トップエスイー修了制作

ソフトウェアの老化を放置させないための ヘルスチェック

富士通株式会社　東　勲　higashi.isao@jp.fujitsu.com

開発における問題点

システムは、繰り返し修正することにより、複雑化し保守性が低下する（ソフトウェアの老化）。保守性を向上させる方法は、既存の研究成果で提案されているものの、機能追加が優先される運用現場ではいつ対処を実施するかの判断が困難であり、最終的に再構築が必要になるまで放置されることがある。

手法・ツールの提案による解決

システムの老化傾向を把握し、老化に対する適切な対処方法と実施時期を明確にするために、システムそのものとシステムを取り巻く環境に対する評価指標を作成し、その推移を確認することで放置を防ぐソフトウェアのヘルスチェックを提案する。

運用システムの状態を把握する指標の作成

①老化を進める要因の洗い出し

- C1 システムに対する理解不足
- C2 ソースコードの品質不足
- C3 時間の経過による環境の変化
- C4 機能追加・改修の頻度

②評価指標の作成

- D1 システム理解度合
  - システム内の把握度合により評価
- D2 変動性の受入度合
  - 機能追加時や初期要件からの変更に対する柔軟性をもたせているかで評価
- D3 サポート度合
  - システムが利用しているOS 、言語、フレームワークのサポート状況から評価
- D4 明瞭度合
  - ソース複雑度の逆を表す指標として定義
- D5 システムの安定度合
  - バグの割合や、機能追加要望数により評価

③各指標の基準（Level1～5）を決定

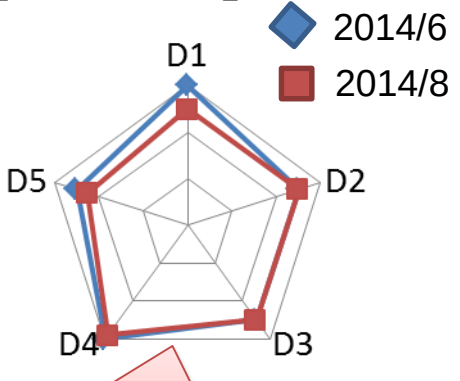
システム理解度合（一例）

L5 41~50	30分以内に機能から、対応ソース・ライブラリを特定可能
L4 31~40	一日以内に機能から、対応ソース・ライブラリを特定可能
L3 21~30	2, 3日かかるが、ソース・仕様書などにより把握可能
L2 11~20	ソースはあるが，仕様書の欠如などにより一部把握不可能
L1 1~10	一部ソースが欠損しており把握不可能

実証実験

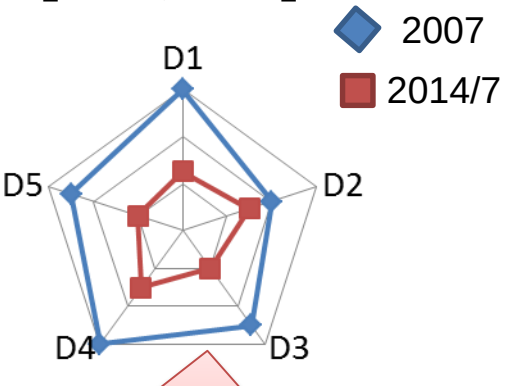
社内開発者向けの2つの運用システムに適用

【システムS】



2か月だが組織変更などにより変化している。低くなった部分を対処すればよい

【システムM】



放置され続けた結果、全体的に低く、どこから手を付けるかの判断が難しい

効果と今後の展開

効果

- 修正が行われていない場合でも、組織の体制変更などにより運用システムとして状態が悪化していることを確認
- 悪化しやすい部分を一目で把握でき、対処の優先順位ならびに適切な対処の判断が容易になった

今後

- 診断結果と生産性の関連を継続的に蓄積し、老化によるコスト増の関係をモデル化可能
- 工数見積りの精度向上へと期待できる