ソフトウェア環境の変化がメトリクスに及ぼす影響について

神 谷 年 洋

Effects of Changing Software Environment on Metrics

Toshihiro Kamiya

1. 自己紹介

大阪大学の井上研究室，メトリクスチームに所属しています．我々は，ソフトウェアメトリクス（尺度）を用いた品質管理をテーマとして研究を行っています．

我々が扱っているメトリクスは，現在，FP（ファンクションポイント）と複雑度メトリクスです．FP はソフトウェアの要求仕様書からソフトウェアの機能量を計測するもので，開発コストを予測するために用いられます．複雑度メトリクスは設計書やソースコードからソフトウェアの複雑度を計測するもので，エラーの発生を予測するために用いられます．

これまでの研究には，ある要求仕様書作成支援ツールの出力から（半）自動的に FP を求めるツールを作成し，その評価を行う研究，C++のソースコードに対して複雑度メトリクスを適用し，エラーの予測精度を実験的に評価する研究などがあります．今後は，ソフトウェア開発の各段階によって，これらのメトリクスがどのように変遷しているかの研究などを予定しています．

2. 議論したいこと，問題意識

- メトリクスによる予測はどの程度実用化できるか？
  メトリクスによって，開発コストやエラーの発生を予測する際には，統計的な予測を用いる．乱暴に単純化すれば，メトリクスによる予測は「以前，あるプロジェクトをメトリクスで計測した時の計測値は○ ○ であり，そのプロジェクトの完了には △ △ 日かかった．今度のプロジェクトの計測値は × × だから，だいたい□ □ 日かかるだろう」式で行われる，といえる．
  予測の正確さは，統計的予測が可能なくらいデータを積み上げられるかどうかにかかっている．しかしながら，現実のソフトウェアの開発においては，開発者個人の能力差，開発プロセスやツールや言語といった環境の変化，開発対象となるソフトウェアのドメインの違いなどにより，統計的に均質な母集団というものは想定しがたい．かといって，能力差，環境，開発対象などを調節する実験（比較実験，contorolled experiment) はコストがかさむため，どうしても小規模な開発を実験対象にすることになる．小規模な開発から得られた結論を一般の大規模な開発に適用することには危険が伴う．
  このような困難を乗り越えるすべはあるだろうか．
- フレームワークやコンポーネントによる再利用はメトリクスの適用を困難にするのではないか？
  フレームワークはドメインを限定することで，コンポーネントは粒度を大きくすることで，再利用を促進する．フレームワークは，その応用範囲を特定のドメインに限定することで，特殊化された部品をライブラリとして用意する．開発対象のアプリケーションがその応用範囲に入っていれば，再利用率を上げることができる．コンポーネントは，”large grained object”，すなわち大きな機能をもつ部品であり，コンポーネントがうまく使えれば，大量のソースコードを再利用することになる．

† 大阪大学 大学院基礎工学研究科 情報数理系専攻 ソフトウェア科学分野
Division of Software Science, Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University

再利用が行われることで，たとえば FP の場合，アプリケーションの持つ機能量から，新規に開発しなければならないソースコードの量を見積もることが困難になるのではないか？複雑度メトリクスの場合なら，再利用されるソースコードと新規開発されるソースコードの質的な違いが，エラーの発生を予測する際に問題にならないだろうか．また，再利用される部品のドキュメンテーション (ソースコードは提供されるか．設計書は提供されるか．仕様書は提供されるか .) がメトリクスに及ぼす影響はどのようなものか．

- ソフトウェア開発プロセスの違いはメトリクスの有効性に影響するか？
  従来のウォーターフォールモデルでは，要求仕様，設計，コーディング，テスト，というフェーズをこの順序に行うことでソフトウェアが開発されると仮定している (フェーズを逆行するのはエラーが発見されたとき，すなわち「何か問題が発生したとき」である) . 近年では，ウォーターフォールモデル以外の開発プロセスとして，スパイラル開発のアプローチも用いられる . スパイラル開発のアプローチでは，仕様や設計のフェーズは何度も繰り返し行われ，そのたびに要求仕様書や設計書が部分的に，あるいは全面的に修正される . このようなドキュメントを繰り返し変更することを前提とした開発において，メトリクスによる予測はどの程度有効だろうか .
- 「ソフトウェアの進化・発展を計測する」メトリクスというものはあるだろうか？
  ソフトウェアの進化・発展の仕組みによって，ソフトウェアシステムの寿命はより長くなる . すなわち，システムを構成するコンポーネントは頻繁に入れ替わったり追加されたりするかもしれないが，システム自体は長期間にわたって保守されることになる .
  従来のメトリクスは，ソフトウェアの機能や構造が一定であると仮定し，それを元に開発コストやエラーの発生を予測するものである . 進化・発展により，ソフトウェアの機能や構造が動的に変化するようになったとき，なお通用するメトリクスといったものはあるだろうか .

(平成？年？月？日受付)
(平成？年？月？日採録)