

ソフトウェアメトリクスの研究動向

大阪大学
大学院情報科学研究科
コンピュータサイエンス専攻
楠本真二
kusumoto@ist.osaka-u.ac.jp



Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

内容

- ソフトウェアメトリクス
- ソフトウェアメトリクスの分類と例
- 研究動向
 - ▶ International Symposium on Software Metrics (ソフトウェアメトリクスに関する国際シンポジウム)を中心に
- メトリクスを用いた研究事例



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

ソフトウェアメトリクス

● 定義[1]

- ▶ ソフトウェア開発プロセス, 資源, 成果物の実態がソフトウェアを開発するという点から見て適当であるかどうかを計測し, 予測するための手段

● 目的

- ▶ 現在開発中のソフトウェアや進行中のプロセスの品質が, 開発標準と比べて高いか, 低いかを定める.
- ▶ 将来の品質が, 開発標準と比べて高いか, 低いかを予測する.
- ▶ ソフトウェア品質と開発プロセスの効率を改善する.

[1] R.G. Reynolds and E. Zannoni: Metrics, Encyclopedia of Software Engineering, Wiley-Interscience (1994).



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

様々な意味

- ソフトウェア工学・メトリクスの文献において, “metrics”という用語は, 様々な意味で用いられている.

- ▶ Number(数値)
- ▶ Scale(尺度)
- ▶ Attribute(属性)
- ▶ Model(モデル)

N. E. Fenton, “*Software Metrics: A Rigorous Approach*,” Chapman & Hall, 1991.



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

メトリクス: 数値

- 数値 (A *number* derived from a product, process or resource)
 - ▶ ソフトウェアプロダクト, プロセス, 開発に要する資源から得られる具体的な数値.
 - ▶ 例: プログラム行数, ファンクションポイント数, リリース後の故障数. 一月当たりの開発行数



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

メトリクス: 尺度

- 尺度 (A *scale* of measurement)
 - ▶ 計測のものさし, 分類基準.
 - ▶ 例: 混入工程に基づくバグ分類. 重大さに基づく故障分類

重大度	記述
1	System is unusable for any purpose.
2	Important areas of functionality are unavailable.
3	System features and functionality are unavailable but a work around is achievable.
4	No immediate impact on users e.g. general systems advice.
5	Situation where cosmetic change has been reported.

D. Lawson and G. Coleman: "Investigating software measures to improve product reliability", Proc.of the 2002 ACM symposium on Applied computing, pp. 1031 - 1035 (2002).



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

尺度の種類

- 比例尺度 ratio scale
 - ▶ 数値の差と共に数値の比にも意味がある
- 間隔尺度 interval scale
 - ▶ 数値の差のみに意味がある
- 順序尺度 ordinal scale
 - ▶ 順序のみに意味がある
- 名義尺度 nominal scale
 - ▶ 観察される変数と数値を対応させる(分類として記号の意味を持つだけ)
- 下位の尺度データに適用できる全ての統計手法は, 上位の尺度データにも適用できる. ただし, 逆は成り立たない.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

メトリクス: 属性

- 属性 (An identifiable *attribute*)
 - ▶ プロダクト, プロセス, あるいは開発に要する資源を識別できる性質.
 - ▶ 例: モジュール間結合度, ソフトウェア信頼性 (一定期間内に故障を起こさない確率)
 - モジュール間結合度
 - 非直接結合
 - データ結合
 - スタンプ結合
 - 制御結合
 - 外部結合
 - 共通結合
 - 内容結合



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

メトリクス: モデル

- モデル (A theoretical or data driven *model* describing a dependent variable as a function of independent variables, with the intention that the model is used for prediction purposes)
 - ▶ プロダクト, プロセス, あるいは開発に要する資源の間関係を表す理論的モデル, あるいは実測データに基づいて構築されたモデル.
 - ▶ 例: ソフトウェア信頼度成長モデル, COCOMOモデル

$$E = a(Size)^b$$
$$D = cE^d$$

organicモード: a=2.4, b=1.05, c=2.5, d=0.38

embeddedモード: a=3.0, b=1.12, c=2.5, d=0.35

semi-detachedモード: a=3.6, b=1.20, c=2.5, d=0.32



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

計測対象に基づくメトリクスの分類

- 成果物(product)
 - ▶ 開発の結果得られたもの
 - ▶ ドキュメント, プログラム, テストデータ等
- 資源(resource)
 - ▶ 開発作業の入力となったもの
 - ▶ ハードウェア, ソフトウェア, ドキュメント, 人的資源, 知識等
- プロセス(process)
 - ▶ 成果物を作り出すための作業
 - ▶ ライフサイクルの全てのフェーズ(要求分析, 設計, コード生成, テスト, 品質保証等)



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

内部属性と外部属性

● 内部属性

- ▶ 計測対象の内部的な状態．直接計測可能な場合が多い．
- ▶ 例：プログラムの行数

● 外部属性

- ▶ 計測対象とそれが置かれている環境との関係から得られる性質．
- ▶ 直接計測は難しいが，内部属性によって間接的に計測されることが多い．
- ▶ 例：プログラムの理解容易性，再利用性



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

成果物メトリクス(1)

- これまでに提案されている大部分のメトリクス
- プログラムコードを対象とするものが多い
 - ▶ 記録される成果物がほとんどコードのため
 - ▶ ツールの普及により他の成果物を対象とするものも増加
- コードメトリクス
- 設計メトリクス



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

コードメトリクス

- 行数
 - ▶ LOC
 - ▶ 定義は数多く存在する
- Software Science by Halstead
 - ▶ ソースコード中に含まれる演算子(operator)と非演算子(operand)の種類数, 出現回数からプログラムの大きさ, プログラミング労力等を計測する.
- 複雑さメトリクス
 - ▶ データ構造やデータフローの複雑さ:
 - Span: 同一識別子が参照される距離
 - ▶ 制御構造の複雑さ
 - サイクロマチック数



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

設計メトリクス

- モジュールの結合度
 - ▶ 非直接, データ, スタンプ, 制御, 外部, 共通, 内容
- モジュール凝集度
 - ▶ 機能的, 情報的, 連絡的, 手順的, 時間的, 論理的, 暗号的
- ファンクションポイント
- C&Kメトリクス
 - ▶ オブジェクト指向設計に対する複雑さメトリクス
 - ▶ クラス内部, 継承, 結合の点から評価する



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

プロセスメトリクス・資源メトリクス

- プロセスメトリクス
 - ▶ 時間, コスト工数

 - ▶ プロセス実行時に発生するイベント数
 - 設計変更, コード修正, 欠陥発見
 - ▶ 時間, 工数の計測は難しい
 - 開発者は複数プロジェクトに携わっている
- 資源メトリクス
 - ▶ 人的資源
 - ▶ 生産性
 - 投入された工数と結果として得られた出力



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

メトリクスの選択

- トップダウンアプローチ
 - ▶ ソフトウェア開発プロジェクト全体の目的の設定と共に行われるべき.
 - ▶ メトリクスは, 目的が達成されているかどうかを評定するための道具
 - ▶ 抽象度の高い目的とメトリクスの関連を決める必要がある.
 - GQMパラダイム
- ボトムアップアプローチ
 - ▶ 計測可能なメトリクスを全て計測する.
 - ▶ 評価したい特性との関連を調べる.
 - ▶ 関連のあったメトリクスのみを計測する.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

GQMパラダイム

- V. R. Basiliらによって提案された総合的なソフトウェア計測の枠組み
 - ▶ 計測目的とメトリクスの対応関係が明確であり、何のためにデータを収集するのかの確認が容易である。
 - ▶ データ収集のよい動機付けとなる。
- ▶ GQM関連Webページ
 - <http://irb.cs.uni-magdeburg.de/sw-eng/us/java/GQM/>

V. R. Basili and D. M. Weiss: A methodology for collecting valid software engineering data, *IEEE Transactions on Software Engineering*, Vol.SE-10, No.6, pp.728-738, 1984.



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

計測原理

- 計測はトップダウンで行われるべきである。
 - ▶ まず計測の目標があって、その目標を遂行するために尺度が定義され、計測が行われなければならない。
- データ分析は何らかの目的や仮説に基づいて行われるべきである。
 - ▶ 例えば、コスト予測の改善に役立つのか、それとも、コストを評価するのか、その目的を明確にした上で分析を行わなければならない。

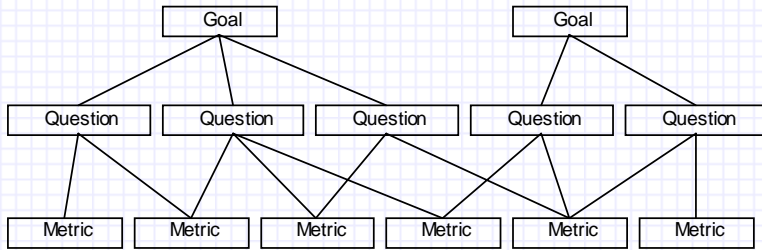


2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

GQMモデル

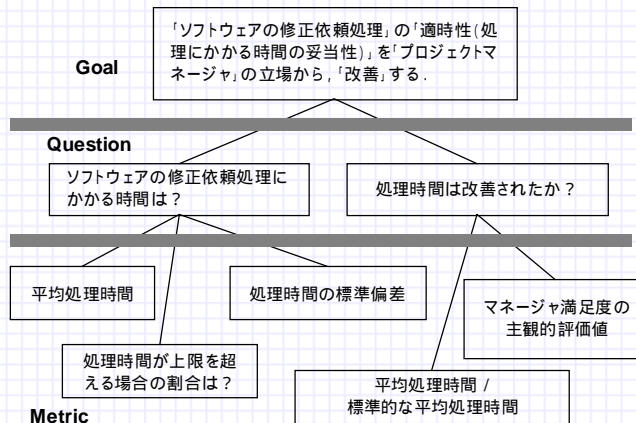


2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

GQMモデルの例



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Goal(目標:概念レベル)

- 計測の目標を
 - ▶ 計測対象
 - プロダクト
 - 要求仕様書, 設計ドキュメント, プログラム, テストデータ, ...
 - プロセス
 - 仕様作成, 設計, コーディング, テスト, ...
 - 資源
 - 作業員, ハードウェア, ソフトウェア, 作業空間, ...
 - ▶ 計測理由
 - ▶ 品質モデル
 - ▶ 視点
 - ▶ 環境

に基づいて明確にしたもの。



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Question(質問:操作レベル)

- 特定のGoalを評価,あるいは,達成する方法を明確にしたもの。
- 計測対象(プロダクト, プロセス, 資源)の特性を品質の観点から明らかにすることができる。



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Metric (尺度: 量的レベル)

- Questionに定量的に答えるためのデータの集合.
- データは, 客観的データと主観的データに分かれる.
 - ▶ 客観的データ: 計測対象のみに依存し, 計測の観点には依存しないデータ.
 - 例: ドキュメントのバージョン数, 作業工数, プログラムサイズ, ...
 - ▶ 主観的データ: 計測対象と視点の両方に依存するデータ.
 - 例: テキストの信頼性, 顧客満足度のレベル, ...



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

GQM適用プロセス

1. Goalの設定
2. Questionの生成
3. Metricの明確化
4. データ収集法の開発
5. データの収集, 妥当性確認, 分析
6. データの事後分析



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

Goal作成用テンプレート(タイプA)

- 目的: 「計測対象(プロダクト, プロセス, 資源): 設計ドキュメント, テスト工程, 保守担当者, ...」の「理由: 特徴付け, 評価, 予測, 動機付け, 改善, ...」を行うために,
 - 観点: その「焦点: コスト, 正しさ, バグ率, 変更回数, 信頼性, 使いやすさ, 適時性, ...」を「視点: ユーザ, 管理者, 開発担当者, 開発組織, ...」の立場から,
 - 環境: 「コンテキスト: パイロットプロジェクト, 実プロジェクト, 開発現場, 実験室, ...」において分析する.
- 例:
- ▶ 目的: 「設計ドキュメント」の「特徴付け」を行うために,
 - ▶ 観点: その「正しさ(ユーザ要求を正確にもれなく実現しているかどうか)」を「ユーザ, 開発者, およびテスト実施者」の立場から,
 - ▶ 環境: 「我々の開発環境」において分析する.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

Goal作成用テンプレート(タイプB)

- 対象: 「ソフトウェアの修正依頼処理, システムテスト, 最終プロダクト, 設計者, ...」の
 - 論点: 「コスト, 正しさ, バグ率, 変更回数, 信頼性, 使いやすさ, 適時性, ...」を
 - 視点: 「ユーザ, 管理者, 開発担当者, 経営者, ...」の立場から,
 - 目的: 「特徴付け, 評価, 予測, 動機付け, 改善, ...」する.
- 例:
- ▶ 対象: 「ソフトウェアの修正依頼処理」の
 - ▶ 論点: 「適時性(処理にかかる時間の妥当性)」を
 - ▶ 視点: 「プロジェクトマネージャ」の立場から,
 - ▶ 目的: 「改善」する.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

Goal設定に必要な情報

- 組織の方針や戦略 「論点」や「目的」
- プロダクト, プロセス, 組織の定義 「対象」
- 組織のモデル 「視点」



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Questionの主要パターン

- 「対象(プロダクト, プロセス, 資源)」そのものを「目的」から見て明確にするための質問。
 - ▶ ソフトウェアの修正依頼に対する現在の処理速度は？
- 「対象」の属性を「観点(視点)」から見て明確にするための質問
 - ▶ 修正依頼の実際の処理時間は, 予想とどれくらい食い違っていますか？
- 「対象」の特徴を「観点(視点)」から見て評価するための質問
 - ▶ 修正依頼の処理能力はプロジェクトマネージャから見て満足のいくものですか？



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Metric設定上の注意点

- 既存データの量と質
 - ▶ 信頼できる既存データが存在する場合には、できるだけ利用する。
- 計測対象の成熟度
 - ▶ 形式的に定義でき計測方法もある程度確立されている対象に対しては客観的尺度を用いる。
 - ▶ 形式的な定義が現段階では難しく、計測方法も確立されていない対象に対しては、主観的尺度を用いる。
- 学習
 - ▶ GQM モデルは常に洗練し、さまざまな場合に適応させていく必要がある。特定のQuestionに答えるためだけの尺度でなく、GQMモデルの信頼性評価にも役立つ尺度を用いる。



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

メトリクスの研究動向

- International Symposium on Software Metrics(METRICS)
- 主催: IEEE-CS TCSE
- 1993年から開催
- 今年で10回目
- 出席者は70 ~ 200人
 - ▶ 180人 100-120人 75人(2004)
- 他の会議と併設されることが多い
 - ▶ ICSE, ISSRE, ICSM, ASMA



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

1993年(1)

- Case studies
 - ▶ Reuse, Measurement program
- Code and maintenance metrics
 - ▶ Oometrics
- Metrics framework
- Future directions
 - ▶ Reuse, Dynamic comp.

1996年(3)

- Theoretical foundations
- Experience report
- Object oriented metrics
- Empirical studies
- Software reliability
- Testing



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

1997年(4)

- Process Improvement:
- Theory and Methodology
- Metrics Applied to Testing
- Application/Phase/Paradigm Specific Measurement
- Uses of Software Measurement

1998(5)

- Architectural Metrics
 - ▶ Code clone
- Software Process Improvement
 - ▶ CMM, PSP
- Error Models and Measures
 - ▶ Web-based review
- Object-Oriented Measures
 - ▶ Coupling, cohesion, FP
- Program Management Measures
- Formal Models
- Testing Measures
 - ▶ Inferring change effort



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

1999年(6)

2001年(7)

- Setting up Software Measurement Programs
- Studies of Object-Oriented Systems
- Project Management
 - ▶ FP
- Novel Measurement Approaches
- Data Analysis Methods
- Maintenance Studies
- COTS Studies
- Technology Evaluation
 - ▶ Clone, Inspection
- Ethics

- Software Effort and Cost
- Software Inspections
- Software Projects
- Software Modules
- Testing
- Maintenance and Evolution
- Software Measurement Infrastructure
- Object-Oriented Systems and COTS
- Industrial Application and Experiences
- Prediction



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

2002年(8)

2003(9)

- Software Project Prediction
- Measurement for Object-Oriented Systems
- Maintenance and Evolution
- Software Inspections
- Web and Network Technology
- Models and Abstractions
- Fault Prediction
- Methodological Issues

- Web Development
- Design Patterns
- Estimation
- V & V
- New Metrics
- Measurement Frameworks
- Empirical Research
- Management
- Prediction
- Maintenance



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

2004年 9/14-16(Chicago)

- General Chair:
 - ▶ Prof. Linda Ott, Michigan Technological University
- Program Chairs:
 - ▶ Prof. Martin Shepperd, Bournemouth University
 - ▶ Dr. Audris Mockus, Avaya Labs Research
- 採択率: 50% (36/71)
- 新しい試み
 - ▶ Late Breaking Papers



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

セッションタイトル

- Studies of Maintenance and Evolution
- Theory and Measurement
- Component-based Software Engineering
- Data Analysis Methods
- Advances in Empirical Software Engineering
- Project Management
- Empirical Studies of Inspections
- Software Size Measurement
- Assessment
- Effort Prediction for Web Development
- Defect analysis



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

Open Source Software Development: A Case Study of FreeBSD

Trung Dinh-Trong, James M. Bieman

● 目的

- ▶ オープンソース開発の特徴調査(Apach, Mozillaの調査結果に対する追証実験)

● 仮説

- ▶ コアメンバー(10 ~ 20人)が存在する .
- ▶ 作業の調整のために形式的な取り決めがある
- ▶ コアメンバーより多くのcontributerが存在する .
- ▶ 商用システムよりも品質が高い
- ▶ etc.

● 評価方法

- ▶ Core team memberへのアンケートとCVS, E-mail, Bug report データの利用



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンビカルソフトウェア工学研究会

Applying Function Point to Unified Modeling Language: Conversion Model and Pilot Study

G. Cantone, D. Pace, G. Calavaro

● 目的

- ▶ UML仕様書からのファンクションポイントの自動計測

● 手段

- ▶ ユースケース図, クラス図, シーケンス図より, FPを計測
 - ユースケース図: ユーザと計測境界
 - クラス図: データファンクション
 - シーケンス図: トランザクションファンクション

● 評価結果

- ▶ CFPSの計測結果とほぼ等しい(61 vs. 64)
- ▶ 但し, 分類はかなり違う.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンビカルソフトウェア工学研究会

A Controlled Experiment for Evaluating a Metric-Based Reading Technique for Requirements Inspection

B. Bernárdez, M. Genero, A. Durán, M. Toro

● 目的

- ▶ 著者らが提案しているMetric-based reading 技術(種々のユースケースモデルから計測されるメトリクス値に基づいてレビューすべき箇所を特定する)の評価

● 手段

- ▶ GQMパラダイムを用いた実験設定
 - Analyze: チェックリストとMBR, Purpose: 評価, Wrt.: 効率と効果, Viewpoint: 研究者, Context: 学生実験
 - 準備, 実行, データ検証

● 結果

- ▶ 効果(発見バグ数/総バグ数) $MBR > CBR$
- ▶ 効率(発見バグ数/時間) $MBR = CBR$



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

基調講演(1)

● Empirically Evolving Software Engineering Techniques by Victor R. Basili

- ▶ Improvement paradigm
- ▶ GQM paradigm
- ▶ Experience factory
- ▶ 適用事例



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

基調講演(2)

● Systematic Reviews by Barbara Kitchenham

- ▶ 定義: 明確に定義された疑問について, 関連する研究を特定し, 選択し, 批判的に吟味し, 得られるデータを解析してまとめる手法
- ▶ 目的: 対象となる疑問についての現在利用可能な全ての研究データをまとめること
- ▶ 特徴: データを収集するという膨大な努力, 論文の批判的評価, 事前に定められた質評価基準を満たす研究を統合して結論が導かれる.



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

主なテーマ

- Inspection
- Object-oriented metrics
- Size measurement
- Maintenance
- Project management

- 既存のメトリクスのカスタマイズ
- 評価手法の重視
 - ▶ コンテキスト, Validity



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

パネル

- Now We are Ten: Our 10th Symposium
 - ▶ James Bieman, Lionel Briand, Jyrki Kontio, Marvin Zelkowitz

 - ▶ 貢献
 - Validationの重要性
 - コミュニティ創成
 - ▶ 問題
 - コミュニティの成長が頭打ち
 - 他の国際会議との競合 (ISESE)
 - 企業からの参加が少ない



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

ISESE2004

- Software Changes and Evolution
- Software Testing
- Programming Practices
- Defect Management
- Infrastructure and Tool Support for Empirical Studies
- Cost Estimation
- Software Requirements
- Predictive Models
- Reading Techniques for Inspections
- Empirical Studies Methodology



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

METRICS2005

- 19-22 September, 2005, Como, Italy
- General Chair
 - ▶ Sandro Morasca, Università degli Studi dell'Insubria
- Program Chairs
 - ▶ Filippo Lanubile, University of Bari
 - ▶ Carolyn Seaman, University of Maryland Baltimore County
- 原稿締切
 - ▶ Abstract: Feb. 23, 2005
 - ▶ Paper: March 15, 2005 (ISESE April 5) (1/3が共通PCメンバー)



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

メトリクスを用いた研究事例

- 凝集度・結合度を用いたモジュール評価
 - ▶ 構造化設計法の導入によるモジュール凝集度・結合度の改善
- レビュープロセス評価
 - ▶ テストコストの削減率に基づくレビュー効果
- プロセス改善
 - ▶ 組込ソフトウェア開発プロセスの改善
- faultyクラスの予測
 - ▶ メトリクスを利用したクラスのfault-proneness評価
- ファンクションポイント計測
 - ▶ 種々のプロダクトからのFP自動計測



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

Fallacy on measurement (1)

- You can't manage what you can't measure. -

- 計測できないものはマネジメントできない。
- 成熟度の高い組織以外では、計測が無視されることが多い。
 - ▶ マネジャーがメトリクスの重要性を認識していない。
 - ▶ 計測が難しい。
- 必要項目を計測するためのコスト(SEL)
 - ▶ プロジェクト全体の7-9%(3% データ収集と検証, 4-6% データ処理と分析)

Robert L. Glass: "Facts and Fallacies of Software Engineering", Addison Wesley (2002).



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Fallacy on measurement (2)

- You can't manage what you can't measure. -

メトリクス使用頻度top10とbottom5

- | | |
|------------------------|------------------------|
| ● Top10 | ● Bottom5 |
| ▶ 出荷後のバグ数 (61%) | ▶ モジュール複雑度 (24%) |
| ▶ 変更,あるいは,変更要求の数 (55%) | ▶ 出荷したコード行数 (22%) |
| ▶ 顧客満足度 (52%) | ▶ ドキュメントのサイズ・複雑性 (20%) |
| ▶ 開発中の抽出バグ数 (50%) | ▶ 再利用コード行数(16%) |
| ▶ ドキュメントの完全性/正確性 (42%) | ▶ ファンクションポイント(10%) |
| ▶ バグ特定/修正時間(40%) | |
| ▶ 種類別バグの分布(37%) | |
| ▶ 機能毎のバグ数(32%) | |
| ▶ テストにおける仕様カバレッジ(31%) | |
| ▶ コードのテストカバレッジ(31%) | |



2004/12/14

第4回エンビカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Fallacy on measurement (3)

- You can't manage what you can't measure. -

- 経験豊富なマネジャーは、定性的に物事を把握してうまくマネジメントしている。
- データに基づいてマネジメントする方が、何も無いよりは正確にできる。
 - ▶ 様々なスポーツ
 - ▶ 株価
- 計測がソフトウェアのマネジメントで重要な要素であることは本当。
- 正しくは、“You can't control what you can't measure.”
 - ▶ Management = プロジェクトをうまく成し遂げるために行う作業(planning, organizing, staffing, directing, controlling)
 - ▶ Controlling = プロジェクトに影響を与えるなんらかの事象が生じた場合に、適切に対処すること、それに伴う作業。

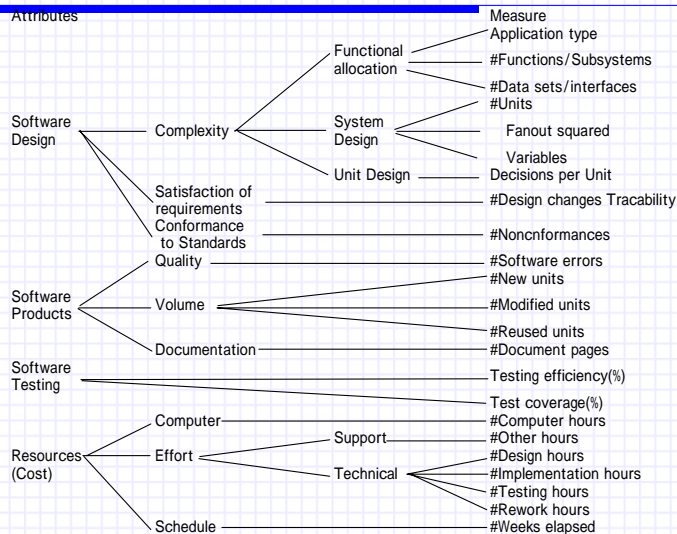


2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

Basic software measurement set



D. Card: Measuring Software Design Quality, Prentice Hall(1990).



2004/12/14

第4回エンピリカルソフトウェア工学研究会

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

まとめ

● ソフトウェアメトリクス

- ▶ ソフトウェアとその開発 / 利用過程を対象とした (定量的評価) 尺度 .
- ▶ 利用者の求める「機能」「性能」「品質」をソフトウェアが備えているかどうかを定量的 , かつ , 客観的に明らかにための道具 .
- ▶ 利用目的をトップダウンに議論するならGQMパラダイムなどが参考になる .
- ▶ ソフトウェアとその開発プロセスのモデル , データ収集システムなども必要となる .



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会

参考文献

- 松本健一 : “ソフトウェアメトリクス” , EPM研修資料(2004).
- 井上克郎 , 松本健一 , 飯田元 : ソフトウェアプロセス , 共立出版 (2000).
- Encyclopedia of Software Engineering, Wiley-Interscience (1994).
- N. E. Fenton, *Software Metrics: A Rigorous Approach*, Chapman & Hall(1991).
- Proceedings of International Symposium on Software Metrics (METRICS), IEEE Computer Society Press.
- D. Card: *Measuring Software Design Quality*, Prentice-Hall(1990).
- Robert L. Glass: *Facts and Fallacies of Software Engineering*, Addison Wesley (2002).



2004/12/14

Software Engineering Laboratory, Department of Computer Science, Graduate School of Information Science and Technology, Osaka University

第4回エンピリカルソフトウェア工学研究会