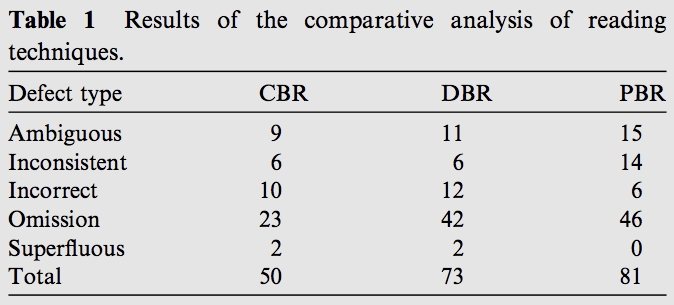
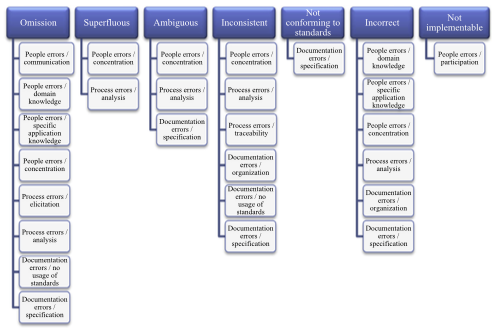
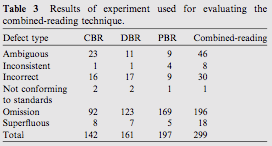
Detecting defects in software requirements specification

* どんな論文か
  + ソフトウェア要求仕様書の欠陥を見つけることに関する論文
  + 要求分析に良いツールを提供
  + 世の中に欠陥の分類や、検出技術はあんまりない
  + 欠陥や検出技術についての知識は、要求フェーズの欠陥を発見するために、編集と再評価が必要
  + この論文では要求欠陥の分類と発生原因を示す
  + 分類は欠陥とその原因に基づいて行なっている
  + 要求の欠陥のためのcombined-reading techniqueを提案
  + 要求仕様とソフトウェア品質の質を改善する
* 目的
  + 問題の原因や、欠陥や検出技術を可能にすることなど広範囲に渡り理解すること
* 分類の比較の成果
  + 例外クラスの分類の目的
    - [5]software development life cycle の欠陥を分類することでSWの品質向上
    - [6]要求仕様書のレビューを支援のためにチェックリストを構築したい。そのためにsoftware requirements specification で欠陥タイプを分類すること
    - [7]エラーの防止と探索のために要求フェーズで欠陥を引き起こすエラーを分類
  + アプローチ
    - [6,7]経験的な研究アプローチをとった。[5]は不明
  + 対象
    - [5]フェーズやタイプに焦点を絞っていない
    - [6]要求フェーズのsoftware requirements specification
    - [7]要求仕様書に対して
  + 定義
    - [5]異常の定義をしていない。いつどこで発生したのかも
    - [6]例を含めて異常を明示した。原因は不明
    - [7]例を挙げたが欠陥とエラーの関係は示していない
  + 問題解決について
    - [5]指定はなし
    - [6] root-cause-analysis methodを使用。問題と目的の違いを議論。
    - [7] root-cause-analysis methodを使用。問題と目的の違いを議論。
* 欠陥検出(読み取り技術)の比較
  + 仮説?
    - 事象を引き起こしている欠陥を見つけることができる能力によって、欠陥の検出技術は変えるべき
  + 方法
    - ATM、選挙、Web出版システムの３つを用意した
  + 結果
    - 分析と書いていたが、表の事実を述べているだけ？4.3章
* 関連研究
  + 要求仕様書がソフトウェア品質の重要な要因になる[1]
  + 要件の欠陥に関する研究が３つ
    - 異常クラスを分類[5]
    - 要求仕様書の欠陥タイプの分類[6]
    - 欠陥の原因となった要求のエラーについて[7]
  + 欠陥検出(読み取り技術)について
    - Checklist-Based-Reading technique (CBR)
      * 特別な誤りを探すためにチェックリストor報告書を作る[18]
      * 典型的な欠陥についてのyes,noの質問をする[10]
      * メリット
        + 点検(チェック？)プロセスで欠陥のあるとこを探し出すことができる
        + 検査項目が提供される
      * デメリット
        + 質問が他の組織からの情報を収集していない
        + 確実ではない
        + どのように使うか、指示がない
        + 同じことを何回も聞くから無駄につながる
    - Defect-Based-Reading technique (DBR)
      * 特定の欠陥に着目
      * 要求はSoftware Cost Reductionによって書かれる
      * 欠陥を分類し、そのクラスごとに質問をする
      * チェックリストによるシナリオが作られる
      * 特定の欠陥を見つけるためにそのシナリオでレビューをする
    - Perspective-Based-Reading technique (PBR)
      * レビュワーが見つける共通の欠陥を最小にすることで効率を良くする
      * ソフトウェア開発やメンテナンスで人がする役割を確認する
      * レビュワーが反復して行うシナリオがあり、それをレビュワーが答える
        + 特定の視点からの理解を深めることができる
      * 欠陥を見つけるのを助ける[20]
      * 利点[8]
        + 特定のステークホルダーの視点からソフトウェアドキュメントをレビューするのでeffective, systematic, focused, goal-oriented, customizable and transferableである
        + CBRより共通して発見することが少ない
        + ソフトウェアのことが詳しくない人でもわかるからCBRよりいい
    - 論文
      * Checklist-Based-Reading technique[8,9]
      * Perspective- Based-Reading techniques[8,9]
      * アドホックベース[10-12]
      * Checklist-Based-Reading[13,14]
      * Usage-Based-ReadingとChecklist-Based-Reading techniqueの比較
* 欠陥とならない例外がいくつかある

‘‘improvement desirable anomaly’’ and the ‘‘editorial anomaly’’[5]

‘‘misplaced’’, ‘‘redundant’’ ,‘‘typo or formatting’’, ‘‘Misplaced’’, ‘‘typo’’ or ‘‘formatting’’ [6]

* 提案する分類
  + omission, ambiguous, inconsistent, superfluous, incorrect, not-conforming to standards, not-implementable and risk-prone (表２)
* 要求のエラーを確認—＞３大エラーがわかった。(SRSドキュメントの形式と一致)
  + 人に関係するもの(要求仕様書の作成の関係者)
  + プロセスに関するもの(要求分析)
  + ドキュメントに関するもの(要求を書くプロセスでおかしかったもの)
* エラーと原因の関係を調べるために実験
  + 条件
    - ATM、選挙、Web出版システムの３つを用意
    - CBR,PBR,DBRを使用し欠陥を抽出(適材適所)
    - それを詳細な三大エラーに分類
    - ↑70個ぐらいのイエスノーの質問をした
  + 実験
    - ４つのケーススタディーを使用
      * ガレージ制御システム
      * ABCビデオシステム
      * オンラインショップ
      * オンラインゲーム
    - 結果
      * 
      * 不具合は全部で417個
      * 提案手法がいいのは表から明らか
    - 提案手法の利点
      * 点検する項目が明確
      * 訓練がいらない
      * SWを知らなくてもOK
      * 欠陥をたくさん見つける
      * 原因が明確にわかる
    - 提案手法の欠点
      * 全部が見つかるわけではない
      * 誰が悪いかを見つけるものではない
      * 自動化していない