

ソフトウェア開発におけるメトリックス環境

1G-1

大西 論 松本 健一 杉山 裕二 鳥居 宏次

(大阪大学基礎工学部)

1. まえがき

ソフトウェア・メトリックス(以下、メトリックスと略称)とは、ソフトウェア生産活動とソフトウェア自身の一般的な性質に関する計量化(定量化)である¹⁾。その目的は、ソフトウェア生産の客観的な認識のほかに、ソフトウェア生産活動の制御を行い、ソフトウェアの生産性と品質を効率的に向上させることにある。

今日まで多くのメトリックスが提案されてきたが、

- ・必要なデータの収集およびその分析が煩雑である。
- ・広く一般に適用できるメトリックスがない。
- ・メトリックスを用いることにより得られる恩恵が明白でない。

といった理由から、広く認められたり普及しているものがほとんどないのが現状である。

本稿では、こうした問題点を解決し、ソフトウェアの開発過程において、その生産性と品質の向上にメトリックスが利用できる環境について考察する。

2. メトリックス環境

2.1 目的

メトリックス環境とは、ソフトウェアの開発過程において、プログラマがメトリックスを積極的に活用できるような環境のことである。したがって、メトリックスに必要なデータを自動的に収集するだけが目的ではない。収集したデータとメトリックスの分析結果(メトリックスデータ)を、プログラマにフィードバックすることによって、ソフトウェアの生産性と品質を向上させることが目的である。

2.2 特長

(1) データの自動収集と一元管理

必要なデータは、できる限り自動的に収集する。自動収集に向かないものも、メニュー形式でキーボードから入力するなど、プログラマの負担ができるだけ少ない方式で収集する。収集したすべてのデータは、利用しやすいようにデータベース化し一元管理する。

(2) メトリックスツール群とコンストラクション機能

プログラマが利用できるメトリックスは複数とする。プログラマは各自の開発環境や必要に応じてメトリックスを選択でき、複数のメトリックスを組み合わせることで、より信頼性の高い結果を得ることもできる。

コンストラクション機能は、メトリックスを改良・作成し、プログラマが自分なりのメトリックスを作り上げるためのものである。プログラマは実状に合せ、各メトリックスツールに変更を加えることができる。また、メトリックスデータの中から入力パラメータを選び、内部処理を記述することにより、新しいメトリックスツールを作成できる。このとき、他のメトリックスの分析結果を入力パラメータにできるので、メトリックスを階層的に構成し、より抽象度の高いメトリックスへと発展させることが可能である。

(3) 半助言システム

メトリックスを用いることによる恩恵は、プログラマが直接受けるべきである。したがって、メトリックスデータは、プログラマに直接フィードバックされる。その手段として、プログラマに対して対話的にアドバイスをを行う半助言システムを用いる。プログラマは対話的にアドバイスを受け、その情報をもとに自らのソフトウェア開発過程を管理、その生産性と品質が向上するよう努力する。

(4) 分散処理システム

メトリックスの処理にはかなりの手間が必要であり、ソフトウェア開発を行う計算機上でその処理も行うことは物理的にかなりの負担になる。また、計算機一般の流れからみても、分散処理システムが主流になると考えられる。そこで、メトリックス専用の計算機をネットワークを通じてつなぎ、そこで必要な処理を行う。これにより、ホスト計算機の負担を最小限に抑えることができる。

3. システム構成

システムの基本構成は次のとおり(図1参照)。

○メトリックスシステム

・データ収集ツール群

メトリックスに必要なデータを、プログラマの負担ができるだけ少ない形式で収集する。たとえば、プログラムテキストの履歴、コマンドの使用状況、テストの経歴、バグ情報などを収集する。

・メトリックスデータベース

収集したデータやメトリックスの分析結果など、システムで用いるデータを一元管理する。

・メトリックスツール群

各種のメトリックスツールと統計処理ツールを含む。メトリックスツールには、コスト・工数の予測、複雑性・信頼性の評価を行うものなどがある。統計処理ツールは、メトリックスの分析結果の有効性を検証するためのものである。

・メトリックス作成ツール群

メトリックスの改良・作成の手助けをするコンストラクションツールと、改良・作成したメトリックスの分析を行うツールを含む。

○半助言システム

・プロセス管理ツール群

ソフトウェアの開発過程を管理するツール群。プログラムテキストの変更過程を管理するコンフィギュレーション管理ツール、コスト・工数を事前に予測した値と比較していくコスト・工数管理ツール、テスト資源の配分やテスト完了の判断を行うテスト管理ツールなどを含む。

・プロダクト管理ツール群

生産物としてのソフトウェア自身を管理するツール群。保守性を考える場合重要となるプログラムの複雑さや読み易さを管理する複雑性・読解性管理ツール、信頼性をより高くするための信頼性管理ツールなどを含む。

Metrics Environment on the Software Development

Satoshi ONISHI, Kenichi MATSUMOTO, Yuji SUGIYAMA, Koji TORII

Osaka Univ.

4. 現状

現在、システムのうち実現しているものは、データ収集ツール群、メトリックスツール群の一部である。その概要について述べる。

- プログラムテキストの履歴収集ツール
エディタ使用時に、そのテキストの内容を自動的に記録する。また、エディタ使用理由などを記録でき、簡単なドキュメントを残せるようになっている。
- コンパイル、テストデータの収集ツール
コンパイルの状況、テストの実行時間、テストデータなどを収集する。
- コマンド情報収集ツール
- Halstead のプログラム計測ツール

このほかにも、データの収集状況、分析結果、開発の進捗状況などをわかりやすく表示するツールなどを実現している。

これらのツールは、メトリックス用ワークステーションとして使用している Sun-2 やホストの VAX-11/750 上に構成されている。これらのツールを使用して、実際に学生実験でのデータを収集した²⁾。

5. 予想される効果

現在は、一部のメトリックスデータが収集できる程度であるが、本システムが完成すると、メトリックスを用いる際のプログラマの負担が最小限になるため、メトリックスの積極的な利用が期待される。そうした状況のもとで、メトリックスとその利用方法の研究が進み、ソフトウェアの生産性と品質の向上が一層はかられると考える。

6. おわりに

今後、本稿で提案したメトリックス環境を順次整備していき、効率的かつ信頼性の高いソフトウェア開発環境を構築していく予定である。

参考文献

- 1) 日本情報処理開発協会：“システムの高信頼性技術に関する調査研究（電子応用システム）成果報告書”，(1985)。
- 2) 松本，小田，大西，工藤，杉山，鳥居：“ソフトウェア信頼性モデルの比較とプログラム作成能力の評価基準について”，昭和61年度電子通信学会総合全国大会，1709 (1986)。
- 3) Leon J. Osterweil：“TOOLPACK - AN EXPERIMENTAL SOFTWARE DEVELOPMENT ENVIRONMENT RESEARCH PROJECT [1]”，6TH ICSE, pp.166-175 (1982)。
- 4) 宮本勲：“ソフトウェア・エンジニアリング：現状と展望”，T B S出版会 (1982)。

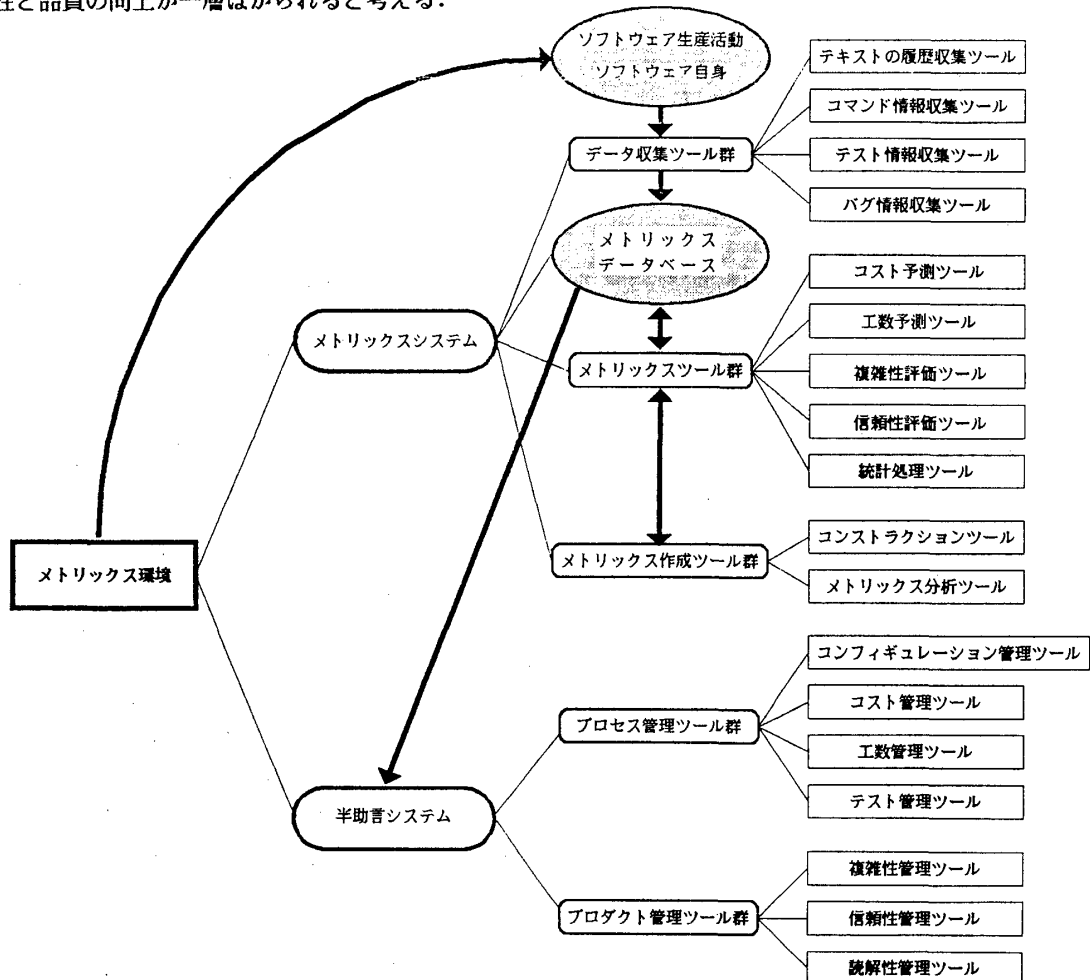


図1 システム構成図