



Pairwiseテスト導入の 効果と課題

Pairwiseは大規模パッケージソフトウェアで
効果的に運用できるか

株式会社ワークスアプリケーションズ
アドバンスト・テクノロジー&エンジニアリング本部
品質エンジニア 森山 貴士

テスト設計の技術

～テストケースだけでなく途中計算もお願いします～

だそうです、
元ネタがわからん・・・OKWAVE?

知ってたら誰かつぶやいて！

たぶん
みんなが
思ってること

**過去・現在・未来
すべてのバグを
生まれる前に
消し去りたい！**

残念ながらこの願いを
叶える白い小動物には
なれなかったので、
事例発表をします。

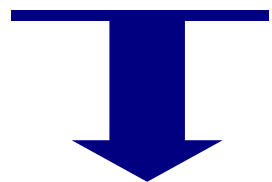
pairwise testing

pairwise?

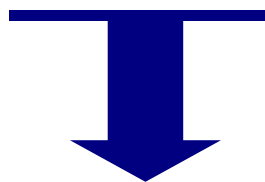
無限に増える組み合わせを実行可能な
レベルまで少なくするテスト手法

組み合わせテストの基本概念：因子と水準

- ✓ 「天気」が「晴れ」なら洗濯物を干す



因子



水準

pairwise.

Pairwiseと並列に扱われるもの

- ・ 直交表
- ・ HAYST法※

多くの場合、2 因子までの不具合がほとんど

参考 : Cumulative Percent of Faults Triggered by n-way Conditions

FTFI No.	RAX convergence	RAX correctness	RAX interf	RAX engine	POSIX modules	Medical Devices	Browser	Server	NASA GSFC
1	61	72	48	39	82	66	29	42	68
2	97	82	54	47	*	97	76	70	93
3	*	*	*	*	*	99	95	89	98
4	*	*	*	*	*	100	97	96	100
5	*	*	*	*	*		99	96	
6	*	*	*	*	*		100	100	

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 6, JUNE 2004 D.
Richard Kuhn, Senior Member, IEEE, Dolores R. Wallace, Member, IEEE Computer Society,
and Albert M. Gallo Jr.

Pairwise!

- ✓ どんなときに使える？
- ✓ どういう風にしたら運用できる？
- ✓ 他のメリットは？
- ✓ 問題は？
- ✓ 何から始めたらいい？

自己紹介

森山 貴士

Moriyama Takashi



- ✓ 1986年 大阪生まれ 25歳
- ✓ 立命館大学政策科学部 卒
- ✓ 2009年ワークスアプリケーションズ入社
- ✓ 品質技術系の部署に配属されて3年目
- ✓ facebookもやっています

弊社の紹介



株式会社ワークスアプリケーションズ

- ✓ 事業内容：

大手企業向けERPパッケージ「COMPANY[®]」
の開発・販売・サポート

- ✓ 企業理念：

「日本の情報投資効率を世界レベルへ」

「日本のクリティカルワーカーに活躍の場を」

今回の事例となる製品

COMPANY 就労・プロジェクト管理

人事発令

各種申請

勤怠管理

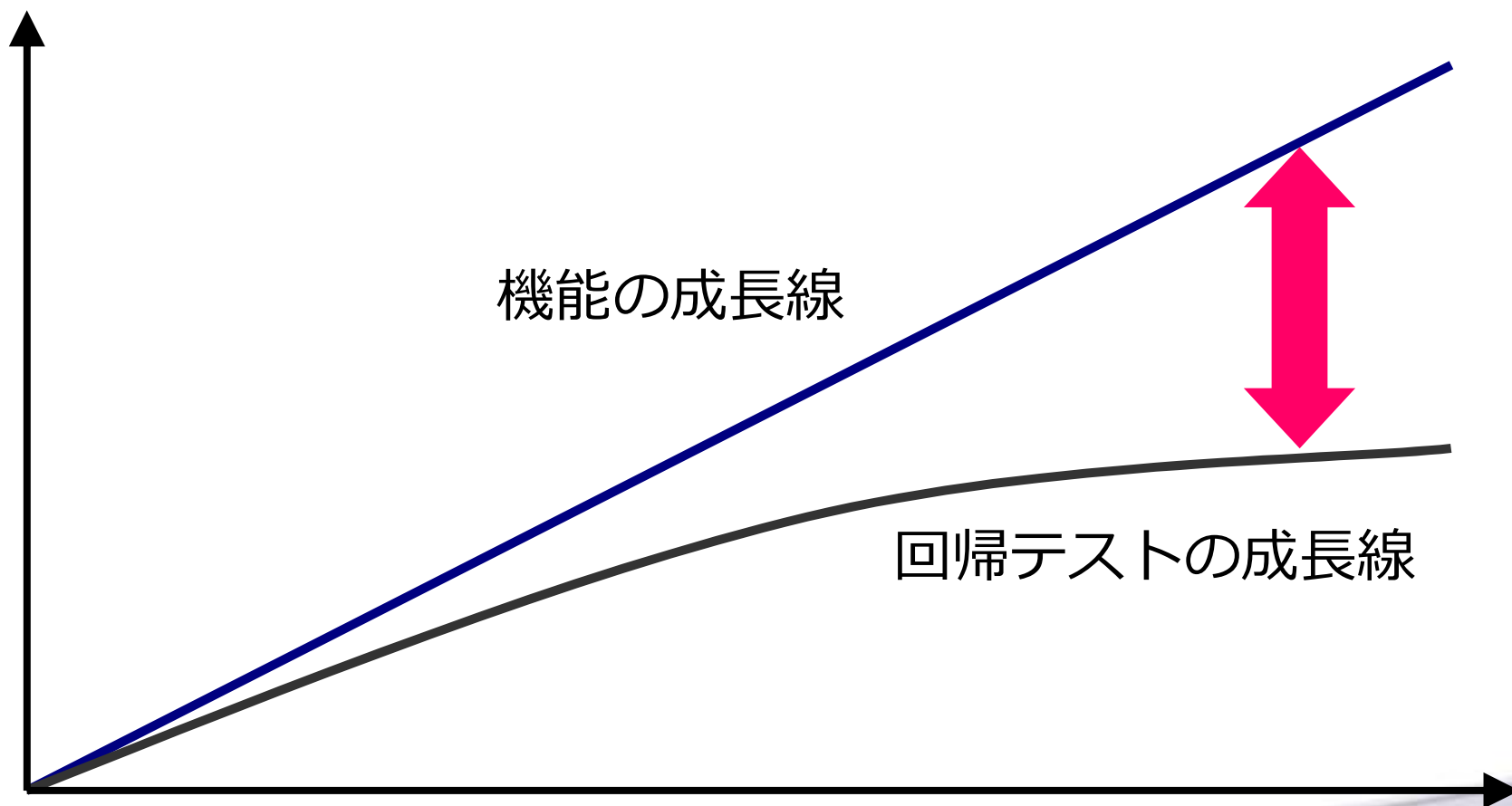
給与集計

会計処理
(ACシリーズ)

- ✓ **COMPANY HRシリーズ**のうちひとつ
- ✓ 日々の出勤退の打刻や休暇管理など、**勤怠**に関わる業務を扱う
- ✓ 開発規模：**500万**ステップ
- ✓ バージョンアップ：年3回以上

背景

機能の成長に回帰テストの充実が間に合わない



背景

機能が複雑になり、組み合わせが増大

テスト数が増え、メンテナンスコスト増大

バージョンアップごとの回帰テストが不十分になる

デグレードのリスクが高まってしまう



MISSION



デグレードしないように
網羅的な回帰テストを作って運用しろ！

ハハーツ!!

問題と仮説

わかっている問題点

スピードが追いつかない

機能に対してカバーできている範囲が限定的

やり方が属人化している

全体として実行テストを管理できていない

デグレードのリスクが高まってしまう

Pairwiseに着目



Zoom In!

Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

組み合わせは自動出力なので非属人化できる

考えが揃うので管理もしやすくなる

網羅的な回帰テストが充実！

先行調査

不具合修正レポートを調査

不具合の修正レポートの発生条件から、過去の不具合の因子数を調査

- ✓ 過去1年間のリリースノートからレポート200件を無作為抽出
- ✓ うち機能追加を除いた不具合の145件について調査

不具合現象

どのような不具合が発生したのか記述してください。
発生条件および影響範囲も記述してください。

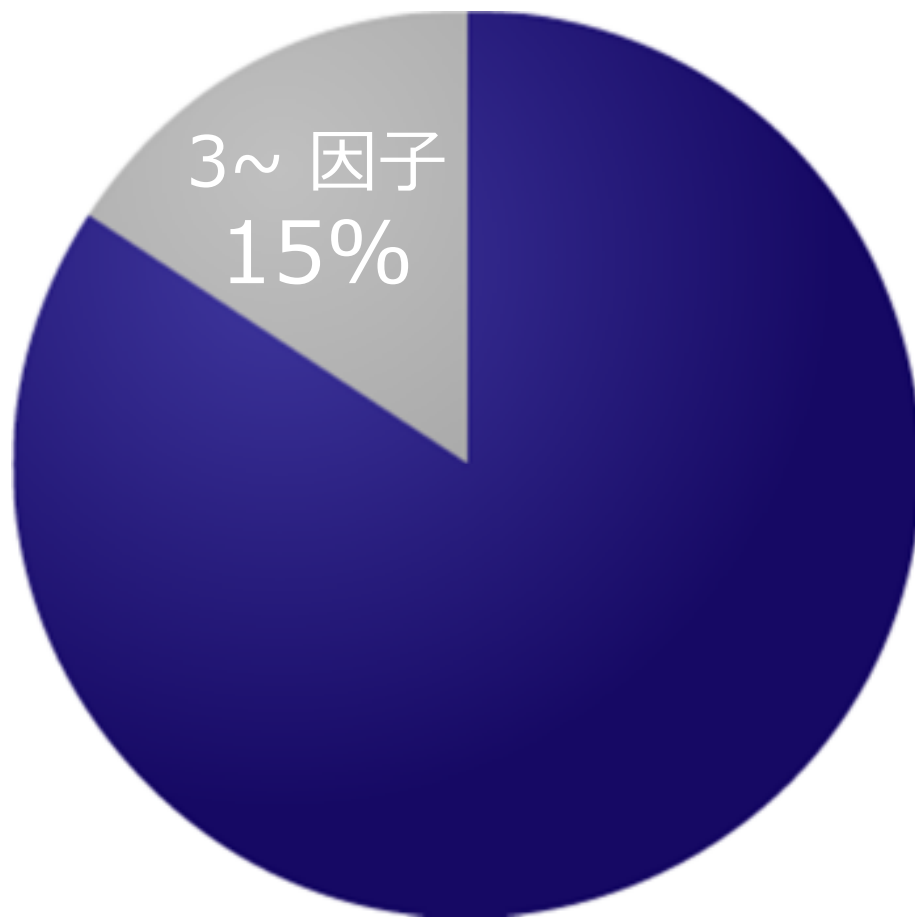
▲▼

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----

【起動経路】
【クライアント】
就労システム⇒休暇管理⇒休暇付与管理⇒休暇付与

【発生条件】
以下の全ての条件を満たす場合。
(1) 休暇付与の申請が承認されている。
(2) 休暇付与の申請が承認されている。

不具合の85%が2因子間までで発生

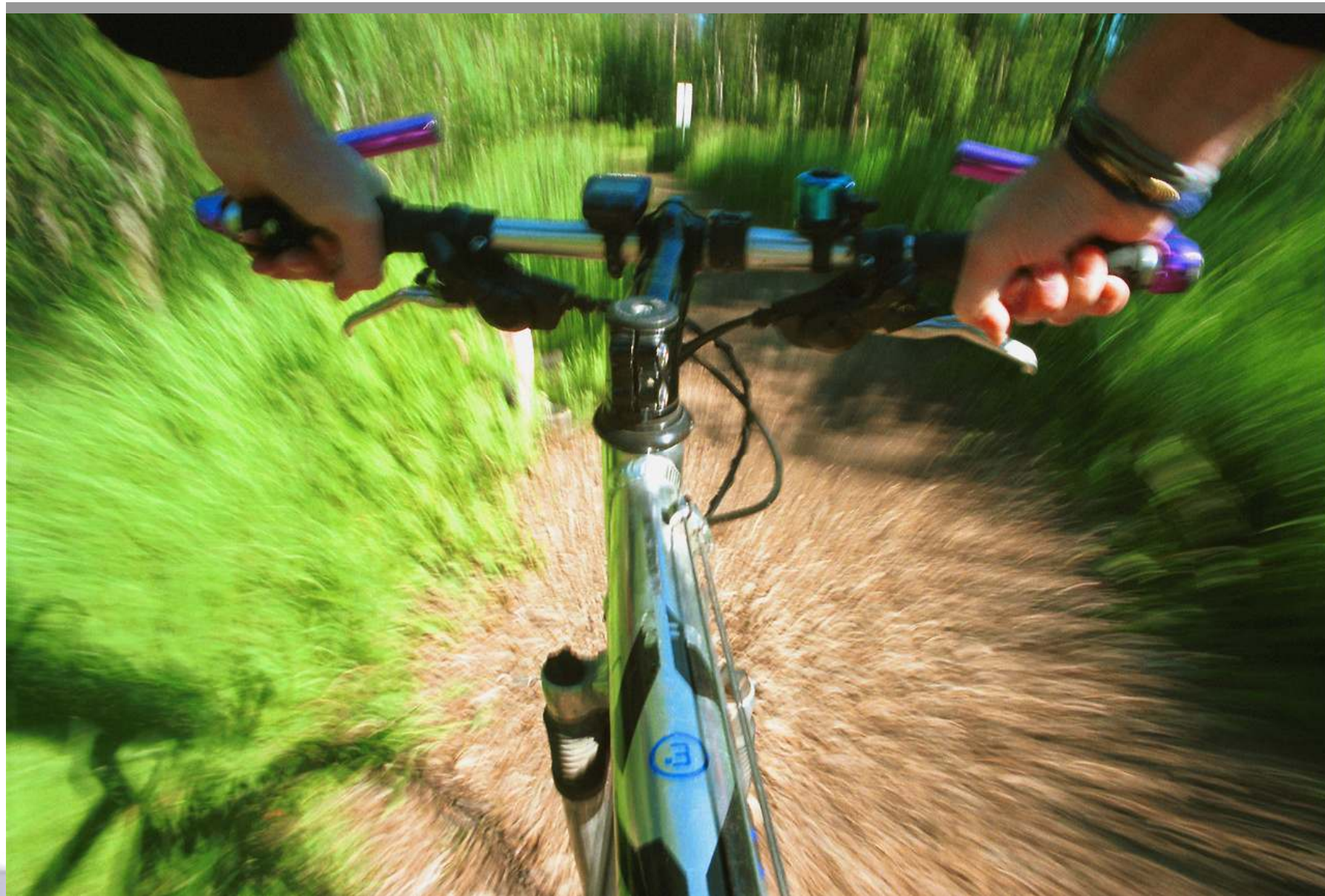


~2 因子

85%

(n=145)

スピード重視でPairwiseを採用



実施の概要

検証方法

- ✓ 休暇取得サブシステムについてのそれぞれの機能に対して、機能テストを実行
- ✓ 既存の方法とPairwiseを利用した方法の双方で検証を実施
- ✓ 双方の運用方法を比較して評価を行った

既存の方法で実施した期間	2010.7~12
Pairwiseを用いて実施した期間	2011.1~6

テストの組み合わせ作成方法

PictMasterを利用して組み合わせを作成

PictMaster

大項目 No.		大項目名	
小項目 No.		小項目名	

パラメータ	値の並び
申請したサービス	付与休暇申請00-09, 年次有給休暇届, 代休]
申請サービスの承認段階	無段階, 1 段階, 2 段階
申請した休暇	付与休暇, 時間代休, 代休, 公認休暇
申請した休暇の時間帯	全日, 午前, 午後, 時間, 分数
休暇モード	新, 互換
取消時の申請書のステータス	最終承認済(7), 1次承認済, 未承認, 差戻し

PICT + PictMasterでできること

- ✓ 因子と水準から因子間を網羅した組み合わせを生成
- ✓ 「天気」が「晴れ」なら「洗濯物」は絶対「干さない」のような制約指定
- ✓ 「晴れ」は多めに出現させたいなどの重み付け
- ✓ 組み合わせ条件からの想定結果の記述

その他、他にも色々あります。

テストケース作成のルール

網羅することを軸に設定

- ✓ 対象機能を決める
- ✓ 仕様として関わる因子を抽出する
- ✓ また、過去の不具合レポートから対象機能に影響があった因子をすべて抽出する
- ✓ 設定上できない、業務上ありえないもののみを制約条件として指定
- ✓ 重み付け等は任意で実行

実施の結果

Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

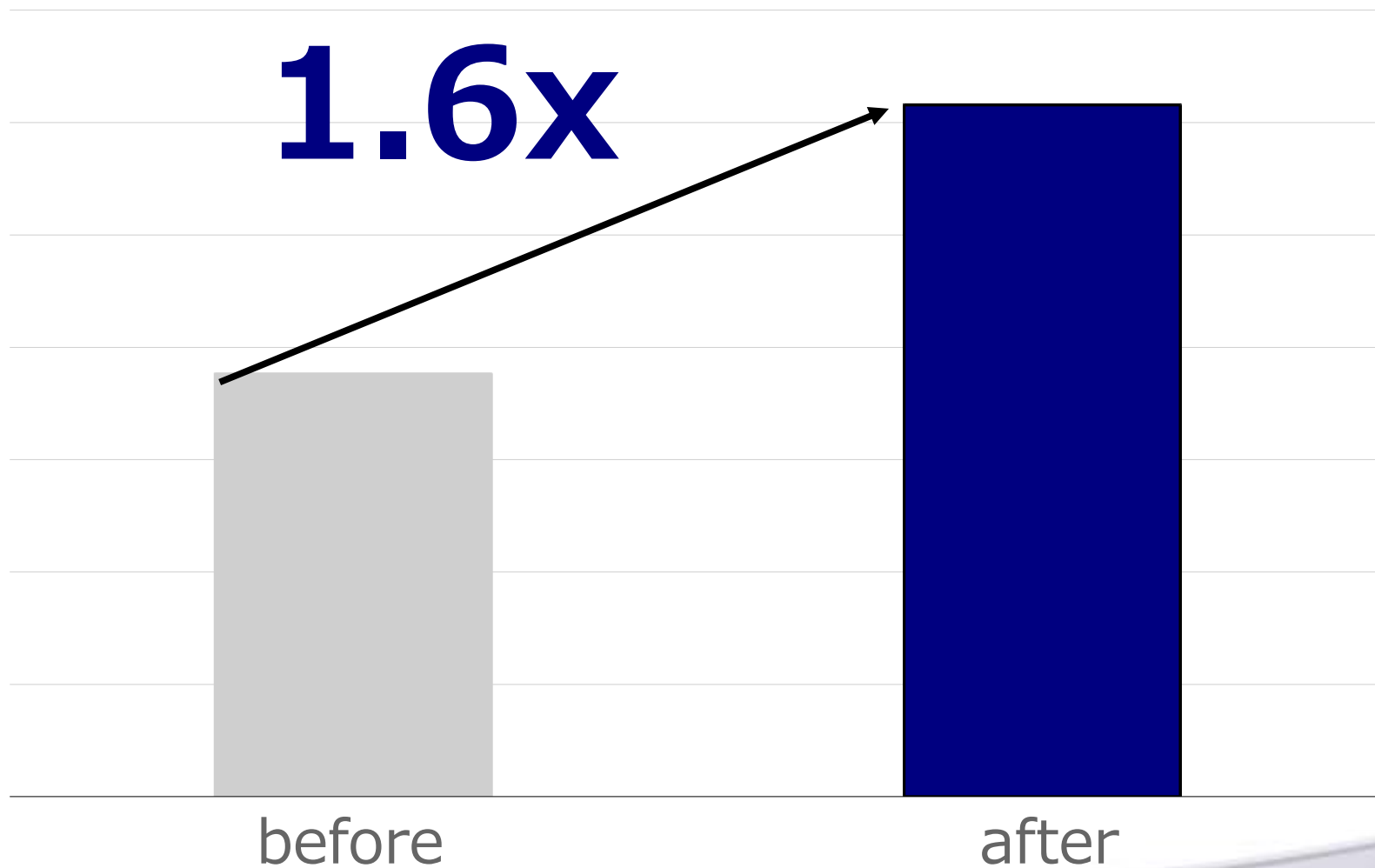
組み合わせは自動出力なので非属人化できる

考えが揃うので管理もしやすくなる

網羅的な回帰テストが充実！



○ テストケースあたりの不具合検出率が向上



Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

組み合わせは自動出力なので非属人化できる

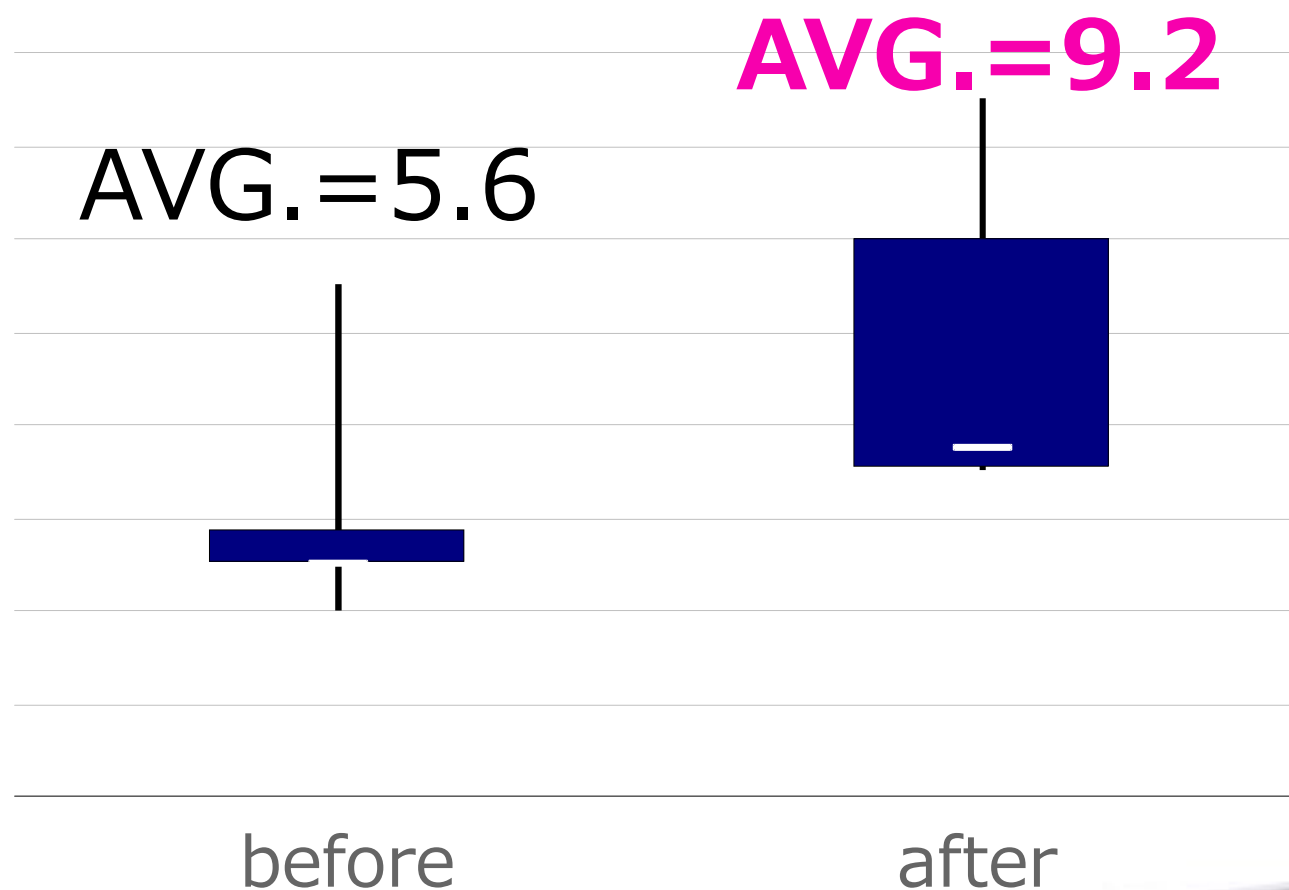
考えが揃うので管理もしやすくなる

網羅的な回帰テストが充実！



○ 1機能あたりの実行範囲も拡大

1機能に対するテストあたりの因子数の変化



Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

組み合わせは自動出力なので非属人化できる

考えが揃うので管理もしやすくなる

網羅的な回帰テストが充実！



△ きちんとしたレビューはやはり必要

良かったところ

やり方自体はかなり統一しやすい

課題

慣れていないとPairwiseすることが目的になり
軸がぶれる

制約条件の指定が思ったより複雑

Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

組み合わせは自動出力なので非属人化できる

考えが揃うので管理もしやすくなる

網羅的な回帰テストが充実！



今後の課題で話します

まだ準備中…



その他の取り組み

ありませんか？手順書の記述ミス

このパターンは本当は半日では？

いやいや想定結果が間違いでは？

操作

- 1) COMPANYにログイン
- 2) 年次有休休暇届（就労）に入る
- 3) **全日**休暇を新たに申請・承認する
- 4) 本人用予定入力画面に入り、データが期待される結果と相違ないことを確認する

半日の年次有給休暇の予定が指定した日付に作成されている

Pairwiseを使うことでパターンのマスタができる

	A	B	C	D	E	F
1	申請したサービス	申請サービス	申請した休暇	申請した休暇	休暇モード	取消時の
2	公認休暇届（就労）	2段階	付与休暇	午前	新	最終承認
3	公認休暇届（就労）	無段階	時間代休	午後	互換	最終承認
4	失効積立休暇届	2段階	代休	時間	互換	最終承認
5	特別休暇届	1段階	公認休暇	全日	新	最終承認
6	公認休暇届00-99	1段階	公認休暇	分数	互換	最終承認
7	年次有給休暇届	無段階	代休	午後	新	最終承認
8	失効積立休暇届	無段階	付与休暇	午前	互換	最終承認
9	代休取得申請	1段階	時間代休	時間	新	差戻し
10	付与休暇申請00-09	無段階	時間代休	分数	新	最終承認
11	公認休暇届00-99	2段階	代休	全日	新	最終承認
12	特別休暇届	2段階	公認休暇	午前	互換	最終承認
13	失効積立休暇届	1段階	時間代休	全日	互換	最終承認

これを参照して手順をつくってみる

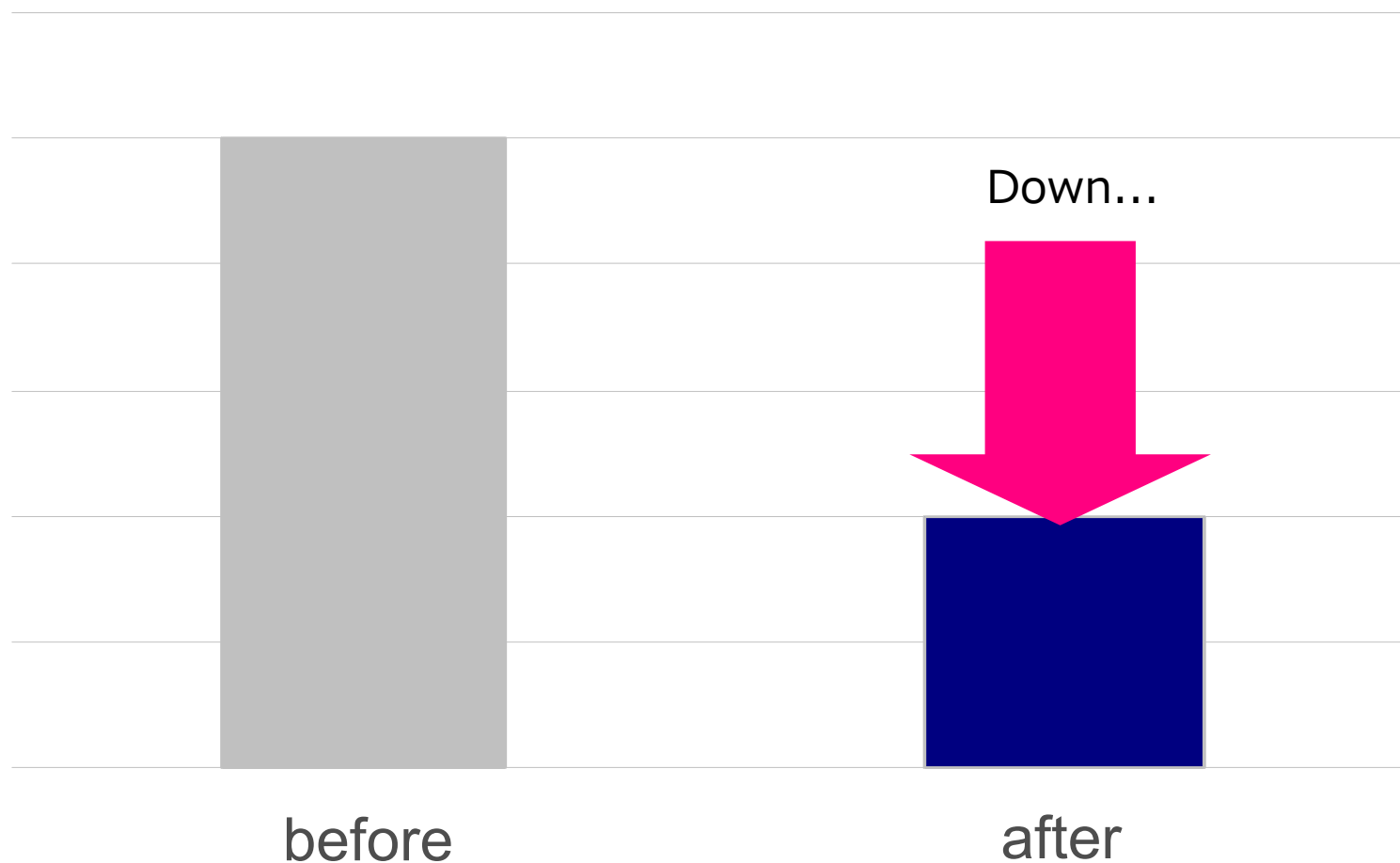
= "1)COMPANYにログイン"&CHAR(10)&"2)年次
有休休暇届（就労）に入る"&CHAR(10)&"3)"&
B3&"休暇を"&C3&"を"&E3&"申請・承認する"&
CHAR(10)&"4)本人用予定入力画面に入り、データ
が期待される結果と相違ないことを確認する"

= 手順書でコピペミスしない！

問題点 = これがEXCELだってこと。

発生した課題

設定項目が増えた分、実行速度が落ちた



遅くなった理由

因子が増えてテスト実施が複雑化

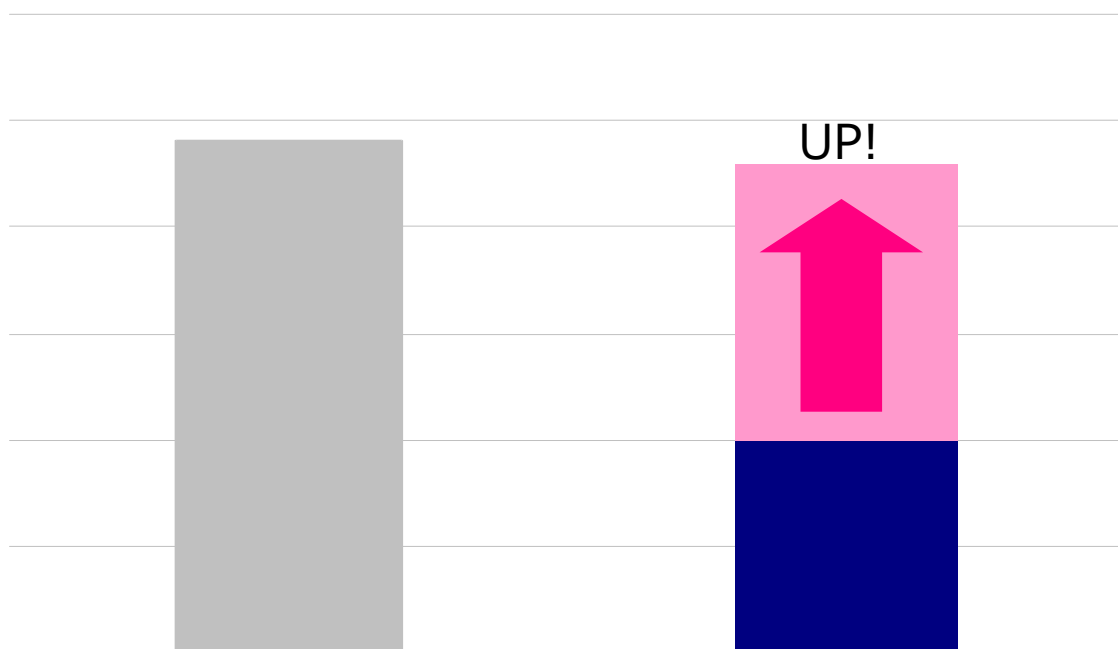
- ✓ 含める因子が多くなった分、設定項目が増大
- ✓ 単純に実行コストが増大
- ✓ 複雑な処理の中で作業ミスが増大



実行速度の改善

+2~3時間の作業でほぼ変わらないスピードに

- ✓ 事前設定部分をバッチ化
- ✓ 作業者は
業務画面のみ
実行して
テストを実施



まとめ

まとめ

過去の因子を調べるだけでも
設計はしやすくなる



まとめ



パッケージ・組み込み系など機能が複雑になりやすいソフトウェアで使いやすいと思われる

まとめ

ルールは決めやすいが完全な
ルーチン化ができるわけではない



まとめ

Pairwiseは何度も使うテストに利用すべし



今後の展望

Pairwise利用をした場合（仮説）

より効率的に不具合検出できる

網羅範囲を拡大できる

組み合わせは自動出力なので非属人化できる

考えが揃うので管理もしやすくなる

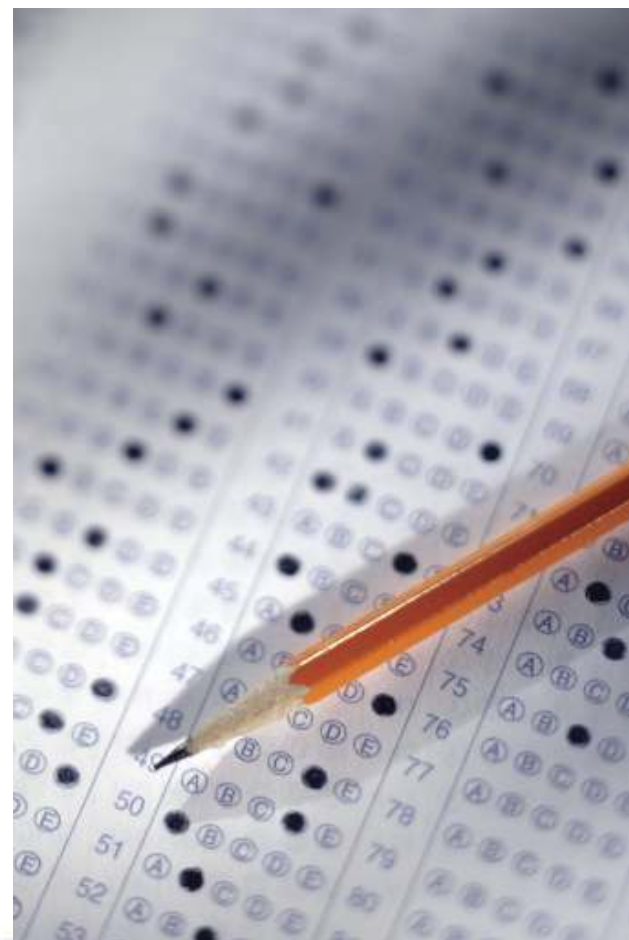
網羅的な回帰テストが充実！



今後の展望

大量のブラックボックステストを体系的に管理

- ✓ 個々の要素が全体もれなく構成することを示す
- ✓ ある機能に手を入れたときに実行すべき範囲は？
- ✓ 実施～結果レポートまでを自動で実行できる
- ✓ どこでどんなテストがされているか可視化



今後の展望

プロトタイピングとツールの検討

Fabre

メニュー▼

データリンク

各種マスタリンク

勤務実績出力リンク

+ 新規の要求の追加

#3 勤務実績出力リンク

要求の詳細

締められた実績を集計する処理

親

+ 親をセット

タグ

+ タグを追加

仕様の一覧

ID	内容
#0000001	[一定のスケジュールごとに自

+ 新規のグループの追加

#0000001 [一定のスケジュールごとに自

より詳細

年・月・日・時間ごとにスケジュールを設定するこ

TestLink



提案用プロトタイピング

最後に

品質技術業界をもっと盛り上げたい・・・！

- ✓ 製品の品質向上と自己の成長
- ✓ 技術や取り組みの向上や共有
- ✓ 特に若い人で盛り上げたい

以上となります。

ご清聴ありがとうございました！
今年もよろしくお願いします。