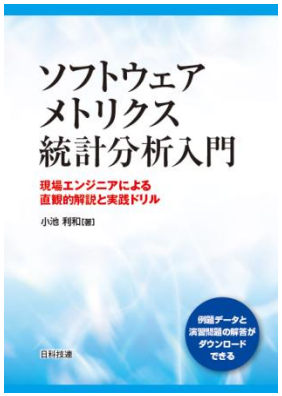
ソフトウェアメトリクスの測定精度の課題にどう対処するか

～測定精度を考慮した統計分析の実践～

ヤマハ(株)
小池 利和

自己紹介



小池 利和

ヤマハ(株) 品質保証部

【経歴】

1998年～ SEPG、SQAとしてソフトウェアメトリクスの実践活用に従事。
2013年～ 電子楽器製品全般の品質保証を担当。
2016年～ 全社のソフト品質改善活動に従事。

【学会活動、執筆、資格等】

日科技連SQiP研究会委員長、メトリクス演習コース主査、SQiP運営委員
即活用！「レビューの質チェック票」 http://thinkit.co.jp/article/856/1
QC検定1級（2013年9月成績上位者として日本規格協会Web上で表彰）
『データ指向のソフトウェア品質マネジメント』（日経品質管理文献賞受賞）
『ソフトウェアメトリクス統計分析入門』執筆

【コミュニティ活動】

データ分析勉強会
https://sites.google.com/site/kantometrics/
同人誌『メトリクス公団Vol.1』『メトリクス公団Vol.2』を発行

アジェンダ

- ソフトウェアメトリクスの分析に統計手法を使う意義
- 2群の母平均の差のt検定の解説
- リリース後品質を予見するメトリクス
  ～t検定を実務で活用した事例～

「メトリクスとは何か」や「収集方法」を扱う講義ではありません。
メトリクス分析を有効なものとする強力な手段の1つとして
統計手法の初歩を解説します。

ソフトウェアメトリクスの分析に
統計手法を使う意義

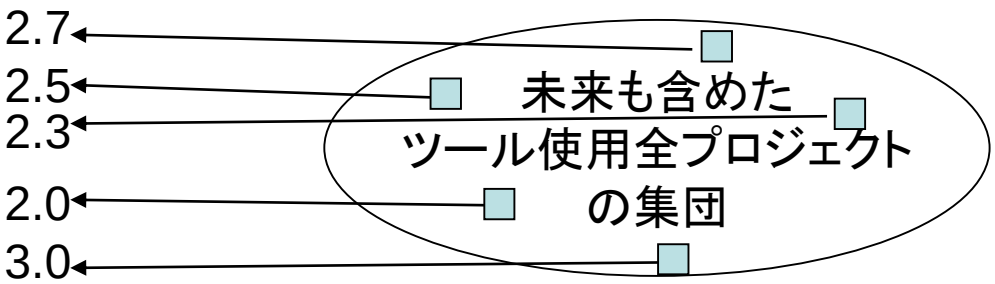
こんな時どうしますか？

- 品質向上を目的に静的解析ツールをトライアル導入しました。
- 品質向上の効果は、開発終盤のシステムテストでの欠陥密度（欠陥件数/システム規模KLOC）がどれだけ低下したかで判断することにします。
- ツール未使用/使用の5プロジェクトの欠陥密度は以下の通りでした。
  - **ツール未使用：2.8，3.2，3.0，2.5，3.5（平均＝3.0）**
  - **ツール使用　　：2.7，2.5，2.3，2.0，3.0（平均＝2.5）**
- ツールを全開発者が使用できるようにするには追加ライセンス購入で多大な費用が発生します。
- この結果から、ツールの効果ありと判断して、本格導入を進めるべきでしょうか？

測定データの平均と真の平均は違う

- ツールを使用した5プロジェクトだけで考えるならば、確かに平均は2.5となっていて、効果はあったと主張できます。
- しかし、本当に主張したいのは、ツールを使用することで、この組織の実力が上がり、将来のプロジェクト含めて欠陥密度が低いはず、ということです。この5つで低下しても、その後のプロジェクトで上がっては意味が有りません。
- 真の実力としての欠陥密度は、未来に実施されるプロジェクトも含めた全データで算出する必要がありますが、それは不可能です。
- 未来分も含めた全プロジェクトからサンプリングされた5つのデータを使って、真の実力値の平均を推し測らなければなりません。

ツール使用の集団からサンプリングされた5プロジェクトデータで平均を算出



測定データの平均=2.5 → **推測** → **真の実力の平均=？**

意思決定に統計手法を活用する意義

- データ分析結果に対する2つの反応について考えてみましょう。
  a. データで示されると説得力があるので、何となく信じてアクションを取る（あわて者）
  b. ソフトウェアメトリクスなんて精度が悪いので、分析結果は信用できず、アクションなど起こす気になれない（ぼんやり者）
- 今回得られた3.0、2.5という平均値は、たまたま得られた5プロジェクトのデータから算出されたもので、欠陥密度の真の実力値を示す訳ではありません。
- あわて者、ぼんやり者どちらの判断が正しいかは神のみぞ知ることで、ある意味ギャンブルです。ならば、確率が高い方に掛け続ければ、長い目で見れば勝てます。
- アクションすべきか、静観すべきかの判断に確率論をベースにした統計手法を活用すれば成功の可能性が高まります。

ソフトウェアメトリクスでの統計手法活用の意義

- 前頁でソフトウェアメトリクスは測定精度が悪いとありました。
  では、精度が悪いデータは本当に使えないのでしょうか？
- ソフトウェアメトリクスは自然科学分野のデータ（長さ、重さなど）に比べて、測定誤差が大きいのは事実です。
- しかしながら、経済学や心理学など人間の行動を研究対象とする社会科学分野では測定データの誤差が大きいのは当たり前のこととして、調査結果から結論を導くのに統計手法の活用は必須となっています。
- ソフトウェアメトリクスは人間の活動を測定している側面が大きいので、社会科学分野のデータに近い性質を持ちます。
- 精度が悪いから使えないと嘆くのではなく、だからこそ、統計手法を駆使して、誤差に埋もれない重要な情報を読み取る必要があるのです。

社会科学分野の研究例（その１）

夫婦の関係満足度および生活充実感における規定因の検討
（赤澤淳子 2005 社会心理学研究, Vol.21(2), 147–159.）

夫婦関係の安定や崩壊に関する有力な理論の一つに，社会的交換理論がある。この理論を基に，性別役割観・収入・子の年齢等の要因と，性別役割分業の投入および成果との関係について検討を行っている。ここでは，配偶者との関係にどのくらい満足しているかについて「非常に不満である（1 点）」から「非常に満足している（7 点）」の 7 件法の単一尺度で回答を求めたデータへの t 検定をとり上げる。

第 1 群：夫, mean=5.76, sd=1.36, n=236
第 2 群：妻, mean=5.36, sd=1.43, n=236
対応なし, t=4.32, $p < .001$, $\widehat{es}$=0.398

引用：豊田秀樹『検定力分析入門』（東京図書、2009）P66

- 要は配偶者に対して満足しているかどうかを夫/妻にアンケートして、満足度の違いがどうかを調査したもの
- 満足度の回答は1（不満）～7（満足）の7段階で、
  夫の平均が5.76、妻の平均が5.36という微妙な結果
- このデータに対して、対応の無い2群の母平均の差のt検定という統計手法を用いて、夫と妻の満足度に高度に統計的有意差ありという結論（満足度は妻＜夫、やはりというべきか・・・）

社会科学分野の研究例（その2）

「恋人」という間柄を意味する諸行為の記号学的分析 （山根一郎 1987 社会心理学研究, Vol.2(2), 29–34.）

異性間で交換されうる諸行為のなかで恋人という間柄に対して有標な行為を特定し，各行為の有標性の強さが行為の送り手の含意と受け手の推定において異性間で異なるかどうか検討している。ここでは“相手の夢を見たことを話す”という行為について恋人同士だという意味合いを「全く込めない/感じない (1)」から「非常に強く込める/感じる (7)」の単極次元で回答させたデータの，女性の含意と男性の推定における t 検定をとり上げる。

第 1 群：女性含意, mean=3.88, sd=1.76, n=84
第 2 群：男性推定, mean=4.39, sd=1.26, n=69
対応なし, t=2.06, $p < .05$, $\widehat{es}$= 0.335

引用：豊田秀樹『検定力分析入門』（東京図書、2009）P67

- 恋人同士で「相手の夢を見たことを話す」という行為の受け止め方の違いを調査したもの
- 女性含意とは女性がその行為に対して恋人同士という意味を込めたかどうかの度合で、男性推定とは逆に受け止め方の度合
- 回答は1（込めない/受け止めない）～7（込めた/受け止めた）の7段階で、女性含意の平均が3.88、男性推定の平均が4.39
- 前頁と同じ手法で、統計的有意差ありと結論（女性はそのつもりはないのに男性が勘違いするケースが多い・・・）

社会科学分野の研究例（その3）

テスト形式の違いによる学習方略と有効性の認知の変容 （村山航 2004 心理学研究, Vol.75(3), 262–268.）

テスト形式の違いが学習者に与えるテスト期待効果について，実際の授業場面を用いて実験的に検討した。被験者にとって未学習のテーマで全 5 回の授業を行い，毎回授業の最後にその日の授業で習った内容に関して 5 分間の確認テストを実施した。被験者は，確認テストが空所補充型テストで与えられる群と記述式テストで与えられる群にランダムに割り当てられた。ただし，最終回の授業では，両群とも，空所補充型問題 8 問と記述式問題 1 問から成る同一のテスト問題が与えられた。この最終確認テストの空欄補充型問題の得点について，群ごとに平均点を求め，対応なしの t 検定を行った。

第 1 群：空所補充型テスト群の最終確認テストの得点, mean=5.08, sd=1.93, n=24
第 2 群：記述式テスト群の最終確認テストの得点, mean=4.13, sd=2.09, n=24
対応なし, t=-1.65, $n.s.$, $\widehat{es}$=0.476

引用：豊田秀樹『検定力分析入門』（東京図書、2009）P64

- 授業最後の確認テストは穴埋式と記述式のどちらが学習効果が高いかを調査（記述式の方が高いと主張したかったと思われる）
- 穴埋式/記述式で確認テストを行った2つの群に対して、最後は同じテストを受けさせてその平均点の違いを見ている
- 穴埋式の平均点が5.08で、記述式が4.13という結果
- 前頁と同じ手法で、統計的有意差なしと結論（つまり、どちらの確認テスト方式も学習効果に差があるとは言えない）

3つの研究結果の違い

	平均の差	標準偏差	データ数	t値	P値	結論
その1	0.40	1.36、1.43	236、236	4.32	<0.001	高度に有意差あり
その2	0.51	1.76、1.26	84、69	2.06	<0.05	有意差あり
その3	0.95	1.93、2.09	24、24	−1.65	>0.05	有意差なし

- 面白いことに「平均の差」と「結論」の解釈が逆転しています。
  - 平均の差は、その3（0.95）＞その2（0.51）＞その1（0.40）
    結論は、その1（高度にあり）＞その2（あり）＞その3（なし）
  - 結論はt検定という統計手法を用いて、平均の差が誤差の範疇かどうかを確率計算をして、客観的に判定しています。
  - つまり、t検定を用いずに平均の差だけで見たのでは、結果を見誤る恐れがあるということです。
  - 次の章でt検定について詳しく解説します。

「統計手法を使う意義」まとめ

- 得られた測定データの平均が真の平均とは限りません。真の平均を得るのは不可能か、現実的ではないことが多いので、得られた（多くの場合少ない）測定データで判断するしか無いのです。
- 更に、ソフトウェアメトリクスは自然科学分野のデータ（長さ、重さなど）に比べて、人間の活動を測定している側面が大きいので、測定精度が悪いのは仕方ありません。
- 人間の行動を対象とする社会科学研究の例で、そのような測定基準も精度も曖昧なデータでも、統計手法を適用することで、研究成果として客観的な結論を導き出していることを示しました。
- ソフトウェアメトリクスを活用する場面で、「データが少ないから」、「測定精度が悪いから」使えないと嘆くのではなく、だからこそ統計手法を適用して客観的な結論を見出すことが重要です。

2群の母平均の差のt検定

３つの研究結果のように夫婦、男女、A方式／B方式のように2つの群に分けてデータを収集して、その平均に偶然とは言えない差があるかどうかを検証する統計手法です。
ソフトウェアメトリクスならば、例えば改善前後、品質問題有無といった形で2つに分けて、その違いを検証するといったことに適用できます。

3つの研究結果の違い（再掲）

	平均の差	標準偏差	データ数	t値	P値	結論
その1	0.40	1.36、1.43	236、236	4.32	<0.001	高度に有意差あり
その2	0.51	1.76、1.26	84、69	2.06	<0.05	有意差あり
その3	0.95	1.93、2.09	24、24	−1.65	>0.05	有意差なし

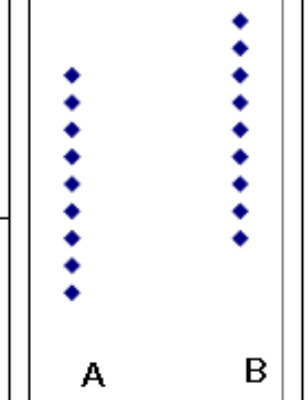
- 面白いことに平均の差と結論が逆転しています。
  - 平均の差は、その3（0.95）＞その2（0.51）＞その1（0.40）
    結論は、その1（高度にあり）＞その2（あり）＞その3（なし）
- 差を見るには2つの群の「平均の差」、「標準偏差」、「データ数」を総合して判定する必要があり、その総合指標をt値と呼びます。
  t値の絶対値が大きいほど差が大きいと捉えます。
- 推定誤差を加味しても意味の有る差かどうかをP値で判定します

4パターンの2群データ

パターン1

Data	
0.6	A
0.7	A
0.8	A
0.9	A
1	A
1.1	A
1.2	A
1.3	A
1.4	A
0.8	B
0.9	B
1	B
1.1	B
1.2	B
1.3	B
1.4	B
1.5	B
1.6	B

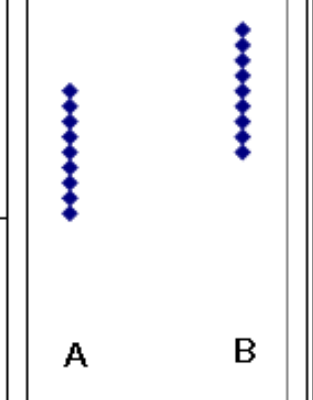
A: 平均 1.00, 標準偏差 0.27
B: 平均 1.20, 標準偏差 0.27



パターン2

Data	
0.8	A
0.85	A
0.9	A
0.95	A
1	A
1.05	A
1.1	A
1.15	A
1.2	A
1	B
1.05	B
1.1	B
1.15	B
1.2	B
1.25	B
1.3	B
1.35	B
1.4	B

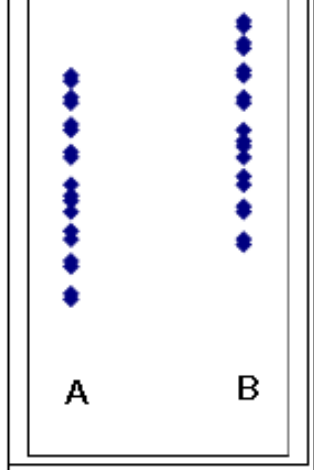
A: 平均 1.00, 標準偏差 0.14
B: 平均 1.20, 標準偏差 0.14



パターン3

Data	
0.6	A
0.7	A
0.8	A
0.9	A
1	A
1.1	A
1.2	A
1.3	A
1.4	A
0.58	A
0.72	A
0.83	A
0.94	A
0.96	A
1.12	A
1.22	A
1.32	A
1.38	A
0.8	B
0.9	B
1	B
1.1	B
1.2	B
1.3	B
1.4	B
1.5	B
1.6	B
0.78	B
0.92	B
1.03	B
1.14	B
1.16	B
1.32	B
1.42	B
1.52	B
1.58	B

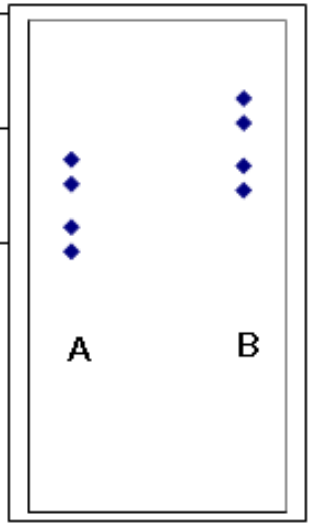
A: 平均 1.00, 標準偏差 0.27
B: 平均 1.20, 標準偏差 0.27



パターン4

Data	
0.85	A
0.93	A
1.07	A
1.15	A
1.05	B
1.13	B
1.27	B
1.35	B

A: 平均 1.00, 標準偏差 0.14
B: 平均 1.20, 標準偏差 0.14



これらの4パターンの2群データは全て平均の差が0.2です。
AとBの差がより顕著なのはどれでしょう？
または、同じですか？

t検定のt値とは

t検定は、単純に平均の差を比べるのではなくt値で差をみます。
概念的にはt値の計算式は以下の通りです。

d：平均の差
s：2群の標準偏差を合併（加重平均）したもの
n：2群のデータ数

$$t = \frac{d}{s/\sqrt{n}}$$

dが大きくなると分子が大きくなりtが大きくなる

$s/\sqrt{n}$ が小さくなると分母が小さくなりtが大きくなる

sが小さくなると分母が小さくなりtが大きくなる

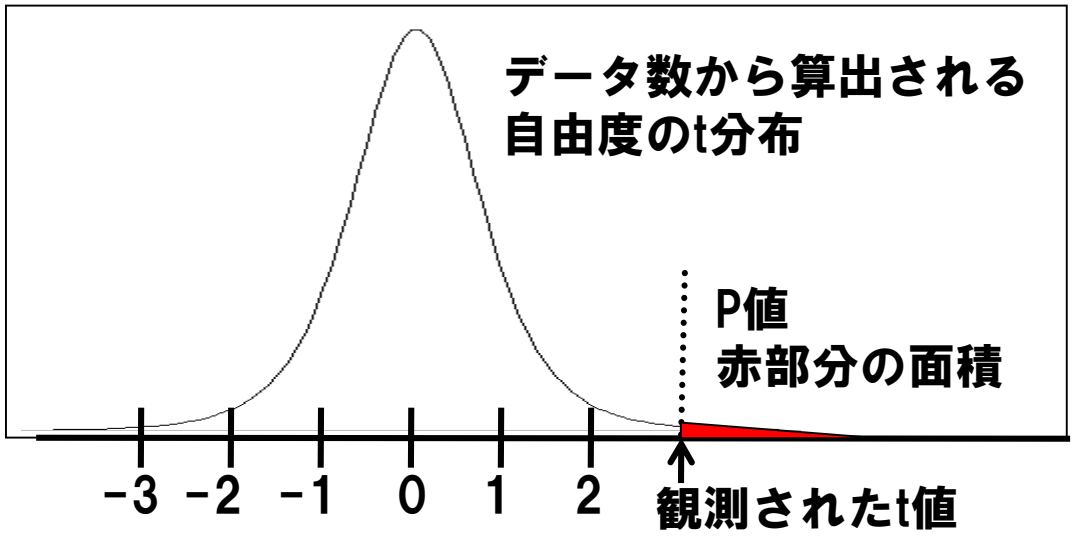
nが大きくなると分母が小さくなりtが大きくなる

標準誤差と呼ぶ
ソフトウェアメトリクスは測定誤差（sに相当）が大きく、データ（n）が少ないので使えないと言われる正体

t値（の絶対値）が大きければ、2つの平均に顕著な差があると判定しますが、その判定基準はどうするか？

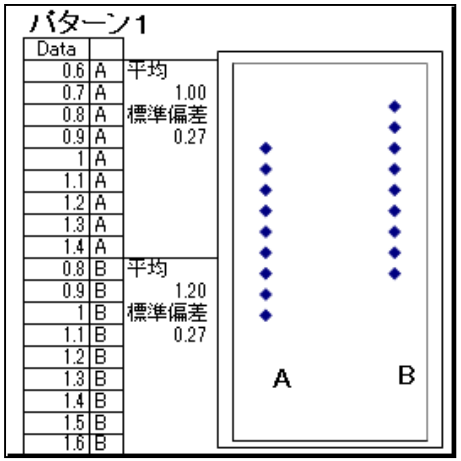
P値の小ささで判定

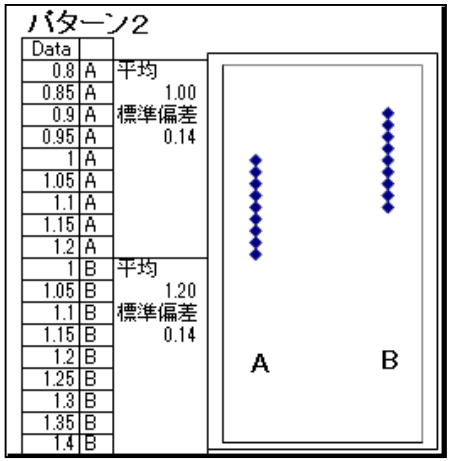
- 平均の差をベースに計算したt値が0に近い値ならば、2つの群の平均はほぼ等しいと見なせるし、0から離れた値ならば違っていると判断できます。
- その離れ具合を評価するのにP値を用います。P値が小さいほど離れていると言えます。
- 十分に離れている＝等しくない＝有意差あり
と判定する基準は慣例的に5%とします。

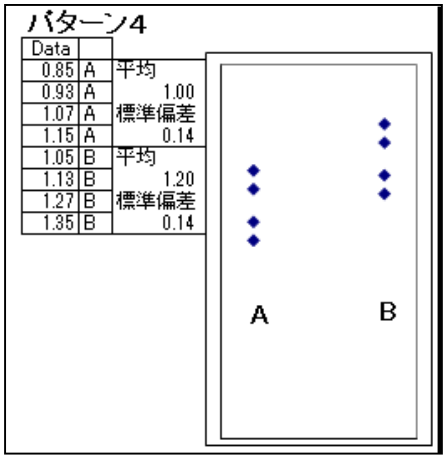


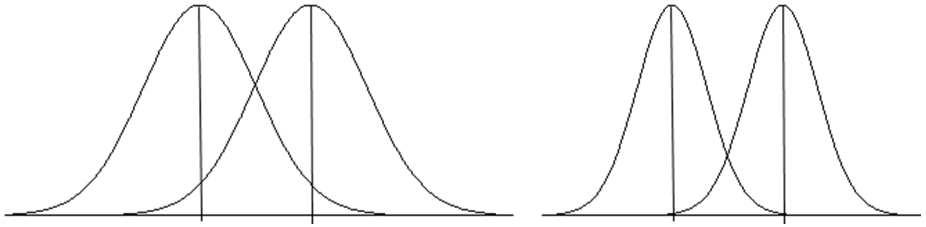
P値が小さいならば、分布から外れた値という意味になる。
つまり平均は等しくないと確信持って判断することができる

4つのパターンのP値の違い









ばらつきが小さいほど差が顕著（分離性高い）

データが多いほど差が顕著（信憑性高い）

	平均の差	標準偏差	データ数	t値	P値	結論
パターン1	0.2	0.27	9	1.55	0.070	有意差なし
パターン2	0.2	0.14	9	3.10	0.003	高度に有意差あり
パターン3	0.2	0.27	18	2.26	0.015	有意差あり
パターン4	0.2	0.14	4	2.09	0.041	有意差あり

「2群の母平均の差のt検定」まとめ

- 改善前/後、品質問題有/無、A方式/B方式といった形で、データを2つの群に分けて、その平均の差に誤差範囲ではない違いがあることを検証する場面で適用できます。
  （簡単に言えば、違いが「たまたま」or「そうではない」の検証）
- 2群の平均の差を評価するには以下の2要素も絡みます。
  - データのばらつき
    大きいほど違いがぼける（つまり、違いが小さい）
  - データ数
    多いほど平均の確からしさ（推定精度）が上がり、
    違いが明確になる（つまり、違いが大きい）
- 平均の差だけで一喜一憂するのでは、客観的な判断を見誤ることがあるので注意が必要です。
- t検定の原理をきちんと理解するのは難しいですが、まずはp値が5%以下になれば良いと覚えましょう。

リリース後品質を予見するメトリクス

『データ指向のソフトウェア品質マネジメント』3.2節

2つの品質メトリクスのどちらが、リリース後の品質と関連が強いかを母平均の差のt検定を活用して調べた事例です。

分析目的と使用メトリクス

- プロジェクト完了時に算出していた以下の2つのメトリクスが製品リリース後の品質に関連があることを検証すると同時にどちらの方が関連が強く品質指標として妥当かを確認する。
  - 不具合密度＝ST不具合数/新規開発行数
  - 不具合検出率＝ST不具合数/ST工数

※ST：システムテスト

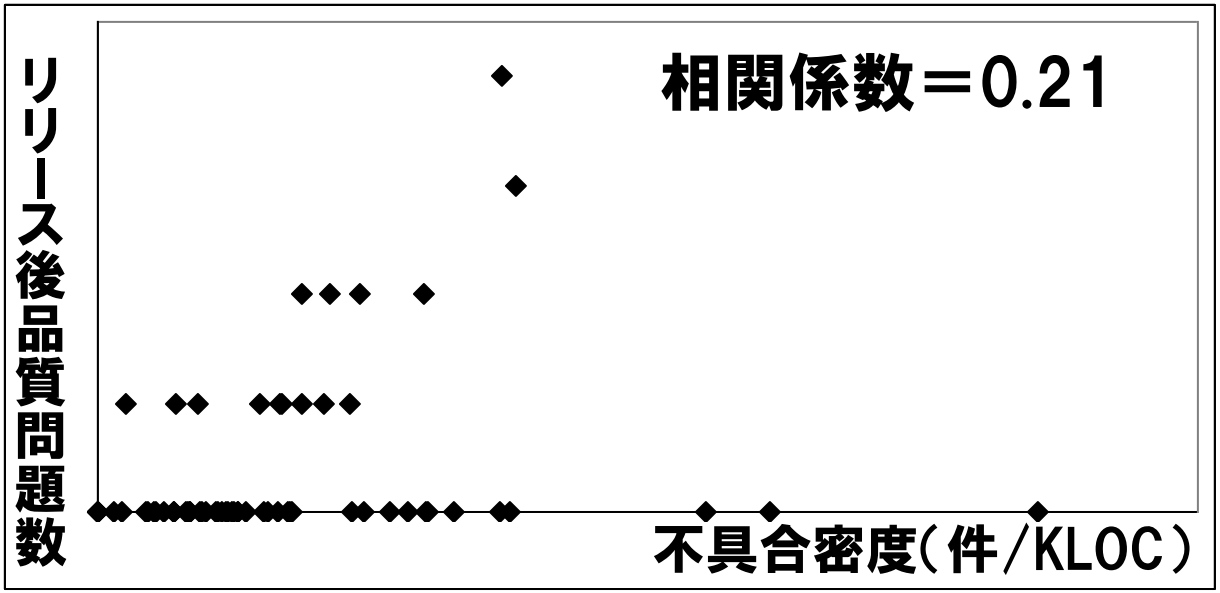
- 元々は、不具合密度を品質指標としていましたが、以下の理由で妥当ではないと感じていたことも背景となっています。
  - 不具合がテストが不十分で低いのか、品質が良くて低いかが区別できない
  - 信頼性工学では、MTBFなど時間と不具合の関係を見ることが一般的

用いたデータセット（抜粋）

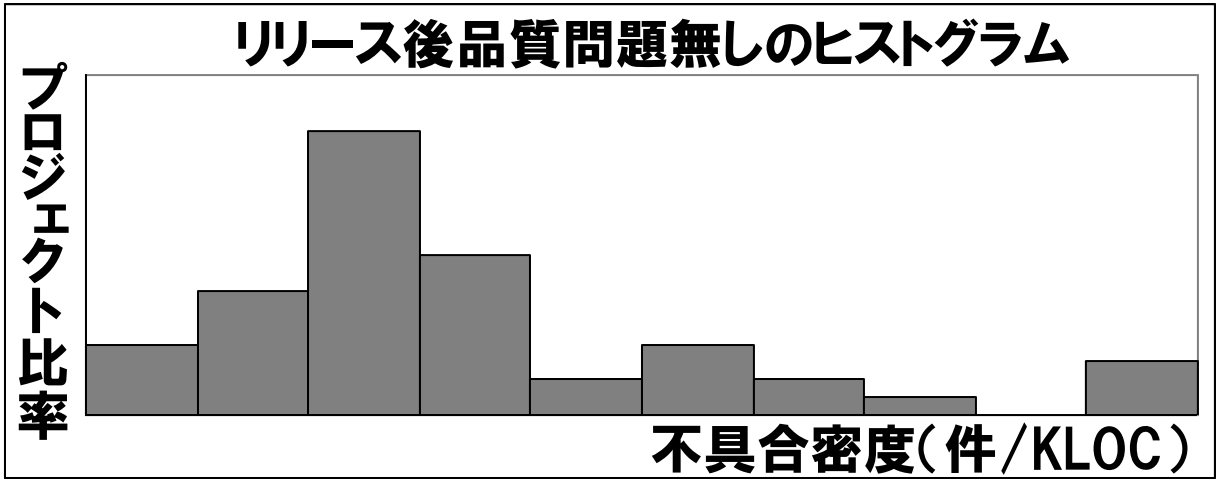
Project	不具合密度	不具合検出率	リリース後品質問題数	問題有無
Project01	0.97	0.07	0	無し
Project02	0.65	0.03	0	無し
Project03	0.00	0.02	1	有り
Project04	0.12	0.00	0	無し
Project05	0.46	0.03	1	有り
Project06	0.34	0.01	0	無し
Project07	0.04	0.05	0	無し
Project08	0.06	0.02	0	無し
Project09	0.13	0.00	0	無し
Project10	0.14	0.00	0	無し
Project11	0.29	0.02	0	無し
⋮	⋮	⋮	⋮	⋮

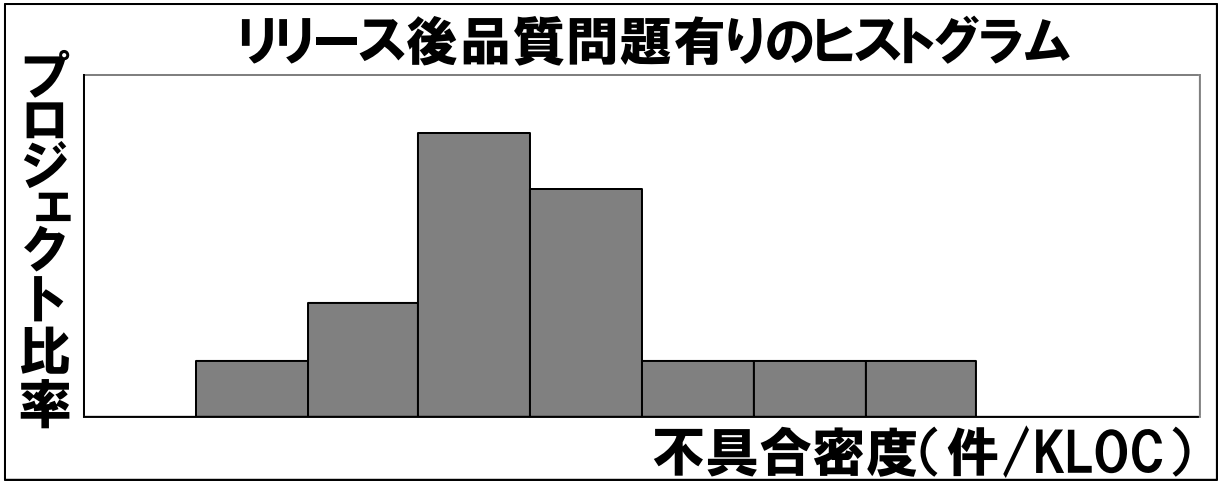
不具合密度と品質問題数の相関分析

- あくまでも本事例のデータでの結果に過ぎませんが、
  相関はそれほど高くは有りませんでした。
- ですが、全く関連が無いとは言い切れないので、
  （諦めずに！）違うアプローチを検討しました。
- データを品質問題が「無」と、1件でも発生したら「有」の2つに
  分けて、「無」と「有」のデータ群の分布の違いを見ることに。



品質問題有無の不具合密度分布の違い





- 無しの方が左に分布が寄っているが、右側にもデータがばらついており、明らかに差が有ると言えるかは微妙
- 母平均の差のt検定を行う

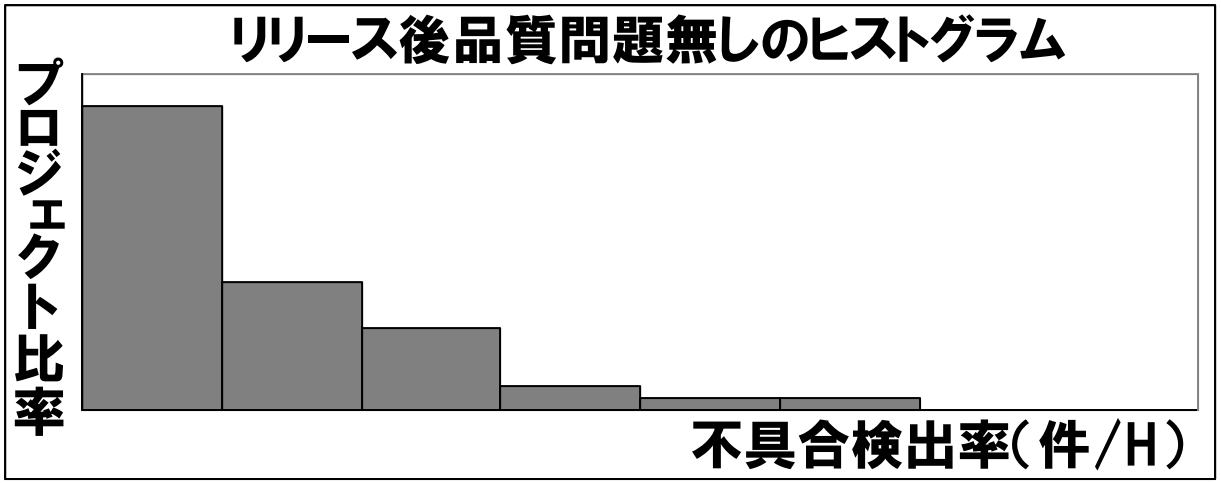
不具合密度の母平均の差のt検定

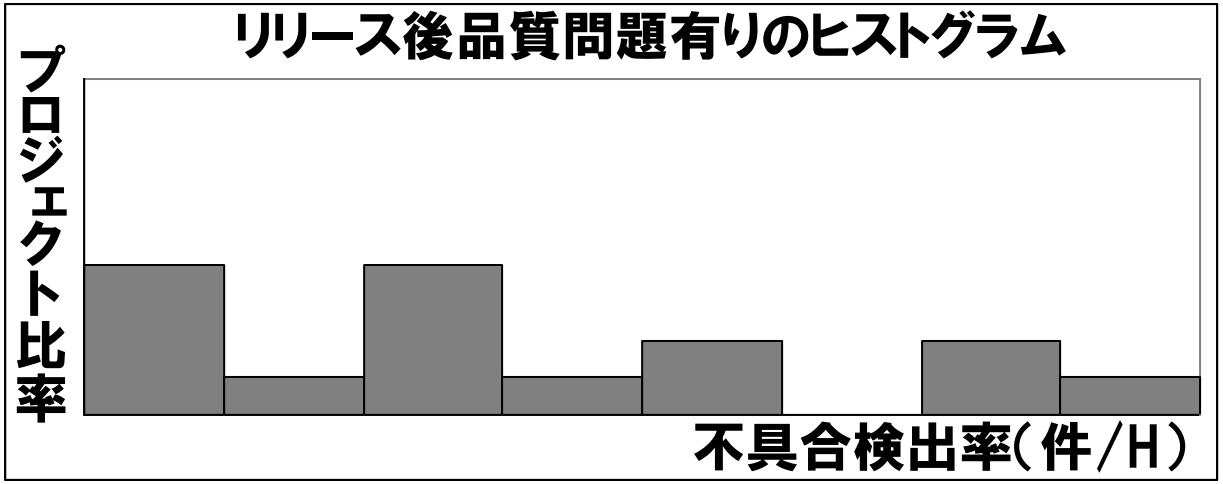
	平均値	自由度	標準誤差	t値	P値
品質問題無し	1.32	40	0.28	−0.93	**0.36**
品質問題有り	1.58				

- P値が0.36（36%）で有意水準5%よりも大きいため、平均値の差は統計的に有意差無しと判断、つまり品質問題無し/有りの分布の平均値に差があるとは言えないと判定します。
- 残念ながら、この分析結果では、不具合密度と品質問題有無の関連を示すことは出来ませんでした。

品質問題有無の不具合検出率の違い

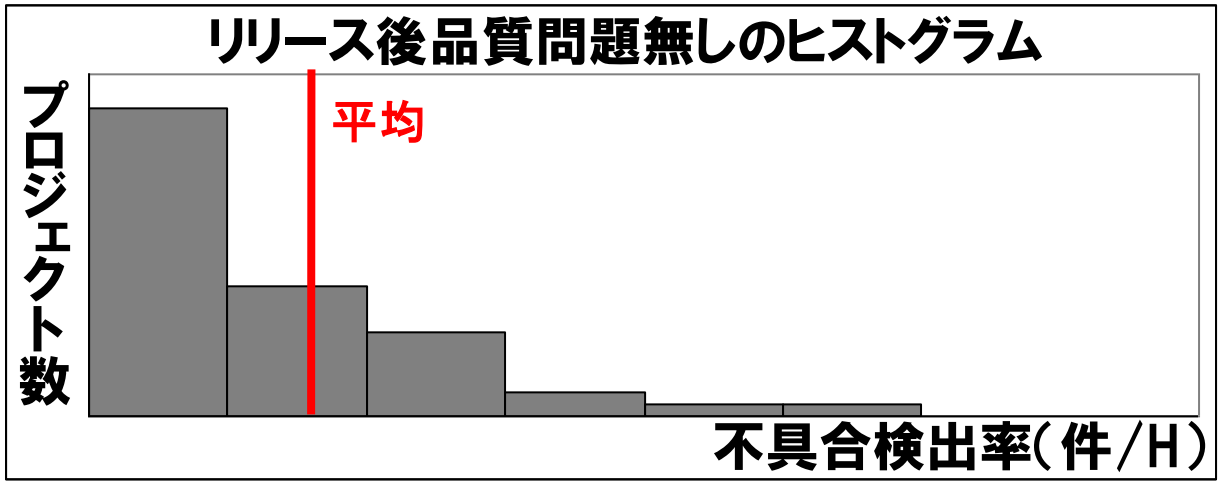
■ 次に不具合検出率で同様の分析を進めますが、相関分析は不具合密度と同様に傾向性が見られなかったので、省略しています。



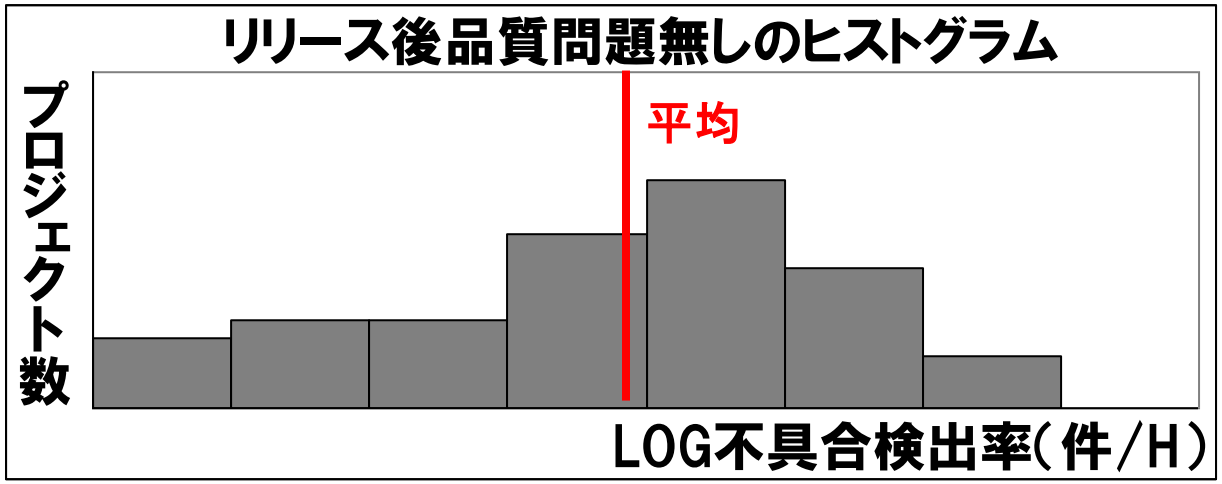


- 無しの分布が明らかに右に歪んでいます(左に偏り)
- このまま分析すると具合が悪いので、対数変換をします。

対数変換

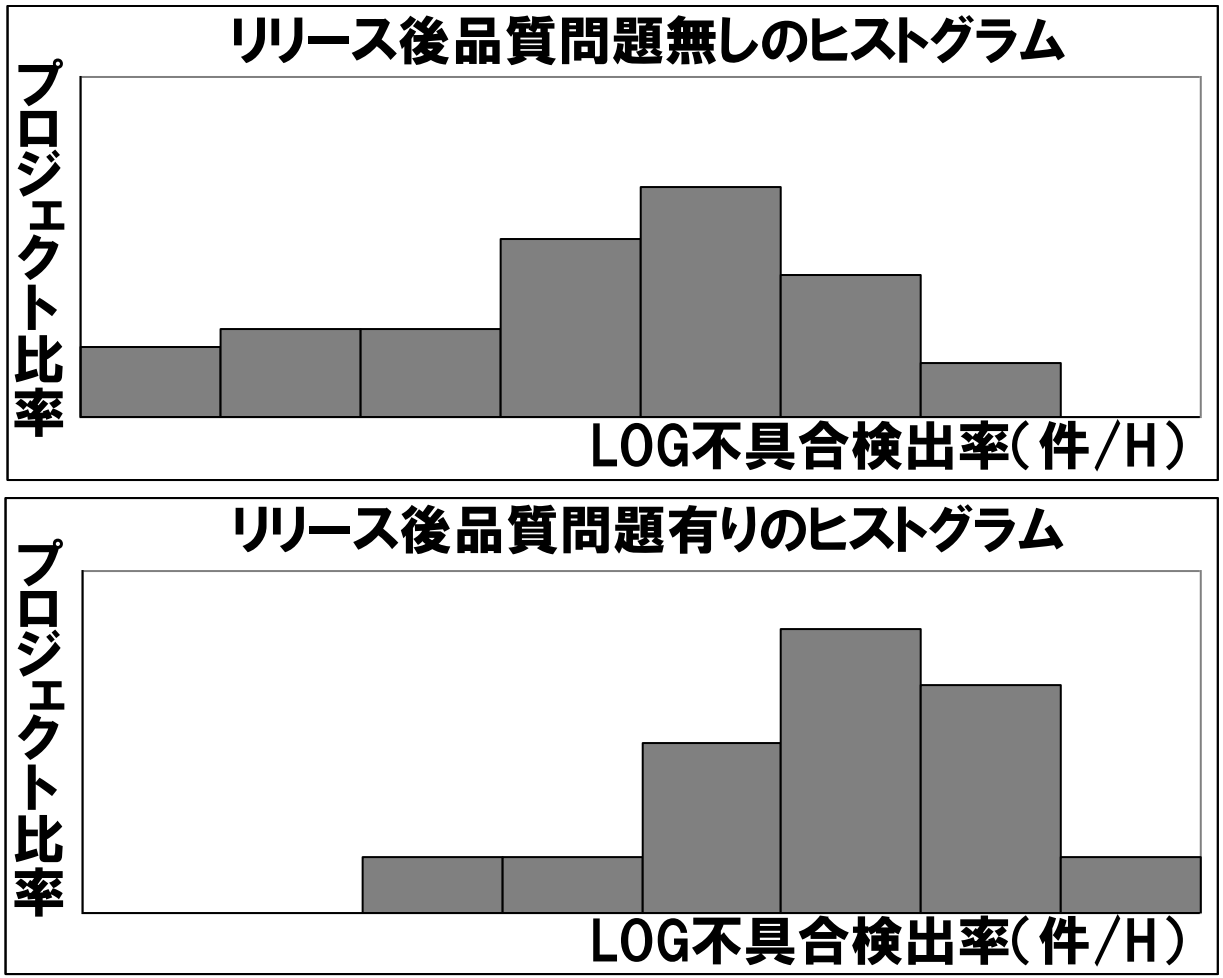


対数変換で ⇒ **偏りを補正**



- ソフトウェアメトリクスはレンジの広いデータを扱うことが多く、上図のように左に偏った分布になる場合があります。
- 偏った分布で平均を扱う分析を行うのは望ましくありません。
- このような場合に対数変換を施すことで、分布の偏りを補正することができます。

対数変換後不具合検出率の分布の違い



- 不具合密度の時よりも、無しの分布が左に寄った傾向を示しています。
- それでも、明らかに平均の差があるというほどではないので、やはり母平均の差のt検定を行うことに

対数変換後不具合検出率の母平均の差のt検定

	平均値	自由度	標準誤差	t値	P値
品質問題無し	−1.44	61	0.15	−3.31	**0.002**
品質問題有り	−0.95	※対数変換後の値での計算結果なので、平均値がマイナスとなります			

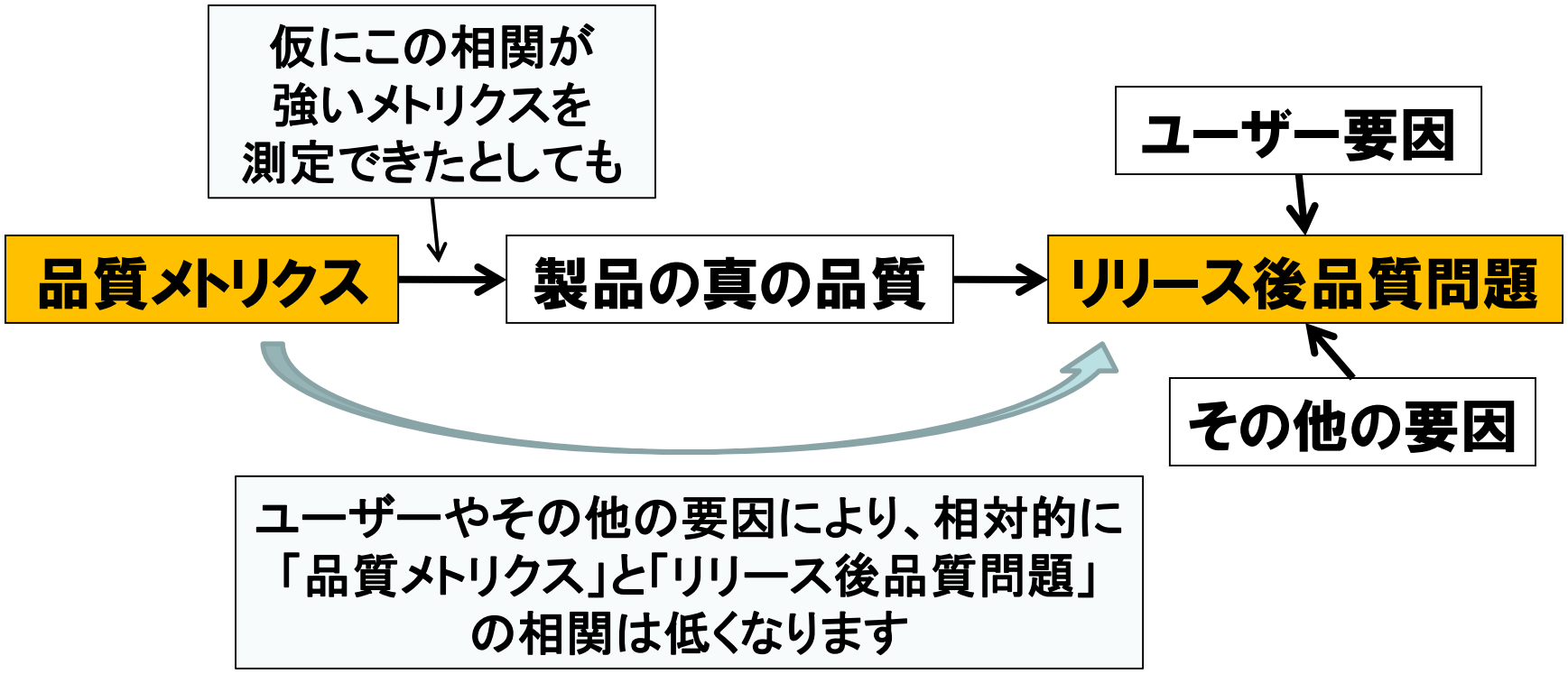
- P値が0.002（0.2%）で有意水準5%よりも小さいため、平均値の差は統計的に有意差有りと判断、つまり品質問題無し/有りの分布の平均値に違いがあると判定します。
- この結果、不具合検出率が大きい方がリリース後に品質問題を発生する可能性が高いことがデータにより支持されました。

本分析の結論

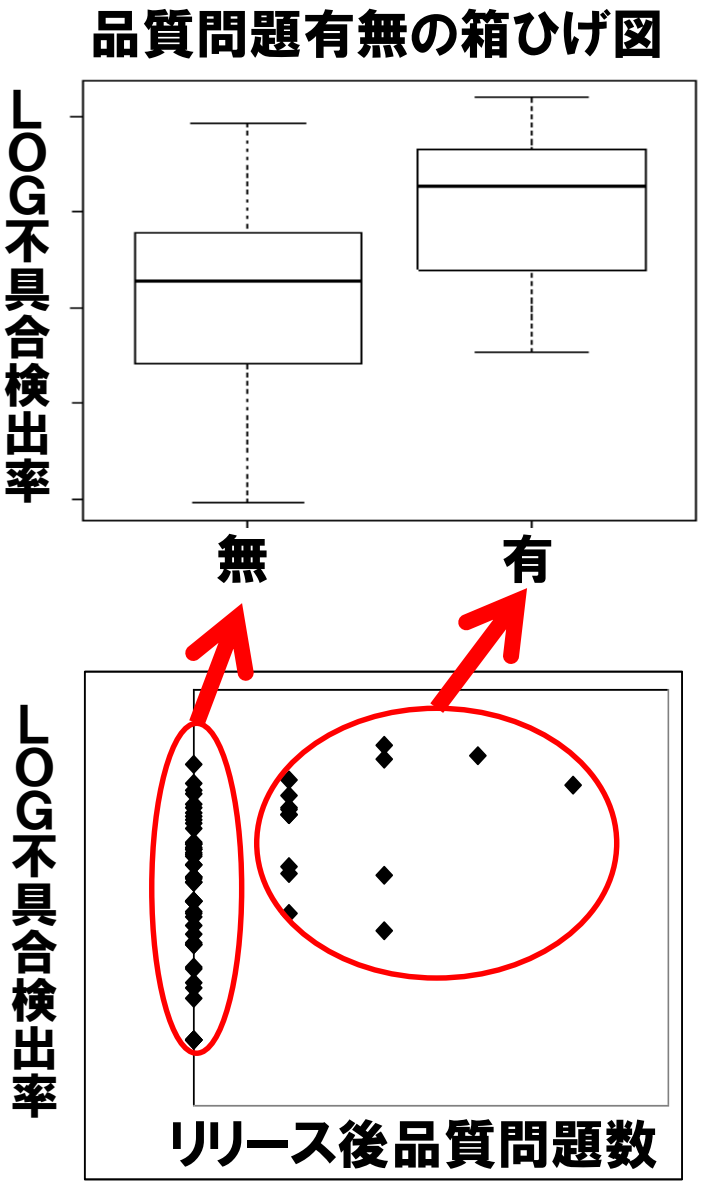
- 不具合密度と不具合検出率について、リリース後の品質問題数と関連性を分析しました。
- 不具合密度と不具合検出率ともに、リリース後の品質問題の件数との明確な相関を見出すことは出来ませんでした。
- しかしながら、リリース後に品質問題が無いプロジェクトの方が平均して不具合検出率が小さいことが統計的に証明されました。
- あくまでも平均的な結果なので、不具合検出率の大小だけで、リリース後に品質問題が有無を占うことは出来ませんが、不具合検出率が大きい方が品質問題が起きる可能性は高いと考えられます。
- 今後プロジェクト完了時の品質を判断するには、不具合密度よりも不具合検出率を見る方が有効であることが分かりました。

（補足）なぜ相関分析ではうまくいかない？

- リリース後に品質問題が発生するかどうかは、製品の品質だけに依らず、ユーザーやその他の要因の影響も大きいです。
- プロジェクト完了時の品質メトリクスにそれらの要因まで加味することは難しいです。



（補足）相関分析と母平均の差のt検定の違い



- 母平均の差のt検定はデータ群で比較するので、多少の誤差によるデータ変動では結果は変わりません。つまり、鈍感（ロバスト）です。

誤差の多いソフトウェアメトリクスは
よりロバストなアプローチを試みる

- 相関分析は個々のデータの動きを見るので、誤差などのデータ変動に敏感です。

「リリース後品質を予見するメトリクス」まとめ

- リリース後を予見するメトリクスとして、本分析結果では、不具合密度よりも不具合検出率の方が有効であることが分かりました。（一般論ではありませんので、留意してください）
- 不具合検出率とリリース後品質問題数の相関分析では、明確な関連性を示すことができませんでした。そこで、リリース後品質問題数を問題有/無の2つに分けて、2群の母平均の差のt検定を実施したところ関連性を示すことができました。
- 原因系（今回はリリース直前の不具合検出率）と結果系（今回はリリース後品質問題数）が、時間的、構造的に因果関係を見出すことが難しい場合に相関分析はうまく行かないことが多いです。
- 相関が無いから役に立たないと嘆くのではなく、手法の引き出しを増やし、色々なアプローチを試みることが重要です。