

アット スコー プ
AtScope活用シーン 2
リファクタリングスコア

2016年11月10日

ビースラッシュ株式会社



構成要素と設計構造をスコア化



- 100点満点で設計の改善箇所を減点

特徴

- 設計改善に直結し、ソフトウェア資産価値が高まる
 - － 「要素点」が低い場合は、リファクタリング、が効果的
 - － 「構造点」が低い場合は、リバーズ設計、が効果的
- 組織設計力の把握ができる
 - － 徐々にスコアが下がる組織は、設計力は低い
 - － 設計図でピアレビューできる組織作り

■ 算定方法

- 正常値を満たさないものを100点から減算
- 要素点と構造点の算定項目は下のとおり

要素点

項目	正常値
ファイル長さ	1000ステップ未満
ファイル内の関数の数	20未満
関数の長さ	30ステップ未満
フィールド変数の数 (※)	7個以内

構造点

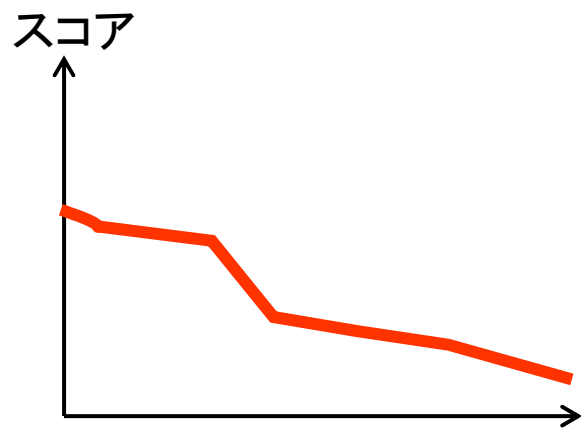
項目	正常値
相互依存	0
ファンアウト数	10個未満 (関数コール)
データスコープ (※)	0 (ファイル外アクセス)

(※)は、解析設定で「変数アクセスを含める」を選択した時のみ算出

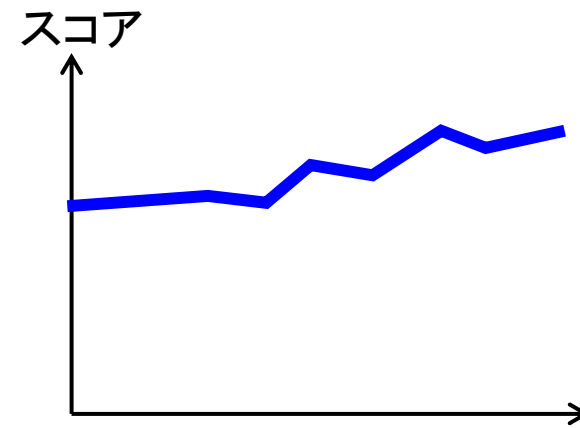
スコアの使い方①: 資産価値の向上



- 相対的なスコアの推移を定点観測
 - － 徐々にスコアが上がる仕掛けを作る
 - 設計図でのピアレビュー、など



アドホックな変更の積み重ね



変更時にリファクタリング
や構造設計を実施

スコアの使い方②: 設計力の向上



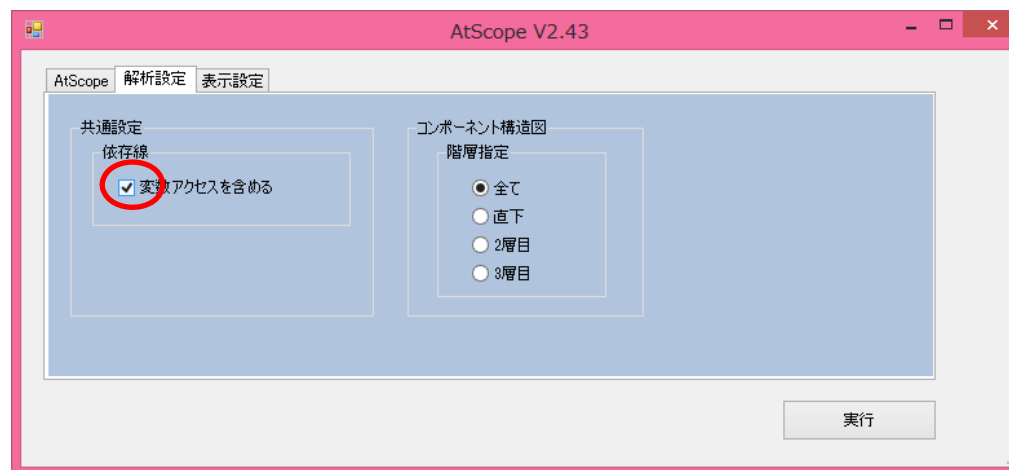
- 要素点と構造点から、設計力の強化施策を練る

要素点 \ 構造点	高い	低い
	高い	低い
高い	ピアレビュー (さらなる設計力向上)	リバーズ設計
低い	リファクタリング	設計技法の見直し 構造化設計の学習

変数アクセスを含むスコアの出し方



- グローバル変数がある場合は、解析設定で「変数アクセスを含む」をチェック



スコアの使い方③: 戦略的な設計改善



■ 戦略的な設計構造改善

- トップ5リストから実施する項目を決めて、設計改善する
- 改善しなくても良い項目も決める

＜＜スコア＞＞
総合：94点

要素：99点
構造：89点

＜＜要素の警告リスト＞＞

/* 関数の長さ */
44 : Lcd Ic_init()

要素点

＜＜構造の警告リスト＞＞

/* ファンアウト数 */
17 : InspectionProcess.c
15 : DiagnosisStep.c
14 : EepRom.c
14 : Lcd.c
12 : Inspector.c

/* 循環依存 */
2 : InspectionProcess 8 : CurrentMeter
3 : InspectionProcess 3 : DiagnosisStep
1 : Receptionist 3 : DiagnosisStep
2 : UserPerson 1 : Receptionist

構造点

＜＜日付＞＞
2015年06月09日 14時03分

計測対象:
C:\01_C_sample_code\01b_TaiSosei

リファクタリングスコアのおおよその目安



一人で担当する範囲で計測すると、次のような目安になります。

- 絶対値はあくまで参考としてお使いください
- 相対的な使い方を推奨します

スコア	評価	コメント
80点～100点	優	他の人でも変更可能なソフトウェア
60点～79点	良	引継ぎ説明が少しで済むソフトウェア
0点～59点	可	引継ぎ期間が必要なソフトウェア
マイナス点	不可	熟知していないと修正することが大変なソフトウェア

局所的な修正を繰り返しているとスコアは徐々に低下していきます。マネージャは、スコアの推移を管理して、スコアを徐々に上げていくための体制や仕組みを作ること、ソースコード劣化を未然防止することができます。

また、受発注における納品条件(検収条件)の目安として使うこともできます。