

機能間の依存関係解析を用いたソフトウェア保守支援

ソフトウェアシステムの保守段階における変更要求

システムは様々な理由でプログラムの変更を繰り返しながら運用される

- ・ 要求の変化によるシステムの機能追加・削除・編集
- ・ 不具合の発生によるバグ修正
- ・ システムの統合・刷新

保守作業コストの大半を占めるプログラム理解作業

システムの要求 - 機能 - 実装とその対応関係を把握する必要がある

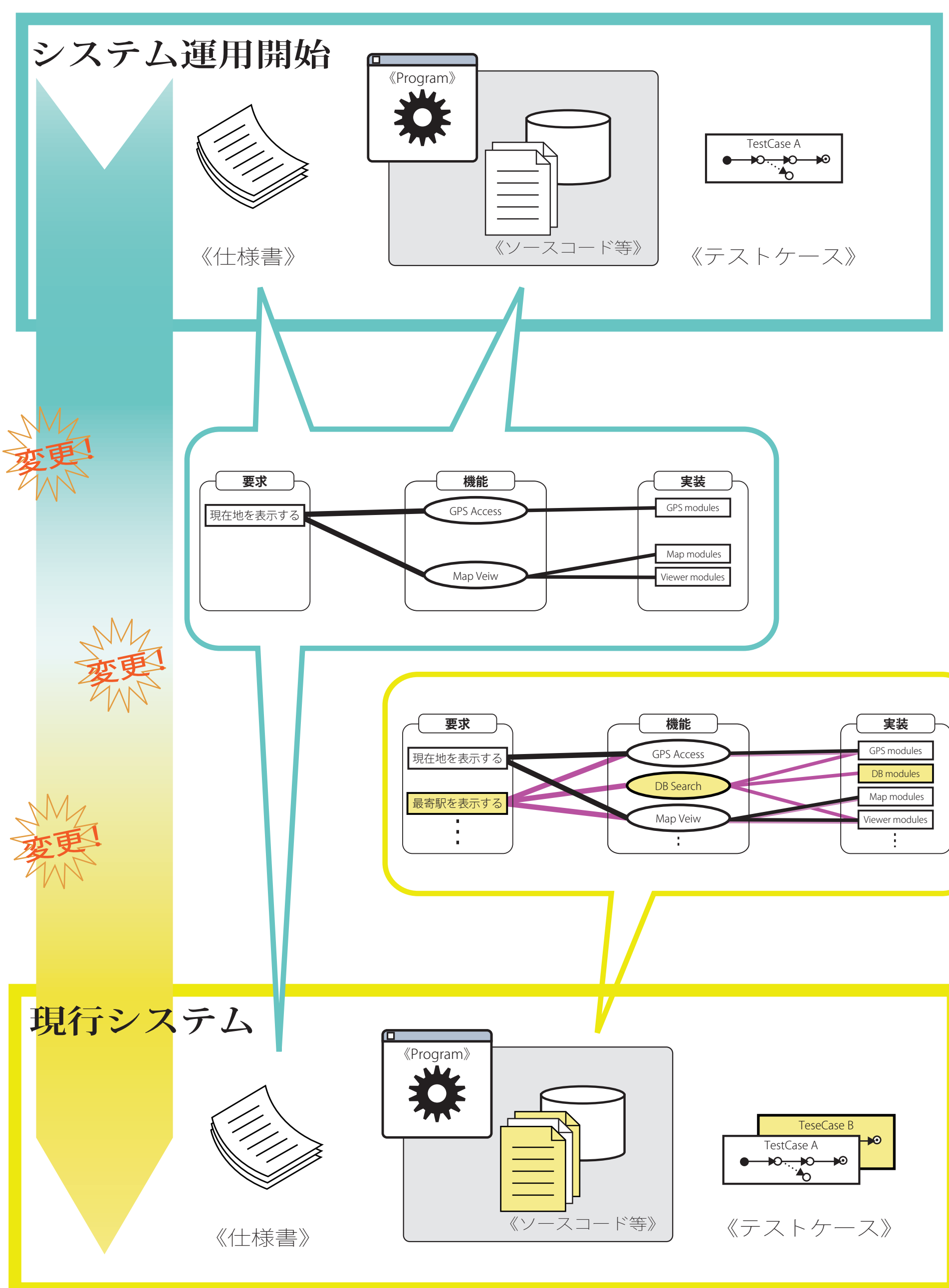
- ・ システムはどんな機能を持つのか？
- ・ ユーザーからある要求を実現する機能はどれ？
- ・ システムを構成するプログラム・モジュールは？
- ・ ある機能を実装しているプログラムモジュールは？
- ・ ある機能を実現する際のプログラムの振る舞いは？

変更箇所の特定, 変更内容の決定, 変更後のテストの選出等に必須の知識

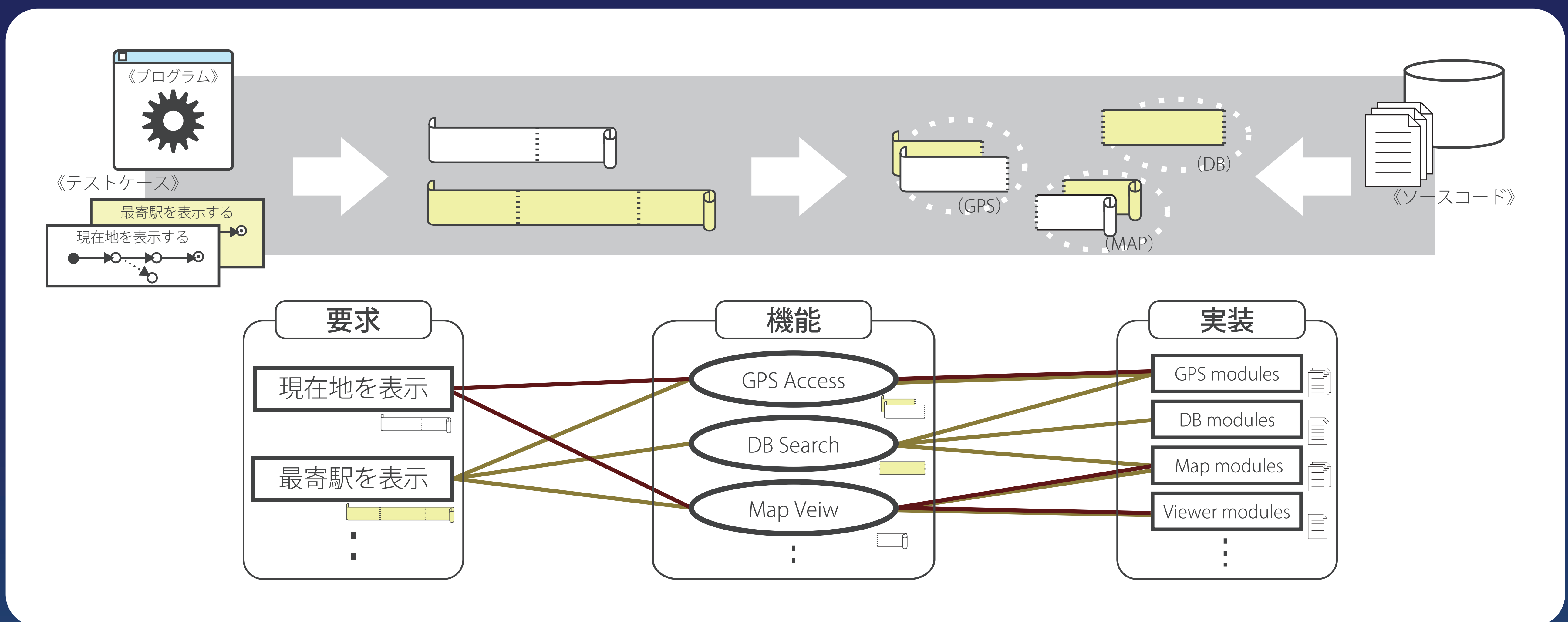
繰り返し変更されるソフトウェアの理解支援の必要性

現行システムの持つ機能と要求 - 機能 - 実装の対応関係情報の紛失

- × 設計工程で機能と対応関係を定義・記述した仕様書に、保守工程での変更内容が反映されない
- × プログラムへの変更を繰り返す際に、意図しない機能間の依存関係が作りこまれる
- × 現行システム上の機能とその依存関係を手作業で調査・復元するコストが膨大



【提案手法】 現行システムの機能と要求 - 機能 - 実装の対応関係の復元



1. プログラムの実行時情報から、システムの持つ機能を抽出する

- (1) “要求”を統合テスト時のテストケース等から実行シナリオとして入力する。
- (2) システムが各要求を実現した際の実行時情報を、各機能の実行区間毎に分割する。(機能フェイズの自動検出手法)
- (3) 分割された実行区間を、処理内容ごとに分類し、分類されたクラスタを機能に対応付ける。

2. 要求から機能、機能から実装への対応関係を再構築する

- (1) 実行シナリオと分割された実行区間の属するクラスタとの対応から、要求と機能との対応関係を構築する。
- (2) 分割された実行区間とその区間で実際に動作したプログラムモジュールとの対応から、機能と実装の対応関係を構築する。

Java プログラムで構成されるシステムへの適用実験

- ・ UI ベースの解析では検出できない機能間の依存関係を抽出
- ・ 実装モジュール間の動的な依存関係も検出可能
- ・ クラスタから機能の抽象名を推定・提示する工夫が必要

今後の展望

- ・ より大規模なソフトウェアシステムに対する適用実験による評価
- ・ テストからの不具合探索範囲の絞り込みや、変更後の回帰テストにおけるテストケースの順位付けへの応用