

The Influence of Organizational Structure on Software Quality: An Empirical Case Study



Nachiappan Nagappan

Brendan Murphy

Victor R. Basili

論文の説明をする前に...

Conwayの法則ご存知ですか？

Conwayの法則とは？

- メルヴィン・コンウェイが1968年に提唱
- 『システムを設計する組織は、その組織のコミュニケーション構造と同じ構造を持ったシステム構造を設計をする』という意味
- 組織が実際にどれくらいソフトウェアの品質や構造に影響を与えるかは未知

目的とアプローチ

- 目的
 - Conwayの主張と品質の関係を明らかにする
- 前提
 - 組織の複雑さが調和の困難さを招くであろう...
- アプローチ
 - Vista開発時の構造とリリース後の欠陥を比較

品質をどう測るか？

- クラッシュログを欠陥として各バイナリを「エラーが起きやすい」とそうでないものに分類
 - Vistaに含まれるバイナリ(.dll, .sys, .exe)
- 1つの欠陥で数千人のユーザがクラッシュしてもそれは1つの欠陥であるとする

組織の複雑さをどう測るか？

- バージョン管理システムのデータから、どの開発者がどのファイルを変更したのかを明らかにし、ビルド情報からどのソースがどのバイナリに使われているのかを特定
- 上記のデータから8つのメトリクスを計測

組織の複雑さをどう測るか？

- メトリクス

- 開発者の人数

- 何人の開発者が特定のバイナリを変更したのか
 - ただし、現在も社員であり続けていること
 - エラーの起きやすいバイナリほど高くなるのでは？

- 元開発者の人数

- 特定のバイナリを変更したけどもリリース前に会社を去った開発者の人数
 - 途中で辞めた人が多いほど欠陥が多くなるのでは？

組織の複雑さをどう測るか？

- メトリクス

- 変更回数

- 変更が多いほどほど品質は低いのでは？

- 主所有の深さ (Depth of Master Ownership)

- バイナリに対する作業がどれくらい分布しているのか
 - 主所有: 75%以上の変更を行った人
 - 主所有の人が組織のどれだけ下のレベルにいるのか
 - 共通のマネージャを持つ開発者ばかりで行われる変更が多ければDMOの値は高くなる

組織の複雑さをどう測るか？

- メトリクス

- 開発に寄与する組織の割合

- DMOのレベルに報告を行う人の組織中での割合
 - あるバイナリを所有する組織に100人の人がいて、そのうち25人が主所有者に報告を行う場合, 0.25

- 組織上のコード所有レベル

- 特定のバイナリを所有している組織が行う変更の割合
 - 値が高いほどエラーが少なくなるのでは？

組織の複雑さをどう測るか？

- メトリクス

- 組織上の所有権

- 特定のバイナリに変更を行いかつ主所有者に報告を行った技術者の割合
 - エラーの起きやすいバイナリほど低くなるのでは？

- 組織の交差因子

- 特定のバイナリに少なくとも10%の変更を行う組織の数
 - 値が高いほど、多くの組織に変更が分散している

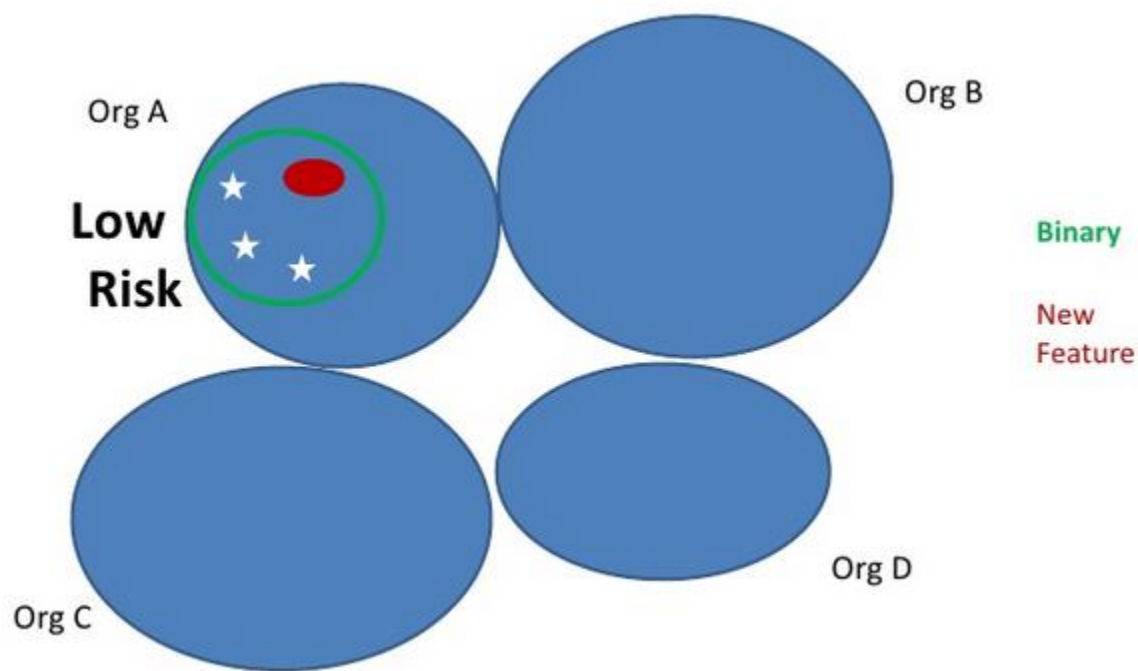
分析結果

- ロジスティック回帰により，組織構造のメトリクスからエラーの多さを予測できるのか？

モデル	正確性	再現性
組織構造	86.2	84.0
コード変更量	78.6	79.9
コードの複雑さ	79.3	66.0
依存関係	74.4	69.9
テストカバレッジ	83.8	54.4
リリース前のバグ	73.8	62.9

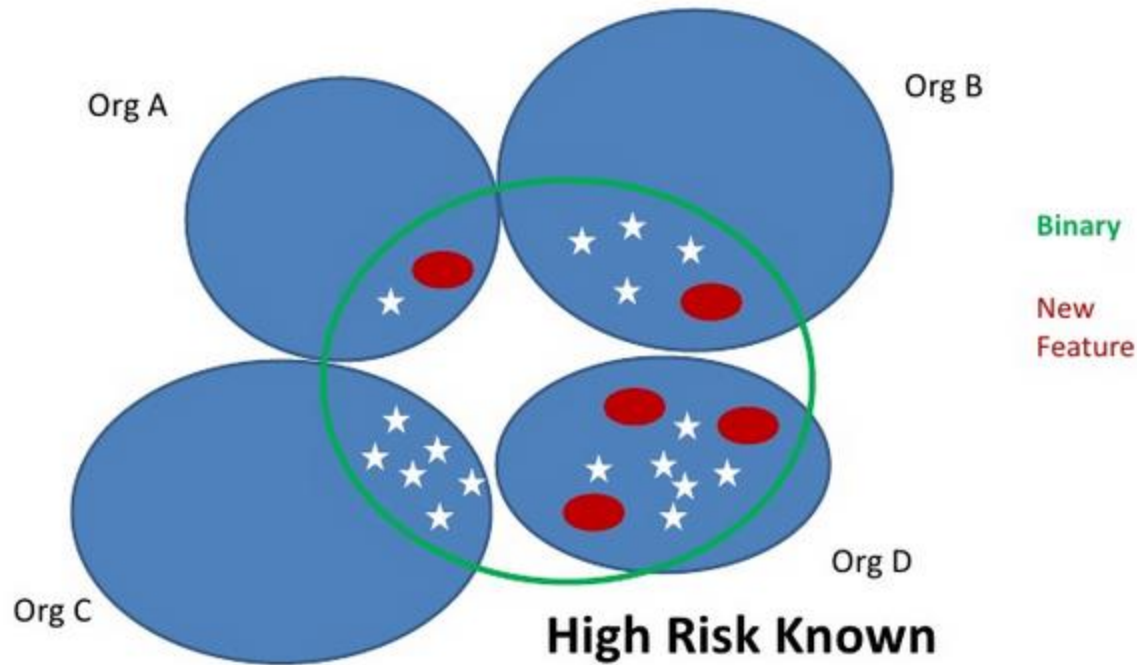
追加分析

- バグが出るリスクの低い例



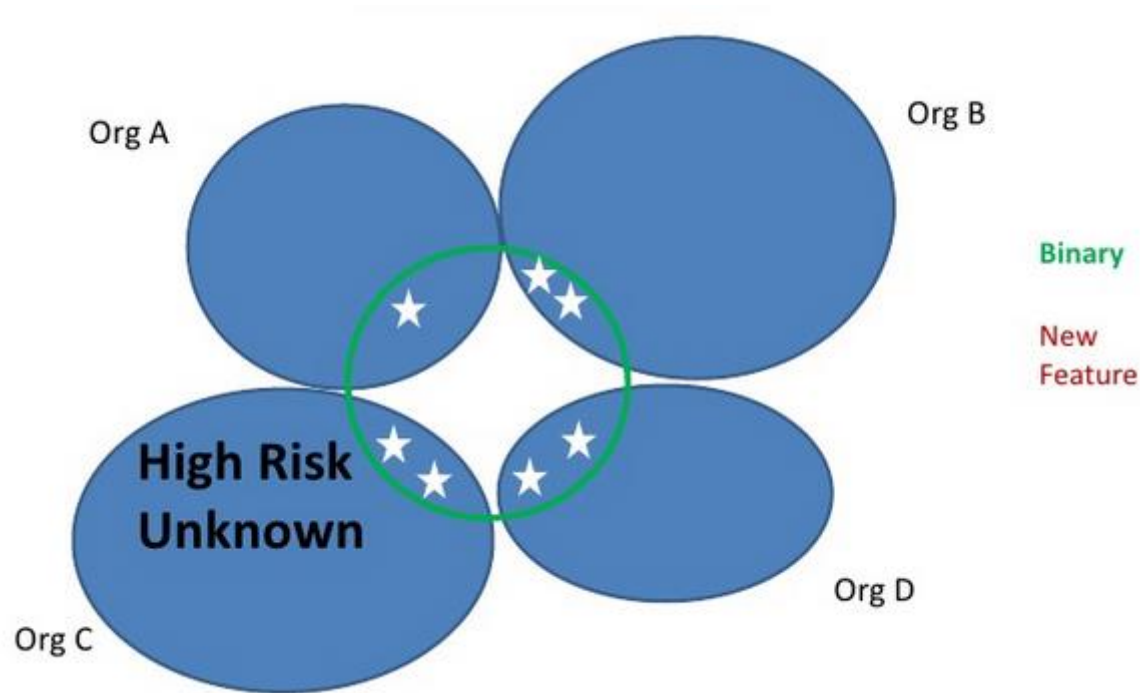
追加分析

- バグの発生するリスクが高い例



追加分析

- 潜在的なバグを含んでいるリスクが高い例



まとめ

- リサーチクエスション
 - Conwayの主張とプロダクトの品質の関係はどうか？
- 分析方法
 - Windows Vista開発時の組織構造とリリース後の欠陥について調査
- 結果
 - 組織構造の指標が品質に一番関係していた

コードを見るより組織構造を見た方がよくわかる