

何をもちて品質目標値を変えるのか？CK メトリクスでしょう！

How do you change the quality target value? CK metrics!

小島 義也

Yoshiya.kojima@avasys.jp

エプソンアヴァシス株式会社 品質管理部

発表要旨：

システム開発において品質目標値を見直す場合、適切な判断であれば、進行しているプロジェクトの改善にはとても有効である。問題はその適切な判断を行うために何を基準とするか？である。

今回はコード分析では定評のある CK メトリクスを採用し、その判断基準に使用できるかを試みた。

CK は 6 種のメトリクスからなる、クラスに対する設計と複雑度を評価する尺度である。採用した理由としては、設計成果であるコードの情報から判断基準を作成すれば、その時点での品質状況を推測でき、かつ開発途中の目標の変更を実施できるのではないかと考えたからである。

本発表では、実際の開発プロジェクトにこの CK メトリクスをどのように適用し、今回の目的である品質目標値を変更する基準作成がどの程度まで実施できたのか、この一連の取組みについて紹介する。

キーワード：

CK メトリクス、コードメトリクス、品質目標の見直し、品質状況

想定している聴衆

品質管理、設計開発

発表者の紹介（全角 100 文字）：

元 ISMS 主任審査員／プライバシーマーク審査員／日科技連カフェ・ソフトウェアクオリティ世話人／「評価者による ODC を使用した不具合分析の現場展開～属人化を排除していく試み」(JaSST Tokyo 2015)



何をもって品質目標値を
変えるのか？
CKメトリクスでしょう！

小島 義也

エプソンアヴァシス株式会社

e-mail:yoshiya.kojima@avasys.jp

EPSON AVASYS CORPORATION

いつ品質目標値を見直しますか？

- 要件確定後→大規模開発なら。。
- 設計完了後→定量的な情報が少なく、リスクを盛る傾向で落ち着いてしまう。
- 実装完了後→開発物(成果物)が計測できる。
- 単体結合テスト完了後→開発物の完成度が高い。でも、ちょっと遅い？

開発物を計測してみる

- 派生開発のため、複数プロジェクトのデータを計測する必要があった。
- そのため、継続的に比較可能なメトリクスを選択する必要性から、定評のある**CKメトリクス**を採用した。

WMC: 複雑度(メソッド数)
DIT: 継承ツリーの深度
NOC: 子クラスの数

CBO: 結合度
RFC: 呼び出すメソッド数
LCOM: 凝集度の欠如

- ChidamberとKemererが1994年に提案した、オブジェクト指向ソフトウェアに対応した6種の複雑度の尺度。

自分たちの基準値をつくってみた

- プロジェクトの総合結果とコードメトリクスの結果を比較して、対象プロジェクトの判断基準を作り、その傾向から品質状態を推測することを試行した。
- まずは、ゾーン分析を試してみた。
- 分析結果をもとに、実プロジェクトの開発中に目標値変更を試行した。

基準値外 ←CK値		→基準値内		基準値	
悪い	高低1	良い	高低2	AVG	MED
WMC	↑	WMC	↓	32	13
DIT	→	DIT	→	2	2
NOC	→	NOC	→	1	0
CBO	↑	CBO	↓	16	11
RFC	↑	RFC	↓	23	15
LCOM	↑	LCOM	↓	52	64
WMC	↑	WMC	↓	32	13
DIT	→	DIT	→	2	2
NOC	→	NOC	→	1	0
CBO	↑	CBO	↓	16	11
RFC	↑	RFC	↓	23	15
LCOM	↑	LCOM	↓	52	64
	悪い	良い		AVG	MED
基準値外 ←CK値	→基準値内		基準値		

① [レビュー欠陥が高い]
[CK値が基準値外]

↑
欠陥
↓

少ない

② [レビュー欠陥が低い]
[CK値が基準値外]

③ [レビュー欠陥が高い]
[CK値が基準値内]

↑
欠陥
↓

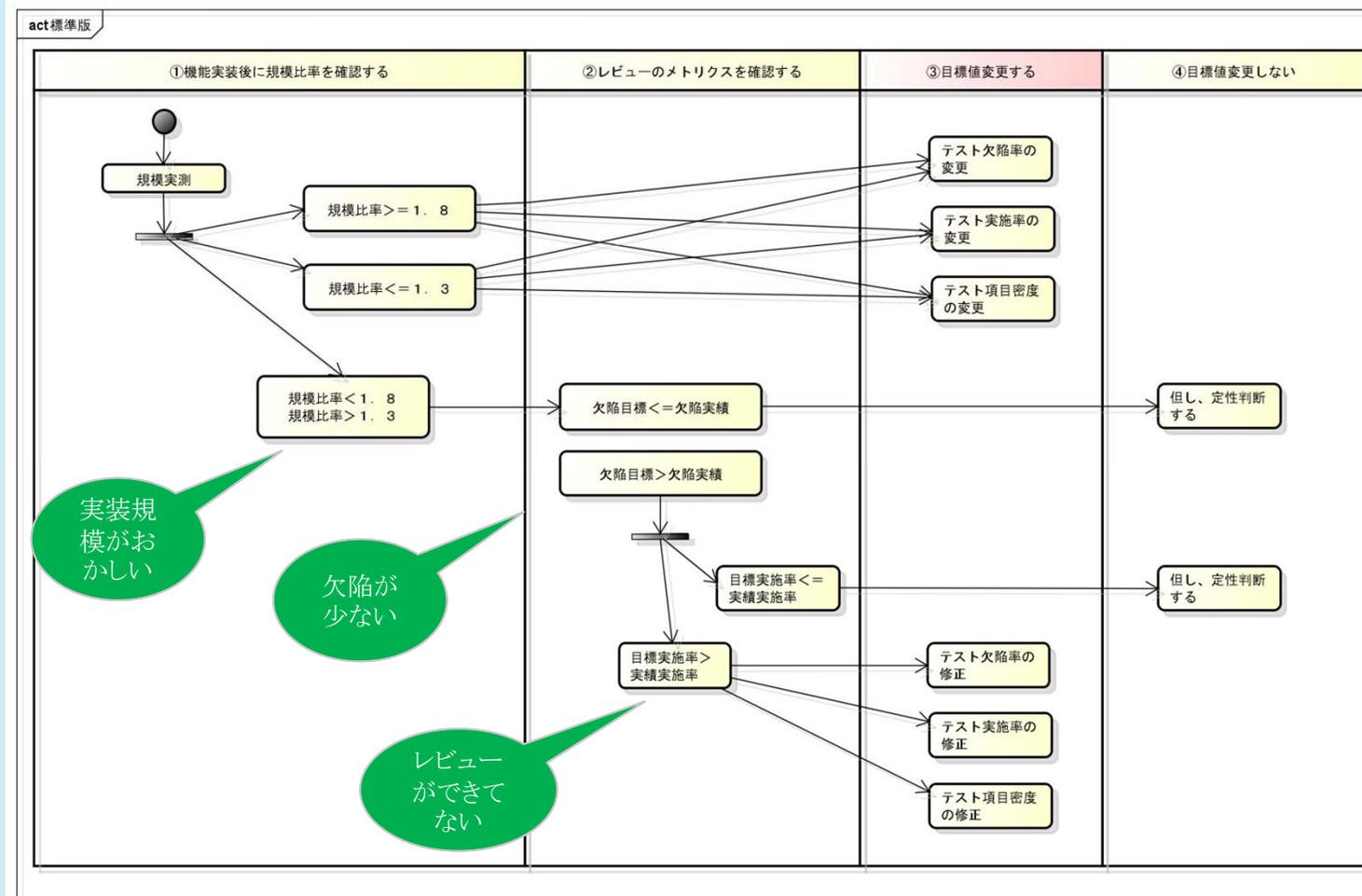
少ない

④ [レビュー欠陥が低い]
[CK値が基準値内]

ゾーン分析はいまひとつか？

- 基準値はこれまでのプロジェクトで計測したCK値の平均、中央値で比較した。
- 四象限の各々に係数を設定し、目標値に係数を掛ける方法で設定値の変更を行った。
- この方法は簡易的に実施するには検討工数もかからず、目標値の変更ができて良い面もあった。
- ただ、各テスト工程の欠陥密度、実施率、項目密度に同じ係数を掛けるのは、いまひとつ。
- また、規模が大きい開発の場合、この方法では当てはまらなかった。

プロセス・メトリクスの子実で品質目標の変更を判断



まず開発規模の子実をみて⇒レビュー欠陥密度の子実をみて⇒レビュー実施率の子実をみて、目標値を変更するか否かを判断する。

品質目標の変更をPMと合意する

- 目標値と実績値とのズレが生じた場合は、まず原因分析を行い、目標値を変更することがプロジェクト遂行にとって妥当であることの可否、そして理解と納得を、現場に示すことが必要。
- 規模の実績値で品質目標を変更する場合、次にCKメトリクスの数値を参照し、**テスト欠陥密度、テスト項目密度、テスト実施率**の目標変更を実施。
- 上記はPMと一緒に検討し、合意した上で実施することが肝要。

テスト欠陥密度の目標値を補正する

- 過去プロジェクトの成果物と現在プロジェクトの実装後のコードからCKメトリクスを計測する。
- 過去プロジェクトの成果物から各複雑度(各CK)の基準値を算出し、現在プロジェクトの各複雑度と比較する。
- 各複雑度の基準値はCKのメトリクスの6種毎に算出する。
- 各複雑度の基準値には、各CK値のZスコアを使用する。
- 各CKメトリクスを、その複雑度の意味から対応するテスト工程に設定する。
- 各テスト工程に各複雑度の増減値を割り当てる。割り当てた増減値をテスト欠陥密度の目標値の増減と仮定して、目標値を補正する。

各CK値のZスコアから基準値を設定する

- 過去プロジェクトのデータから、各CK値のZスコアを算出し、基準値とした。
- その基準値と現在PJの実装完了後の各複雑度を比較した。
- 基準値を超えたプロジェクトは品質が悪いとの判断をした。

LCOM

Project名	平均値	合否判定	PJ優劣
Project01	61.92	NG	劣
Project02	46.21	OK	優
Project03	48.92	OK	劣
Project04	52.04	NG	優
Project05	58.93	NG	劣
Project06	45.91	OK	優
Project07	57.08	NG	劣
Project08	51.87	OK	優
Project09	61.12	NG	劣
Project10	55.01	NG	劣

※過去のプロジェクトが優勢であったか、劣勢であったかのか、PMの定性的な判断から、PJ優劣を設定した。

LCOMの例⇒同様にCKの6メトリクスに対して基準値から修正目標値の増減を算出する

平均値の平均	53.90
平均値の標準偏差	5.85
基準割合	0.63
基準Zスコア	-0.33
平均値の基準値	51.96
OK数	4.00
PJ優勢数	4.00
今回PJの平均値	54.49
今回PJのZスコア	0.10
平均値の基準値からの差異	2.53
Zスコアの差異	0.43
欠陥密度の目標値	3.00
欠陥密度の修正目標値	1.30

- ①OK数とPJ優劣数が同件数となる割合から基準値を求めた
- ②過去PJのZスコアを基準値と仮定する
- ③今回PJの平均値を算出する
- ④今回PJのZスコアを算出する
- ⑤今回PJのZスコア-基準Zスコアを品質が悪くなっている補正係数とする
- ⑥結合テストの目標値
- ⑦結合テストの欠陥密度×Zスコアの差異⇒修正目標値の増分と仮定した

CKメトリクスと対象テストを定義する

CK略称	CKメトリクス	テストフェーズ	略称	複雑度の種類	テスト内容との意味付け
WMC	循環的複雑度	単体テスト	UT	経路の複雑度	経路数による機能関連のテスト
RFC	呼出メソッド数	単体テスト	UT	応答の複雑度	呼び出すメソッドのテスト
CBO	クラス結合度	結合テスト	LT	結合の複雑度	結合クラスのテスト
DIT	継承ツリーの深 度	結合テスト	LT	継承の複雑度	継承されている変数 やメソッドのテスト
NOC	派生クラス数	機能テスト	FT	派生の複雑度	サブクラスのテスト
LCOM	凝集度の欠如	機能テスト	FT	過度の複雑度	機能の影響範囲のテ スト

- CKメトリクスの複雑度の意味をもとに、各テスト工程との紐付けを考えた。
- 必ずしも上記の当てはめが正しいとは言えないため、他のプロダクトメトリクスでの検証も必要であると考えている。

テスト欠陥密度の目標値を補正する

- 各CKメトリクスを、その複雑度の意味から対応するテスト工程を設定し、複雑度の増減値を割り当てた。

WMC(循環的複雑度)	経路の複雑度	単体テスト	UT
RFC(メソッド数)	応答の複雑度	単体テスト	UT
CBO(クラス結合度)	結合の複雑度	結合テスト	LT
DIT(継承ツリー深度)	継承の複雑度	結合テスト	LT
NOC(派生クラス数)	継承の複雑度	機能テスト	FT
LCOM(凝集度の欠如)	過度の複雑度	機能テスト	FT

- 各テストに割り当てた増減値をテスト欠陥密度の目標値の増減とし、目標値を補正した。

	目標[欠陥密度]	増分		修正目標[欠陥密度]
単体テスト	2.0	WMC	0.30	2.1
		RFC	-0.18	
結合テスト	1.5	CBO	1.33	3.4
		DIT	0.53	
機能テスト	2.5	NOC	0.25	4.1
		LCOM	1.30	

- 各テストの欠陥密度 × 割り当てたCK値のZスコアの増分 ⇒ 修正目標値

テスト項目密度の目標値を補正する

- テスト項目密度の対象テストを定義する。
- CKメトリクス間の相関係数を算出する。
- 相関係数からテスト作業量の割合を推測する。
- テスト項目密度の増減を推測する。
- テスト項目密度の目標値を補正する。

テスト項目密度の対象テストを定義する

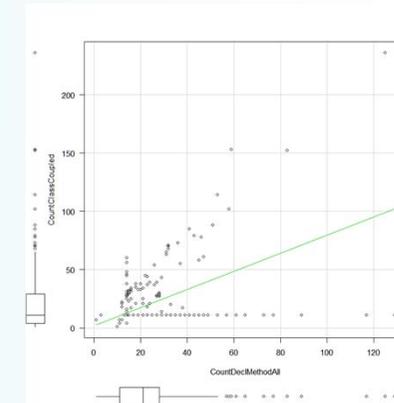
WMC(循環的複雑度)	経路の複雑度	単体テスト	UT
RFC(メソッド数)	応答の複雑度	単体テスト	UT
CBO(クラス結合度)	結合の複雑度	結合テスト	LT
DIT(継承ツリー深度)	継承の複雑度	結合テスト	LT
NOC(派生クラス数)	継承の複雑度	機能テスト	FT
LCOM(凝集度の欠如)	過度の複雑度	機能テスト	FT

WMC(循環的複雑度)	DIT(継承ツリー深度)					
RFC(メソッド数)	UT	NOC(派生クラス数)				
CBO(クラス結合度)	UT-LT	UT-LT	CBO(クラス結合度)			
DIT(継承ツリー深度)	UT-LT	UT-LT	LT	RFC(メソッド数)		
NOC(派生クラス数)	UT-FT	UT-FT	LT-FT	LT-FT	WMC(循環的複雑度)	
LCOM(凝集度の欠如)	UT-FT	UT-FT	LT-FT	LT-FT	FT	LCOM(凝集度の欠如)

- CKがどのテスト工程において関係性が強いかを想定して、テスト項目密度の目標値を補正する。
- 複雑度が相関していれば、テストの影響範囲は見直しせざるを得ないと考えた。

CKメトリクス間の相関係数を算出する

WMC(循環的複雑度)	WMC(循環的複雑度)					
RFC(メソッド数)	0.404	RFC(メソッド数)				
CBO(クラス結合度)	0.664	0.573	CBO(クラス結合度)			
DIT(継承ツリー深度)	-0.295	-0.223	-0.0477	DIT(継承ツリー深度)		
NOC(派生クラス数)	NA	NA	NA	NA	NOC(派生クラス数)	
LCOM(凝集度の欠如)	0.207	0.248	0.62	0.443	NA	LCOM(凝集度の欠如)



- ここでは、CKメトリクス間の相関係数を求めている。
- もちろん、2つの変数間の因果関係を説明できるものではないが、可能性としてのCK(各複雑度)間の影響発生を表現した。

相関係数からテスト作業量の割合を推測する

- ▶2つの変数(例えばWMC/RFC)と一緒に働いたときの作業量の割合(増加または減少)として相関係数を利用した。
- ▶但し、相関係数は順序尺度であるため、そのまま作業量の計算には使用できない。
- ▶そのため、まず相関が強いかわ弱いかわだけを各テスト工程において再設定した。
- ▶これは、CKの因果関係の分析は難しくできていないためであり、CKは互いに原因と結果の関係にはあると前提した。

WMC(循環的複雑度)	WMC(循環的複雑度)					
RFC(メソッド数)	+強い	RFC(メソッド数)				
CBO(クラス結合度)	+強い	+強い	CBO(クラス結合度)			
DIT(継承ツリー深度)	-弱い	-弱い	ない	DIT(継承ツリー深度)		
NOC(派生クラス数)	NA	NA	NA	NA	NOC(派生クラス数)	
LCOM(凝集度の欠如)	+弱い	+弱い	+強い	+強い	NA	LCOM(凝集度の欠如)

テスト項目密度の増減を推測し、目標値を補正する

テスト	項目密度	修正目標項目密度
単体テスト	100	150
結合テスト	50	90
機能テスト	200	240

- 相関の強い／弱いに対して、補正係数を設定した。
- 目標値に対して補正係数を掛け、各テスト工程のテスト項目密度の増減を算出した。

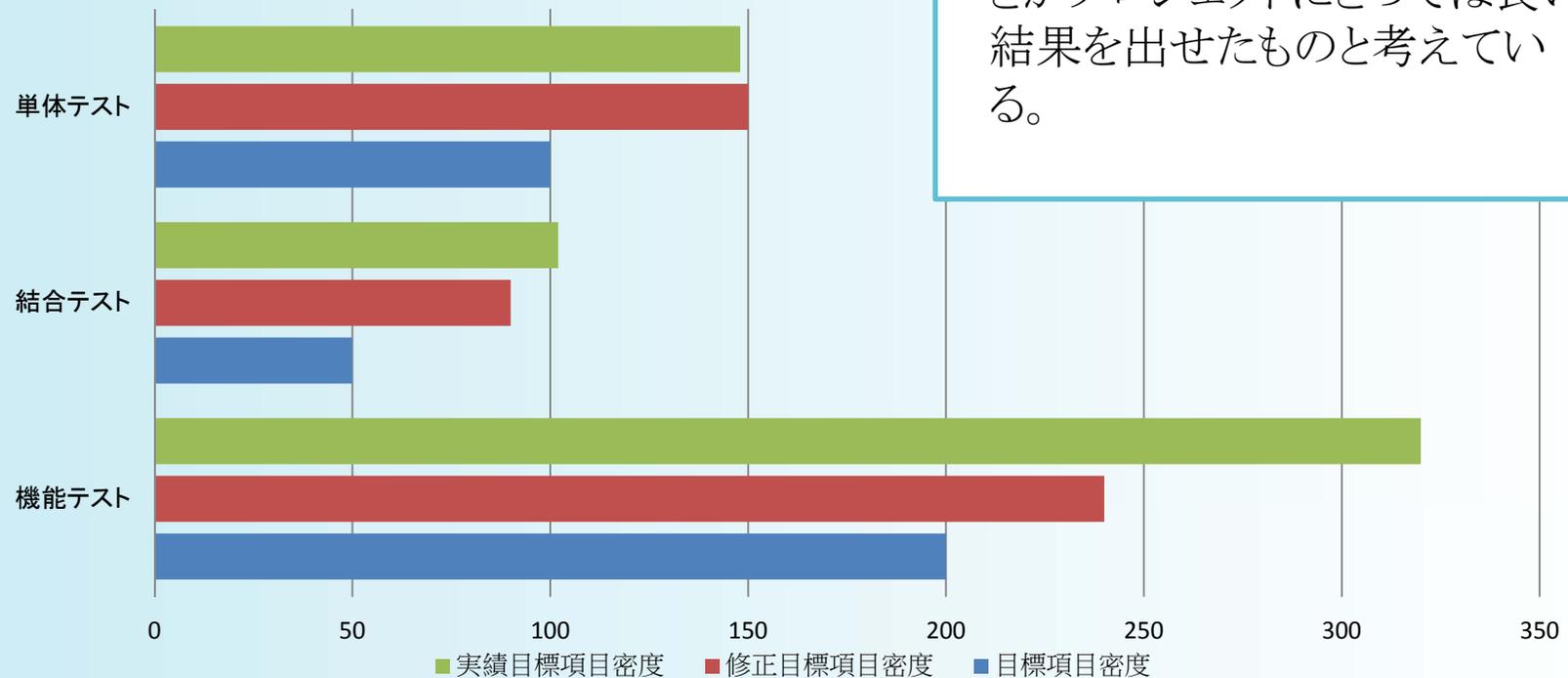
WMC(循環的複雑度)	WMC(循環的複雑度)	修正目標項目密度	修正係数	修正目標項目密度	修正係数
RFC(メソッド数)	50	RFC(メソッド数)	1.8	90	1.8
CBO(クラス結合度)	50	50	1.0	50	1.0
	25	25	0.9	22.5	0.9
DIT(継承ツリー深度)	-10	-10	0.9	-9	0.9
	-5	-5	0.8	-4	0.8
NOC(派生クラス数)	NA	NA	NA	NA	NA
	NA	NA	NA	NA	NA
LCOM(凝集度の欠如)	10	10	2.5	25	2.5
	20	20	10.0	200	10.0

PJ完了後のテスト項目密度の結果

テスト	目標項目密度	修正目標項目密度	実績目標項目密度
単体テスト	100	150	148
結合テスト	50	90	102
機能テスト	200	240	320

- ▶修正目標が各テストにおいて役に立っていることは実績から見える。
- ▶推測の正確さよりも、修正目標値に対しての改善策を立てたことがプロジェクトにとっては良い結果を出せたものと考えている。

テスト項目密度比較結果

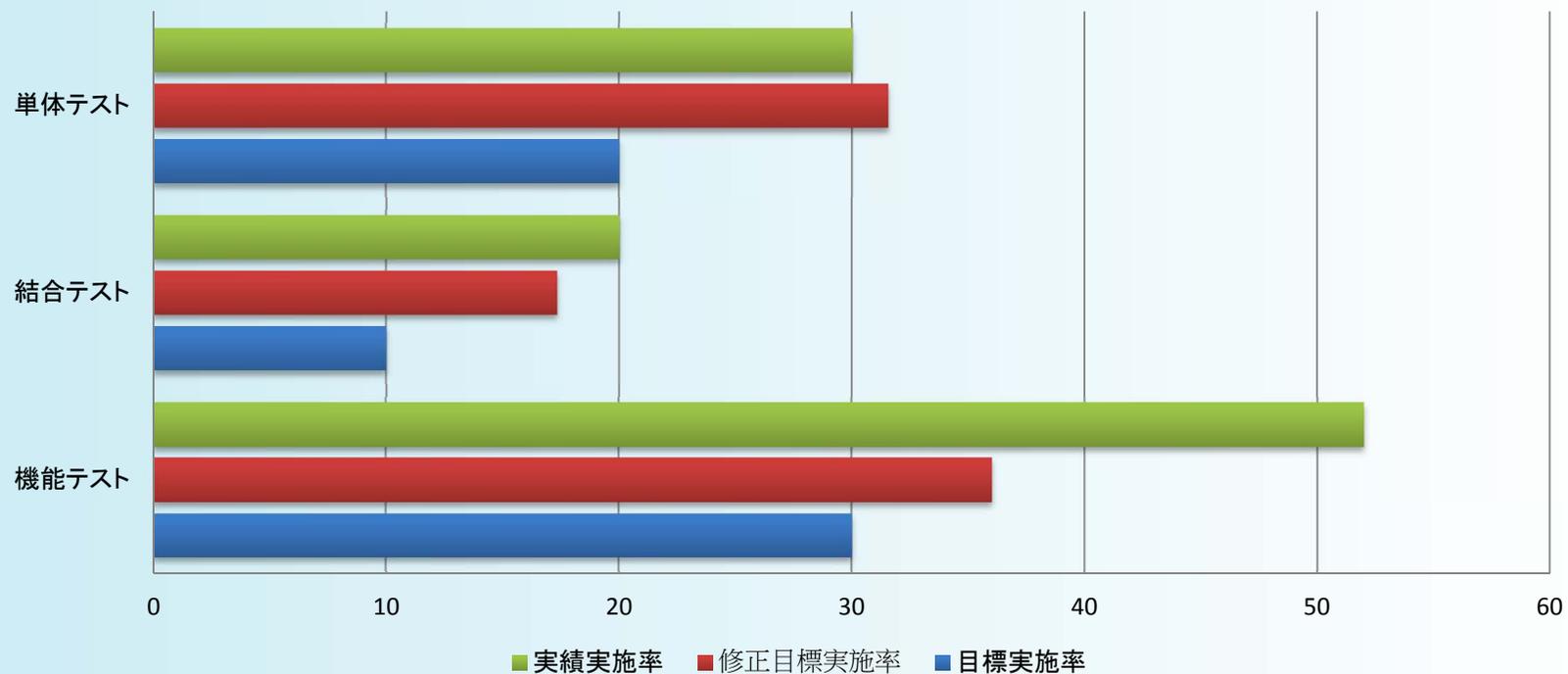


PJ完了後のテスト実施率の結果

テスト	目標実施率	修正目標実施率
単体テスト	20	32
結合テスト	10	17
機能テスト	30	36

➤ テスト実施率に関しては、テスト項目密度で追加した係数をそのまま利用した。

テスト実施率比較結果

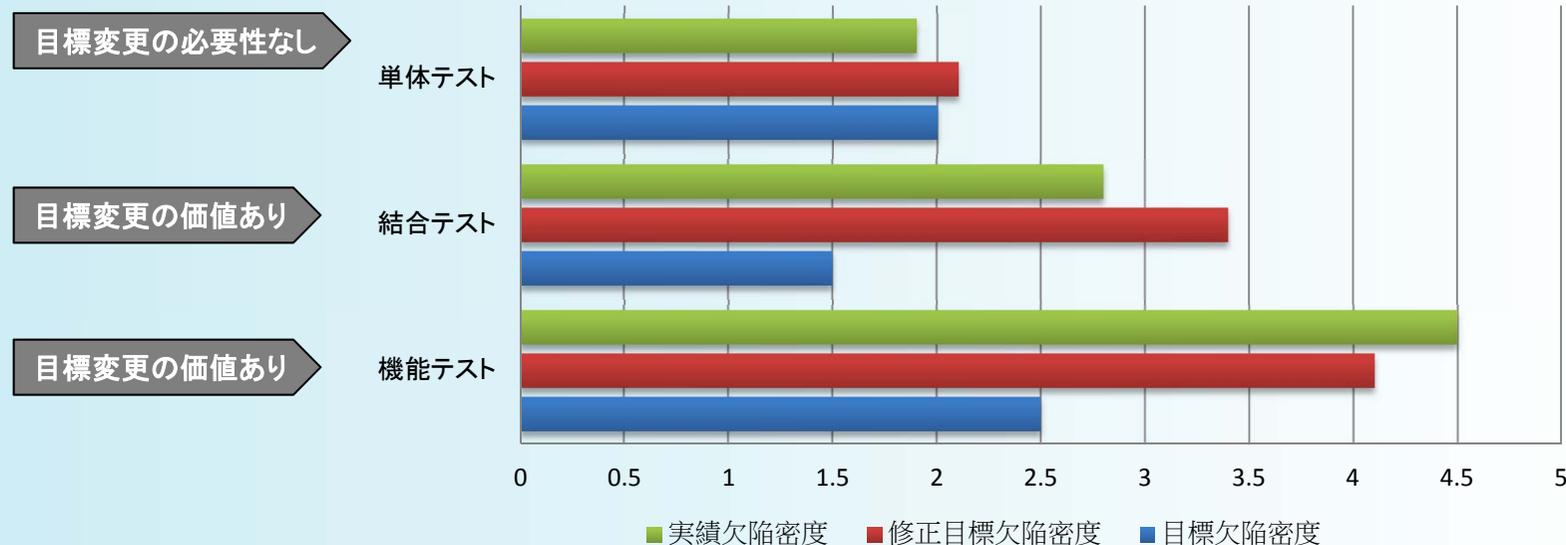


PJ完了後のテスト欠陥密度の結果

	目標[欠陥密度]	増分		修正目標[欠陥密度]
単体テスト	2.0	WMC	0.30	2.1
		RFC	-0.18	
結合テスト	1.5	CBO	1.33	3.4
		DIT	0.53	
機能テスト	2.5	NOC	0.25	4.1
		LCOM	1.30	

- 修正目標が各テストにおいて役に立っていることは実績から見える。
- 変更値を欠陥予測量としてみれば、目標変更には価値がある。

テスト欠陥密度比較結果



まとめ

- 実績としては、規模計測による成果物の判断は有効性が高い。
- CKメトリクスの関係も目標値変更の有用性を示せる結果となった。
- 但し、実際には、WMC,CBO,LCOMが意味のある数値を示しており、他のプロダクトメトリクスの検証も必要だと考えている。
- また因果関係が示せていないため、限定プロジェクトでの成果とも言えなくはない。
- 上記のような課題はあるが、成果物の数値を直接測定し、品質の状況を推測すること自体に意義があり、一步前進したと考えている。
- 品質はプロセスのメトリクスから判断する機会が多いが、プロジェクトの直接の成果物であるコードからの測定値から、現状の品質を推測するアプローチも重要なことである。