥ プロジェクトメンバーは必要なスキル(業務知識、現行の理解、技術など)を有していますか？	5	1
	⑦ ソースの修正部分に対応して実施すべきテスト項目が抽出できるようになっていますか？	4	2
(3) プロセスの整備と活用状況	⑧ 改修要件の確認プロセスが明確で、順守できていますか？また有効に機能していますか？	5	3
	⑨ レビュープロセスが明確で、順守できていますか？また、有効に機能していますか？	5	4

「事例」 サンプル

項目：①ソースは解析しやすい(見やすい)ようになっていますか？

評価コメント

レベル5になったプロジェクトは、保守を開始して間が無いが、そうでなければよほど管理が行き届いている、あるいは複雑な改修があまり無いものと推察されます。

メンバーの入れ替わりや改修を繰り返すうちにルールが徹底されなくなり、分かりやすいコメントを付ける余裕も無くなってしまうのが普通で、レベル3くらいを何とか保っているというケースが多いのではないのでしょうか。

熟練メンバーは慣れていて問題無いかも知れませんが、新規メンバーを受け入れた時に大きな障害を発生しがちです。レベル1に近づかないよう、定期的にルールの見直しを行いましょう。

事例1： この会社では下記のような仕組みを導入し、ソースの解析し易さを維持している

データ項目名称とそれを構成する名詞要素が登録されており、未登録データ項目名称の使用や命名規則を守らない名称登録が機械チェックされるようになっている。このチェックにより、名称サーチによる使用箇所特定の精度が向上する。

プログラム構造のひな型を数種類に限定し、スケルトン(骨格コード)に基づいたコーディングを義務づけており、これにより可読性と解析性が向上する。

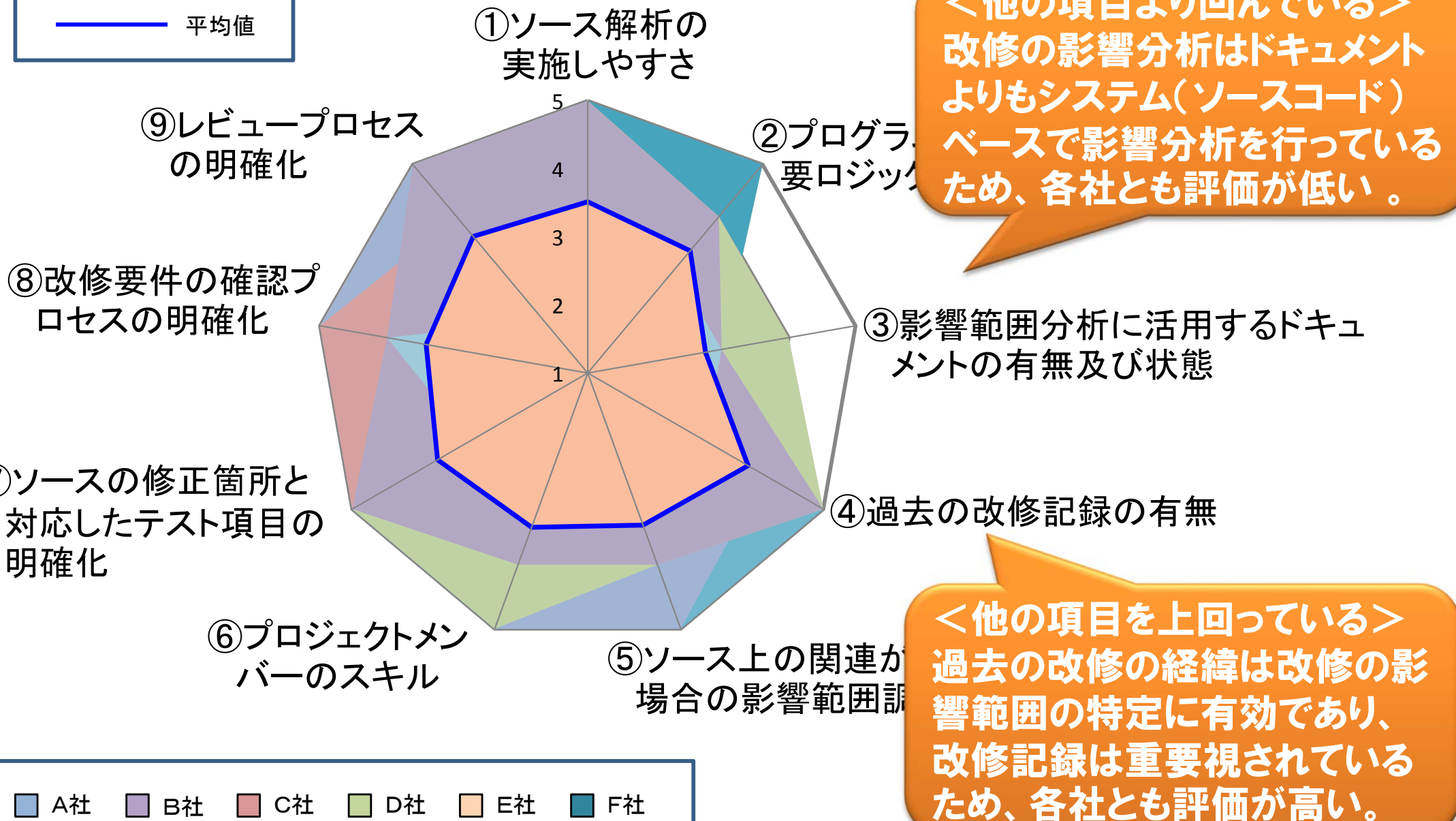
プログラム仕様書にモジュール構造図を漏れなく添付している。これにより可読性と解析性が向上する。

また、特定のシステムではあるが、SEが作成するジョブ説明書とプログラミング構造を分離。SEは日本語で仕様を記述し、内部構造はプログラマが独自の内部設計ドキュメントを作成して管理することで、ドキュメントの可読性を向上している。

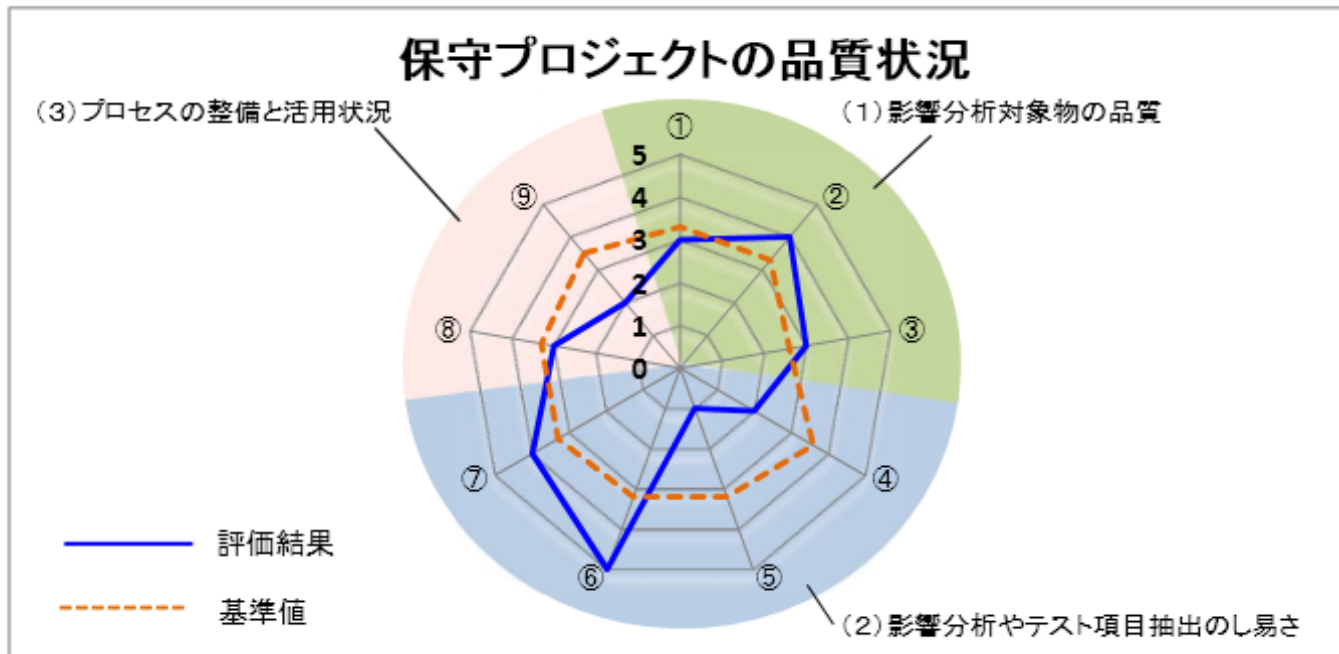
事例

参加企業の適用結果

— 平均値



適用事例



【参加企業のプロジェクトへの適用事例】

例に挙げた保守プロジェクトで、改善が必要な項目を以下に挙げる。費用対効果を考えて、対応を検討する。

- ① ソースの解析(見やすさ)は若干の改善が必要。
- ④ 過去の改修記録の維持・参照は大幅な改善が必要。
- ⑤ 業務や法規制からの観点での影響範囲の特定は大幅な改善が必要。
- ⑧ 改修要件の確認の明確化について若干の改善が必要。
- ⑨ 改修手順(レビュー)の明確化について改善が必要。

「保守プロジェクトの品質状況」の使い方

- (1) 組織(自社や部門など)の実力値を把握する(基準値)
- (2) プロジェクト毎に評価し、組織のレベルに至っていない項目に着目する
- (3) 組織レベルに至っていない項目について、費用対効果を考えて、適切な対応(対策)を検討する

チェックリストの試行結果から考察できること

合計点が高いプロジェクトには以下の傾向がある

ソースが見やすく、かつ分析に活用するドキュメントが維持・整備されている。

⇒基本的な成果物の維持管理ができているプロジェクトは影響分析、テストも適正に行えていると言える。

プロジェクトプロファイルとの比較では以下の傾向がある

・「重要度高」「月当たりの改修工数大」の場合、ソース上の関連が無い箇所も網羅的に影響分析ができている。

⇒高い意識で継続的に保守を行うことで、業務・システムの理解が深まることが考えられる。

・「分析対象物の品質」があまり良くない状態でも、保守時の影響分析がそれなりにできているプロジェクトがある。

⇒プロジェクト全体平均と比べ、システムの規模に対して保守要員が多い傾向にある(人海戦術による対応)。



振り返りと今後の展望

今回の活動の振り返り

チェックリスト試行時に寄せられたコメントには、
「自プロジェクトを客観的に見ることができた」
「改善のきっかけになる」
といったものが多かった。



環境品質を見える化するチェックリストが、意外と存在しておらず、今回の活動が広く保守プロジェクトに対して有効なものではないかと感じた。

今後の展望

- ・設問の補足説明や事例が有効というコメントが多かった。
今後、さらなる事例の充実化を図り、保守プロジェクトへ外部事例として提供することが必要である。
- ・「環境品質」に加え、「作業品質」の評価指標を検討し、保守活動のレベルアップに寄与する活動にチャレンジしたい。