ソフトウェア開発企業における開発タスクの自動計測

Automatic Measurement of Software Development Tasks in Software Companies

門田 暁人[*1] 上野 秀剛[*2] 荒木 健史[*3] 山田 欣吾[*4] 松本 健一[*5]

あらまし 本稿では, 開発行動記録システム TaskPit を開発組織に適用した結果について報告する．12 名の開発者を 6 日間計測した結果，(a) リーダが開発作業に従事しすぎている, (b) メールによるコミュニケーションが少なすぎる人がいる，(c)短時間に顧客あての長文メールを何本も出していると推定される人がおり，メールの書き方に改善の余地がある可能性がある，といった改善の糸口を発見できた．

1 はじめに

著者らはこれまでに，「計測に基づくプロセス改善」を実践し，世の中に広めていく一手段として，ソフトウェア開発行動記録システム TaskPit[5][6]を 2008 年に開発し，普及に努めてきた．TaskPit は，開発者もしくは開発チームが，日々の開発タスクに従事した時間や作業量を計測するツールであり，Eclipse や Visual Studio を使った「プログラミング」，Word を使った「文書作成」，ブラウザ上の Web メールを使った「メールの読み書き」等の，アプリケーションやウィンドウにひも付けられたタスクを自動計測できる．

TaskPit をとりまくコミュニティは次第に広がり, 有志による可視化ツール TaskVieweR の開発，クイックマニュアルの作成が行われ，2013 年 6 月現在の TaskPit 1.0.0～1.0.3 の総ダウンロード回数は 1200 を超えるに至った[6]．今後は，TaskPit ユーザ向けに，実開発における TaskPit の適用事例およびその知見の蓄積が求められている．

本稿では，TaskPit を，ソフトウェア開発組織に適用し，9 日間（6 営業日）にわたって 12 名の開発者を計測した結果について報告する．12 名の内訳は，開発メンバー7 名，リーダ 3 名，顧客窓口 2 名である．計測データの分析は，調査対象部署の作業概要を把握している他部署のベテラン社員 1 名が行った．

2 TaskPit

2.1 開発の動機と特徴

Tom DeMarco の「計測できない物は制御できない」という格言に代表されるように,

*1 Akito Monden, 奈良先端科学技術大学院大学
*2 Hidetake Uwano, 奈良工業高等専門学校
*3 Kenji Araki, 株式会社アクセス
*4 Kingo Yamada, 株式会社ファインバス
*5 Kenichi Matsumoto, 奈良先端科学技術大学院大学

開発プロセスの制御や改善には計測が必須である[1]．開発現場で生じる問題の多くは人的要因に起因する[2]ものであるから，プロダクトやプロセスを計測するよりも，開発の主体である人間やその作業を計測し，改善につなげることが自然であると考えられる．その一つの手段である Personal Software Process (PSP) / Team Software Process (TSP)は，開発者および開発チームが日々のタスクに関する情報を記録し，自らのプロセス改善を行う方法として広く知られている[3][4]．ただし，PSP/TSP のデータ計測は人手で行う必要があることから，導入の敷居は高く，広く普及するには至っていない．

著者らは，「開発現場ですぐに使え，計測結果も解釈しやすい」ことを実現する自動計測ツールとして，TaskPit を開発し，改良を行ってきた[6]．TaskPit は，(1)どのタスクにどの程度の時間を費やしているか，(2)どの程度の作業量を各タスクに費やしているか，(3)各タスクの成果物の量，を個人またはチーム単位で自動計測できる．

計測にあたっては，(1)各タスクは，それぞれ異なるアプリケーションやウィンドウ上での作業であると捉える．各アプリケーションやウィンドウ上での作業時間（ユーザが何らかの入力を与えていた期間）を記録することで，各タスクに費やした時間を計測する．また，(2)各タスクの作業量は，各アプリケーションやウィンドウに対するキーストロークやマウスの操作量（回数）として記録する．さらに，(3)各タスクの成果物は，特定のディレクトリ下にファイルとして出力されると捉える．ファイルサイズの増減を一定時間ごとに調べることで，各タスクの成果物の量の推移が計測できる．

2.2　タスクと成果物の定義と計測

異なるタスクであっても，同一のアプリケーションを用いる場合があるため，TaskPit では，アプリケーション実行時のウィンドウ名（に含まれる文字列）によってタスクを区別する． タスクは，実行中の 1 つ以上のアプリケーションの組として定義され，アプリケーションは，実行ファイル名とウィンドウ名の組として定義される．また，成果物は，1 つのディレクトリと 1 つ以上のファイル拡張子の組として定義される．

TaskPit では，計算機のユーザが，各アプリケーションの使用を開始／終了した時刻を記録する．ここで「アプリケーションを使用している」とは，ウィンドウにフォーカスが当たっている状態を指す．ただし，フォーカスされている場合であっても，一定時間（例えば 3 分間）計算機に対する入力（マウス，キーボード）が行われなかった時点で，いずれのアプリケーションも使用していないとみなす．

2.3　システム構成

図 1 に示すように，TaskPit は計測部とそのバックエンドとなるデータベース，可視化部，設定ファイル，ログファイル，作業日報ファイルから構成される．設定ファイルでは，タスクと成果物の定義，ログファイルの出力時間間隔などの指定を行う．また，計測部は，図 2 に示すように，「タスク」タブでは，各タスクの累積の実行時間，打鍵数，クリック数が表示される．「成果物」タブでは，各成果物のファイルサイズとファイル数の増減が表示される．計測結果は，一定時間（例えば 10 分間隔）でログファイルに出力される．ログファイルの出力先を共有フォルダとすることで，多数の開発者の計測結果を容易に集計できる．可視化部は，設定ファイルとログファイルを入力とし，指定さ

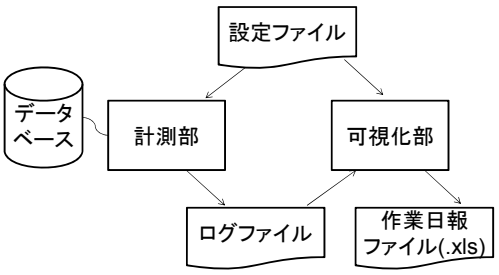

図 1. TaskPit のシステム構成

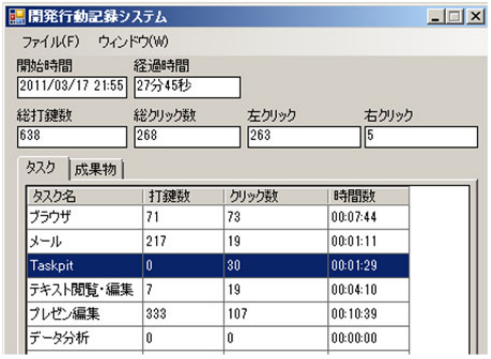

図 2. 計測部の GUI

れた期間の計測結果を様々な側面からグラフ化できる．また，1 日毎の計測結果の概要を作業日報として（Excel の.xls 形式で）出力できる．

2.4 実装

TaskPit 1.0.x は，Windows XP/Vista/7 の.Net Framework 上で動作し，データベースには SQLite を用いている．タスクの特定は，Windows API を用いて行っている．具体的には，GetForegroundWindow でアクティブなウィンドウのハンドルを取得し，GetWindowText によりウィンドウタイトル取得している．また，System.Diagnostics.Process クラスの GetProcessById メソッドによってプロセスを取得し，実行ファイル名を得ている．

3 TaskPit の開発現場への適用

3.1 計測の目的

ソフトウェア開発部署における問題の糸口を発見し，改善につなげることを目的とする．計測結果の分析を行うのは，開発管理・支援部門のベテラン社員 1 名であり，全社的なプロジェクト管理手法の標準化，品質管理，人材育成の観点から開発支援またはコンサルティングを行うことを想定している．分析者は，各作業者の役割を把握しているが，作業内容までは把握していない．

3.2 計測対象と期間

計測対象は，ソフトウェア開発部署の 12 名である．表 1 に示す通り，部署における各人の役割は，リーダ 3 名，開発（A プロジェクト）3 名，開発（B プロジェクト）4 名，顧客窓口 2 名である．A プロジェクトは納期が近いため多忙であった．メンバーのうち B1 は，他部署から 1 か月前に移動してきたため，必ずしも業務に慣れていない．

この部署ではプロジェクト管理ツールとしてブラウザ上で Trac を使っており，Trac 上での資料管理とコミュニケーションを行うことが奨励されている．

データ計測においては，開発作業に外乱を与えないために，GUI を表示しないように改造したサイレント版 TaskPit 1.0.1 を用いた．また，計測に先立って，この部署で使っているアプリケーションについてヒヤリングを行い，タスクとアプリケーションのひも付けを行った．計測は 9 日間（6 営業日）にわたって行った．

表 1　計測対象の開発者

役割	メンバー	備考
リーダ	L1, L2, L3	
プロジェクト A の開発作業（開発 A）	A1, A2, A3	A1 は 4,5 年目．A2，A3 は 1 年目
プロジェクト B の開発作業（開発 B）	B1, B2, B3, B4	B1 は他部署から異動してきてから 1 ヶ月
顧客窓口	CS1, CS2	二人ともに時短勤務

```
ブラウザ = iexplore.exe|firefox.exe|Safari.exe|chrome.exe
メール = thunderbird.exe|Outlook.exe|iexplore.exe:gmail
ファイル操作 = explorer.exe
エディタ　= sakura.exe|noediter2.exe
ＤＢ操作　= SqlWb.exe
エクセル = EXCEL.EXE|soffice.exe
ワード・パワポ = WINWORD.EXE|POWERPNT.EXE
文書閲覧　= AcroRd32.exe
テスト　= mstsc.exe|Beyond32.exe|DF.exe|reverse.exe|reverseserver.exe|perfmon.exe|….
プログラミング・デバッグ = eclipse.exe|devenv.exe|VPC.exe|VMWindows.exe|hh.exe
```

図 3. 設定ファイル（抜粋）

3.3　設定ファイル

計測に用いた設定ファイルの一部を図 3 に示す．図に示されるように，1 つのタスクに対して複数のアプリケーションが割り当てられている．

3.4　計測結果と分析

図 4 は，各タスクの従事時間，全タスクの合計時間，勤務時間をそれぞれ 1 日あたりの平均値で示したものである．勤務時間は，TaskPit の起動・終了時刻から，1 日の勤務時間を推定したものである．得られた主な結果は次の通りである．

- 図 4 に示されるように，いずれの開発者においても登録外の作業があった．その原因の一つとして，この部署では，勤怠管理システムを用いていたが，設定ファイルに登録されていなかった．また，TaskPit 実装上の問題として，仮想 OS 上でのテストやプログラミング・デバッグについて，計測できていないことが明らかとなった．（TaskPit 1.0.3 では，設定ファイルに登録していないアプリケーションについても計測できるよう拡張を行った．）
- 納期を間近に控えた A1, A2, A3 の勤務時間，および， PC 上での作業時間は長くなっている．一方，B1, B2, B3, B4 は仕事が順調に進捗しており、勤務時間はプロジェクト A に比べ短くなっている．TaskPit により，プロジェクトの忙しさをある程度把握できるといえる．
- 勤務時間・PC 上の作業時間ともに，最も多かったのは A1 である．従事時間が多い順にプログラミング・デバッグ（184 分），エディタ（106 分），テスト（94 分）

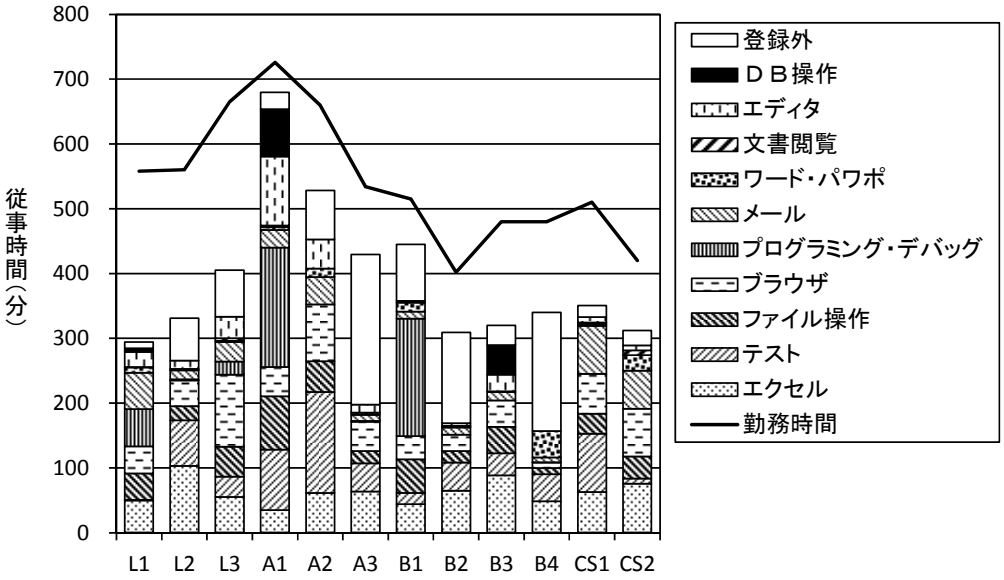

図 4　各タスクの平均従事時間（1 日あたり）

であり，コーディングを中心に作業を行っていることが伺える．一方，A2 は，同じプロジェクトに属するが，プログラミング・デバッグは 0 分であり（登録外を入れても 76 分），テスト（156 分）を担当していることが伺える．TaskPit により，各開発者の作業内容を把握できるといえる．

- リーダ 3 名は，開発者や窓口の者よりも勤務時間に対する PC 上の作業時間が少ない．これは，作業指示，指導，管理等の業務を行っているためである．ただし，リーダ L2 は他のリーダと比べて，勤務時間に対する PC 上の作業時間がやや大きくなっている．タスクの内訳をみると，リーダ L2 はテストに平均 71 分従事し，また，テストケース管理に関すると思われるエクセル操作に 103 分も従事している．リーダとしては開発作業に従事しすぎているといえる．
- 顧客窓口の 2 名（CS1, CS2）は，他者よりもメールの従事時間が多く，ブラウザの使用時間も多い．メールによって顧客とのやり取りを行っていることと，ブラウザ上で Trac による資料管理とコミュニケーションを行っていることが伺える．
- メールタスクに着目すると，L2，A3，B1, B2, B3, B4 はいずれも 15 分未満であった．この部署では，コミュニケーション手段として電話よりもメールを使用することが奨励されている（記録を残すため）．このことから，特にリーダ L2 は，メールの使用時間が少なすぎるといえる．その後の調査で，L2 は電話を多用していることが分かった．
- メールタスク，プログラミング・デバッグタスクの従事時間と打鍵数を図 5 に示す．メールタスクについて 1 分あたりの打鍵数を比較すると，CS1, CS2 の値が高い．この 2 名は打鍵数自体も多く，メールの本文が長い傾向があることが伺える．顧客とのメールのやりとりでは，なるべく簡潔に要点を押さえて書くことが望ましく，ま

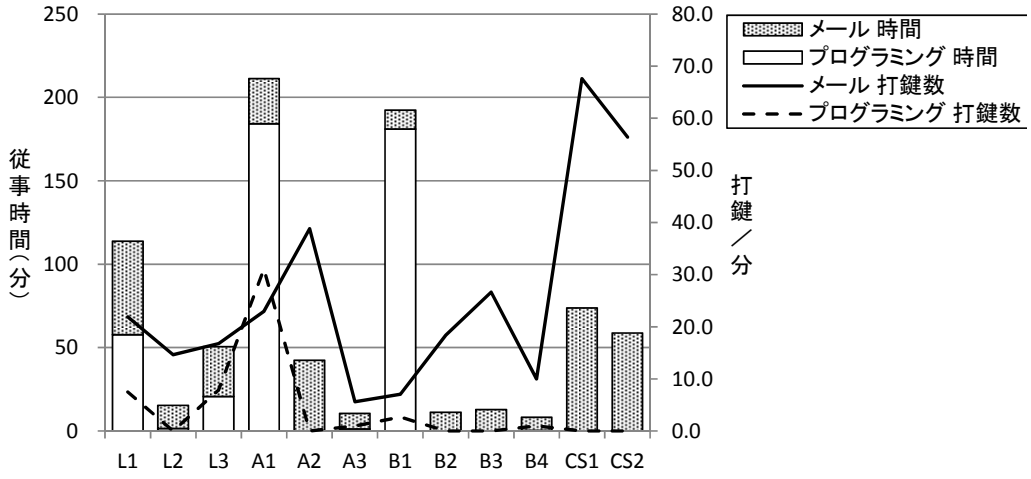

図 5　メールとプログラミングの従事時間と打鍵数（1 日あたり）

た，記述内容をよく吟味・推敲することが求められるため，メールの書き方に改善の余地がある可能性がある．

- 図 5 において，プログラミング・デバッグのタスクと打鍵数に着目すると，A1 は，打鍵数，1 分あたりの打鍵数ともに多く，品質を別にすれば生産性の高い開発者であるといえる．

4　まとめ

本稿では，開発行動記録システム TaskPit を開発現場に適用し，6 営業日にわたって 12 名の開発者を計測した．その結果，開発管理・支援部門の立場から，プロジェクトの忙しさや各開発者の作業内容を把握できることが分かった．また，(a)リーダが開発作業に従事しすぎている，(b) メールによるコミュニケーションが少なすぎる人がいる，(c)短時間に顧客あての長文メールを何本も出していると推定される人がおり，メールの書き方に改善の余地がある可能性がある，といった改善の糸口を発見できた．著者らは，今後も引き続き，TaskPit の適用事例を増やし，その知見を公開していく予定である．

参考文献

[1] T. DeMarco, "Controlling Software Projects: Management, Measurement & Estimation," Yourdon Press, New York, USA, 1982.
[2] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター, "IT プロジェクトの「見える化」中流工程編," 日経 BP, 2008.
[3] W. S. Humphrey, "A discipline for software engineering," Addison-Wesley, 1995.
[4] W. S. Humphrey, "パーソナルソフトウェアプロセス入門," 共立出版, 2001.
[5] 門田 暁人, 亀井 靖高, 上野 秀剛, 松本 健一, "プロセス改善のためのソフトウェア開発タスク計測システム," ソフトウェア工学の基礎 XV, 日本ソフトウェア科学会 FOSE2008, pp.123-128, 2008.
[6] ソフトウェア開発行動記録システム TaskPit, http://taskpit.jpn.org/