インクリメンタルな開発の測定事例の報告

(株)オージス総研
藤井　拓、細川　亮二
中清　桂子、市橋　幸治

追加

発表のアウトライン

- 研究の背景
- どのような測定、分析を行いたいか（追加）
- 提案する測定手法と分析システム
- 測定結果（初期段階）
- 現状の評価、課題、まとめ

今回のプレゼン資料は配布資料に追加、変更を加えています！

反復型 vs ウォーターフォール型

- 日本では、ウォーターフォール型がいまだに大勢を占めている
  - 開発依頼者と開発者の関係
    - 契約を介した関係
    - 開発依頼者がリスクをとりづらい
  - 反復型の効果
    - 適用した効果がはっきりと示されていない

中間に解があるか？

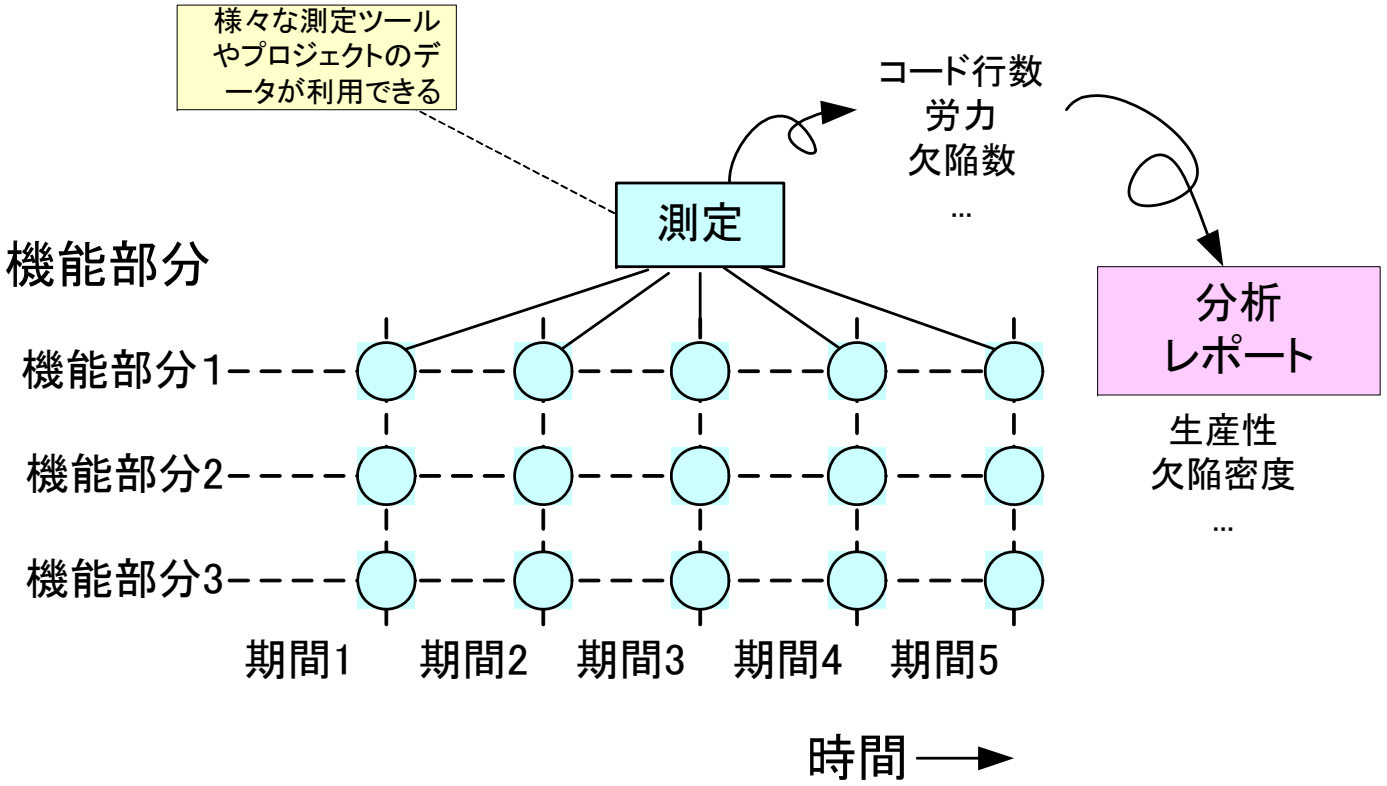
- 日本の状況を考えた場合、両極端以外によりよい解があるかもしれない
  - インクリメンタルな開発
    - 複数のリリースを介した開発
  - 開発プロセスの実装、プラクティス
    - 例えば、テストの自動化などの開発の効率化

もっと有効な開発方式の探求には

- 様々な開発プロセス/開発方式の比較を行い、それらの間の違いを分析するための仕組みが必要
- そのような仕組みに必要なこと
  - 開発結果の測定
    - 実装方法等に依存しない
  - 開発過程の分析

追加

やろうとしている測定と分析

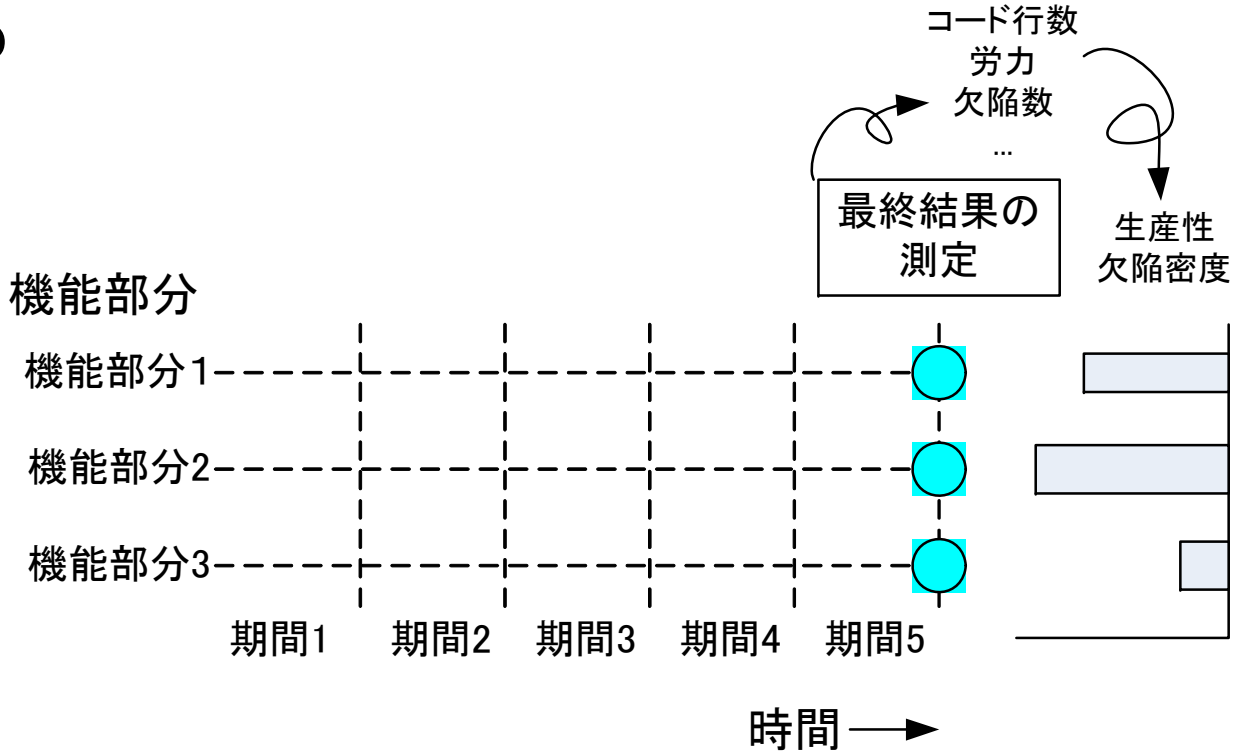


機能部分は、ユーザ機能やコンポーネントを表す

追加

最終結果の測定と分析

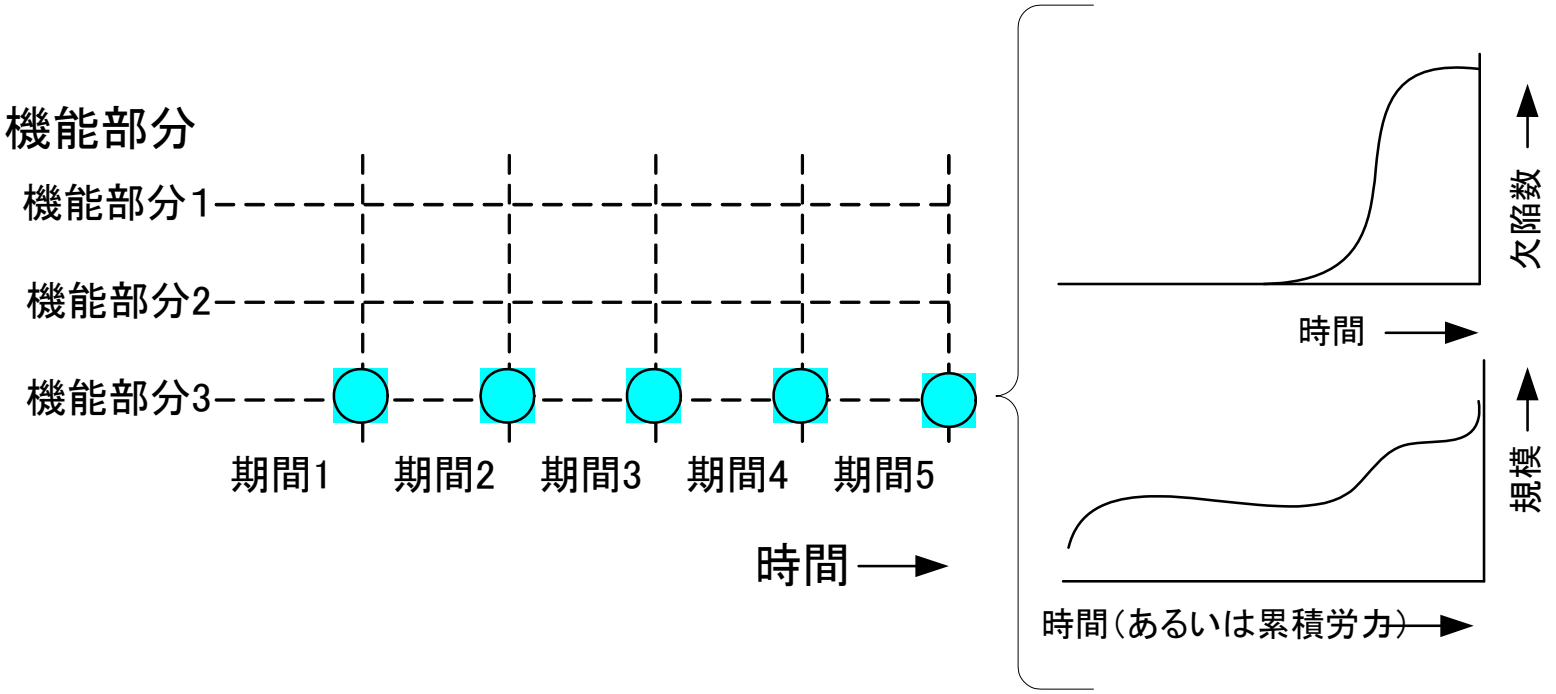
- 最終結果からは機能部分ごとのバラつきが得られる



追加

開発途上の測定と分析

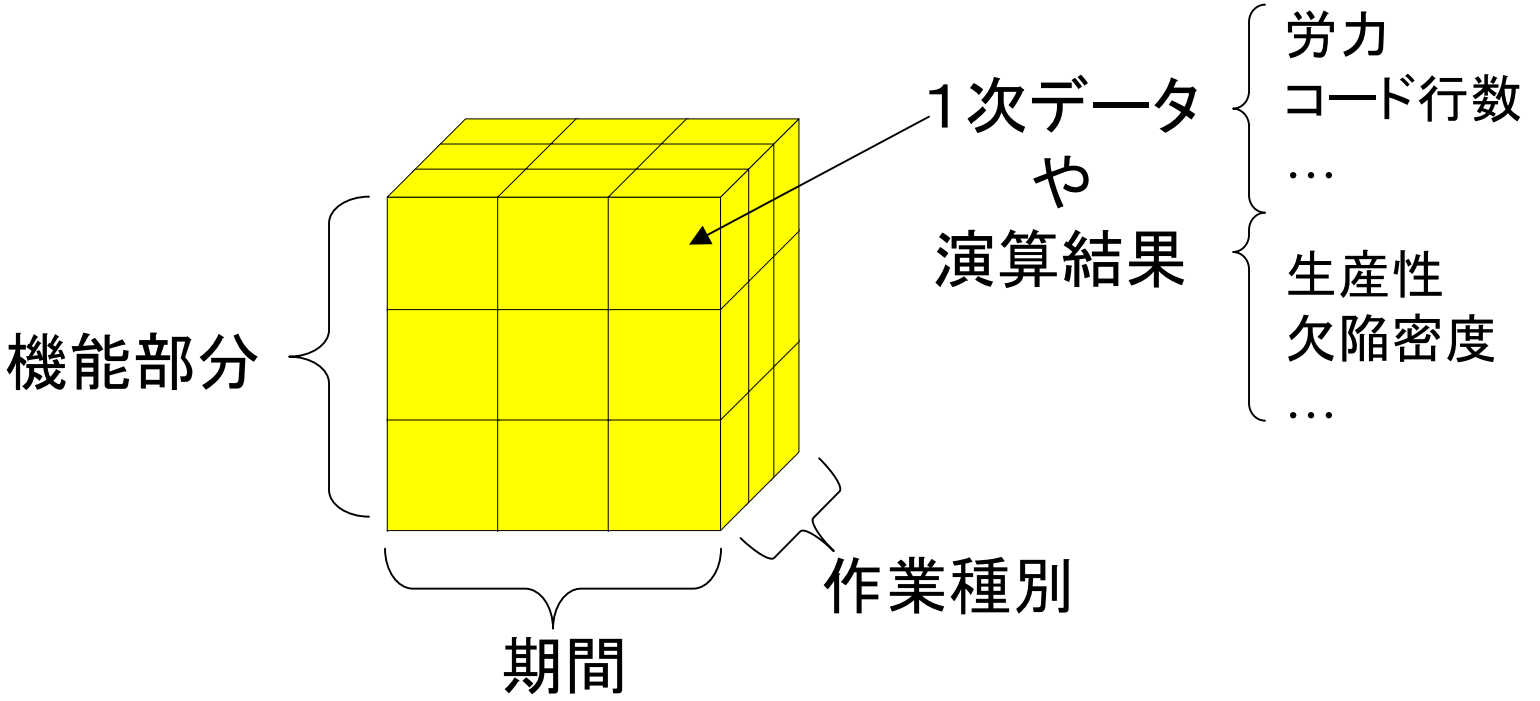
- バラつきの原因が開発途上で起きたことに起因するかどうか調べる



追加

最終結果と途上は切り口の違い

- 最終結果と途上は測定結果を異なる軸や切り口で見たもの



追加

課題

- 分析システム
  - 様々なデータソースから簡単にデータが取り込める
  - 測定結果、計画など観点（粒度）の異なるデータが機能部分、期間など複数の軸（切り口）を切り出せ、演算できる
- 規模の測定
  - プロジェクトや機能部分間での比較を行うためにコード行数以外の規模の測定方法が使いたい
  - 開発初期段階に規模の測定を行いたい

変更

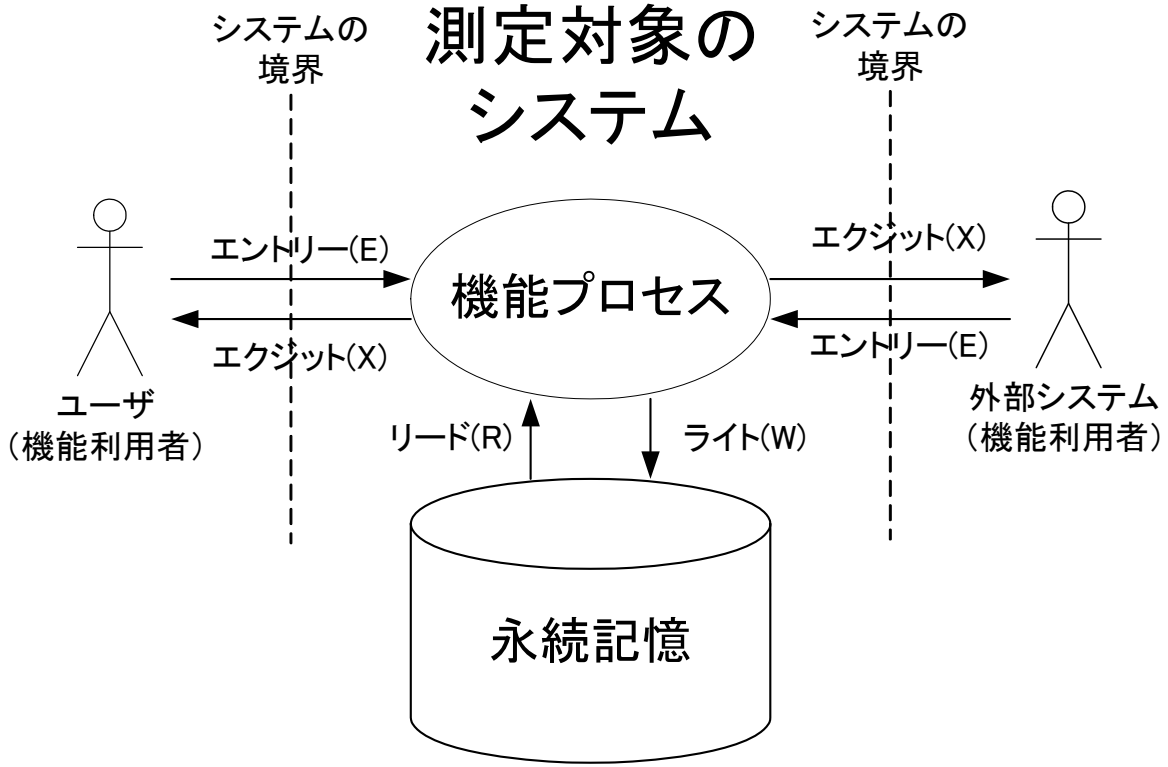
本研究の目的

- 以下の２点に基づく、開発プロセス等を比較、分析する方法の有効性を検証する
  - 機能規模測定手法COSMIC法
  - プロジェクト分析システムIDeAn
    - 開発結果と開発過程の分析

今回の発表では、本方法を1つの開発プロジェクトに適用した初期の結果を報告し、~~本方法の有効性を議論~~する

COSMIC法とは

- データ移動に基づいて機能規模を測定する手法



COSMIC法の特徴

- 要求や既存システムに基づいて機能規模を測定できる
- データ移動に注目しているので、データベースアクセス以外に通信も測定可能
- 測定方法が比較的シンプル、かつ一般に公開されている

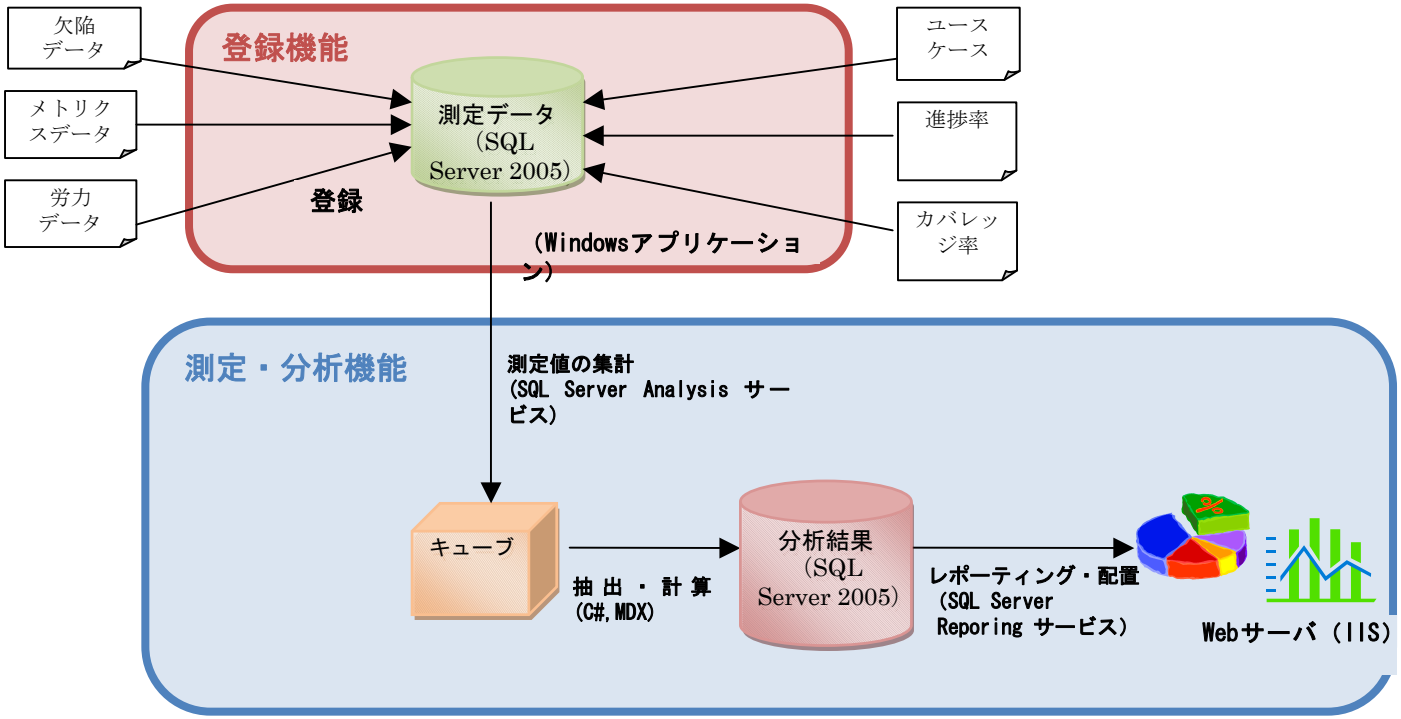
COSMIC法の長所と短所

- 長所：ソフトウェアの規模を実装言語や実装/設計構造と独立して測定することが可能になる
  - 開発生産性などを実装言語や実装/設計構造
- 短所：開発途上の規模の測定には使いづらい

開発途上の規模の測定はコード行数を併用したほうがよい

IDeAnとは

- 反復的な開発プロジェクトのモニター/分析システム



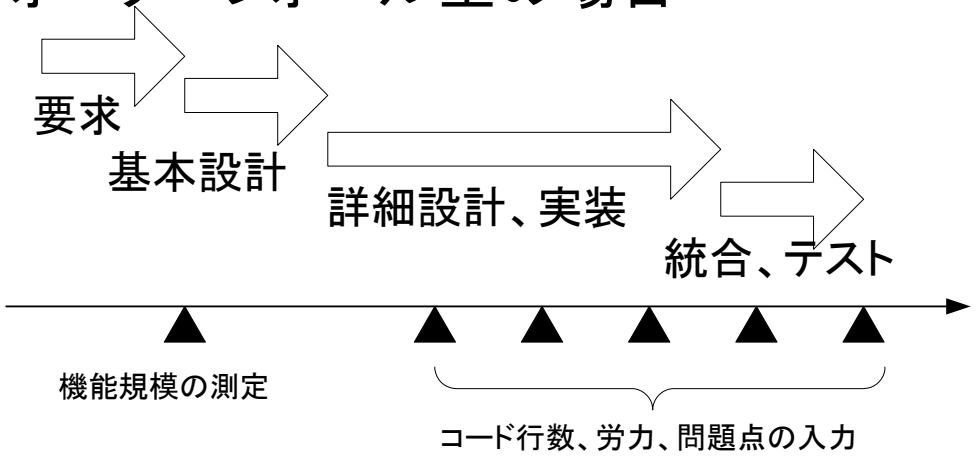
IDeAnの特徴

- 規模、労力、問題点管理データなどのデータを統合し、分析結果を出力する
  - 規模は、機能規模とコード行数の両方
- 規模、労力、問題点管理などのデータは、以下の2つの次元で入力される
  - 時間（期間）
  - 機能部分（ユースケース/画面）

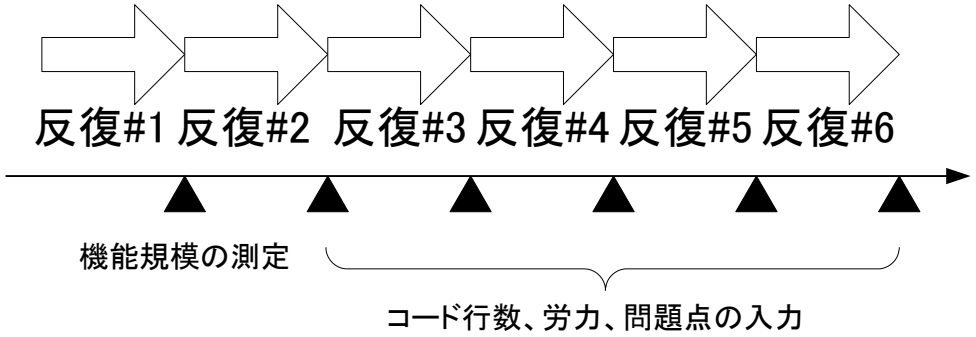
開発結果と開発過程の両方による分析を支援

COSMIC+IDeAnによる測定

ウォーターフォール型の場合



反復型の場合



COSMIC+IDeAnによる分析 :開発結果の分析

- COSMIC法で測定した機能規模に基づき、プロジェクト全体と機能部分毎の分析結果を求める
  - 開発生産性
  - 機能規模当たりの作業労力
  - 欠陥密度(これだけLOCも用いる)
  - 欠陥あたりの労力

COSMIC+IDeAnによる分析 :開発過程の分析

- コード行数規模に基づき、プロジェクト全体と機能部分毎の開発途上での測定値の推移を求める
  - 開発生産性
  - 欠陥密度
  - 作業種別ごとの労力

開発結果に影響を及ぼす開発過程の因子を探る

測定対象のプロジェクト

- 対象：ビジネス系のWebアプリケーション
- 開発言語：Java、JSP
- アーキテクチャ：J2EE
- 体制：社員＋協力会社
- 開発プロセス：インクリメンタルな開発プロセス

目的：アプリケーションの基本機能を実装し、評価する

変更

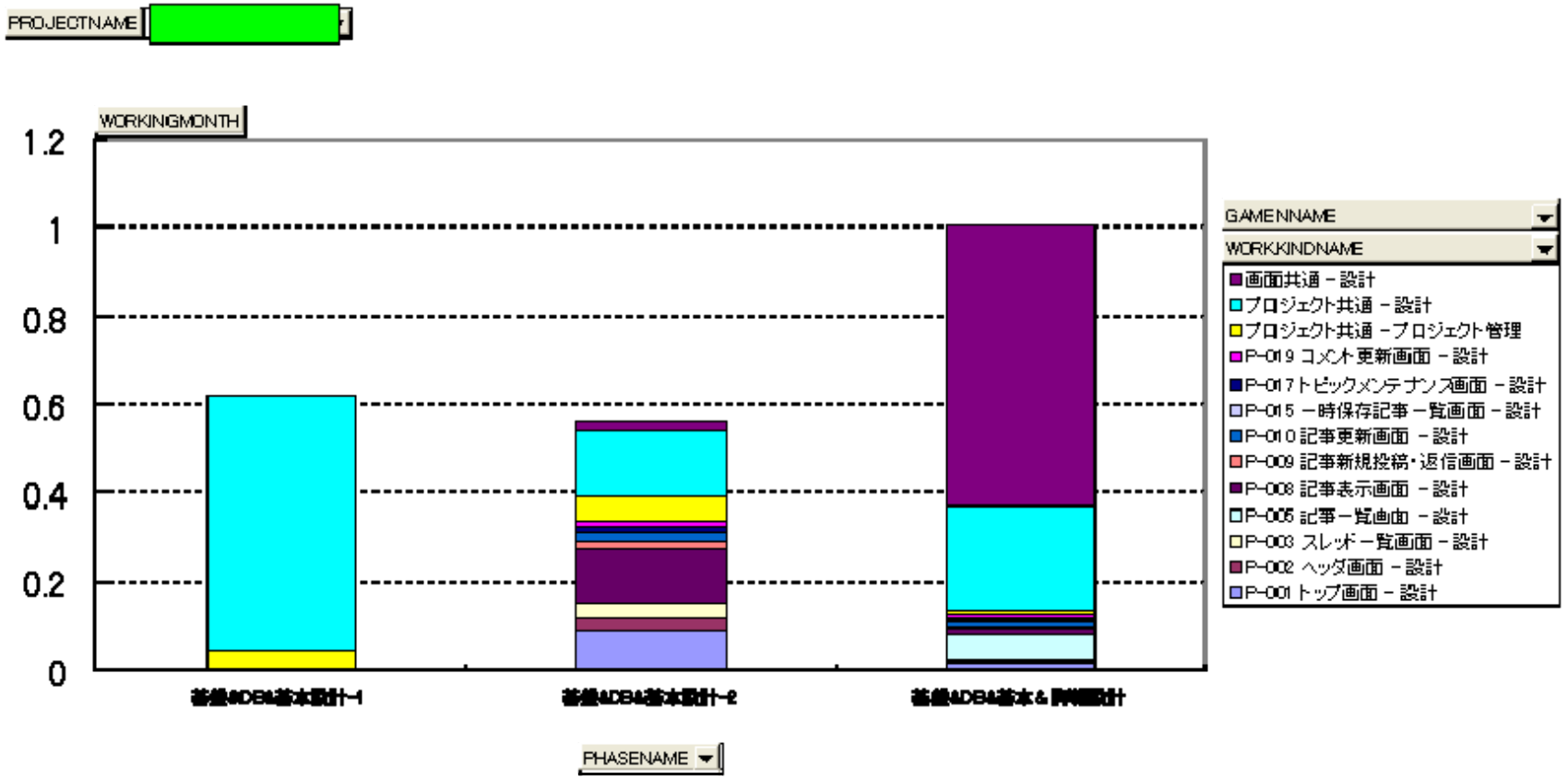
機能規模の測定結果（確定版）

画面ID	画面単位 CFP
P-001	10
P-002	0
P-003	15
P-005	11
P-008	28
P-009	20
P-010	9
P-015	5
P-017	19
P-019	11
合計	117

このプロジェクトでは、機能部分を画面単位に設定している

変更

作業種別、機能部分単位の労力の推移
~~開発生産性等~~の測定結果



追加

現在までの評価と課題

- COSMIC法
  - 画面定義書と簡単なユースケース記述に基づき、半日程度で測定できた
  - 今後、コード行数や労力との相関を求め、有効性を評価する必要がある
- IDeAn
  - 今回のプロジェクトの測定は問題なく行えている
  - 現在の実装は拡張性が低いので改良が必要

まとめ

- 以下の２点に基づく、開発プロセス等を比較、分析する方法を提案した
  - 機能規模測定手法COSMIC法
  - プロジェクト分析システムIDeAn