

Eclipse 寄付バッジによるバグレポート応答時間の影響分析

中才 恵太郎^{1,a)} 畑 秀明^{1,b)} Raula Gaikovina Kula^{1,c)} 松本 健一^{1,d)}

概要: 規模の大きなオープンソースソフトウェアプロジェクトの維持に資金確保は重要であり寄付が集められている。その寄付収集のため、オープンソースソフトウェアプロジェクトの Eclipse はバグ管理システム Bugzilla において一定額以上の寄付者にバッジ（寄付バッジと呼ぶ）を付与するという特典を用意している。寄付バッジを持つ貢献者からのバグレポートが早く対応されるとすれば、効用のある寄付バッジを得るために寄付を促進させることができる。しかし寄付バッジの影響はこれまで明らかになっていない。本稿では、バグレポート提出後、最初の応答までの時間が寄付バッジの影響を受けるのかを明らかにする。寄付バッジをもたないバグレポートと比較して、寄付バッジをもつバグレポートに対する開発者の応答時間を Quantile Difference in Differences (QDID) という手法で分析した。その結果、寄付バッジが応答時間を中央値で約 2 時間短縮することが明らかとなった。また、Eclipse コミュニティを対象としたアンケート調査では、寄付バッジの有無はバグレポート対応への優先度に大きな影響を与えないとの回答を得た。以上の結果から、Eclipse の寄付バッジは、明示的な優先事項ではないがシグナリング理論のシグナリングとして働き、寄付者に好ましい効用があるということがわかった。

1. はじめに

寄付はオープンソースソフトウェア（OSS）プロジェクトで重要な役割を果たす。OSS プロジェクトである LibreOffice は、2016 年に 3 年間で 20 万件の寄付を受けたと報告した。そして、OSS コミュニティの多様なエコシステムと個人の寄付者に頼ることにより、大規模な OSS プロジェクトにはただ一つの大規模な企業スポンサーは必要ないことを強調した [11]。このことから、持続可能な OSS プロジェクトを維持するためには効果的な寄付プログラムの管理が重要であることがわかる。

OSS における寄付は重要ではあるが、OSS プロジェクトへの寄付についてはほとんど研究されていない。寄付に影響を及ぼす原因についての研究は、SourceForge のデータを用いた研究 [6] があり、寄付の意思決定には、OSS プラットフォームとのコミットメント関係の影響を受けることが報告されている。著者らの以前の研究 [7] では、少数ではあるがコミッタは他のグループよりも、一人当たりの寄付額が高いことがわかっている。しかし、これまでの研究では、寄付者への寄付の実用的な効果は、明らかになっ

ていない。

Eclipse は寄付プログラム「Friends of Eclipse」を 2007 年 12 月に開始した。2014 年 11 月より、この寄付の特典として、バグ管理システム（Bugzilla）にバッジが表示され、寄付者がバッジにより区別できるようになっている。現在、35 ドル以上の寄付をした寄付者は、Friend of Eclipse の資格を 1 年間取得することができ、Bugzilla で表示される Friend of Eclipse Badge（以下寄付バッジ）を利用できる。しかし、寄付バッジが寄付者にどのようなメリットをもたらすのかは明らかではない。

本稿では、統計的因果推論に基づいて、寄付バッジステータスを持つバグレポート提出者がバグレポートを提出した時、寄付バッジを持つことで応答時間（最初にコメントがつくまでの時間）に影響を与えるかを明らかにする。分析の結果、寄付バッジは、中央値において、約二時間応答時間を短縮する効果があることが明らかとなった。さらに、Eclipse コミュニティを対象とする調査では、寄付バッジはバグレポートの対応の優先度に大きな影響を与えないことがわかった。以上の結果から、Eclipse の寄付バッジは、明示的な優先事項ではないが、シグナリング理論のシグナリングとして働き、寄付者に好ましい効用があるということを考察した。

本稿の分析の一部は IEEE Software に採録予定 [8] である。本稿は IEEE Software の分析を拡張し、さらに Eclipse コミュニティに対して実施したサーベいをくわえ考察した。

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

a) nakasai.keitaro.nc8@is.naist.jp

b) hata@is.naist.jp

c) raura-k@is.naist.jp

d) matumoto@is.naist.jp

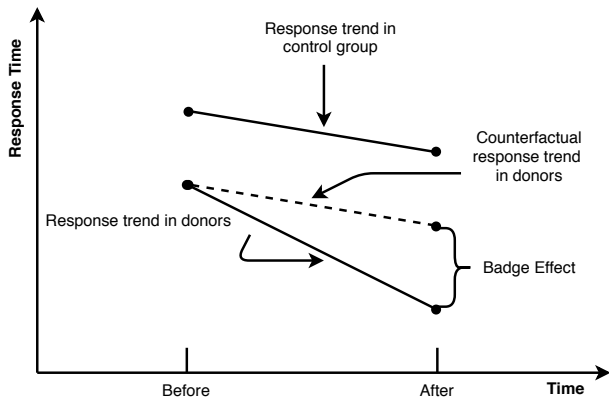


図 1 DID モデルの因果効果の例 (response time vs. before and after donation badge introduction)

2. 統計的因果推論

本節では、寄付バッジの効果の分析に使用する、統計的因果推論とその条件について述べる。統計的因果推論は、医療や、政策分野など幅広い分野で利用されている。

2.1 統計的因果推論

統計的因果推論とは、ある要因が結果に及ぼす影響の度合いを、統計データに基づいて判断するための方法論である [1]。調べたい要因が、ある結果に影響を与えていることを明らかにしたい場合、結果に影響を与える他の要因（交絡因子）を考慮した方法で分析しなければ、因果関係を明らかにすることはできない。調べたい要因のみを完全に切り除く制御された測定が可能であるなら交絡因子を完全に切り除くことができるため因果関係を明らかにすることができるが、人や生物が関わる問題の場合、このような制御された測定を行うことは不可能である。

そのため、統計的に交絡因子を排除し、因果関係を検証する方法として、ランダム化比較試験という方法があり、実験と呼ばれている。対象をランダムに 2 つの群に分け、一方だけに評価しようとしている治療や、政策などの介入を行い、両群を比較することにより介入の因果を推論する。ただし、この手法は実験コストが多くかかり、測定されている既存データ（観察データ）には適用できない。

一方、擬似実験という、観察データから因果関係を導き出す方法がある。擬似実験では、観察データから実験に類似する状況を作成することで、因果関係を示す。本稿では、観察データ（過去のバグレポート、寄付者リスト、開発履歴）を使用して、擬似実験を行うことにより寄付バッジの効果を分析する。

2.2 差分の差分分析

本稿では、差分の差分分析 (Difference in Difference) を観察データに適用することにより寄付バッジの効果を明

表 1 QDID で使用する説明変数

Metric	説明
Donor	寄付者であるか
Period	提出時間が寄付バッジの導入後か
Badge	寄付バッジを持つか
Enhancement	バグ深刻度 (Severity) が Enhancement か
Windows	報告対象が Windows OS か
Linux	報告対象が Linux OS か
MacOS	報告対象が Mac OS か
Component	属する component の応答日数の中央値
Community	属する component の応答日数の中央値
Time	寄付バッジ導入日を基準とした、提出時期
Relationship	最初の応答者とバグレポートのペアの出現回数

らかにする。差分の差分分析とは擬似実験の手法の一つであり、トレンドの影響を除外することにより因果推論を行う手法である。図 1 に差分の差分分析のモデルを示す。図 1 の control group は寄付者ではないが、寄付者のバグレポート提出者と類似するバグレポート応答時間のトレンドを持つ群である。一方、donors が寄付者のバグレポート提出者のバグレポート応答時間の群である。y 軸がバグレポート応答時間を示し、x 軸が寄付バッジ導入前と後を示す。実線が各グループの実際のバグレポート応答時間を示す。DID モデル (差分の差分分析で使用するモデル) では、control group の実際の測定値から、寄付バッジが導入されなかった場合の donors の測定値を仮定する (破線)。寄付バッジが導入されなかった場合の donors のバグレポート応答時間を仮定した値から実際の寄付バッジ導入後の測定値の差 (Badge Effect) を求めることで、寄付バッジの効果を推定する。本手法を適用するためにはある条件を満たす必要がある。本稿では 2.5 節に条件に関する議論を記述している。

2.3 傾向スコアマッチング

本稿では、差分の差分分析を行うための条件を満たすため (2.5 項)、擬似実験の手法の一つである、傾向スコアマッチングを利用している。傾向スコアマッチングはマッチング法の一つである。マッチング法とは、結果に影響する条件 (共変量) が類似するペアを作り、条件が同じグループを作成することで、比較可能な状態にする手法である。傾向スコアマッチングは共変量が多くある場合に利用される手法である。複数の共変量を一つの得点とした傾向スコアを使用し、割り付けられる確率が同じ (傾向スコア) ようなペアを作成する手法である。本手法によって、図 1 のように control group と donors を作成した。

2.4 分位点回帰分析

差分の差分分析を行う際、通常は線形回帰分析を用いて分析を行うが、本分析では線形回帰分析の代わりに分位点回帰分析を行う。分位点回帰分析で差分の差分法を行うこ

とを Quantile Difference in Differences (QDID) とよぶ。線形回帰分析は平均値に着目する場合に用いるのに対し、分位点回帰分析は分布に着目する場合に用いる。ソフトウェア開発データの分析には分布に偏りがあるため、中央値に着目して分析する場合が多い。本分析においても、バグレポートの応答時間は分散が大きいため、分位点回帰分析を行った方が、適切であると考えられる。分位点回帰分析の場合でも線形回帰分析と同じように、目的変数と説明変数を指定し、分析を行う。今回の場合は、目的変数をバグレポート提出時間からコメントが得られるまでの時間とする。表 1 に示す変数を説明変数とする。特に、差分の差分分析では (Donor, Period, Badge) の変数に着目して分析を行う。偏回帰係数 Badge が寄付バッジの効果量を示す。残りの説明変数はバグレポートの応答時間を予測する際に、調整のために用いた変数である。

2.5 因果推論を行う上での分析条件

SUTVA 条件: 因果効果を推定する場合、(a) 個体を特徴付ける潜在的な結果が他の個体の受ける処置に依存しないことかつ、(b) 処置内容を明確にすること以上 2 条件の仮定が置かれている。これを SUTVA (stable unit treatment value) 条件と呼ぶ。本分析においては、(a) ある寄付バッジを持つバグレポートの提出したバグレポートに対する応答時間は他のバグレポートが寄付バッジを持つことにより、干渉を受けるかどうかと (b) が論点となる。

寄付バッジを持つバグレポートの割合が多い場合、バグレポートは他のバグレポートの干渉を受けると考えられる。ただし、分析期間においては、Eclipse のバグ貢献者は 4,872 人に対し、寄付バッジを持つバグレポートは 55 人 (2016 年 12 月) [7] である。寄付バッジを持つバグレポートは少ないため、個人の寄付バッジを持つバグレポートへの干渉はごくわずかであると考えられる。そのため、(a) の条件は満たしている。(b) についてはバグレポート提出時以前、一年間において 35 ドル以上を寄付しているかどうかを条件としており、処置内容は明確である。

差分の差分分析の条件: 寄付バッジの因果効果を推定するため、差分の差分分析を行う。差分の差分分析によって、因果効果を正しく推定するためには、平行トレンド仮定と共通ショック仮定を満たす必要がある。平行トレンド仮定を満たすには、寄付バッジをもつバグレポートが提出したバグレポートに対する応答時間群と寄付バッジを持たないバグレポートが提出したバグレポートの応答時間群において、もし、Bugzilla 上で寄付バッジを表示するという実装がされなかった場合に両群は平行したトレンドとなる必要がある。共通ショック仮定を満たすには、寄付バッジの実装前の測定と寄付バッジ導入後の測定との間に結果に影響を与える可能性のある別の要素が起きていない、もしくは起きている場合でも両群に対し、同じく作用している必要

がある。

本分析では、平行トレンド仮定を満たすため 2 件の工夫をしている。1 件目は、仮想的に寄付バッジの有無以外は同じ経験を持つバグレポートを傾向スコアマッチングにより抽出してそのバグレポートを比較している。2 件目は、バグレポートを比較する際、応答時間に影響があると思われる要素を分位点回帰モデルに追加して、分析を行っている。以上の工夫により、両群では寄付バッジの有無以外には他の要素は同じである状態であると考えられるため、統計的に平行トレンド仮定を満たしていると考えられる。

また、共通ショック仮定においては、著者の知る限りでは寄付バッジの有無以外に結果を与える要素は存在しない。また、両群共、同じ集計期間、同じ Eclipse のバグレポートを計測対象としている。そのため、仮に結果に影響を与えるような要素が存在していたとしても、両群共に同じく作用していると考えられるため、共通ショック仮定においても満たしていると考えられる。

3. 分析

3.1 Friend of Eclipse と寄付バッジ

Eclipse は、2017 年 12 月に「Friend of Eclipse」という寄付プログラムを開始した。Friend of Eclipse は Linux の寄付プログラム [2] に触発されて、寄付者に多くの特典を提供した。特に、2014 年 10 月のピーク時には、T シャツや、技術書の割引などを提供していた。しかし、寄付者は特典に関心がないという Eclipse が実施したアンケート調査 [4] の結果から、現在は、寄付の特典は寄付バッジと Friend of Eclipse のロゴが使用できる権利だけが提供されている。

Eclipse では、年間 35 ドル以上寄付をすることで寄付バッジが寄付者に提供される。寄付バッジとは、Eclipse のバグ管理システム (Bugzilla) 上のバグレポートに氏名と共に表示されるものであり、バグレポート利用者は、バグレポート貢献者から寄付者を認識することができる。寄付バッジの提供に関する議論は Bugzilla 上で行われ、寄付バッジは 2016 年 11 月 13 日から提供が始まった [3]。

3.2 分析手法

分析概要について図 2 に示す。図 2 に示すように、本分析は control group と donors のマッチング (図上部) とバグレポートの差分の差分分析 (図下部) で構成されている。以下、各工程ごとに説明を行う。

Step 1: バグレポートの収集. 寄付バッジが最初に導入された日は 2014 年 11 月 13 日 [3] である。差分の差分分析を行うためには、寄付バッジ導入前後のデータが必要である。本分析では、2012 年 11 月 13 日から 2016 年 11 月 12 日までの 4 年間のバグレポートを分析期間とした。前後で同じ集団で比較を行うため、寄付バッジ導入の前後の両期間で少なくとも 1 回以上のバグレポートを提出したバグレ

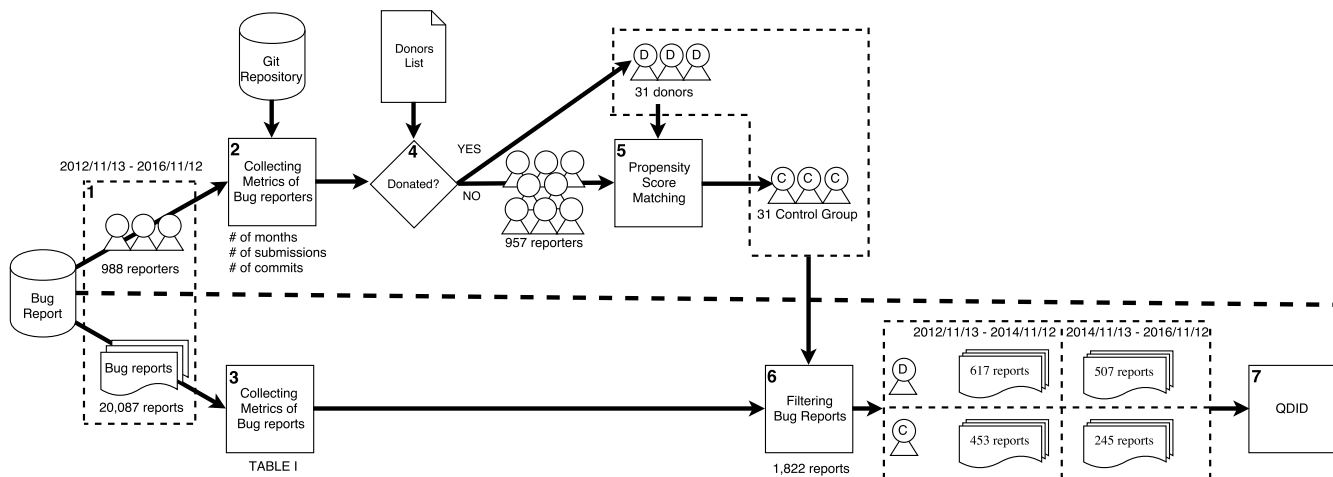


図 2 データ収集と統計的因果推論の分析プロセス

ポータのバグレポートのみを対象とした。バグレポートの応答時間を適切に測るため、バグレポータとバグレポートに割り当てられた人物 (assignee) が同じ場合はそのバグレポートを除外した。また、忘れられたり、または意図的にレスポンスを延期したバグレポート (数週間から 1 年以上経ってからレスポンスがあるバグレポートが存在する) を除外するために最初のレスポンスが 3 日以内に來なかったバグレポートも除外した。以上の操作からバグレポートは全体の 60% となった。

Step 2: バグレポータのメトリクス。 バグレポータのメトリクスは、傾向スコアマッチング (Step 5) で使用する。バグレポート報告経験月数、導入前と後それぞれのバグレポート投稿回数、コミット回数をバグレポータの計測メトリクスとした。

Step 3: バグレポートのメトリクス。 バグレポートのメトリクス (表 1 に示す) は QDID で使用される。差分の差分分析に必要な (Donor, Period, Badge) に加えて、開発者の応答時間に影響を及ぼす可能性のある要因について検討した。以下に検討したメトリクスについて説明する。バグ深刻度 (Severity) の高いバグは速く修正され [13], バグが発見されたオペレーティングシステム (OS) はバグ修正時間に影響を与える [12] と報告されている。そのため、7 レベルの Severity と OS に関連するメトリクスを検討した。Component は、バグレポートのコンポーネントの応答日数の中央値を測定した。パイロットスタディによって、開発者の応答時間はコンポーネントによって異なることがわかった。いくつかのコンポーネントでは応答時間は中央値で半日以内であるが、応答時間が中央値で 10 日を超えるものも存在する。Community は、コンポーネントのコミュニティ内の貢献者のサイズを示す。Time [14] は、寄付バッジを導入日を基準とした、提出時期を示す。時間の影響を考慮するためこのメトリクスを追加した。Relationship は、レポータと応答した開発者との間の社会的および個人的な

関係を考慮するため、用意した。Ortu らは、感情的なコメントがバグ修正に影響を与える可能性があることを報告した [9]。ただし、既存のセンチメント分析ツールはソフトウェア工学のドメインには必ずしも適用されないと報告されているため、感情的な要素は追加しなかった [5]。上記のメトリクスから、Akaike 情報基準 (AIC) に基づいてモデルに使用するメトリクスを選択した。表 2 に、最小 AIC 値のメトリクスを示す。応答時間の計測には、バグレポートを提出してから、バグレポータ以外の開発者からの最初のコメントが得られるまでの間の時間を使用している。

Step 4: 寄付者の分類。 Eclipse の寄付サイトには、寄付者の名前、寄付をした日、寄付額がリストとして公開されている。この情報から、各寄付者のバグレポートにおける寄付バッジの出現期間を特定することができる。ただし、バグレポータの名前が同じかつ異なる電子メールアドレスのバグレポートが存在した時、バグレポータと寄付者の名前を一意に関連付けることができないため、そのバグレポートは除外した。その結果、31 人の寄付者が特定された。

Step 5: 傾向スコアマッチング。 傾向スコアマッチングでは、よく知られている最近傍探索アルゴリズムを使用した。寄付をしていない 957 人のバグレポータから、31 人の donors とマッチングされた 31 人のバグレポータが control group として選択された。

Step 6: バグレポートのフィルタリング。 図 2 上部で選択されたバグレポータの提出したバグレポートのみを抽出する。図 2 に示すように、これらのバグレポートは、2 つの期間と 2 つのグループで構成される。

Step 7: Quantile Difference in Differences (QDID)。 本研究では、比較的早く対応されたバグレポートに焦点を当てるため、時間単位で影響を分析した。表 1 に示す収集されたバグレポートのメトリクスを使用して、QDID を行う。QDID では、0.1 分位点刻みの Badge の偏

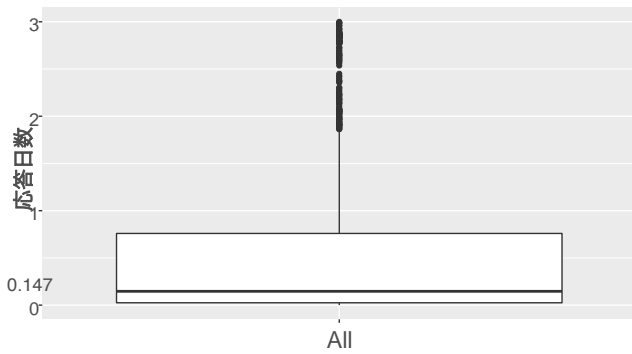


図 3 分析対象のバグレポートの応答日数

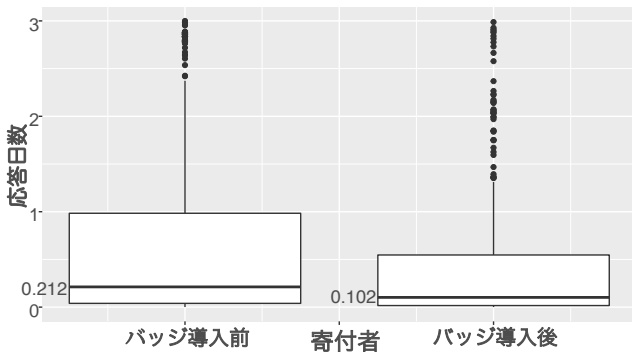


図 4 donors のバグレポートの応答日数

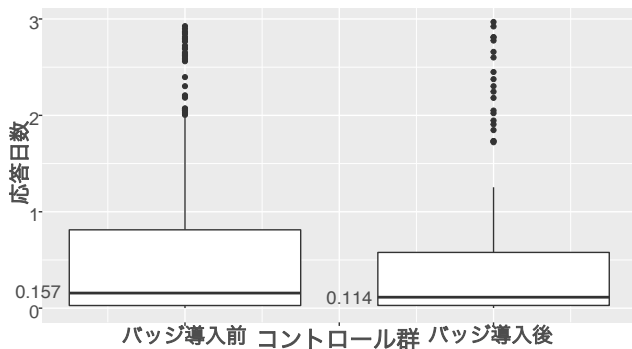


図 5 control group の応答日数

回帰係数の結果と 0.5 分位点に着目して結果を報告する。

4. 結果

4.1 寄付バッジの効果

分析対象のバグレポートの応答日数の分布を図 3 に示す。分析対象全体の応答時間の中央値は 3 時間 32 分であった。donors のバグレポートの応答日数の分布を図 4 に示す。donors の寄付バッジ導入前の応答時間の中央値は 5 時間 5 分、導入後の応答時間の中央値は 2 時間 26 分であった。control group の応答日数の分布を図 5 に示す。control group の寄付バッジ導入前の応答時間の中央値は 3 時間 46 分、導入後の応答時間の中央値は 2 時間 44 分であった。以上の結果から、寄付者も control group も寄付バッジ導入後の応答時間のトレンドは下降傾向にあることがわかった。

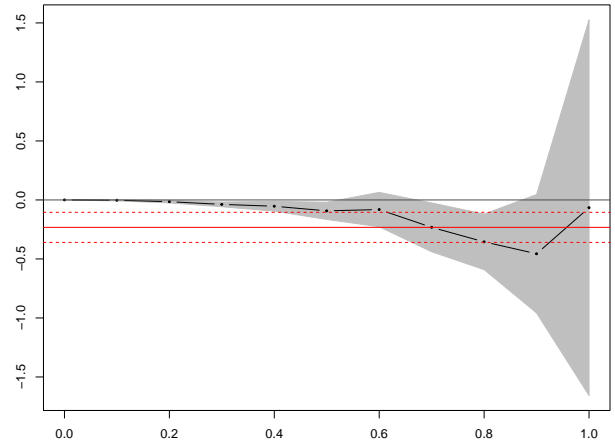


図 6 偏回帰係数 Badge の各分位点の結果

表 2 QDID の 0.5 分位点の結果

Metric	Coeffs (Errors)	t value	Pr (> t)
(Intercept)	2.237 (1.120)	1.998	0.046
Donor	2.259 (0.834)	2.708	0.007
Period	-1.598 (1.227)	-1.302	0.193
Badge	-2.219 (1.061)	-2.092	0.037
Enhancement	0.668 (0.815)	0.820	0.412
Windows	1.119 (0.692)	1.617	0.106
Linux	0.675 (0.949)	0.711	0.477
MacOS	0.793 (0.907)	0.875	0.382
Component	0.317 (0.198)	1.603	0.109
Community	0.000 (0.001)	0.105	0.917
Time	0.152 (0.117)	1.303	0.193
Relationship	0.005 (0.005)	0.894	0.371

AIC = 4,025, Pseudo R^2 = 0.011

また、donors のバッジ導入前の応答時間は他のグループと比べ長いことがわかった。QDID の結果を図 6 と表 2 に示す。図 6 は、0.1 分位点ごとに分位点回帰分析の Badge の結果をプロットしたものである。灰色の範囲は誤差範囲を示している。全ての分位点において、Badge の偏回帰係数はマイナスである。ただし、分位点が大きくなるほど誤差範囲が大きくなる。特に、0.8 分位点が一番 Badge の効果が大きい。表 2 は 0.5 分位点における分位点回帰分析の結果である。Badge の偏回帰係数の値は -2.219 (時間) であり、統計的に有意な結果となった。以上の結果から、全体のバグレポートの中央値 3 時間 32 分に対し、寄付バッジが導入されたことにより、寄付バッジを持ったバグレポートの提出したバグレポートは中央の分布で約二時間最初の応答時間が短縮されるということがわかった。

5. 考察

寄付バッジの分析結果の理由を探り、フォローアップするため、Eclipse コミュニティに対し、サーベイを行なった。以降、サーベイ内容を記述する。

5.1 サーベイ

どのようなバグレポートを優先的に扱うのかについて、Eclipse コミュニティに対し、アンケート調査を行った。

調査方法: Google フォームで、調査フォームを作成し、Eclipse の Marketing Specialist に調査フォームの広報を依頼した。2018 年の 4 月 12 日と 6 月 7 日の 2 回に渡り、Eclipse の保有する公式 web サイト、ソーシャルメディア (Twitter, Facebook, LinkedIn) 上で、調査フォームのリンクをポストしてもらった。さらに、7 月 3 日に Eclipse が毎月発行する Eclipse Newsletter のイントロダクションで取り上げてもらった。集計期間は 2018 年 4 月 12 日から 7 月 6 日である。回答者は 27 人。うち、有効回答 23 人であった。

調査項目: 調査項目は以下のとおりである。

- 回答者の属性
- バグレポートを読むことに影響する要因
- バグレポートにコメントをすることに影響する要因
- バグを優先的に直すことに影響する要因 (バグを修正したことがある人のみ回答)
- 寄付バッジの認知度
- 寄付バッジに対する意識

5.2 サーベイの結果

寄付者の属性: 回答者は 23 人であり、そのうち 12 人がバグを直したことがある経験があると答えた。また、End-user が一番多く、次に多い属性はバグレポート貢献者であり、両方ともその割合は 60% 以上であった (複数回答)。寄付をした経験がある回答者は 6 人で、現在、寄付バッジを利用する権利を持っている人は 2 人であった。

バグレポートの扱いの優先順位: 図 7 にバグレポートに回答することに影響をする要因についての結果を示す。バグレポートを読むこと、コメントをすること、直すことに影響する要因について質問した。どの質問の場合も、よく知られたバグレポートによって提出されたバグレポート (well-known reporter) と寄付バッジがついたバグレポート (donation badge) は肯定意見より、否定意見の方が多かった。自分がよく知っているコンポーネントに関するバグレポート (components) とバグ深刻度が高いバグレポート (high bug severity) はどの質問においても肯定意見の方が多く、他の要因と比べても肯定の結果が上位になっている。肯定的な感情を伴う簡潔で丁寧で完璧な方法で書かれたバグレポート (expression) はどの質問においても肯定意見の方が多く、バグを直すことに関しては肯定意見の割合は小さくなっている。成熟して、追加のフォローアップが必要のないバグレポート (mature) と自分がよく知っているバグレポートによって提出されたバグレポート (reporter is familiar) は、読むこと、バグを直すことに関しては肯定意見が多かったが、コメントをする場合には否

定意見が多い。

寄付バッジの認識: 寄付バッジについて知っていた人は 23 人中 10 人で、半数もいなかった。寄付バッジについて情報を提示した上で、寄付バッジを知らなかった人の中で、寄付バッジに興味を持ったのは 3 人で、興味を持たなかったのは 13 人であった。

寄付バッジの感じ方: 寄付バッジを見たときに感じることにする質問の結果を図 8 に示す。寄付バッジに対して、friendly メンバーであると認識し、コミュニティに暖かく歓迎されるべき (friendly) という質問に対しては、肯定意見のほうが否定意見よりも多かったがほぼ同じであった。また、寄付バッジに対して、knowledgeable メンバーであると認識し、初心者ではなく、バグレポートの品質が良い (knowledgeable) と respected メンバーであると認識し、高く優先すべきである (respected) という質問に関しては否定意見のほうが多い結果となった。

まとめ: 寄付バッジの効果の分析では、寄付バッジを持つことによって、応答時間を短くなるという結果が得られた。しかし、サーベイの結果では、Eclipse のコミュニティにおいて、寄付バッジの要素ではバグレポートを読む、コメントする、バグを直すことに関して優先的には扱おうとは思っていないことが明らかとなった。他の人に関する項目のよく知っているバグレポートや、よく知られたバグレポートに関しても同様に優先度の割合が低い結果となっている。一方、バグの内容に関する項目であるよく知ったコンポーネントや、バグ深刻度が高いバグレポートに関しては優先的に扱おうと思われていることがわかった。

これらの結果から、寄付バッジを優先的に扱おうとは思っていないが、無意識に寄付バッジを持つバグレポートに対し、少しは優先的に応答しているということがわかった。この無意識に働いている効果はシグナリング効果であると考えられる。シグナリングとは、私的情報を開示する行動である。Tsay らはプロジェクトマネージャは Pull request の受け入れの際に開発者のシグナリング情報を利用して報告している [10]。この寄付バッジを持つというシグナルは、長期的にプロジェクトに関わり貢献したいという意味を示唆するものであると考えられる。

6. おわりに

本稿は QDID によって応答時間が寄付バッジの影響を受けるのかを分析した。その結果バグレポート全体の応答時間の中央値 3.5 時間に対し、寄付バッジを持つことで、中央値で約二時間短くなることがわかった。さらに、どのようなバグレポートを優先的に扱うのかについて、Eclipse のコミュニティに対し、アンケート調査を行なった。その結果、寄付バッジを持つバグレポートは、優先的には扱おうとは思っていないことがわかった。優先的にバグレポートを扱っていないのに関わらず、実際のバグレポートの分

How Eclipse Contributors Respond to Bug Reports

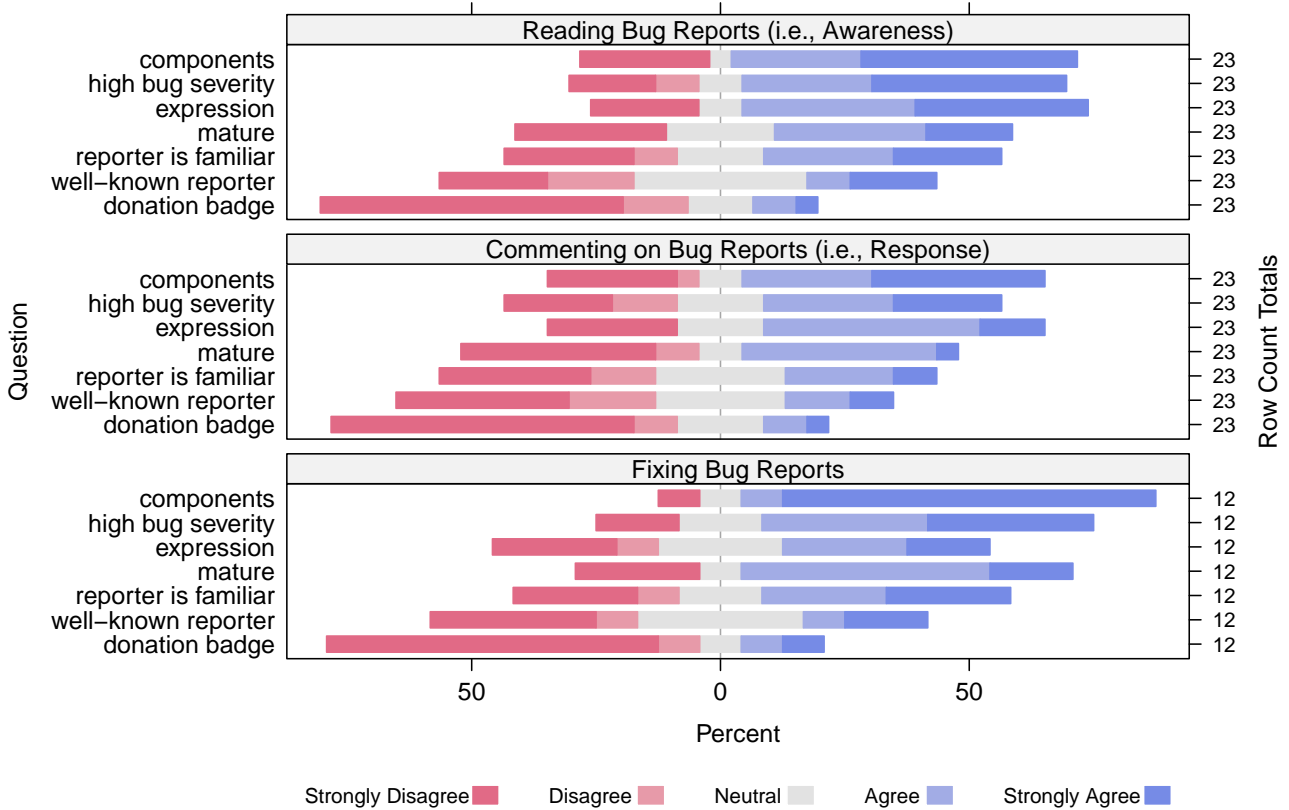


図 7 バグレポートを扱いの優先順位

Perception of the Donation Badges

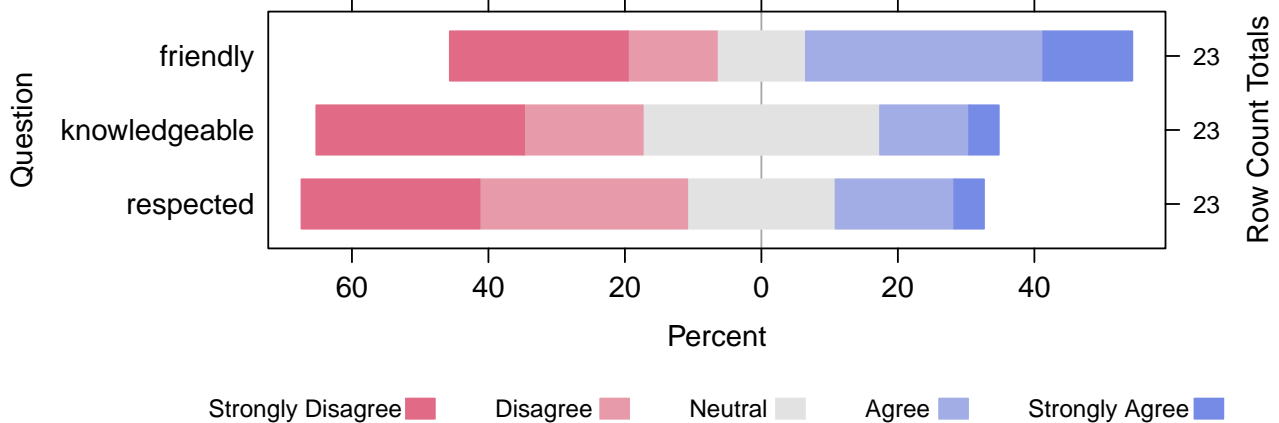


図 8 寄付バッジの感じ方

析で寄付バッジを持つことで、バグレポートの応答時間を短くなっているという結果がでたのは、シグナリング効果で説明できると考えられる。寄付バッジをもつというシグナルは、長期的にプロジェクトに関わり貢献したいという意思を示唆するものと考えられる。

謝辞 本研究は JSPS 科研費 16H05857 と 17H00731 の

助成を受けた。本研究のサーベイの実施に関して、多大なるご支援、ご協力をしてくださった、Eclipse Foundation Marketing Specialist の Roxanne Joncas さんに深く感謝する。また、本サーベイに協力してくださった Eclipse コミュニティメンバーの皆様に感謝する。

参考文献

- [1] Angrist, J. D. and Pischke, J.-S.: *Mostly harmless econometrics: An empiricist's companion*, Princeton university press (2008).
- [2] Bug282088: Expand Friends of Eclipse Program (2009).
- [3] Bug434249: Add decorator for Friends of Eclipse (2014).
- [4] Bug514954: Make "Friends of Eclipse" more appealing (2017).
- [5] Jongeling, R., Sarkar, P., Datta, S. and Serebrenik, A.: On Negative Results when Using Sentiment Analysis Tools for Software Engineering Research, *Empirical Softw. Engg.*, Vol. 22, No. 5, pp. 2543–2584 (online), DOI: 10.1007/s10664-016-9493-x (2017).
- [6] Krishnamurthy, S. and Tripathi, A. K.: Monetary donations to an open source software platform, *Research Policy*, Vol. 38, No. 2, pp. 404 – 414 (online), DOI: <https://doi.org/10.1016/j.respol.2008.11.004> (2009).
- [7] Nakasai, K., Hata, H., Onoue, S. and Matsumoto, K.: Analysis of Donations in the Eclipse Project, *2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP)*, pp. 18–22 (online), DOI: 10.1109/IWESEP.2017.19 (2017).
- [8] Nakasai, K., Hata, H. and Matsumoto, K.: Are Donation Badges Appealing? A Case Study of Developer Responses to Eclipse Bug Reports, *CoRR*, Vol. abs/1803.04129 (online), available from <http://arxiv.org/abs/1803.04129> (2018).
- [9] Ortu, M., Adams, B., Destefanis, G., Tourani, P., Marchesi, M. and Tonelli, R.: Are Bullies More Productive?: Empirical Study of Affectiveness vs. Issue Fixing Time, *Proceedings of the 12th Working Conference on Mining Software Repositories, MSR '15*, Piscataway, NJ, USA, IEEE Press, pp. 303–313 (online), available from <http://dl.acm.org/citation.cfm?id=2820518.2820555> (2015).
- [10] Tsay, J., Dabbish, L. and Herbsleb, J.: Influence of social and technical factors for evaluating contribution in GitHub, *Proceedings of the 36th international conference on Software engineering*, ACM, pp. 356–366 (2014).
- [11] Vignoli, I.: 200,000 thanks (2016).
- [12] Zhang, F., Khomh, F., Zou, Y. and Hassan, A. E.: An Empirical Study on Factors Impacting Bug Fixing Time, *Proceedings of the 2012 19th Working Conference on Reverse Engineering, WCRE '12*, Washington, DC, USA, IEEE Computer Society, pp. 225–234 (online), DOI: 10.1109/WCRE.2012.32 (2012).
- [13] Zhang, H., Gong, L. and Versteeg, S.: Predicting Bug-fixing Time: An Empirical Study of Commercial Software Projects, *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, Piscataway, NJ, USA, IEEE Press, pp. 1042–1051 (online), available from <http://dl.acm.org/citation.cfm?id=2486788.2486931> (2013).
- [14] Zhao, Y., Serebrenik, A., Zhou, Y., Filkov, V. and Vasilescu, B.: The Impact of Continuous Integration on Other Software Development Practices: A Large-scale Empirical Study, *Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017*, Piscataway, NJ, USA, IEEE Press, pp. 60–71 (online), available from <http://dl.acm.org/citation.cfm?id=3155562.3155575> (2017).