

SQIPシンポジウム2009

ソースコードの品質を 直接、詳細に測る方法

2009年9月11日

ヤマハ(株) PA・DMI事業部 DMIソフト開発G

小池 利和



アジェンダ

- メトリクスについて想うこと
- ソースコード品質評価ツールAdquaのご紹介
- ヤマハ(株)での測定分析事例

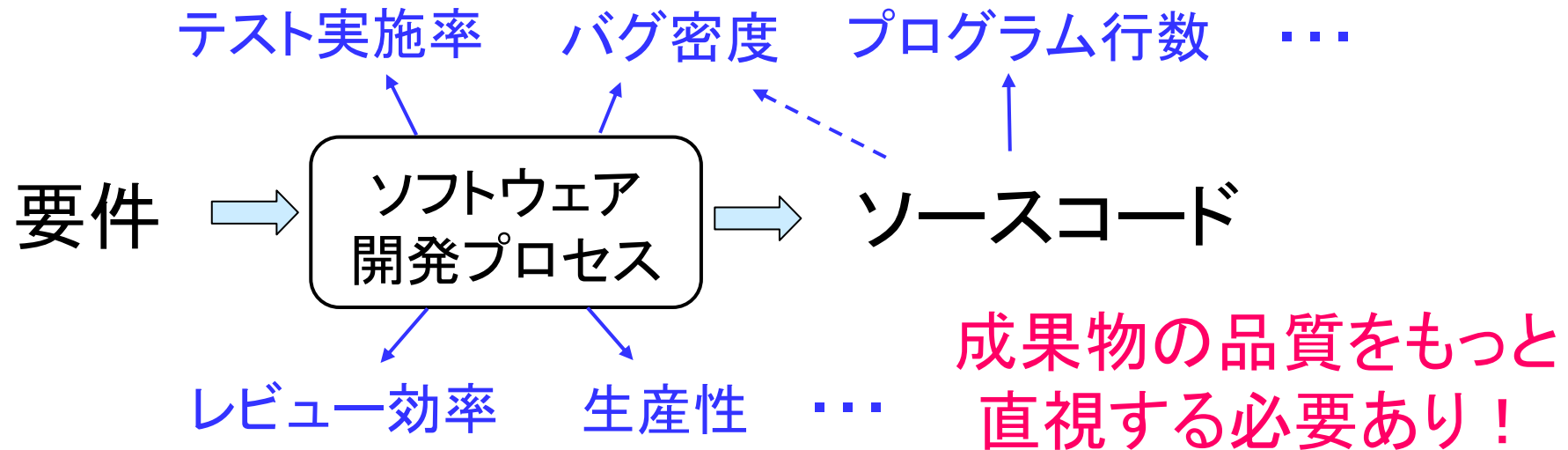
自己紹介

- 現在は、電子楽器のソフトウェア（ファーム、PCアプリ）を開発している部門のSEPGです。
- 昨年までは、品質保証部門に所属し、全社スタッフとしてのSEPGでした。周りのメンバーはいわゆる品質管理屋ばかりで、私だけ浮いた存在でした。
- 10年ちょっと前までは、スポーツ事業部というところで、テニスシューズの開発をしていました。
(もちろんソフト開発とは全く無縁の職場です)

メトリクスについて想うこと

仕事の品質と成果物の品質

様々なメトリクスが測定されているが...



その多くは仕事ないし、仕事の結果を測っている。

ソフトウェア開発の最終成果物とも言えるソースコードについて直接測定しているものは意外と少ない。

仕事と成果物の品質の相関は？



ソースコードは開発のアウトプットだが、派生開発においては次の開発のインプットという見方も出来る。

仕事の品質がソースコードの品質を主導しているように考えがちだが

その逆も考えられ、つまり車の両輪となっている。

もし仕事と成果物の品質に相関関係が無いとしたら
何のためにプロセス改善をしているのか？

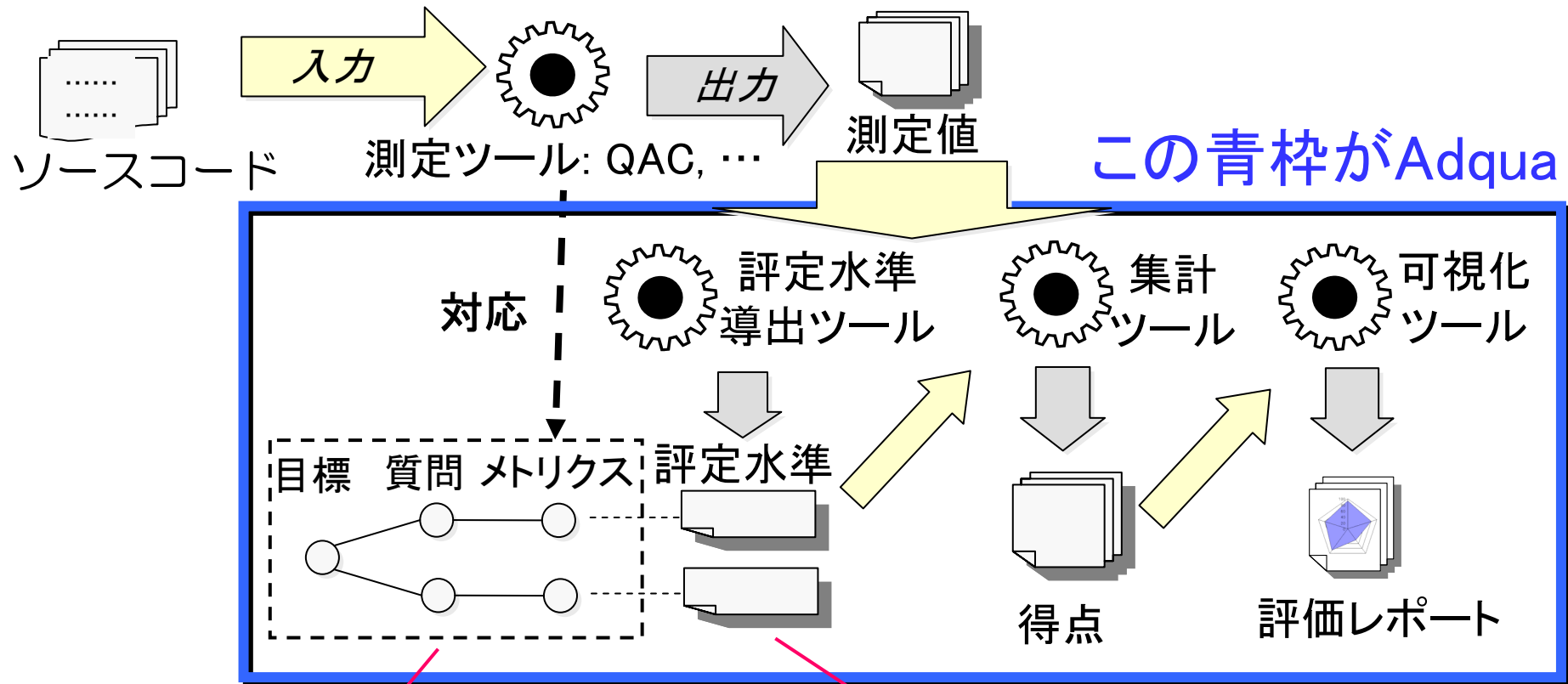
ソースコード品質評価ツール のご紹介

ソースコード品質評価ツールの概要

- オージス総研、早稲田大学、ヤマハの共同研究によって開発されたツール
- 正式名称はAdqua(アドクア)
オージス総研から条件付で無償提供
- ソースコードを解析し、ISO9126に準拠した品質特性の観点で得点を出力

後にポイントを絞って詳細を解説します
まずは実際の評価結果をご覧ください

ツールのメカニズム

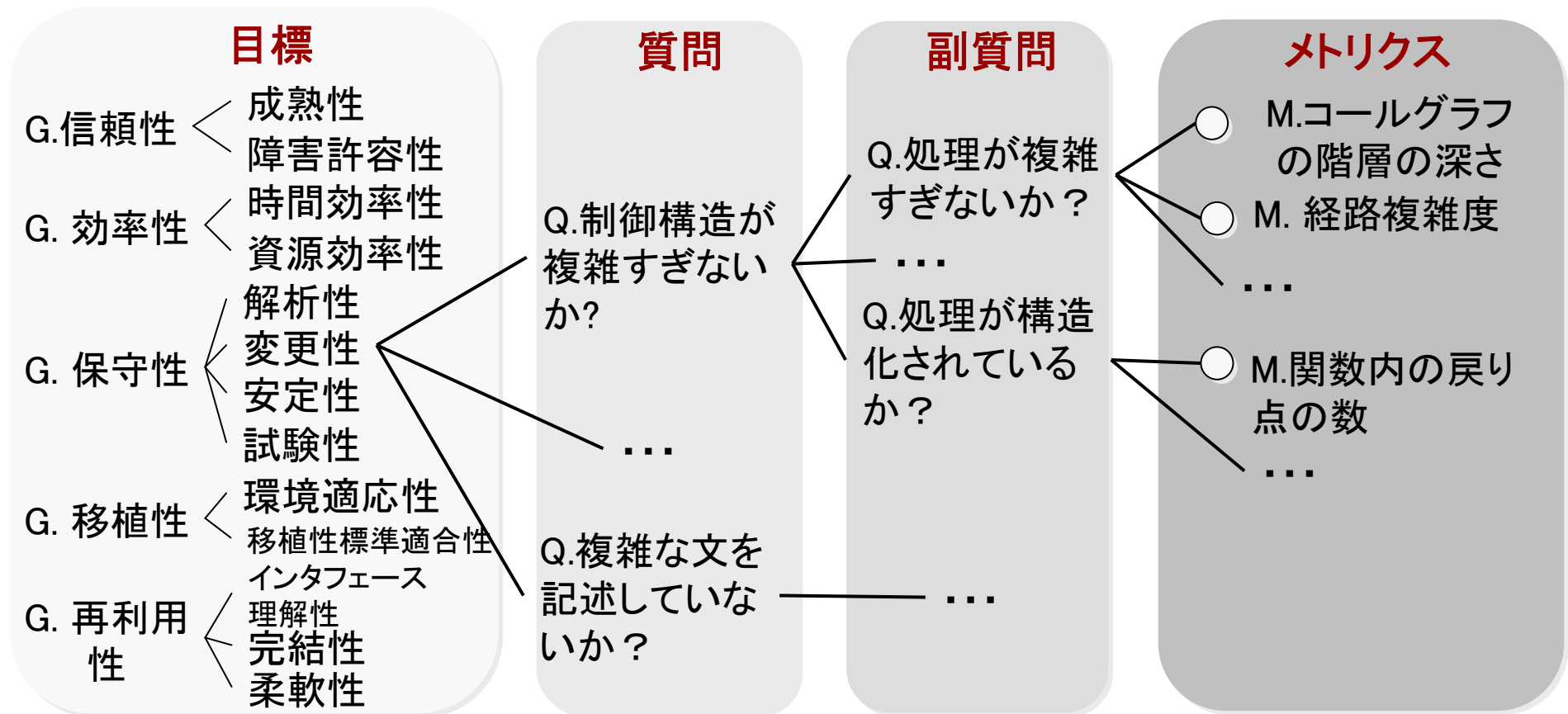


品質スイート
(このツールの肝)

各種メトリクスを0点～100点に
正規化する仕組み(これは秘伝)

品質スイート

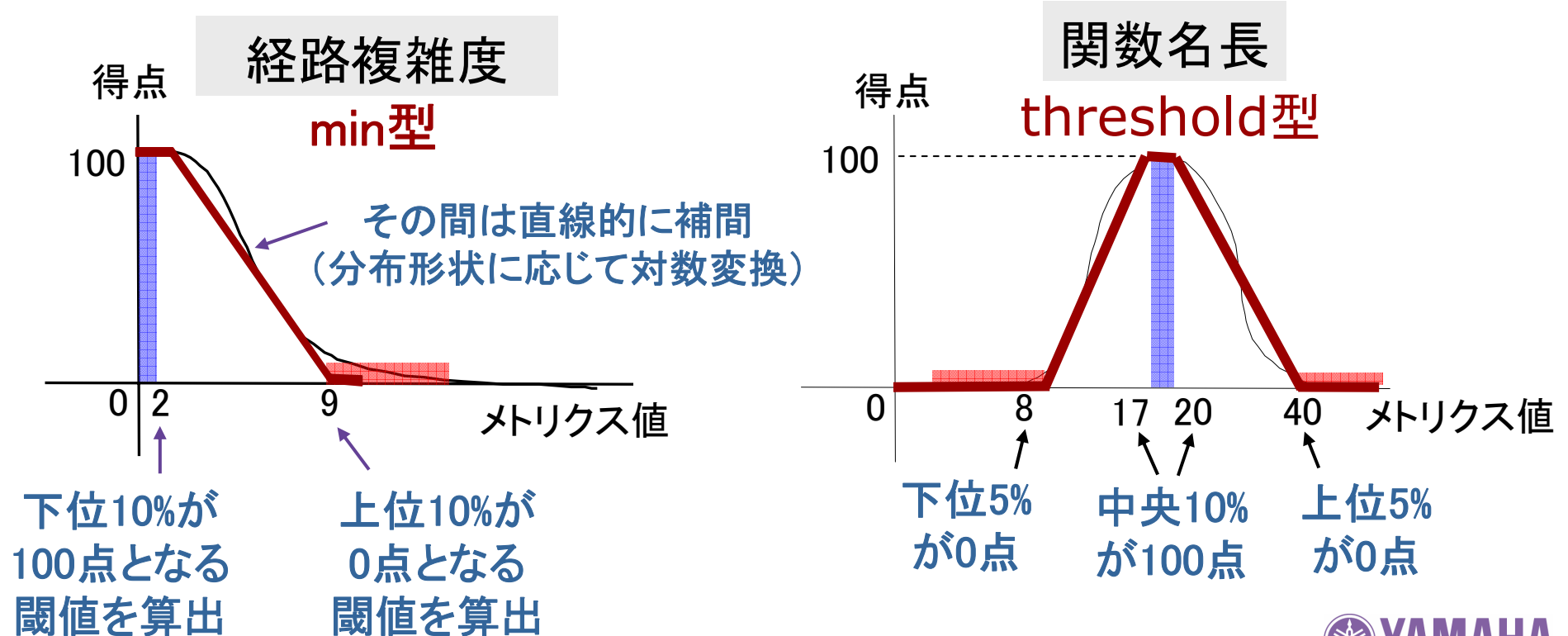
- ISO9126ベースの網羅的品質モデル(再利用性のみ独自追加)
- 43の質問、123の副質問、348(C言語)・433(C++)のメトリクス



評定水準

- C/C++の合計145プロジェクトより測定値分布取得
- 技術的観点と分布形状により8タイプに得点プロファイル型分類
- 実データの分布に基き、統計的に閾値を算出

評定水準の導出例



Adquaの長所

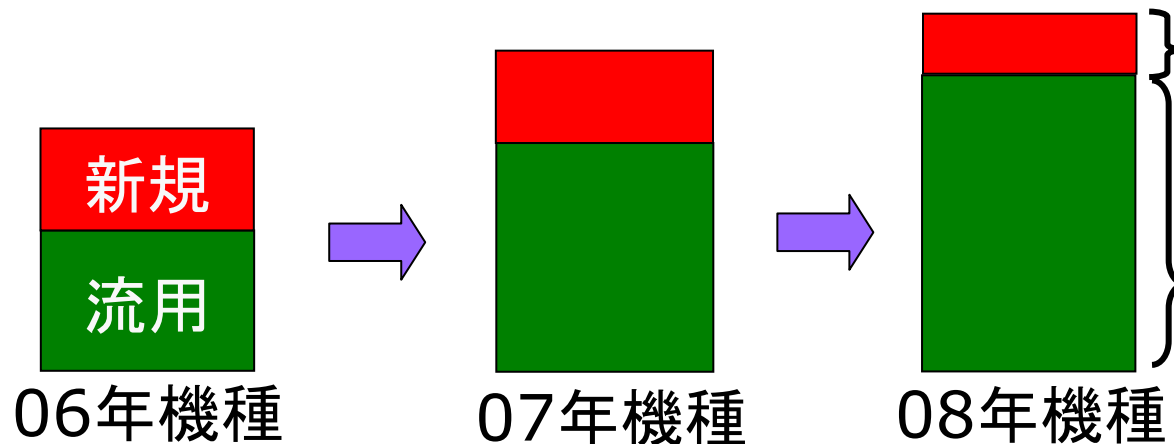
- ソースコードさえ有れば、自動で評価が可能
- システム⇔モジュール⇔ファイル⇔関数、または、品質特性⇔副特性⇔メトリクス、といった具合に目的に応じてマクロ⇔ミクロ自在に品質を確認できる
- 全てのメトリクスが0～100点に正規化されることで、評価結果が直感的で分かり易い
- 大量の実データによって評点水準を導出することで、根拠の無い主観的な評価を排除
- 多種多様なメトリクスを採用していることで、システムの個性などに左右されづらく、ロバストな結果を出力

ヤマハ(株)での 測定分析事例

注ぎ足し開発？？？



- 当該部門の開発は、派生開発がメイン
- ほぼ同じチームで、継続開発



仮に毎年同じ工数で
開発するならば

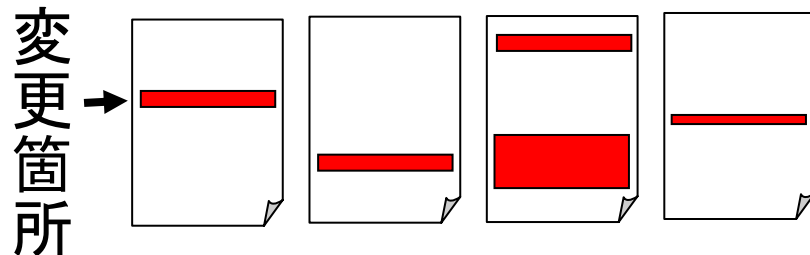
新規開発部分が
目減りしていく！

流用部分が年々
増加していくので

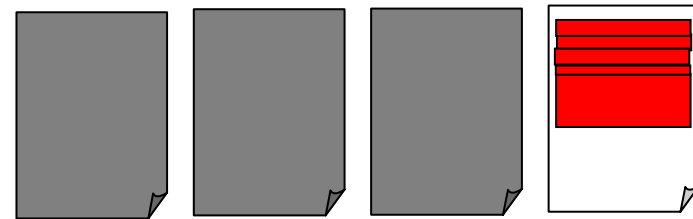
流用部分の品質が開発に多大な影響を与える

保守性、再利用性を何とかしなければ！

保守性が高いとは
変更箇所の特定制→修正→テスト
が効率良く実施出来て、
デグレードしにくい



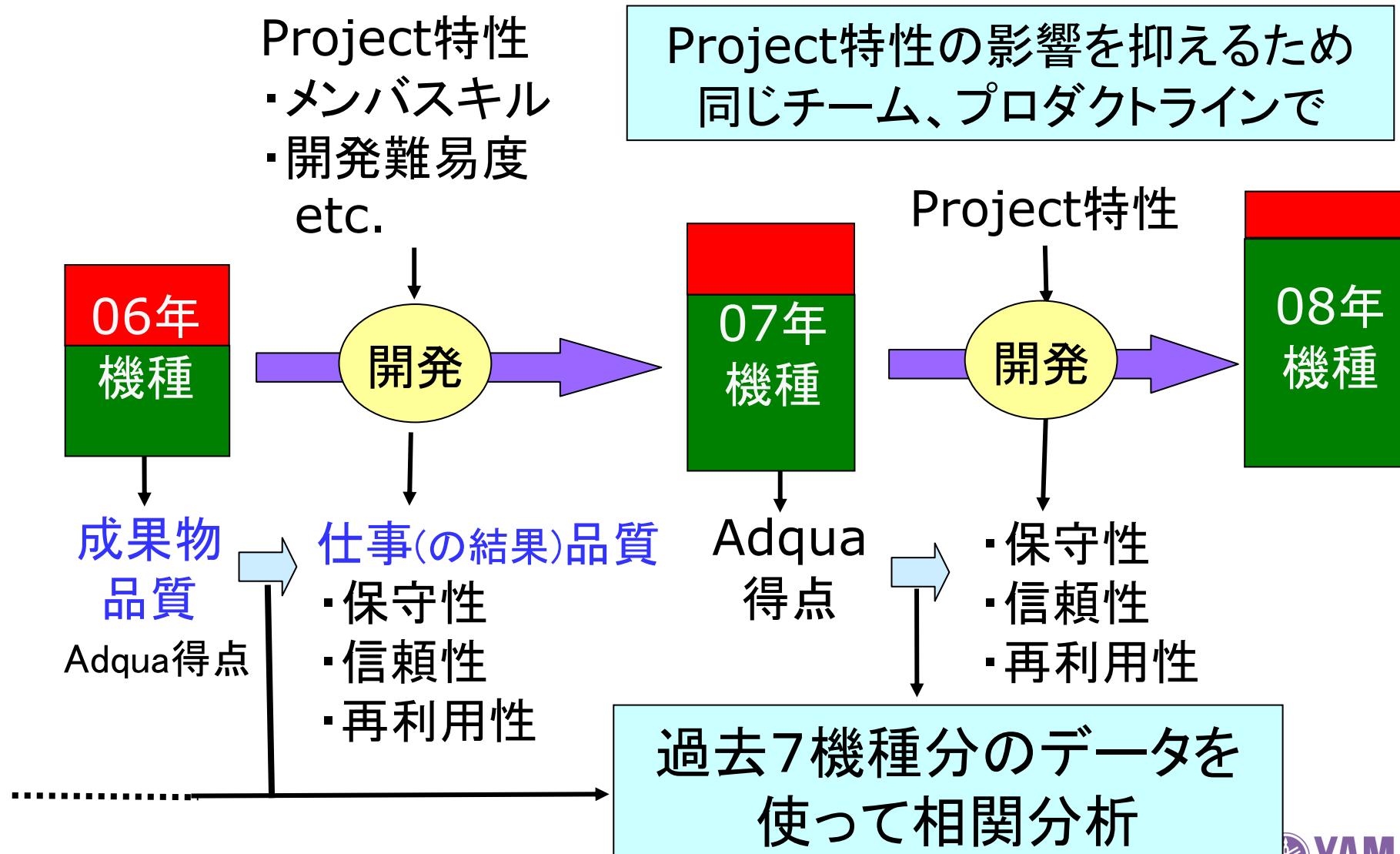
再利用性が高いとは
インターフェースが明確で、
独立性高く、変更箇所が
局所化されている



こんな思いから、縁有ってAdquaを共同開発しました。

しかし、Adquaの得点は開発現場に受け入れられるか・・・

Adqua妥当性検証の方法



保守性の品質指標

$$\text{コーディング \& テスト効率} = \frac{\text{新規開発行数}}{\text{コーディング + テスト工数}}$$

$$\text{バグ密度} = \frac{\text{システムテストバグ件数}}{\text{新規開発行数}}$$

Adquaの保守性の得点と
コーディング & テスト効率は正の相関
バグ密度は負の相関となるはず・・・

保守性の分析結果

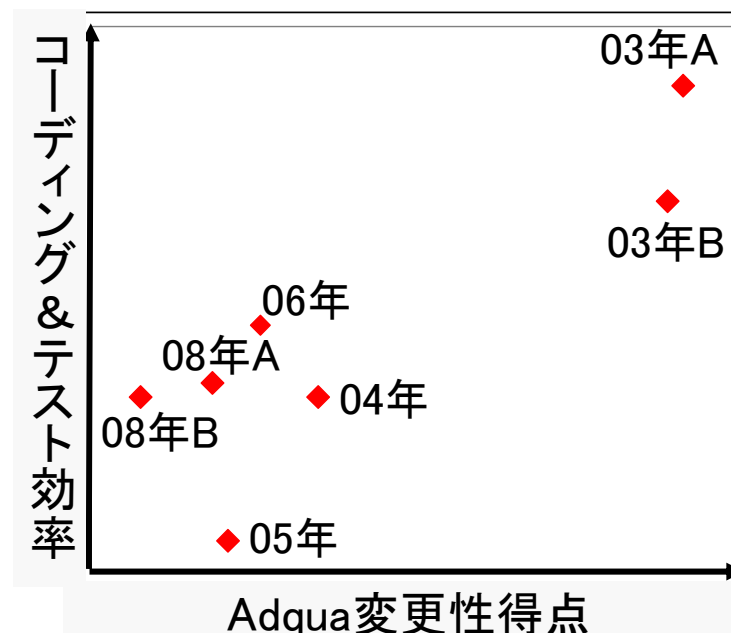
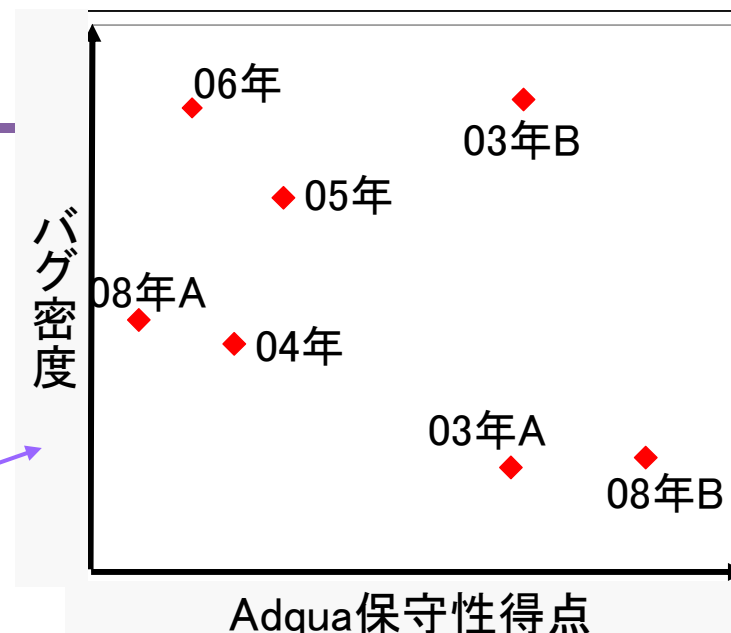
Adqua得点と品質指標の相関係数

	コーディング &テスト効率	バグ密度
保守性	0.43	-0.80
解析性	0.41	-0.82
変更性	0.86	-0.33
安定性	-0.37	-0.43
試験性	0.80	-0.71

網掛け部が、回帰分析で
統計的に有意



ほぼ想定通りの結果！



信頼性の品質指標

- 前述のバグ密度を指標とした
- Adquaの信頼性の得点とバグ密度は負の相関となるはず・・・

信頼性の分析結果

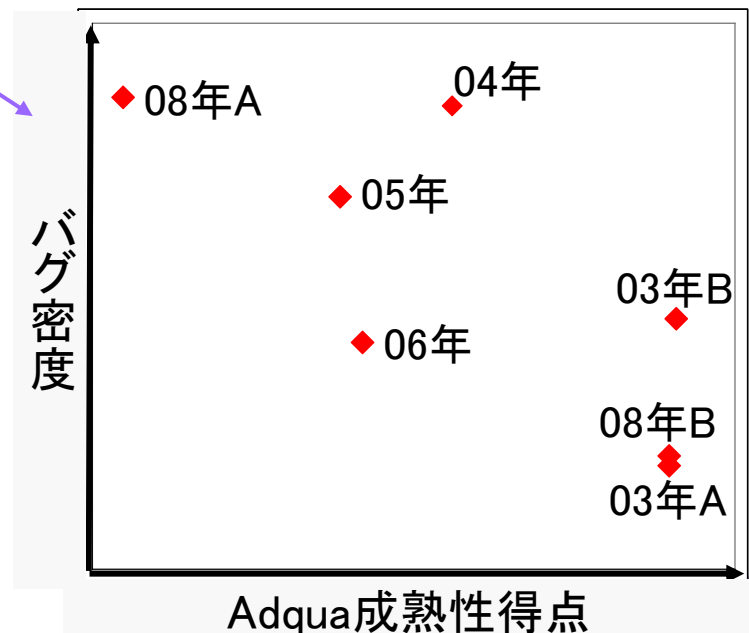
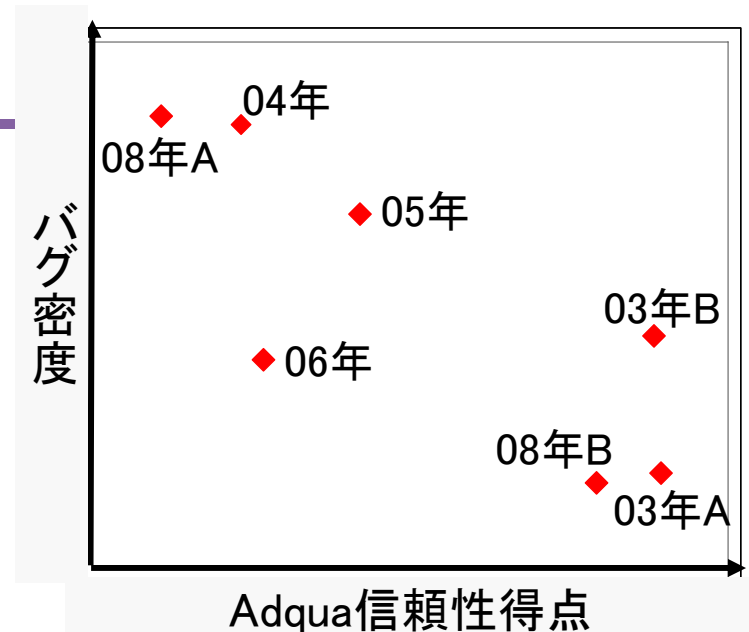
Adqua得点と品質指標の相関係数

	バグ密度
信頼性	-0.81
成熟性	-0.77
障害許容性	-0.34

網掛け部が、回帰分析で
統計的に有意

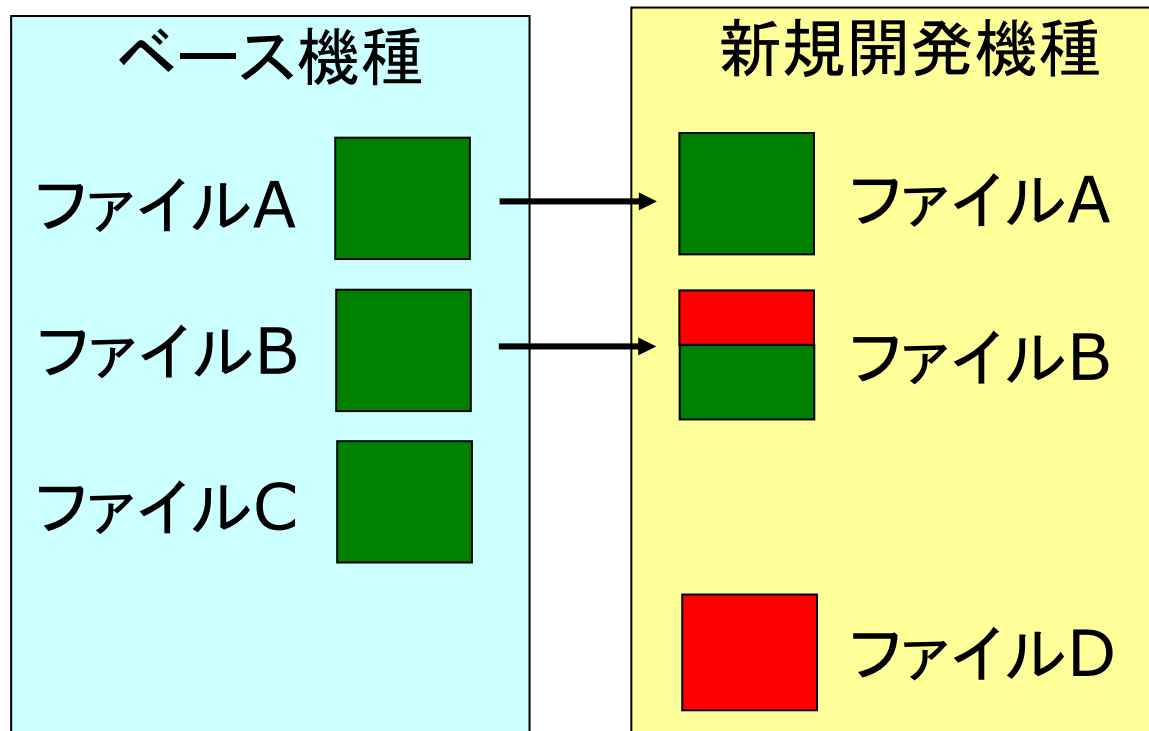


想定通りの結果！



再利用性の品質指標

$$\text{再利用ファイル率} = \frac{\text{無修正で再利用したファイル数}}{\text{流用したファイル数}}$$



流用したAとBのうち
修正無しで、再利用
できたのはAだけ

つまりこの例では
再利用ファイル率が
 $1/2 = 50\%$ となる

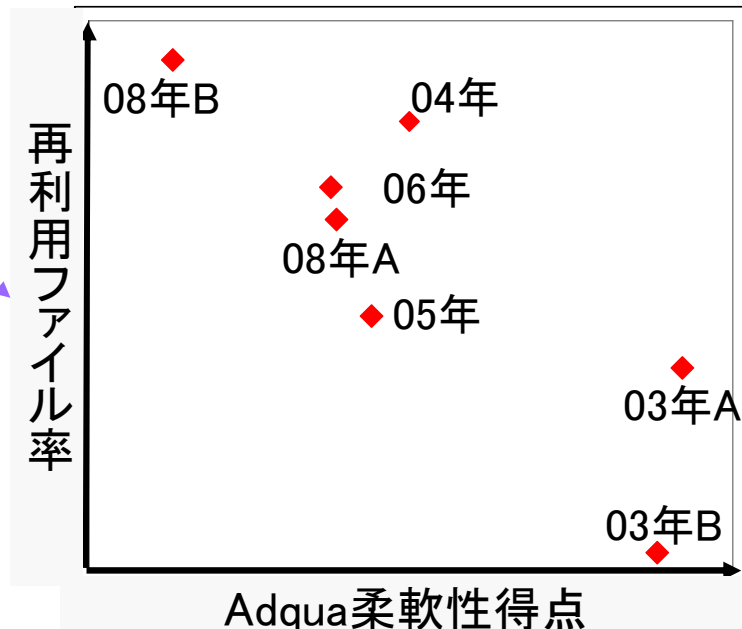
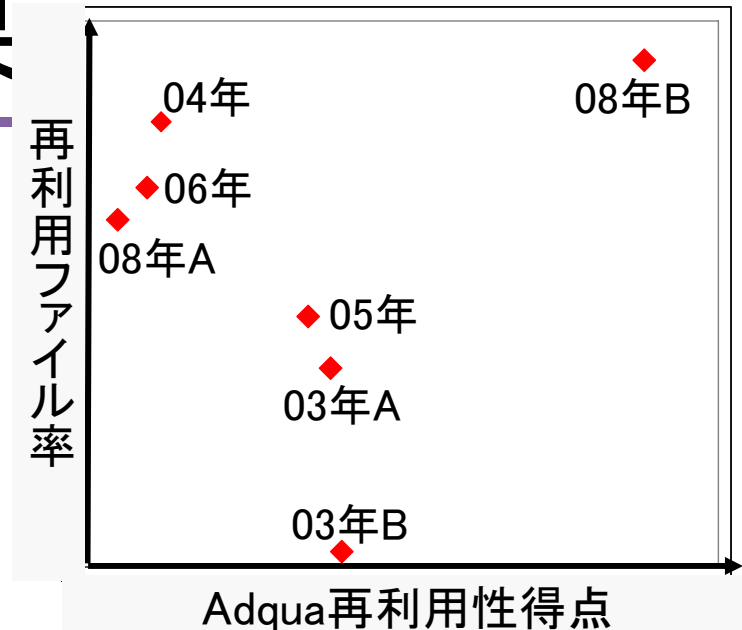
再利用性の分析結果

Adqua得点と品質指標の相関係数

	再利用ファイル率
再利用性	0.08
インターフェース 理解性	0.21
完結性	0.55
柔軟性	-0.85 ?

網掛け部が、回帰分析で
統計的に有意

全てがそうはうまく行かない！



分析結果の考察

- 若干課題も有るが、Adqua得点の妥当性は概ね検証出来た(安心して開発者に薦められる自信が得られた)
- 部分的に想定していた結果が得られなかったのは、Adquaの得点が間違っているのか、品質指標が適切ではなかったのか、残念ながらその調査は出来ていない
- かなり以前のプロジェクトのデータから品質指標を捻出して分析を試みたので、品質指標の方が適切ではなかった可能性は高い
- ただし、想定とは全く逆の結果が出た部分に関しては、Adquaにも改善点が有ると考えている(今後の課題として取り組む予定)

まとめ

- プロセス改善を推進していると、どうしても仕事の品質ばかりに着目しがちです。しかしながら、ものづくりの現場の基本姿勢として、やはり出来上がった成果物の品質を直視することを忘れてはいけないと思います
- 更に、仕事と成果物の品質が互いに寄与しあっていることが確認出来なければ、プロセス改善はただの自己満足となってしまいます
- そのための道具の1つとして、Adquaは大いに活用出来るものに仕上がったと自負しております

主要な参考文献

鷺崎弘宜, 小池利和, 波木理恵子, 田邊浩之,
C言語プログラムソースコードの再利用性測定法とその評価,
ソフトウェアテストシンポジウム JaSST'09 Tokyo, 2009.

鷺崎弘宜, 波木理恵子, 福岡呂之, 原田洋子, 渡辺博之,
プログラムソースコードのための実用的な品質評価枠組み,
情報処理学会論文誌, Vol.48, No.8, pp.2637-2650, 2007.



YAMAHA

感動を・ともに・創る

ご清聴ありがとうございました