



---

# 卒業研究報告書

平成28年度

---

研究題目

タスクの自動計測システムにおける  
導入を妨げる要因の削減

---

指導教員 上野秀剛 講師

---

氏名 藤井滉斗

---

平成29年2月16日 提出

奈良工業高等専門学校 情報工学科

# タスクの自動計測システムにおける

## 導入を妨げる要因の削減

上野研究室 藤井 滉斗

ソフトウェア開発者の作業を計測するシステムとしてTaskPitが存在する。このシステムは、現在使用しているアプリケーション名、あるいはそれに関連付けされた作業名と作業時間、打鍵数、左右クリック数を自動的に記録する。本研究室では以前から、実際のソフトウェア開発現場での作業データを収集するために、企業にTaskPitの導入を依頼している。本研究ではTaskPitをより多くの現場で適用するため、システムの導入を妨げている様々な要因を被験者実験と性能評価実験で検証する。被験者実験では、TaskPitを実際にインストールし、出力データを分析してもらうことで“インストール時に困難な操作・用語が存在する”、“システムが使いにくい”、“動作環境が限定される”という3つの要因を検証する。性能評価実験では、TaskPitを様々なWindows OSのPCにインストールし、動作確認することで“動作環境が限定される”という要因を検証する。また、CPUの使用率とデータ通信量を測定することで、“ハードウェアへの負荷が大きい”、“通信経路を圧迫する”という2つの要因も検証する。被験者実験の結果は、設定ファイルと実行ファイルの場所をTaskPitのインストーラや取扱説明書に明示することで、ファイルを探す時間を短縮できることを示唆する。また、設定ファイルのユーザーインターフェースの改良も必要である。被験者の分析作業から、既に分析済みのデータを出力することで、ユーザの負担が減少すると考えられる。性能評価実験の結果、TaskPitは様々なWindows OSにインストールできた。CPUの使用率を調査した結果、平常時は使用率が1%未満だが、データ送信の前後で使用率が10%程度まで上昇した。また、TaskPitのデータ通信量は非常に小さいため、通信経路は圧迫しないと示唆される。

# 目次

1	はじめに	3
2	関連研究	4
3	システムの導入を妨げる要因	6
3.1	インストール時に困難な操作・用語が存在する	6
3.2	システムが使いにくい	6
3.3	動作環境が限定される	7
3.4	ハードウェアへの負荷が大きい	7
3.5	通信経路を圧迫する	8
3.6	導入効果が明確でない	8
3.7	ソフトウェアの価格が高い	8
3.8	充実したサポートを受けることができない	9
4	実験	10
4.1	被験者実験	10
4.1.1	被験者と実験環境	10
4.1.2	実験タスク	10
4.1.3	タスク中の作業計測	14
4.1.4	アンケート	14
4.1.5	分析方法	15
4.2	性能評価実験	16
4.2.1	動作環境の確認	16
4.2.2	CPU使用率の測定	16
4.2.3	ネットワーク通信量の計測	16
5	結果と考察	17
5.1	被験者実験の結果	17
5.1.1	計測データの平均と標準偏差	17
5.1.2	最小作業量との比較	20
5.1.3	アンケートの結果	21
5.2	性能評価実験の結果	24
5.2.1	動作環境の確認	24
5.2.2	CPU使用率の測定	24
5.2.3	ネットワーク通信量の計測	24
5.3	改善策の提案	26
5.3.1	設定ファイルの場所	26

5.3.2	計測するアプリケーションの設定方法 . . . . .	26
5.3.3	TaskPitの起動 . . . . .	26
5.3.4	作業履歴の出力形式 . . . . .	27
<b>6</b>	<b>おわりに</b>	<b>28</b>
	<b>謝辞</b>	<b>29</b>
	<b>参考文献</b>	<b>30</b>
	<b>付録</b>	<b>32</b>
A	被験者配布資料 . . . . .	32
B	被験者実験アンケート . . . . .	33

# 1 はじめに

ソフトウェア開発者の能力向上や、開発現場での意思疎通を目的として Personal Software Process (PSP) が提唱されている [1]. PSP とは開発者の作業を計測することで、設計やコーディング、テスト、会議など個々の開発作業にどれだけの時間を費やしているのかを分析し、プロセス改善やコミュニケーションに役立てる手法である。

以前から PSP の補助を目的とした、作業記録を容易にする計測システムが複数提案されており、開発者が作業に費やした時間を自動で計測できる。しかし、現在どのような作業をしているのかは、開発者が手動で入力しなければならない。そのため、データの取り忘れや計測そのものに気を取られて作業に集中できないという問題がある。これらのシステムの内容は 2 章で述べる。

一方、コンピュータ上の作業を計測するシステムとして TaskPit が存在する [4]. このシステムは、現在使用しているアプリケーション名、あるいはそれに関連付けられた作業名と作業時間、打鍵数、左右クリック数を自動的に記録する。また、作業の切り替えを自動で判断するため、開発者は現在の作業を手動で入力する必要はない。開発者は、記録された履歴を作業ごとに集計することで、個々の作業にどれだけの時間を費やしているのかを分析できる。

本研究室では以前から、実際のソフトウェア開発現場での作業データを収集するために、企業に TaskPit の導入を依頼している。しかし、多くの現場から、そもそも TaskPit を導入することができなかったという声が返されている。作業データを計測するためには、まず TaskPit を正しく導入する必要がある。そこで、本研究では TaskPit をより多くの現場で適用するため、システムの導入を妨げている様々な要因の削減を目指す。

一般的に、システムの導入を妨げると考えられる要因は、ソフトウェアに起因するものや、ハードウェアに起因するものなどいくつか考えられる。本研究ではその要因を調査し、TaskPit に当てはまるものを考察する。

## 2 関連研究

PSPとはソフトウェア開発者の作業を計測し、個々の開発作業にどれだけの時間や手間を費やしているのかを分析することで、プロセス改善や作業の定量的評価につなげる手法である。PSPを実施するために開発者は、現在行っている作業の目的を定義する必要がある。この作業目的をタスクと呼ぶ。例えば、“設計”、“コーディング”、“テスト”、“会議”のようにタスクを分類し、アプリケーションとの関連付けを行う。タスクごとに作業履歴を分析することで、タスクに対応した作業のプロセス改善につなげる。

PSPの補助を目的とした作業計測システムは、以前から提案されている[3]<sup>1</sup>。Process DashboardやSlimTimerは、開発者が作業に費やした時間を自動で計測できる。しかし、現在どのような目的で作業をしているのかは、開発者が手動で入力しなければならない。そのため、目的の入力ミスや計測そのものに気を取られて作業に集中できないという問題がある。

そこで、作業計測とタスクの判定までを自動化したシステムであるTaskPitが存在する。TaskPitは、Windows環境で動作し、現在使用しているアプリケーション名やウィンドウタイトルの変化を自動的に検出する。そして、あらかじめ設定したタスクに基づいて、作業中のアプリケーションのタスク名、作業時間、打鍵数、クリック数などを記録する。図1にTaskPitで記録した作業履歴の例を示す。作業履歴はcsv形式で保存され、組織の管理者はこの履歴をもとに作業を分析する。

門田らは、TaskPitを開発組織に適用した結果を報告している[5]。実際の開発者の作業履歴から各タスクの作業時間、打鍵数などを抽出し、開発者がどれだけ各タスクに従事しているのかを分析している。その結果、現場リーダーが開発作業に従事し過ぎている、メールの書き方に問題があるなどの改善点を指摘している。また大橋らは、TaskPitの作業目的を自動で予測する手法を提案している[2]。TaskPitに機械学習を取り入れることで、開発者の前後の作業から、現在の作業目的を予測することに成功している。作業目的を自動で予測することにより、開発者が目的ごとに作業分析をする際の負荷軽減を図っている。

タスク名,	開始時刻,	終了時刻,	左クリック,	右クリック,	打鍵回数
設計,	12:00:00,	12:00:01,	1,	2,	3
コーディング,	12:00:01,	12:00:02,	4,	5,	6
テスト,	12:00:02,	12:00:03,	7,	8,	9
会議,	12:00:03,	12:00:04,	10,	11,	12

図1 作業履歴の例

<sup>1</sup>The Software Process Dashboard, <http://www.processdash.com/>

<sup>2</sup>SlimTimer - Time Tracking without the Timesheet, <http://www.slimtimer.com/>

また、TaskPitは専用のサーバアプリケーションを用いて作業履歴の一括管理や遠隔計測ができる。遠隔計測により、管理者の目が届かない場所の作業であっても計測できる。図2にTaskPitの利用例を示す。図2のように、TaskPitはサーバ・クライアントの仕組みをとっている。作業履歴は、インターネットを通じて作業者のコンピュータからサーバに送信される。データ送信の際に作業履歴は暗号化され、改ざんの有無や改ざん日時を検出できる[6]。例えば、他社との共同作業をする際、自社の作業履歴を送信することで作業の進捗を共有できる。

本研究では、TaskPitをより多くの現場で適用するため、システムの導入を妨げているさまざまな要因の削減を目指す。TaskPitの導入を妨げている要因を調査し、その解決策を考案する。

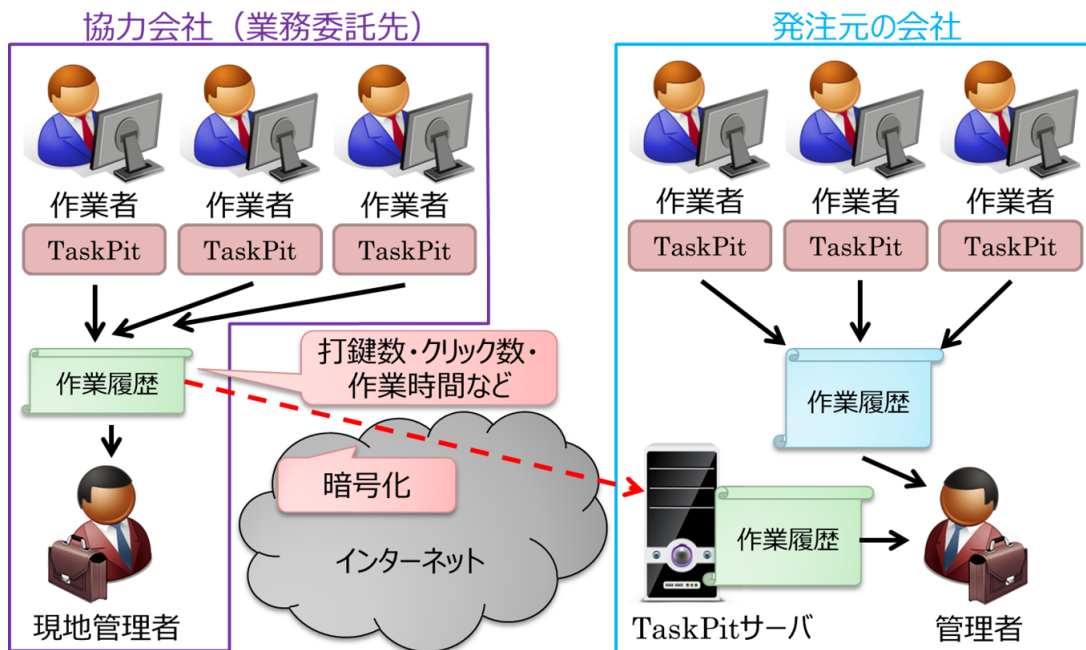


図2 TaskPitの利用例

### 3 システムの導入を妨げる要因

システムの導入を妨げている要因は様々であり、ソフトウェアに起因するものやハードウェアに起因するものなどが考えられる。本研究では、TaskPitの問題を解決するため、ソフトウェア起因の問題点を考察する。本章では、企業へのシステム導入に関する文献[7][8][9][10][11][12]をもとに、一般的にシステムの導入を妨げる要因を整理する。TaskPitは、企業や開発組織への導入を想定したシステムであるため、過去のシステム導入事例を参考にする。

#### 3.1 インストール時に困難な操作・用語が存在する

多くのソフトウェアは使用する際、最初にインストールや初期設定の作業が必要である。TaskPitもその例に漏れず、使用にはインストール・初期設定が必要である。しかし、その作業が困難であると、そもそもソフトウェアを利用できない。

TaskPitインストール時の困難な操作・用語の例として、設定ファイルの操作が挙げられる。TaskPitは、設定ファイルでタスクとアプリケーションの関連付けをしている。またIPアドレスや使用ポートなど、サーバとの通信設定も設定ファイルで行われている。TaskPitを正しく使用するためには、設定ファイルの適切な操作が不可欠である。TaskPitは、主にソフトウェア開発者の使用を想定したソフトウェアである。本研究では、ソフトウェア開発者を模した被験者にTaskPitのインストール・初期設定をしてもらう。その結果から、インストール・初期設定作業において時間がかかった項目、ミスが多かった項目を考察する。

#### 3.2 システムが使いにくい

システムの使いやすさは、ユーザが継続してシステムを利用し続ける上で重要な要素である。新しいシステムを導入したが、使いにくかったために元のシステムに戻す組織も少なくない。なるべく新しい知識を要さないシステムが、使いやすいシステムであるとされている。

現在のTaskPitの仕様上、作業履歴はcsv形式で出力される。この作業履歴はユーザが手動で分析しなければならない。本研究室では以前から、実際のソフトウェア開発現場における開発者の作業データを収集するため、企業にTaskPitの導入を依頼している。その結果、導入に成功したソフトウェア開発企業の作業データを得ている。作業履歴の形式を理解している実験者が、その作業履歴を分析したところ、1人の履歴5日分を分析するために30分かかることが判明した。5日分の作業とは、一般的な企業の1週間の業務に相当する。企業の管理者が1週間に一度、部下の作業を分析すると仮定した場合、かかる時間は「部下の人数×30分」となり、自身の作業時間を著しく圧迫すると予測できる。また、30分という時間



は、作業履歴の形式を理解している者が分析した場合の時間であり、形式を理解していない場合は更に時間がかかると考えられる。本研究では、TaskPitを初めて使用する被験者が、作業履歴の分析にかかる負担を計測する。計測指標は主に、分析作業にかかる時間と、作業後のアンケートによる被験者の主観評価である。計測結果から分析作業の負担が大きいと判断できた場合、TaskPitにおける作業履歴の出力形式に問題があるものとして、その改善策を考案する。

### 3.3 動作環境が限定される

動作環境が限られるシステムは汎用性が低いため、導入が見送られるケースがある。動作環境の代表的な例は、ソフトウェアにおけるオペレーティングシステム(OS)である。TaskPitはWindows環境専用のソフトウェアであるが、Windowsにも様々なバージョンが存在する。

TaskPitは2016年時点で最新のWindows OSである、Windows 10での動作を確認していない。本研究ではWindows 10以前のWindows OSでのインストール及び動作確認をする。また、TaskPitは様々な作業環境での使用を想定している。2016年12月時点で、世界のデスクトップOSシェア[13]は、図3のようになっている。横軸がOSの種類、縦軸がそれぞれのOSのシェアをパーセントで示している。図3より、Windows 7, 8.1, 10は利用者が多いため、検証の必要性が高い。また、Windows XP, Vistaのようなサポート期間が過ぎているもの、著しく利用者が少ないものは、検証の必要性が低い。故に、本研究の被験者実験では、Windows 7搭載のノートPCと、Windows 8.1搭載のデスクトップPCを使用する。本研究の実験機材の都合上、Windows 10環境での被験者実験は見送る。

### 3.4 ハードウェアへの負荷が大きい

ソフトウェアを動作させるためには、ハードウェアが必要である。一般的に高度な処理をするソフトウェアほど、高性能で高価なハードウェアを必要とすることが多い。あまりにも処理が重いソフトウェアは、対応するハードウェアが少なくなるため、開発現場への導入が困難になる。

ハードウェアへの負荷を示す代表的な指標として、CPU使用率が挙げられる。CPU使用率が上昇すると、他のアプリケーションとの共存が難しくなる。TaskPitは作業を計測するという特性上、他のアプリケーションと同時に使用することを前提としている。本研究では、TaskPit自体のCPU使用率を測定し、他のアプリケーションにどれだけ影響を及ぼしているか調査する。

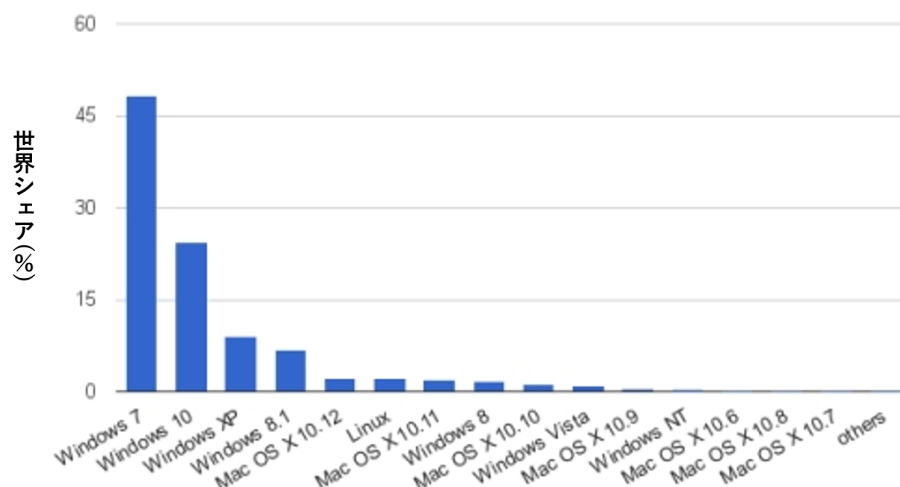


図3 世界のデスクトップOSシェア (2016年12月時点) [13]

### 3.5 通信経路を圧迫する

ソフトウェアの中にはネットワークを使用して通信するものが存在する。TaskPitもインターネットを使用して、作業履歴の送信をする。データ通信量が多すぎるソフトウェアは通信経路を圧迫し、他のシステムにも影響を及ぼす。

本研究では、TaskPitのデータ通信量を測定し、どの程度通信経路を圧迫しているのか調査する。

### 3.6 導入効果が明確でない

システム導入の失敗事例として、得られる効果が明確でないまま導入に踏み切ってしまうケースがある。導入する目的が明確でないシステムは、開発現場に定着しない。

導入効果の明確化は、システムの価値を上げるものとして重要である。しかし、実験には被験者にTaskPitの導入効果を知ってもらう必要があり、検証する方法が困難である。故に、重要な要因の1つであるが、本研究での検証は見送る。

### 3.7 ソフトウェアの価格が高い

ソフトウェアの価格が高い場合、財政的な理由から導入を見送るケースがある。費用対効果の考え方に基づき、価格が高いと費用当たりの効果は薄くなってしまふ。

しかし，TaskPitは無料で利用できるソフトウェアである．従って，TaskPitに該当しない要因のため，本研究では検証しない．

### 3.8 充実したサポートを受けることができない

充実したサポートを受けることができないソフトウェアは，不具合が発生した時にユーザの負担が大きくなるため，導入が見送られるケースがある．一方，豊富なサポートを持つソフトウェアは，ユーザが安心して導入することができる．

しかし，ソフトウェアサポートの充実を検証するためには，企業との連携が必須であり，実験が著しく困難になる．故に，重要な要因の1つであるが，本研究での検証は見送る．

## 4 実験

本研究では，TaskPit 導入の困難さを評価するために，ソフトウェアの被験者実験と性能評価実験を行う。本章では，それぞれの実験方法と分析方法を述べる。表1に，3章で整理したシステムの導入を妨げる要因の内，検証の有無と検証方法を示す。

### 4.1 被験者実験

被験者実験では，3章で述べた“インストール時に困難な操作・用語が存在する”，“システムが使いにくい”，“動作環境が限定される”の3要因を検証する。

#### 4.1.1 被験者と実験環境

被験者実験には奈良工業高等専門学校5年生の学生5名，2年生の学生5名の計10名が参加した。全ての学生は情報工学科に所属している。実験にはWindows 7搭載のノートPCと，Windows 8.1搭載のデスクトップPCの計2台を使用した。実験は2016年12月12日から12月16までの5日間実施した。

#### 4.1.2 実験タスク

実験では被験者に，以下に示す作業1~6を行ってもらう。それぞれの作業は，「TaskPitを導入する開発者（作業1~5）」，「作業履歴を分析する管理者（作業6）」の2つの立場を想定したものである。また，各作業には制限時間を設けている。

作業1: TaskPitのインストール（制限時間10分）

被験者には，デスクトップ画面に用意されたインストーラを使ってTaskPitをインストールしてもらう。被験者がインストールできたと思ったら作業

表1 実験の検証方法

システムの導入を妨げる要因	検証方法
インストール時に困難な操作・用語が存在する	被験者実験
システムが使いにくい	被験者実験
動作環境が限定される	被験者実験&性能評価実験
ハードウェアへの負荷が大きい	性能評価実験
通信経路を圧迫する	性能評価実験
導入することで得られる効果が明確でない	検証しない
ソフトウェアの価格が高い	検証しない
充実したサポートを受けることができない	検証しない

を終了してもらおう。この作業では、インストールの手順がわからない、インストールするフォルダの指定がわからないなどの困難を想定している。

作業2: 設定ファイルのオープン（制限時間10分）

被験者には、インストールしたTaskPitの設定ファイルを開いてもらう。被験者が設定ファイルを開けたと思ったら作業を終了してもらおう。TaskPitの設定ファイルは、隠しフォルダである“AppData”の中に存在するため、エクスプローラーから探し出すことが難しい。この作業では、設定ファイルを見つけられない困難を想定している。

作業3: タスクの設定（制限時間30分）

被験者には、開いた設定ファイルのタスク設定部分を書き換えてもらう。以下の図4に示す7つのアプリケーション登録してもらう。これらのアプリケーション群は、ソフトウェア開発者が業務で使用することを想定したものである。タスクの設定は、アプリケーションのexe名を何らかの方法で調べて登録してもらう。被験者が7つのアプリケーションを登録できたら、終了してもらおう。exe名がわからないアプリケーションは、XXXと入力してもらう。この作業の正しい設定は以下の図5のようになる。図5のように、タスクごとにアプリケーションを列挙することで設定できる。アプリケーションのexe名をパイプラインでつなげて登録する。しかし、登録の際、プログラム名とexe名が一致しないものも存在する。この作業では、登録するアプリケーションのexe名がわからない困難を想定している。

作業4: サーバとの通信設定（制限時間15分）

被験者には、設定ファイルの通信設定部分を書き換えてもらう。使用するサーバのIPアドレス、ポート番号は、予め被験者へ渡す配布資料に示しておく。被験者が通信設定部分を書き換えられたと思ったら作業を終了してもらおう。この作業の正しい設定は以下の図6のようになる。図6にある“server\_connect.enable”は初期値は0であり、そのままではサーバと通信できない。この作業では、サーバを使用しない設定のまま作業を終了する可能性を想定している。

作業5: サーバとの接続（制限時間10分）

被験者には、TaskPitを起動し、サーバとの接続を確認してもらう。サーバ

ブラウザ: InternetExplore, Google Chrome, Mozilla Firefox  
メール: Outlook, Google Chromeから立ち上げたGmail  
プログラミング: Visual Studio 2015, Eclipse

図4 被験者が登録するアプリケーション

```
[tasks]
ブラウザ = iexplore.exe\textbar chrome.exe\textbar firefox.exe
メール = Outlook.exe\textbar chrome.exe:gmail
プログラミング = devenv.exe\textbar eclipse.exe
```

図5登録するアプリケーションの設定例

```
[ipaddress]
ip = 192.168.2.25
port = 54322
server\_connect\_enable = 1
```

図6通信設定の例

画面は被験者各自のPCに表示しておく。また、通信設定を間違えていた場合、被験者は作業4をやり直さなければならない。故に手戻りを防ぐため、作業5の前に設定ファイルは正しくサーバと通信する内容に書き換えておく。被験者がサーバと接続できたと思ったら作業を終了してもらう。この作業では、プログラム中からTaskPitを見つけられず、起動ができない困難を想定している。

#### 作業6: 作業履歴の分析（制限時間30分）

被験者には、3.2節で述べたソフトウェア開発企業の作業履歴を分析してもらう。デスクトップに用意された開発者1名の3日分の作業履歴を分析し、あらかじめ用意された表に結果を記入してもらう。表2に作業履歴の分析時に記入する表を示す。被験者には作業履歴から、開発者のタスク毎の作業時間と、1日の作業時間から見たタスクの割合を記入してもらう。表2は、被験者のデスクトップ画面に用意されたxlsxファイルであり、分析結果は表の色がついた部分に記入してもらう。この作業は管理者が、タスクに費やした時間を分析するシチュエーションを想定している。また、作業履歴の形式を理解している実験者が5日分の作業履歴を分析するのに30分必要であったことから被験者の負担を考慮し、本実験では3日分の作業履歴を分析してもらう。被験者が表の黄色い部分すべてを埋めることができたなら、作業を終了してもらう。この作業では、TaskPitの出力する作業履歴が分析しにくく、分析に時間がかかり過ぎる困難を想定している。

付録Aに被験者へ配布した実験の手順書を添付する。図7に実験に使用したPCのデスクトップ画面を示す。作業環境を統一するため、デスクトップPC、ノートPCともにデスクトップ画面のアイコンは同じ配置にする。また、実際の開発環境に近づけるため、被験者にはインターネットを使用した検索を許可している。

表2 分析データ記入表

タスク名	5月20日		5月21日		5月22日	
	時間(時間:分)	率(%)	時間(時間:分)	率(%)	時間(時間:分)	率(%)
TaskPit						
エクセル						
ファイル操作						
ブラウザ						
プログラミング・デバッグ						
調査						
登録外						
合計	0:00	0.00%	0:00	0.00%	0:00	0.00%

※ 時間は1日の合計作業時間, 率は1日の合計作業時間におけるそのタスクの割合

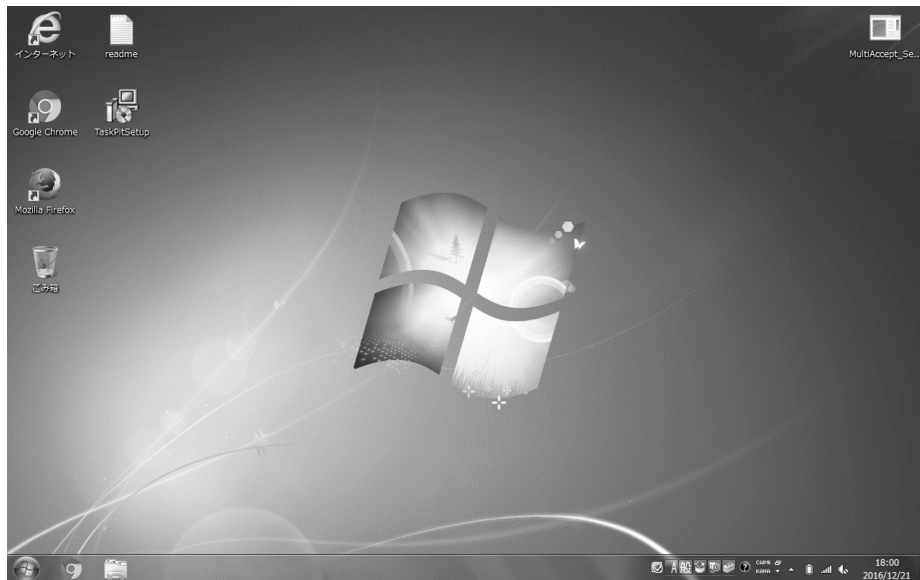


図7 デスクトップ画面

#### 4.1.3 タスク中の作業計測

被験者には1~6の各作業が終わるごとに手を挙げてもらう。被験者の作業中の動作として、作業にかかった時間、ミスの回数、左右クリック数、打鍵数、作業量を計測する。作業にかかった時間は実験者がストップウォッチで計測する。左右クリック数と打鍵数は、“タイプ数カウンター<sup>3)</sup>”を使用して計測する。作業量は、ディレクトリの移動・ファイルオープン・アプリケーションの起動が行われるごとに1ずつ増加する値であり、“ログレコ<sup>4)</sup>”で測定する。また、ミスの回数は、作業3, 4に存在する評価指標である。作業3では、設定する7つのアプリケーションのexe名が間違えている数を示す。作業4では、サーバとの接続に必要な3つの項目”IPアドレス”, ”使用ポート”, ”サーバ使用の有無”の内、間違えている数を示す。

#### 4.1.4 アンケート

すべてのタスク終了後、被験者にはアンケートに回答してもらう。客観的な作業履歴からでは計測できない、作業に対する被験者の主観的評価を得るためにアンケートを実施する。被験者の主観的評価得ることで、ユーザから見た各作業の難易度を考察する。表3にアンケート項目と、各項目に対応する作業を示す。被験者には、表3の7項目に対して、5段階評価をしてもらう。5段階評価の尺度を、図8に示す。評価尺度は、5が最も質問に当てはまる評価となっており、1は最も当てはまらない評価である。また、アンケートでは被験者のコンピュータ使用歴も尋ねている。これはPC使用歴と、作業の速度や正確性との関係性を考察するためである。付録Bに被験者へ配布したアンケートを添付する。

表3 アンケート項目と対応する作業

アンケート項目	対応する作業
TaskPitのインストールは容易でしたか?	作業1
設定ファイルの場所はわかりやすかったですか?	作業2
記録するタスクの設定は容易でしたか?	作業3
アプリケーションのexe名はわかりやすかったですか?	作業3
サーバとの通信設定は容易でしたか?	作業4
TaskPitの起動は容易でしたか?	作業5
ログの分析は容易でしたか?	作業6

<sup>3)</sup>タイプ数カウンター, <http://www.vector.co.jp/soft/win95/util/se399353.html>

<sup>4)</sup>ログレコ、無料PCログ記録ツール - Easy Asset Manager, <http://www.easymanager.com/logreco.html>



5 : 完全に当てはまる
4 : 当てはまる
3 : どちらともいえない
2 : 当てはまらない
1 : 全く当てはまらない

図 85 段階評価の尺度

#### 4.1.5 分析方法

被験者実験では、作業1~6にかかった時間、ミスの回数、左右クリック数、打鍵数、作業量を分析する。例えば、左右クリック数や作業量が多い場合は、ファイルを探すためにフォルダ間を頻繁に移動していると予測できる。打鍵数が多い場合、設定ファイルの操作やインターネットでの検索を頻繁に行っていると予測できる。作業にかかった時間が長い、またはミスの回数が多い場合は、難易度の高い作業であると予測できる。更に、あらかじめ各作業を達成できる最短の作業量を求めておき、被験者の作業量からどれだけ離れているかを算出することで、達成が困難な作業を求める。被験者10人分の作業データの平均、標準偏差をもとに、TaskPitのインストールや作業履歴の分析において困難である作業を予測する。また、アンケートから得られた被験者の主観評価も、作業の困難さを評価する際に用いる。

## 4.2 性能評価実験

性能評価実験では，3章で述べた“動作環境が限定される”，“ハードウェアへの負荷が大きい”，“通信経路を圧迫する”の3要因を検証する．

### 4.2.1 動作環境の確認

“動作環境が限定される”を検証するために，様々なWindows OS搭載のコンピュータにTaskPitをインストールし，その動作を確認する．実験にはWindows 8.1搭載のデスクトップPCと，Windows 7, 10の両方を搭載したノートPCの2台のコンピュータを使用する．デスクトップPCは64bitアーキテクチャであり，ノートPCは32bitアーキテクチャである．実験では，それぞれのPCにTaskPitをインストールし，動作確認のため1時間分の作業を計測する．Windows 7, 8.1, 10全ての環境にインストールできるか検証し，動作確認のため1時間分の作業履歴が存在するかを確認する．

### 4.2.2 CPU使用率の測定

“ハードウェアへの負荷が大きい”を検証するために，TaskPit動作時のCPU使用率を測定する．実験にはWindows 8.1搭載のデスクトップPCを使用した．CPUは第4世代Intel Core i7プロセッサ，メインメモリは8GBのものを搭載している．測定には，Visual Studio 2015のCPU使用率ツールを用いる．測定データは，実験者の卒業研究に関する作業，合計24時間分である．TaskPitの設定ファイルはサーバと通信するように変更する．それ以外の設定は初期状態のまま使用する．その後，Visual Studio 2015のCPU使用率ツールの出力結果より，CPUの平均使用率とCPU使用率のピークが現れる周期を確認する．

### 4.2.3 ネットワーク通信量の計測

“通信経路を圧迫する”を検証するために，TaskPit動作時のネットワーク通信量を計測する．測定データは，実際のソフトウェア開発組織で計測した18人分の作業履歴，合計3000時間分を使用する．まず，18人分の作業履歴ファイルの総データ量を求める．その後，作業履歴からTaskPitが動作している合計時間を算出する．そして，ファイルの総データ量を合計時間で除することで，単位時間あたりにTaskPitが送信するデータ量を求める．求めたネットワーク通信量と，日本や世界の平均インターネット通信速度を比較することで，TaskPitの通信経路の占有率を算出する．

## 5 結果と考察

### 5.1 被験者実験の結果

#### 5.1.1 計測データの平均と標準偏差

被験者実験で得られた計測データの加算平均、標準偏差をそれぞれ表4、表5に示す。

まず、計測データを平均値の観点から分析する。時間に注目すると、作業2, 3, 5, 6には時間がかかっていた。特に、作業6は全被験者が時間内に達成できなかった。作業6の被験者の動きを分析すると、多くの被験者がExcelの使い方を実験中に検索していた。故に、Excelの使い方を熟知していない人にとって、作業履歴の分析は困難であると伺える。打鍵数に着目すると、作業3, 6が特に多かった。被験者の作業履歴を分析すると、いずれの被験者も作業3, 6ではインターネットを使用して検索していた。この結果から、作業3, 6はインターネットを使って調べなければ達成が難しい作業であると考えられる。左右クリック数に着目すると、作業6が特に多かった。作業6はExcelを用いて分析するため、セル移動によるクリック数の増加が考えられる。分析する作業履歴が多いほどクリック数も増えるため、セル移動が少なくなるような作業履歴を出力できれば、ユーザの分析作業に対する負担は軽くなると思われる。作業量に着目すると、作業2, 6が特に多かった。作業2は設定ファイルを探すために、多くの被験者がエクスプローラーの検索機能を使用していた。検索の際にディレクトリの移動が発生し、作業量が増加したと考えられる。1つのファイルを探すために、平均43.1という多くの作業量を必要としており、設定ファイルが探しにくい場所にあると伺える。また、作業6では多くの被験者が、Excelとインターネットの検索画面を行き来していたため、作業量が増加している。故に作業量の観点からも、Excelの使い方を熟知していない人にとって、作業履歴の分析は困難であると考えられる。

表4 各作業の計測データ（加算平均）

作業	時間（秒）	打鍵数	左右クリック数	作業量	ミスの回数
1	102.9	3.0	35.7	13.8	-
2	229.4	29.7	83.4	43.1	-
3	471.3	150.9	83.2	38.6	0.8
4	48.5	28.9	15.3	1.7	0.2
5	192.9	23.6	84.7	37.5	-
6	時間切れ	736.2	500.8	136.8	-

次に、計測データを標準偏差の観点から分析する。各計測データの標準偏差が特に大きかったのが、作業2, 5である。作業2, 5では被験者ごとの時間の偏りが顕著である。被験者ごとの作業2の所要時間を、図9に示す。図9は横軸が被験者、縦軸が作業2の所要時間を示している。B, E, H, Jの4名は設定ファイルを探すために、エクスプローラーの検索機能を使用していた。TaskPitの設定ファイルは、隠しフォルダの中に存在しており、この方法では検索に時間がかかる。一方、その他の被験者はいずれも、Windowsの“すべてのプログラム”から探すことで、短時間で設定ファイルを見つけ出している。この結果から、“すべてのプログラム”の利用が、設定ファイルを開く時間に結びついていると考えられる。故にユーザに対して、設定ファイルを開く際に“すべてのプログラム”を利用するよう促すことができれば、この作業の所要時間は大幅に短くなると思われる。被験者ごとの作業5の所要時間を、図10に示す。図10は横軸が被験者、縦軸が作業5の所要時間を示している。A, E, Jの3名は、TaskPitの起動に著しく時間がかかっている。この3名はTaskPitの実行ファイルを探すために、エクスプローラーの検索機能を使用していた。一方その他の被験者は、“すべてのプログラム”から、TaskPitの実行ファイルを見つけ出している。A, E, Jの3名は、“すべてのプログラム”を使用しなかったため起動に時間がかかっていた。こちらもユーザに対して、TaskPitの実行ファイル探す際に“すべてのプログラム”を利用するよう促すことができれば、所要時間が短くなると考えられる。

表5 各作業の計測データ（標準偏差）

作業	時間（秒）	打鍵数	左右クリック数	作業量	ミスの回数
1	64.4	4.6	17.1	9.4	-
2	187.4	41.6	67.0	41.0	-
3	225.5	87.7	34.1	27.8	2.1
4	17.6	8.1	4.1	1.5	0.6
5	221.7	37.0	87.5	40.0	-
6	時間切れ	362.9	136.2	47.8	-

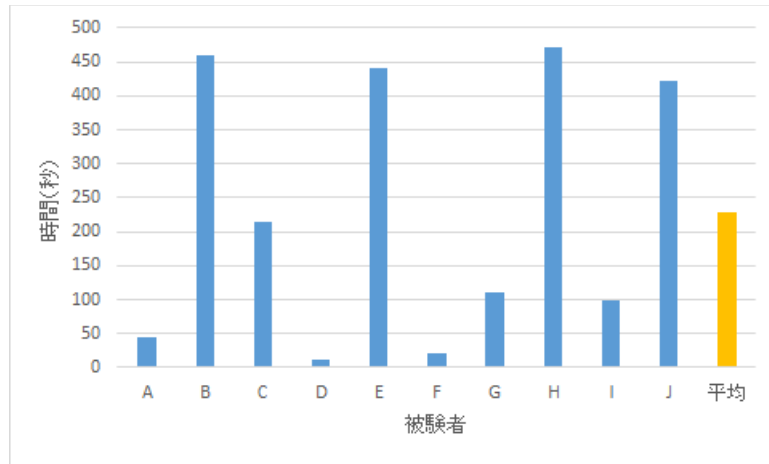


図 9 作業 2 の所要時間

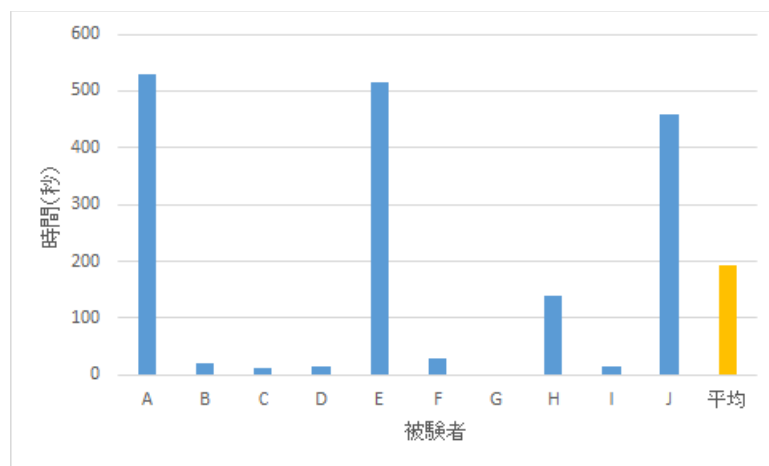


図 10 作業 5 の所要時間

### 5.1.2 最小作業量との比較

被験者の作業と、各作業を最小で達成したものを比較する。比較により、被験者の作業と最短の作業が、どれだけ離れているかわかる。各作業を最小で達成したものである最小作業量のデータを、表6に示す。表6は、被験者実験の各作業1～6を、実験者が最短で達成した時の計測データである。計測データの平均を最小作業量で除したデータを、表7に示す。表7の時間に注目すると、作業2, 5は最小作業量よりも著しく時間がかかっていた。特に作業2の設定ファイルのオープンでは、最短であれば5秒程度で達成できるが、被験者実験では平均200秒近くかかっており、設定ファイルの見つけにくさが伺える。作業量に着目すると、作業2, 3, 5は最小作業量よりも著しく多かった。特に作業3のタスクの設定においては、被験者全員がインターネットで検索していたため、作業量が増加していた。

最小作業量との比較から、まず設定ファイルの見つけにくさが、TaskPitの導入を困難にしていると考えられる。1つのファイルを探すためだけに、多くの時間や作業を費やすのは非効率的である。故に、設定ファイルの場所や、効率の良い開き方をユーザに提示すべきである。また、多くの被験者がアプリケーションのexe名を検索することに時間を費やしていた。このことから、exe名がわからなくともアプリケーションを登録できる仕組みが、作業時間の短縮に効果的であると考えられる。

表6 各作業の最小作業量

作業	時間(秒)	打鍵数	左右クリック数	作業量
1	11	0	12	3
2	5	0	11	2
3	58	68	14	1
4	17	13	12	1
5	8	0	11	2
6	-	-	-	-

表7 計測データの平均÷最小作業量

作業	時間(秒)	打鍵数	左右クリック数	作業量
1	9.4	-	3.0	4.6
2	45.9	-	7.6	21.6
3	8.1	2.2	5.9	38.6
4	2.9	2.2	1.3	1.7
5	24.1	-	7.7	18.8
6	-	-	-	-

### 5.1.3 アンケートの結果

アンケートの結果の平均値を、表8に示す。また、それぞれの質問に対する評価の度数分布を図11~17に示す。アンケート結果より、設定ファイルの場所やアプリケーションのexe名に関して、わかりにくいと感じた被験者が存在しており、改善が必要であると考えられる。ログの分析に関しては、すべての被験者が非常に低い評価をつけており、根本的な改善の必要性が示唆される。

また評価の度数分布から、質問2と質問3において被験者の評価が2分していることがわかる。質問2で低い評価をつけた被験者は、設定ファイルの検索に時間がかかっていた。一方、質問2で高い評価をつけた被験者は、設定ファイルの検索の時間が短かった。質問3で低い評価をつけた被験者は、タスクの設定に時間がかかっていた。しかし、質問3で高い評価をつけた被験者は、タスク設定の時間が短かった。このように設定ファイルのオープンとタスクの設定は、被験者の能力によって評価が大きく変わっている。

表8 アンケートの結果

質問番号	アンケート項目	評価の平均
1	TaskPitのインストールは容易でしたか?	4.9
2	設定ファイルの場所はわかりやすかったですか?	3.8
3	記録するタスクの設定は容易でしたか?	4.3
4	アプリケーションのexe名はわかりやすかったですか?	3.2
5	サーバとの通信設定は容易でしたか?	4.4
6	TaskPitの起動は容易でしたか?	4.3
7	ログの分析は容易でしたか?	1.1

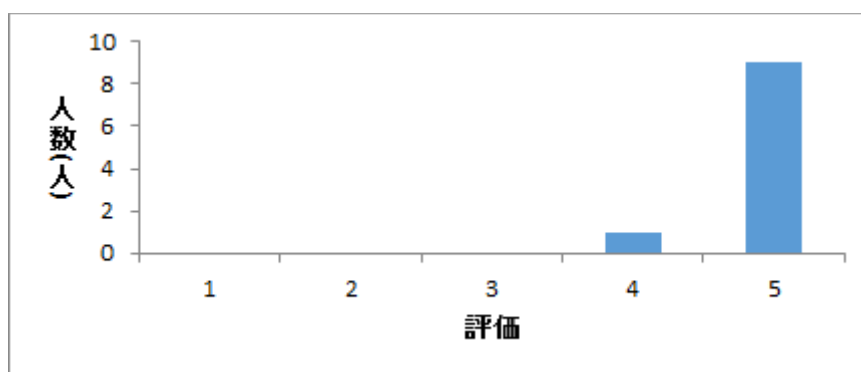


図11 度数分布 (質問1: TaskPitのインストールは容易でしたか?)

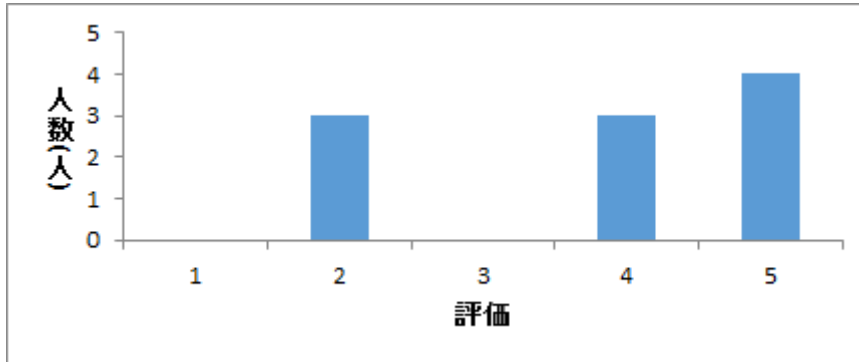


図 12 度数分布 (質問 2: 設定ファイルの場所はわかりやすかったですか?)

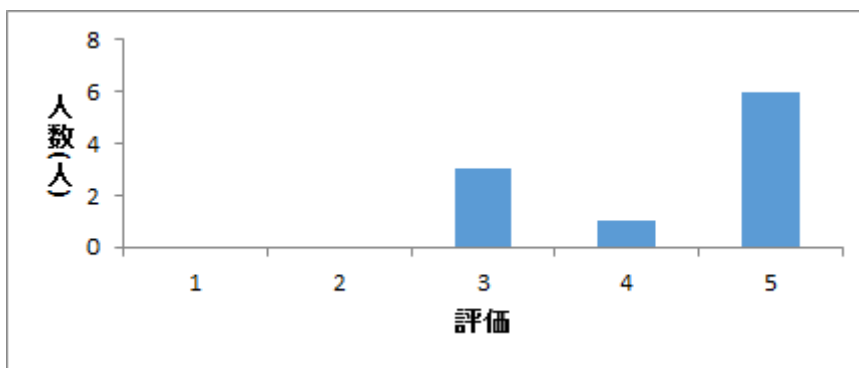


図 13 度数分布 (質問 3: 記録するタスクの設定は容易でしたか?)

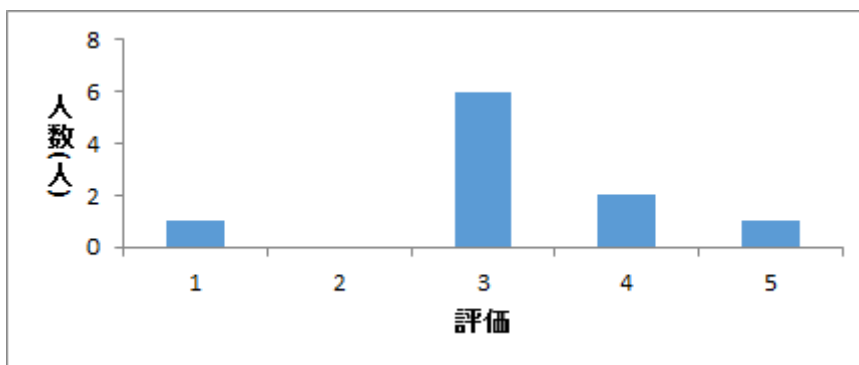


図 14 度数分布 (質問 4: アプリケーションの exe 名はわかりやすかったですか?)



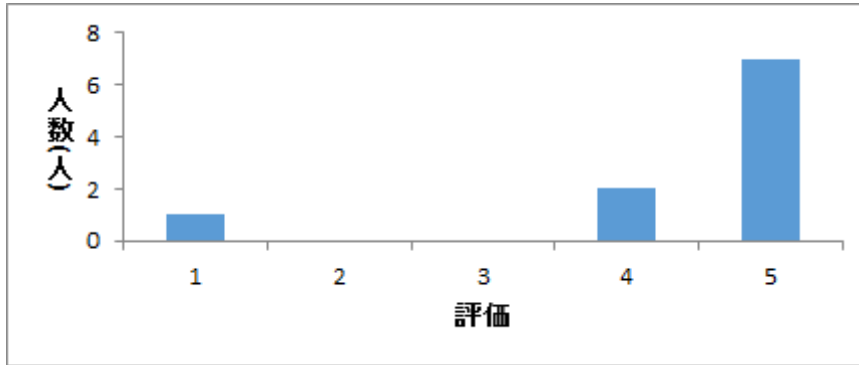


図 15 度数分布 (質問5:サーバとの通信設定は容易でしたか?)

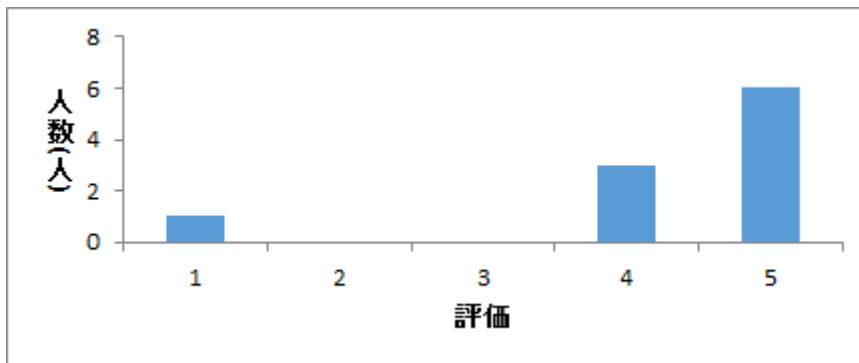


図 16 度数分布 (質問6:TaskPitの起動は容易でしたか?)

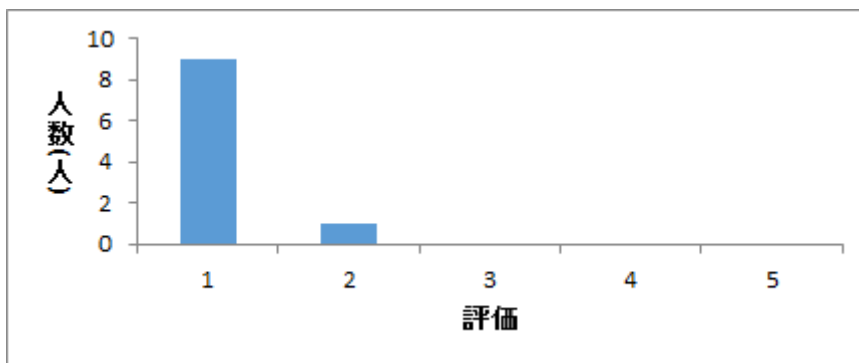


図 17 度数分布 (質問7:ログの分析は容易でしたか?)

## 5.2 性能評価実験の結果

### 5.2.1 動作環境の確認

“動作環境が限定される”を検証するために、様々なWindows OS搭載のコンピュータにTaskPitをインストールし、その動作を確認した。その結果、Windows 7, 8.1, 10搭載のPC全てにインストールすることができた。本実験では、インストール完了の画面が出てきた時点で、インストール作業達成と判断した。また、それぞれのPCにインストールしたTaskPitの動作を確認するため、実験者の卒業研究に関係する作業を1時間分記録した。その結果、すべてのPCに1時間分の作業履歴が存在すると確認できた。故に、OSのバージョンの違いがTaskPitの導入を妨げている可能性は低いと考えられる。

次に、本実験では判断できないような“動作環境が限定される”要因を考察する。現在PCのアーキテクチャは64bitが主流である。しかし、64bitアーキテクチャのPCで検証したOSは、Windows 8.1のみである。故に、64bitアーキテクチャの環境で動作を確認していないWindows 7, 10については、今後検証する必要がある。

### 5.2.2 CPU使用率の測定

“ハードウェアへの負荷が大きい”を検証するために、TaskPit動作時のCPU使用率を測定した。測定の結果、24時間TaskPitを動作させたときのCPUの平均使用率は1.2%、最大使用率は13.8%であった。測定結果の内、ある20秒間を切り出したグラフを、図18に示す。図18の横軸は時間、縦軸はCPU使用率を示している。図18のように、通常時はCPU使用率が1.0%未満であるが、5分ごとに約10秒間、使用率が10%程度まで上昇した。5分という周期は、TaskPitがサーバに作業履歴を送信するタイミングと一致する。この結果から、TaskPitが作業を計測する際のCPU使用率は1.0%未満であるが、サーバと通信する際には一時的に10%程度まで上昇することがわかる。

実験者が卒業研究の作業と並行してTaskPitを使用した際、作業履歴の送信時にPCの動作が重くなることはなかった。しかし実際のソフトウェア開発現場で、どのような影響がでるのかは本研究で検証できていない。故にCPU使用率10%という値が、開発現場においてどれだけの影響を与えるのかは、今後検証する必要がある。

### 5.2.3 ネットワーク通信量の計測

“通信経路を圧迫する”を検証するために、TaskPit動作時のネットワーク通信量を計測した。分析結果を、表9に示す。まず、計測した作業履歴ファイルの総データ量は、678.96[Mbit]であった。また、TaskPitが動作している合計時間は、10735921[秒]であった。この結果から、TaskPitが1秒あたりに送信するデータ量は、6.3242 ×

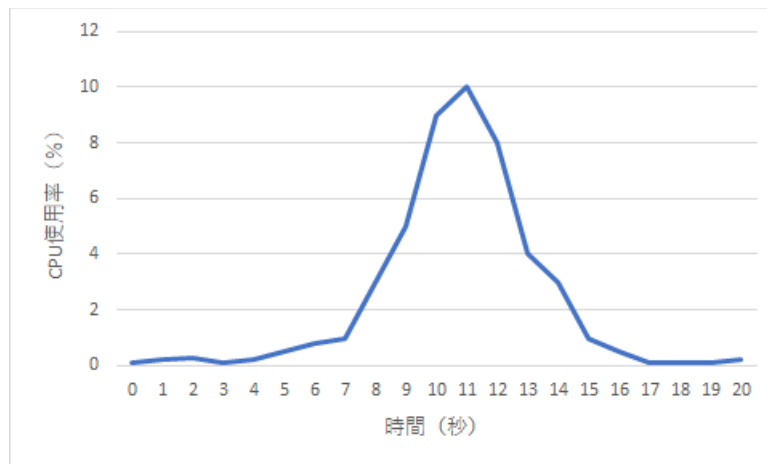


図 18 CPU 使用率の例

$10^{-5}$ [Mbit]となる。しかし、TaskPitは仕様上データの送信は毎秒行わず、初期設定の状態では5分ごとにデータを送信をする。故に、TaskPitが1度に送信するデータ量は、 $1.8973 \times 10^{-5}$ [Mbit]となる。ここで、日本や世界の平均インターネット通信速度と、TaskPitの送信するデータ量を比較する。2016年現在、日本と世界の平均インターネット通信速度は、それぞれ17.1[Mbps]、6.1[Mbps]である[14]。このデータを基準にTaskPitの通信経路占有率を求めると、日本では0.00111%、世界では0.00311%となり、いずれの占有率も非常に小さいことがわかる。故に、“通信経路を圧迫する”という要因がTaskPitの導入を妨げている可能性は低いと考えられる。

表 9 TaskPit の通信経路占有率

項目	日本	世界
TaskPit が1度に送信するデータ量 (Mbit)	$1.8973 \times 10^{-5}$	
平均インターネット通信速度 (Mbps)	17.1	6.1
TaskPit の通信経路占有率 (%)	0.00111	0.00311

## 5.3 改善策の提案

被験者実験と性能評価実験の結果を踏まえて、TaskPitの改善に必要な点を考察する。本研究では、以下の4点の改善が必要であると結論づける。

### 5.3.1 設定ファイルの場所

設定ファイルを探すために、多くの被験者が時間をかけていた。設定ファイルが隠しフォルダの中に存在するため、エクスプローラーを用いた検索にも時間がかかってしまう。“すべてのプログラム”から設定ファイルを探さなかった場合、見つけるまでに非常に時間がかかる傾向にある。

この問題を解決するためには、TaskPitのインストーラや取扱説明書などに、設定ファイルの場所を明記することが効果的であると考えられる。“すべてのプログラム”を利用した被験者は短時間で設定ファイルを開いていた。ユーザに“すべてのプログラム”の利用を促すことができれば、設定ファイルを開く時間を短縮できると推測する。また、インストーラに「設定ファイルのショートカットをデスクトップ画面に生成する機能」を追加することも効果的であると考えられる。デスクトップ画面はユーザの目につきやすい場所であるため、設定ファイルを探す時間を短縮できる可能性が高い。

### 5.3.2 計測するアプリケーションの設定方法

計測するアプリケーションのexe名を知るために、多くの被験者が時間をかけていた。プログラム名とexe名が異なるアプリケーションがあり、この存在が被験者にとって設定を困難にするものであると考えられる。

この問題を解決するためには、設定ファイルのユーザインタフェースの改良が必要であると考えられる。例えば、計測するアプリケーションをドロップダウンリストから選択できるように改良すると、exe名を知らない場合でもアプリケーションを登録できる。この機能は、ユーザのPCにインストールされているソフトウェアを自動で取得し、リストに表示することで実現する。

### 5.3.3 TaskPitの起動

TaskPitの実行ファイルを探すために、多くの被験者が時間をかけていた。設定ファイルと同じく、“すべてのプログラム”から設定ファイルを探さなかった場合、見つけるまでに非常に時間がかかる傾向にある。

この問題を解決するためには、設定ファイルと同様に、TaskPitのインストーラや取扱説明書などに、実行ファイルの場所を明記することが効果的であると考えられる。ユーザに“すべてのプログラム”の利用を促すことができれば、実行ファイル

を探す時間を短縮できると推測する。また、同様にインストーラに対して「実行ファイルのショートカットをデスクトップ画面に生成する機能」を追加することも効果的であると考えられる。

#### 5.3.4 作業履歴の出力形式

実験では、被験者全員が作業履歴の分析を達成できなかった。これには作業履歴の出力形式に問題があると推測する。多くの被験者は、作業中にExcelの使い方を検索していた。このことから、現在の出力形式は分析するユーザに対して、新しい知識を求めていると考えられる。

この問題を解決するためには、作業履歴を分析したデータを出力できるように、TaskPitを改良することが効果的であると考えられる。あらかじめ作業履歴をグラフ化したものや、1日作業時間における割合などを出力できれば、ユーザは分析に時間を費やすことなく作業状況を知ることができる。

## 6 おわりに

本研究では、ソフトウェア開発者のための作業計測システムであるTaskPitをより多くの開発現場に導入するため、導入を妨げている要因を調査した。過去の企業へのシステム導入事例をもとに、システムの導入を妨げる要因を調べ、それらがTaskPitに当てはまるかどうかを検証した。検証のため被験者実験と性能評価実験を行った。被験者実験では、TaskPitを実際にインストールし、出力データを分析してもらうことで“インストール時に困難な操作・用語が存在する”，“システムが使いにくい”，“動作環境が限定される”という3つの要因を検証した。性能評価実験では、TaskPitを様々なOSのPCにインストールし、動作を確認することで“動作環境が限定される”という要因を検証した。また、CPUの使用率とデータ通信量を測定することで、“ハードウェアへの負荷が大きい”，“通信経路を圧迫する”という2つの要因も検証した。

被験者実験の結果、TaskPitの設定ファイルと実行ファイルを探すことに、多くの被験者が時間をかけていた。また、TaskPitが計測するアプリケーションのexe名を調べることに時間をかけていた。この結果から、設定ファイルと実行ファイルの場所をTaskPitのインストーラや取扱説明書に明示することで、ファイルを探す時間を短縮できると考えられる。また、設定ファイルのユーザーインターフェースの改良も必要であると考えられる。登録するアプリケーションをドロップダウンリストから直接選択できるようにすることで、ユーザがアプリケーションのexe名を知らなくとも設定できる。また、分析作業はすべての被験者が達成できなかったため、出力データ形式を根本的に変える必要がある。既に分析済みのデータを出力することで、ユーザの分析に対する負担が減ると考えられる。

性能評価実験の結果、TaskPitは、検証に使用したWindows7, 8.1, 10全ての環境にインストールできた。しかし、64bitアーキテクチャでの動作確認が不十分であるため、今後検証する必要がある。CPUの使用率を調査した結果、平常時は使用率が1%未満だが、データ送信の前後で使用率が10%程度まで上昇した。CPUの使用率が他のシステムにどの程度影響を与えるかは、今後検証する必要がある。また、TaskPitのデータ通信量は非常に小さいため、通信経路は圧迫しないことが示唆される。

本研究の今後の発展として、今回調査できなかった要因の検証が挙げられる。実験の難易度が高いことから、“導入することで得られる効果が明確でない”，“充実したサポートを受けることができない”という2つの要因は検証できなかった。この2つの要因は、実験するにあたって企業との連携が必要である。今後は、実際のソフトウェア開発組織と連携して、これらの今回実験できなかった要因を検証したい。

## 謝辞

本論文の執筆および研究の進行にあたって、多くの方にご協力をいただきました。この場を借りてお礼申し上げます。

指導教員である上野秀剛講師には研究の助言や論文の添削など、常日頃からご指導をいただきました。また、日常生活においても様々なアドバイスをいただきました。心より感謝を申し上げます。

本論文の作成にあたって、松村寿枝准教授には査読をしていただきました。心より感謝を申し上げます。

同じ上野研究室の皆様には発表練習を通じて、的確なアドバイスをいただきました。心より感謝を申し上げます。

またお忙しい中、被験者実験に参加してくださった皆様にも感謝を申し上げます。ありがとうございました。

## 参考文献

- [1] W. S. Humphrey, “パーソナルソフトウェアプロセス入門,” 共立出版.
- [2] 大橋亮太, 上野秀剛, 門田暁人, 荒木健史, 山田欣吾, 松本健一, “自動計測データと機械学習に基づくソフトウェア開発の作業目的の推定,” コンピュータソフトウェア, Vol.33, No.2, pp.139-150, June 2016.
- [3] S. Stumpf, X. Bao, A. Dragunov, T. G. Dietterich, J. Herlocker, K. Johnsrude, L. Li, and J. Shen, “Predicting User Tasks: I Know What You’re Doing!,” InProc, the 20th National Conference on Artificial Intelligence (AAAI), 2005.
- [4] 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, November 2008.
- [5] 門田暁人, 上野秀剛, 荒木健史, 山田欣吾, 松本健一, “ソフトウェア開発企業における開発タスクの自動計測,” 第20回ソフトウェア工学の基礎ワークショップ, Vol.20, pp.257-262, December 2013.
- [6] 池田祥平, 上野秀剛, “第三者によるソフトウェア開発作業評価のための作業記録の保護手法,” 情報処理学会研究報告ソフトウェア工学研究会, Vol.2014-SE-185, No.2, pp.1-8, July 2014.
- [7] “なぜ多くの日本企業がERP導入に失敗したのか: ITpro Active”, <http://itpro.nikkeibp.co.jp/article/Active/20120305/384596/>, 2016/10/27 アクセス.
- [8] “ERP導入失敗事例に学ぶ、3つの導入成功ポイント | ITトレンド”, [http://it-trend.jp/erp/article/failure\\_case](http://it-trend.jp/erp/article/failure_case), 2016/10/27 アクセス.
- [9] “[業務効率化]ERPシステム導入が失敗する10の理由 | Dell 日本”, [http://www.jp.dell.com/content/topics/global.aspx/bizportal/sb360articles/ja/pr\\_enterpresezine\\_detail\\_69?c=jp&l=ja&cs=jpbsd1](http://www.jp.dell.com/content/topics/global.aspx/bizportal/sb360articles/ja/pr_enterpresezine_detail_69?c=jp&l=ja&cs=jpbsd1), 2016/10/27 アクセス.
- [10] “システム開発の失敗事例: システム開発会社評判”, <http://shisutemukaihatsu.net/cat0007/>, 2016/10/27 アクセス.
- [11] “1.なぜ8割の企業がシステム導入に失敗するのか | 効果大のシステム開発で、あなたの実績に。オーシャン・アンド・パートナーズ”, <http://www.ocean-ap.co.jp/attention/failure.html>, 2016/10/27 アクセス.
- [12] “失敗例から学ぶ、POSシステム導入 | ITトレンド”, [http://it-trend.jp/pos\\_system/article/failure\\_case](http://it-trend.jp/pos_system/article/failure_case), 2016/10/27 アクセス.



- [13] “Windows 7/10/XP が増加 - 12月 OS シェア”, <http://news.mynavi.jp/news/2017/01/03/043/>, 2017/1/4 アクセス.
- [14] “世界のインターネット平均速度は6.1Mbps・最大速度の平均は36.0Mbpsで日本は世界第9位 - GIGAZINE”, <http://gigazine.net/news/20161005-akamai-internet-report/>, 2017/1/7 アクセス.

## 付録

### A 被験者配布資料

※デスクトップにTaskPitの説明書であるreadme.txtがあります。お使いください。  
※インターネットは自由に使用して頂いて結構です。  
※各作業1~6が終了するごとにPCから手を放して挙手してください。  
※各作業1~6は1つずつ分かれています。続けて行わないでください。  
※括弧内の時間は制限時間です。ストップウォッチで計測します。

#### 実験1 (インストール)

1. TaskPitをインストールする(10分)

⇒ TaskPitをインストールできたと思ったら作業1終了です。

2. 設定ファイルを開く(10分)

⇒設定ファイルを開くことができたと思ったら作業2終了です。

3. 記録するタスクを以下のように変更(30分)

ブラウザ: InternetExplore, Google Chrome, Mozilla Firefox  
メール: Outlook, Google Chromeから立ち上げたGmail  
プログラミング: Visual Studio 2015, Eclipse

※左から順番に設定をお願いします。

※わからないアプリケーションはXXXと入力し、とぼしてください。

⇒上記の7つのアプリケーションが設定できたら作業3終了です。

4. サーバと通信できるように設定ファイルを書き換える(15分)

※サーバは既に用意されています。下記のものを使用してください。

※IPアドレス: 192.168.2.25, 使用ポート: 54322

⇒サーバとの通信に必要なだと思う部分を書き換えたら作業4終了です。

5. TaskPitを起動し、サーバとの接続を確認する(10分)

⇒サーバとの接続を確認できたと思ったら作業5終了です。

#### 実験2 (分析)

6. デスクトップにあるTaskLog.xlsxから作業履歴を分析し、analysis.xlsx内にある表の黄色い部分を記入する。(30分)

⇒表の黄色い部分がすべて記入できたら作業6終了です。

## B 被験者実験アンケート

学年：      クラス：      氏名： \_\_\_\_\_

### 1. 使いやすさの5段階評価

- 1) TaskPitのインストールは容易でしたか？      (      )
- 2) 設定ファイルの場所はわかりやすかったですか？      (      )
- 3) 記録するタスクの設定は容易でしたか？      (      )
- 4) アプリケーションのexe名はわかりやすかったですか？      (      )
- 5) サーバとの通信設定は容易でしたか？      (      )
- 6) TaskPitの起動は容易でしたか？      (      )
- 7) ログの分析は容易でしたか？      (      )

上記の (      ) の中に，評価に応じて 1 ～ 5 の数字を入れてください。

- 5：完全に当てはまる
- 4：当てはまる
- 3：どちらともいえない
- 2：当てはまらない
- 1：全く当てはまらない

### 2. PC使用歴の確認

あなたが初めてPCを使ってから何年経ちますか？      (      ) 年