

クラッシュレポートの送信頻度と不具合との関連付けに関する実証的評価

小須田 光 亀井 靖高 鷗林 尚靖

本論文では、ユーザによるクラッシュレポートの送信を促すために、ユーザが送信したクラッシュレポートの開発者による利用に関する知見を示す。そうすることで、開発者にとっては不具合発生状況のより正確な把握、ユーザにとっては不利益を被っている不具合の優先的な修正といった恩恵をもたらすことが期待できる。本論文では、ユーザから送信されたクラッシュレポートに対して、クラッシュレポートの送信頻度と関連付けられる割合との関係や、関連付けられているクラッシュレポートが関連付けまでに要するクラッシュレポートの送信件数に関して調査を行った。Firefoxを対象に行ったケーススタディの結果、送信されたクラッシュレポートが関連付けられている割合は約63%であることや、不具合に関連付けられているクラッシュタイプの90%は、1,000件以内のレポートが送信される間に関連付けられていることがわかった。

This paper studies how a developer uses crash reports sent from users in order to encourage the users to send the crash reports. This paper could contribute to what the developers understand the fault occurrence in user environment more precisely (for developers) and what the faults that users suffer would be preferentially addressed (for users). We analyze the relationship between the frequency of the crash reports in a crash repository and the probability of linkages of the crash reports and issue reports in Bugzilla. The results using the dataset collected from Firefox show some findings such as 1) 63% of crash reports are linked to issue reports and 2) 90% of the linked crash reports are linked to issue reports up to the 1,000 crash reports.

1 はじめに

ユーザの環境でソフトウェア欠陥による不具合が発生した場合、ユーザに少なくはない時間的／金銭的損害を与えるだけでなく、開発プロジェクトの評判も落とすことになる。そのため、全ての欠陥がリ

リース前の段階で発見され取り除かれていることが、ユーザと開発者双方にとって望ましい。しかしながら、限られた開発工数と定められた納期の中では、開発者が全ての不具合を事前に発見することは難しい。

そこで、一部の開発プロジェクトでは、クラッシュレポートシステムを導入することで、不具合の発生状況を迅速に認識し、不具合を修正する仕組みを構築している [7]。クラッシュレポートシステムは、ユーザ環境下においてソフトウェアが予期しない停止をした際にスタックトレース^{†1}やユーザ環境等、不具合の再現や修正に有用な情報をレポートにまとめ、開発プロジェクトへ送信するシステムである。送信されたクラッシュレポートは、種類毎に分けられ、サーバに保存される。開発者はソフトウェアの不具合を修正する際、(1) クラッシュレポートの発生頻度から優先的

An empirical study of the relationship between the frequency of crash reports and the linkage with bug reports

Hikari Kosuda, 九州大学大学院システム情報科学府, Graduate School / Faculty of Information Science and Electrical Engineering, Kyushu University.

Yasutaka Kamei, Ubayashi Naoyasu, 九州大学大学院システム情報科学研究科, Graduate School / Faculty of Information Science and Electrical Engineering, Kyushu University.

コンピュータソフトウェア, Vol.29, No.1 (2012), pp.78-84. [研究論文] 2015年1月1日受付.

本論文は第21回ソフトウェア工学の基礎ワークショップ (FOSE2014) の発表論文を発展させたものである

^{†1} クラッシュが起こった際のメソッドの呼び出し過程

に修正する不具合を定めたり、(2) 不具合の修正に役立つ情報（例えば、スタックトレース）をクラッシュレポートから得て利用したりする場合がある。

しかしながら、クラッシュレポートの送信は基本的にユーザの許可が必要であるため、ユーザによっては拒否され、実際にはユーザの環境でどのクラッシュが頻繁に発生しているのかを開発者が知ることは難しい。そのため、(A) 開発者はユーザ環境において本来、頻繁に発生しているクラッシュの状況を正しくつかめていない、(B) ユーザは、ユーザ環境において頻繁に発生している不具合を開発者に正しく伝えきれず、優先的に修正してもらうことが望ましい不具合が優先的に修正されていない、といった問題が発生していることが考えられる。

そこで、本論文では、ユーザへクラッシュレポートの送信を促すために、クラッシュレポートの種類毎の送信頻度と、開発者によるクラッシュレポートの利用との間にどのような関係があるのかについての知見を得ることを目的とする。クラッシュレポートがより多く収集できるようになることで、開発者は不具合の発生状況をより正確に把握し、ユーザが不利益を被っている不具合を優先的に修正することが期待できる。すなわち、前述の(1)が分析対象となる。

本論文の目的を達成するために、オープンソースプロジェクトである Firefox プロジェクトを対象に分析を行った。具体的には、Firefox プロジェクトのクラッシュレポートシステムに蓄積されたクラッシュレポート、及び、不具合管理システムに蓄積された不具合報告データを用いて、1) 送信されたクラッシュレポートが不具合報告へ関連付けされる割合、2) クラッシュレポートの送信総数と関連付けされる割合との関係、3) 不具合に関連付けられているクラッシュタイプが関連付けまでに要する送信件数、及び、4) 一定期間におけるクラッシュレポートの送信頻度と関連付けされる割合との関係に関して分析を行った。

以降、第2章では、クラッシュレポートシステム、及び、これを利用した不具合修正のプロセスについて説明する。第3章ではリサーチクエスチョンを設け、第4章では Firefox プロジェクトを対象に行った分析結果を述べる。第5章では関連研究をまとめる。

最後に、第6章で本論文のまとめと、今後の課題について述べる。

2 クラッシュレポートシステムの仕組み

2.1 クラッシュレポートシステム

ユーザからのフィードバックを得るために、一部のソフトウェアシステムにはクラッシュレポートシステムが組み込まれている。ユーザがソフトウェアを利用している最中に、予期せぬ停止（クラッシュ）が発生すると、クラッシュレポートシステムが実行環境に関する情報をクラッシュレポートにまとめ、開発プロジェクトのクラッシュサーバへ送る（図1(a)）。

クラッシュレポートには、スタックトレースやユーザ環境（OSの種類やバージョン、当該ソフトウェアのバージョン）の情報が含まれている。スタックトレースは、メソッドの呼び出しの順序を記録したものであり、各スタックフレームに、順番、モジュール、メソッドシグニチャ、対応するソースコードへのリンクが記載されている。メソッドシグニチャは、単にメソッドの名前のみを表しているのではなく、メソッドの名前とそのメソッドの引数の組み合わせを表す。

クラッシュレポートは、開発プロジェクトのクラッシュサーバへ送信されると、レポート毎にユニークな番号が割り当てられる。その際、サーバでは、類似するクラッシュレポートを同一種類のレポートとしてまとめる。これは、クラッシュレポートシステムによって集められるクラッシュレポートの数が膨大であり、開発者がどういったクラッシュが頻繁に発生しているかを把握できないためである。同一種類のクラッシュレポートをまとめることは、頻出するクラッシュレポートの種類を容易にし、開発者が優先的に取り組むクラッシュを決定する際に役立つ。Firefox プロジェクトの Socorro の場合、クラッシュレポートはスタックトレースのトップメソッドシグニチャに基づいて自動でグループ分けされる（図1(b)）。

2.2 クラッシュレポートの報告からソースコードの修正までのプロセス

本節では、クラッシュレポートが送られてからソースコードが修正されるまでのプロセスを紹介する。ま

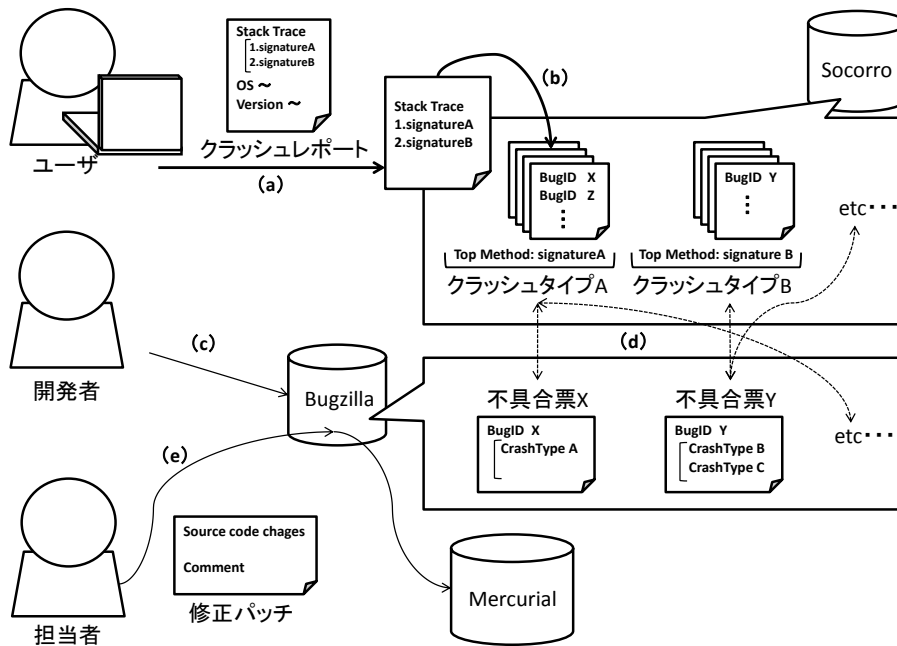


図1 クラッシュレポートシステムの流れ [9]

ず、先にも述べたように、ユーザがソフトウェアを利用している最中に、クラッシュが発生すると、クラッシュレポートが開発プロジェクトのサーバへ送られる(図1(a))。次に、クラッシュサーバは、大量のクラッシュレポートを類似するレポート毎にグループ化する(図1(b))。

レポータと呼ばれる開発者が、クラッシュレポートを調査しており、もし、クラッシュレポートの原因である不具合が、不具合管理システム(例えば、Bugzilla)に登録されていない場合、レポータは、その不具合を不具合管理システムに新たな不具合として登録する(図1(c))。そして、開発者は、クラッシュタイプと不具合票を関連付ける。クラッシュタイプと不具合は、多対多の関係で関連付けられ、一つのクラッシュタイプが複数の不具合と関連付けられる場合や、一つの不具合が複数のクラッシュタイプと関連付けられる場合もある(図1(d))。開発者は議論を重ね、優先的に修正する不具合を選び、修正対象の不具合を決定後、その不具合に修正担当者を割り当てる。開発者が不具合の修正を行う際は、関連付けられたクラッシュタイプのレポートを参照し、不具合

の発生環境等の情報を得ることができる。開発者は不具合を修正するためのソースコード(修正パッチ)を作成すると、不具合管理システムへ提出する(図1(e))。パッチの内容に誤りが無い場合、パッチは開発プロジェクトのソースコードへ統合される。

3 リサーチクエスチョン

本論文では、クラッシュレポートの送信頻度と、開発者によるクラッシュレポートの利用の間に、どのような関係があるのかについて調査する。そのため、1) 送信されたレポートが不具合報告へ関連付けられる割合、2) クラッシュレポートの送信総数と関連付けされる割合との関係、3) 不具合に関連付けられているクラッシュタイプが関連付けまでに要する送信件数、及び、4) 一定期間におけるクラッシュレポートの送信頻度と関連付けされる割合との関係に関して調査を行う。下記の4つのリサーチクエスチョンに取り組む。

- **RQ1.** クラッシュレポートのうち、何割が不具合に関連付けられるか？

2章で述べたように、開発プロジェクトでは、ユー

表 1 データセットの概要

| | クラッシュレポジトリ | 不具合管理システム |
|------------------|----------------------------|----------------------------|
| 対象期間 | 2010/02/25 ~ 2013/10/31 | 1998/04/07 ~ 2013/10/31 |
| レポート数 (不具合票数) | 463,808,045 | 930,000 |

ザから送られてくるクラッシュレポートと不具合報告とを関連付け、クラッシュレポートの内容も活用しながら、不具合の解決に取り組む。しかしながら、全てのクラッシュレポートが不具合と関連付けられているわけではない。

本 RQ では全クラッシュレポートに対する、不具合に関連付けられているクラッシュレポートの割合を調査する。クラッシュレポートと不具合報告が関連付けられるということは、当該不具合の解決に取り組む必要があると開発者が判断したということである。ここでは、クラッシュレポートが不具合の解決の判断にどの程度利用されているかを示す。クラッシュレポートが不具合と関連付けられることと、不具合修正に利用されることとの関係については、第 4 章の本研究の制約で述べる。

- **RQ2.** クラッシュレポート総数が多いと、不具合に関連付けられる割合が大きいのか？

送信されるレポート総数が多いクラッシュタイプのクラッシュは、ユーザ環境で頻繁に発生しているクラッシュである。そこで本 RQ では、クラッシュレポートの送信総数と関連付けされる割合を調査する。クラッシュレポートの送信総数と関連付けされる割合に一定の関係がみられるのであれば、頻繁に起きているクラッシュの原因となる不具合が優先的に修正が必要であると判断されている可能性が示される。

- **RQ3.** 不具合に関連付けられているクラッシュタイプが関連付けまでに要する送信件数はどの程度か？

クラッシュタイプが不具合と関連付けられるまでに一定のクラッシュレポート数が必要であるなら、ユーザのレポート送信はそのクラッシュ

タイプが早期に関連付けられることに貢献する。90%以上の確率で不具合報告に関連付けされるクラッシュレポートの、関連付けられるまでの送信件数を調査し、どの程度の件数が送信された際に関連付けされたのかを理解する。

- **RQ4.** クラッシュレポートの送信頻度が高いと、不具合に関連付けられる割合が大きいのか？

短期間に同種類のレポートが多数送信されるということは、その種類のクラッシュがユーザ環境で頻発しているということである。そのようなクラッシュタイプが不具合に関連付けられやすいならば、開発者はユーザ環境で一定期間内に頻繁に起きているクラッシュの修正に一定の関心があることが示唆される。

4 ケーススタディ

4.1 データセット

本論文では、ケーススタディの題材として、大規模なオープンソースプロジェクトである Mozilla Firefox プロジェクトを選んだ。本論文で対象とするクラッシュデータ、及び、不具合管理システムのデータを表 1 に示す。

表 1 に示すようにクラッシュレポートの期間は、2010/02/25 から 2013/10/31 までである。クラッシュレポートは、Firefox プロジェクトのクラッシュサーバ^{†2} から取得した。ただし、一部の日付のクラッシュレポートは存在せず、取得することができなかった^{†3}。また、不具合管理システムは公開されている範囲（1998 年 4 月 7 日～2013 年 10 月 31 日）の期間を対象とした。2010 年以前の不具合管理システム

†2 <https://crash-analysis.mozilla.com/>

†3 取得できなかった日付は、2010/3/14,4/12-4/15,6/11-6/15,6/17,7/23,8/5-6,9/15,2013/8/28-9/2

表 2 RQ2~RQ4 におけるデータセット

| | クラッシュレポジトリ | 不具合管理システム |
|------------------|----------------------------|----------------------------|
| 対象期間 | 2010/08/24 ~ 2013/10/31 | 1998/04/07 ~ 2013/10/31 |
| レポート数 (不具合票数) | 409,349,461 | 930,000 |

のデータを用いる理由は、2010 年以前に不具合システムに登録されたものが、クラッシュレポートと関連付けられる可能性があるためである。

4.2 分析

(RQ1) クラッシュレポートのうち、何割が不具合に関連付けられているか？

概要. クラッシュレポートと不具合報告が関連付けられるということは、当該不具合の解決に開発者の一定の関心があることを示す。本 RQ ではクラッシュレポートが不具合に関連付けられる割合を調査する。

アプローチ. 全クラッシュレポートのうち、不具合に関連付けられているクラッシュレポートの割合を求める。不具合と関連付けられたクラッシュレポートの総数を全クラッシュレポート数で割ることで、不具合との関連付けがなされているクラッシュレポートの割合を求める。

クラッシュレポートが不具合票に関連付けられているか否かは、次の 2 つの項目に注目し判断した。1 つ目は、Bugzilla の不具合票の *Crash Signature* という項目である。この項目内に各クラッシュレポートのクラッシュタイプ名が記述されている場合、そのレポートは不具合に関連付けられていると判断した。2 つ目は、クラッシュレポートの *bug list* という項目である。この項目内に不具合票の ID が記載されていた場合、このクラッシュレポートと同じクラッシュタイプのレポートは不具合に関連付けられていると判断した。

結果と考察. 全クラッシュレポートのうち、不具合に関連付けられているクラッシュレポートの割合は約 63.0% (292,401,403/463,808,045) であった。送信されたクラッシュレポートのうち、6 割以上が不具合と

関連付けられていた。これにより、少なくとも 63% のクラッシュレポートが、不具合の解決に取り組む必要があるか否かの判断に利用されていることがわかった。不要と判断されたものもあるので、63% 以上のものが判断には利用されていると考えられる。

送信されたクラッシュレポートが関連付けられている割合は約 63.0% である。

(RQ2) クラッシュレポート総数が多いと、不具合に関連付けられる割合が大きいか？

概要. クラッシュレポートの送信総数と関連付けされる割合に一定の関係がみられるのであれば、頻発するクラッシュの原因となっている不具合を開発者が優先的に修正しようと判断している可能性が示される。例えば、総レポート数が 100,000 件と 100 件のクラッシュタイプがそれぞれ 100 種類ずつあるとする。総レポート数 100,000 件のクラッシュタイプは 80 種類が不具合に関連付けられ、総レポート数 100 件のクラッシュタイプは 30 種類が不具合に関連付けられている場合、ユーザのクラッシュレポートの送信は不具合との関連付けに正の影響を与えていることを示唆している。

アプローチ. クラッシュタイプの総レポート数と、不具合に関連付けられる割合を求める。各クラッシュタイプの総レポート数と、不具合との関連付けの有無を調査し、同程度の総レポート数を持つクラッシュタイプが不具合に関連付けられる割合を算出した。例えば、総レポート数が 10,000 件程度のクラッシュタイプが 100 種類あり、そのうち 50 種類が不具合に関連付けられている場合、総レポート数 10,000 件程度のクラッシュタイプは 50% の割合で不具合に関連付

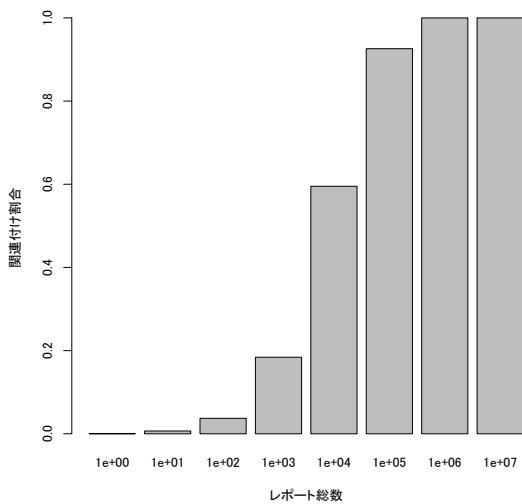


図 2 送信された総レポート数と関連付けの割合

けられていると算出される。

本 RQ を実施するにあたり、データセットのクリーニングを追加する必要が生じた。クラッシュレポートは 2010/02/25 以降のデータしかないため、それ以前にどのクラッシュタイプのレポートが送信されたかを知ることができない。そのため、一部のクラッシュタイプに関しては、総レポート数を正確に知ることができない。

そこで、2010/02/25 から 2010/08/24 までの 180 日の間に送られてきたクラッシュレポートを調査し、その中に含まれないものは、2010/08/24 以前には送られてきたことのないクラッシュレポートであると仮定した。これにより、2010/08/24 以降のクラッシュレポートのみを調査対象とすれば、全クラッシュタイプに対して、送信された総レポート数をより正確に知ることができる。

以上より、この分析におけるクラッシュレポートと不具合票の対象期間は表 2 の通りである。なお、同様の理由で、RQ3・RQ4 についても、同じ対象期間のデータセットを用いて調査を行った。

結果と考察. 調査結果を図 2 に示す。横軸は各クラッシュタイプの総レポート数を、縦軸はクラッシュ

タイプの関連付け割合を表す。例えば、横軸が $1e+3$ 上の棒は、 $1e+3$ と、その 1 つ前の目盛である $1e+2$ との間のレポート数を持つクラッシュタイプ、つまり、総レポート数が 100 件から 999 件の間であるクラッシュタイプの不具合との関連付け割合を表している。関連付け割合は、総レポート数が各目盛の範囲内であるクラッシュタイプを対象とし、不具合に関連付けられているクラッシュタイプの数を、全クラッシュタイプの数で割ることで求めた。

図 2 から、総レポート数が 100 件以上 999 件以下のクラッシュタイプが不具合に関連付けられる割合は 20% にも満たないが、総レポート数が 10000 件以上となるクラッシュタイプは 80% 以上の割合で関連付けられていることがみてとれる。また、各クラッシュタイプの総レポート数とクラッシュタイプの関連付けの割合の間のスピアマンの順位相関係数は、0.994 であった ($p < 0.01$)。

このことから、クラッシュレポートの送信総数と関連付けされる割合に一定の関係（正の相関）があることがわかった。この解釈としては、総レポート数 100,000 件のクラッシュタイプは総レポート数 100 件のクラッシュタイプよりも多くのユーザにより報告され、開発者がその事実を認識したために、クラッシュタイプの送信総数によって不具合との関連付け割合に差が生じたと考える。

クラッシュレポートの送信総数と関連付けされる割合には一定の関係（正の相関）がある。

(RQ3) 不具合に関連付けられているクラッシュタイプが関連付けまでに要する送信件数はどの程度か？

概要. どの程度の件数のクラッシュレポートが送信された際に不具合と関連付けられたのかを理解する。本 RQ によって、クラッシュレポートの送信がクラッシュタイプの不具合への関連付けに影響を与えていることが示唆されるか否かを確認する。

アプローチ. 不具合に関連付けられているクラッシュタイプを対象に、各クラッシュタイプのレポート

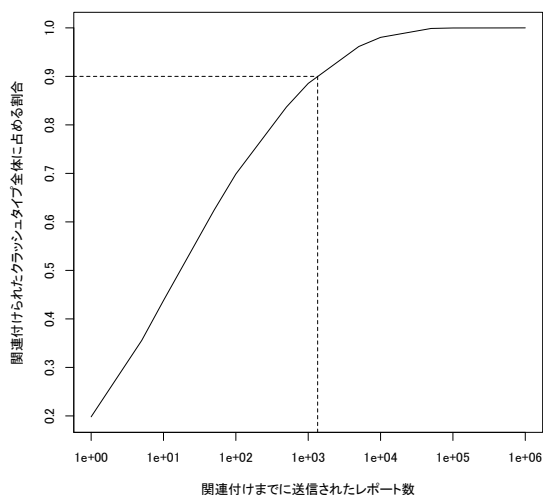


図 3 関連付けまでに必要なレポート数

を送信されてきた順に調査し、初めて不具合に関連付けられた時点で、それまでに送信された当該クラッシュタイプのレポート数を求めた。例えば、クラッシュタイプ A が初めて不具合に関連付けられるまでに、クラッシュタイプ A のレポートが 100 件送信されていたとすると、クラッシュタイプ A が不具合に関連付けられるまでに送信されたレポート数は 100 件であると判断する。

結果と考察. 調査結果を図 3 に示す。グラフの横軸は、各クラッシュタイプが不具合に関連付けられるまでに送信されたレポートの件数を示し、縦軸は、横軸の件数以下で関連付けられたクラッシュタイプが、関連付けられたクラッシュタイプ全体に占める割合を表す。

不具合報告に関連付けられているクラッシュタイプのうち、約 90% が 1,000 件のクラッシュレポートが送信されるまでに関連付けられていることがわかった。どの程度のクラッシュレポートの送信件数で関連付けられているかの目安として、1,000 件程度の送信件数が考えられる。

不具合に関連付けられるクラッシュタイプの約 90% は、1,000 件以内のレポートが送信される間に関連付けられている。

(RQ4) クラッシュレポートの送信頻度が高いと、不具合に関連付けられる割合が大きいか？

概要. 一定期間におけるクラッシュレポートの送信頻度と関連付けされる割合に関係があるならば、開発者はユーザ環境で一定期間内に頻繁に起きているクラッシュの修正に一定の関心があることが示唆される。

アプローチ. 各クラッシュタイプのレポートの 1 日での最大レポート数を調査するために、各クラッシュタイプのレポートの、日付毎の送信件数を集計した。また、クラッシュタイプを不具合に関連付けられているか否かで分類し、1 日での最大レポート数による比較を行う。

本 RQ では送信頻度が 1,000 件以下のクラッシュタイプのみを対象とした調査も行う。RQ3 において、関連付けられるクラッシュレポートのうち約 90% は、クラッシュレポートが 1,000 件送信されるまでに関連付けられていることがわかった。送信頻度が 1,000 件のクラッシュレポートは、最低でも送信件数が 1,000 以上である。そういったクラッシュレポートはそもそも関連付けられる傾向が強いと考え、全てのクラッシュレポートを対象とした分析に加えて、送信頻度が 1,000 件以下のクラッシュレポートのみを対象とした分析も行う。

結果と考察. RQ4 の結果を表 3 に示す。1,000 件以下、及び、全クラッシュタイプを対象としたどちらの場合においても、不具合に関連付けられているクラッシュタイプは、それぞれの 1 日あたりの最大レポート数における平均値と中央値において関連付けられていないものより大きい値を示した。中央値に関してウィルコクソンの順位和検定を行った結果、どちらに対しても統計的に優位な結果が得られた ($p < 0.01$)。また、送信件数の制限がない全クラッシュタイプを対象とした場合は、最大値においても大きい値を示した。

表 3 1日の送信件数の統計量

| | 関連付けの有無 | 最小値 | 中央値 | 平均値 | 最大値 |
|-----------|---------|------|-------|--------|-----------|
| 1,000 件以下 | 有 | 1.00 | 4.00 | 13.07 | 439.00 |
| | 無 | 1.00 | 1.00 | 1.37 | 861.00 |
| 全て | 有 | 1.00 | 18.00 | 341.30 | 298500.00 |
| | 無 | 1.00 | 1.00 | 1.64 | 42700.00 |

これらの結果より、開発者はユーザ環境で一定期間内に頻繁に起きているクラッシュの修正に一定の関心があることが示唆される。

関連付けられているクラッシュタイプのほうが、関連付けられていないものと比べて、一日あたりのクラッシュレポート数が大きい。

4.3 本研究の制約

本論文では、クラッシュレポートの種類毎の送信頻度と、開発者によるクラッシュレポートの利用の間どのような関係があるのかについての知見を得ることを目的として、4つのRQに取り組んだ。これらRQにおける分析では相関関係を扱っており、因果関係について分析できていない点は本研究の制約である。今後、対象プロジェクトの開発者にインタビューを実施し、因果関係についても明らかにすることは今後の課題である。

また、本論文では、不具合の解決に取り組む必要があるか否かの判断に利用されているかの観点で調査を行った。Iftexharら[4]の調査等により、クラッシュレポートが不具合修正に利用されるということは知られている。しかしながら、関連付けられた直後に、スタックトレース等の情報が、修正のために開発者によって利用されているか否かについては実証的評価を行っておらず、その点も今後の重要な課題である。

5 関連研究

本論文と同様、オープンソースプロジェクトを対象としたクラッシュレポートに関する研究が行われている[2][3][6][8]。例えば、Kimら[8]は、クラッシュレポート全体の大半を占めている少数の種類クラッシュ

レポート(トップクラッシュ)が存在することに着目し、それらの原因をできる限り早い段階で解決するため、製品のリリース後の短い期間で送られてきた少数のクラッシュレポートの特徴から、あるクラッシュレポートがトップクラッシュか否かを判定する予測モデルの構築を行い、Firefoxで75%、Thunderbirdで90%の精度を出している。また、Khomhら[6]は、クラッシュレポートによる修正の優先度判定の改善のために、entropy(波及度)に注目し、分析を行った。この研究で、クラッシュレポートの波及度の高低が、関連付けられる不具合の個数や修正時間に影響を与えることが示されている。

送信されたレポートの総数や波及度、レポートの内容に関する研究が行われている一方で、本論文では、クラッシュレポートの不具合票への関連付けに影響する要素として、短期間での送信数に注目した点が新しい。また、従来研究では不具合報告と関連付けられたクラッシュレポートのみを分析対象としている一方で、本研究では全クラッシュレポートを分析対象としている点も異なる。

6 おわりに

本論文では、ユーザへクラッシュレポートの送信を促すために、クラッシュレポートの種類毎の送信頻度と、開発者によるクラッシュレポートの利用の間どのような関係について調査した。具体的には、1) 送信されたレポートが不具合報告へ関連付けられる割合、2) クラッシュレポートの送信総数と関連付けされる割合との関係、3) 不具合に関連付けられているクラッシュタイプが関連付けまでに要する送信件数、及び4) 一定期間におけるクラッシュレポートの送信頻度と関連付けされる割合との関係に関して調査を行った。

Firefox プロジェクトで3年間に蓄積された約460,000,000件のクラッシュレポート、及び、Bugzillaで15年間に蓄積された930,000件の不具合票を対象としたケーススタディの結果得られた知見は、以下の通りである。

- 送信されたクラッシュレポートが関連付けられている割合は約63%である。
- クラッシュレポートの送信総数と関連付けされる割合には一定の関係（正の相関）がある。
- 不具合に関連付けられているクラッシュタイプの90%は、1,000件以内のレポートが送信される間に関連付けられている。
- 関連付けられているクラッシュタイプのほうが、関連付けられていないものと比べて、一日あたりのクラッシュレポート数が大きい。

今後の課題として、送信頻度の高いクラッシュタイプにはどのような特徴があるのか等、クラッシュレポートの内容に関する調査を行うこと、及び、対象プロジェクトの開発者にインタビューを実施し、クラッシュレポートの頻度と関連付けの間の因果関係についても明らかにすることが挙げられる。


謝辞 本研究の一部は、日本学術振興会 科学研究費補助金（若手 A：課題番号 24680003・15H05306、挑戦的萌芽：課題番号 25540026）による助成を受けた。

参考文献

- [1] Bangcharoensap, P., Ihara, A., Kamei, Y. and Matsumoto, K.: Locating Source Code to Be Fixed Based on Initial Bug Reports - A Case Study on the Eclipse Project -, Proc. Int'l Workshop on Empirical Software Engineering in Practice (IWESEP 2012), pp. 10-15 (2012).
- [2] Dang, Y., Wu, R., Zhang, H., Zhang, D. and Nobel, P.: ReBucket: a method for clustering duplicate crash reports based on call stack similarity, Proc. Int'l Conf. on Softw. Eng. (ICSE'12), pp. 1084-1093 (2012).
- [3] Dhaliwal, T., Khomh, F. and Zou, Y.: Classifying field crash reports for fixing bugs: A case study of Mozilla Firefox, Proc. Int'l Conf. on Software Maintenance (ICSM'11), pp. 333-342 (2011).
- [4] Iftekhhar, Ahmed., Nitin, Mohan., and Carlos, Jensen.: The Impact of Automatic Crash Reports on Bug Triaging and Development in Mozilla, The International Symposium on Open Collaboration (OpenSym'14), pp. 1-8 (2014).
- [5] Kamei, Y., Matsumoto, S., Monden, A., Matsumoto, K., Adams, B. and Hassan, A. E.: Revisiting Common Bug Prediction Findings Using Effort Aware Models, Proc. Int'l Conf. on Software Maintenance (ICSM'10), pp. 1-10 (2010).
- [6] Khomh, F., Chan, B., Zou, Y. and Hassan, A. E.: An Entropy Evaluation Approach for Triaging Field Crashes: A Case Study of Mozilla Firefox, Proc. Working Conf. on Reverse Engineering (WCRE'11), pp. 261-270 (2011).
- [7] Khomh, F., Dhaliwal, T., Zou, Y. and Adams, B.: Do faster releases improve software quality? An empirical case study of Mozilla Firefox, Proc. Int'l Conf. on Mining Software Repositories (MSR'2012), pp. 179-188 (2012).
- [8] Kim, D., Wang, X., Kim, S., Zeller, A., Cheung, S. C. and Park, S.: Which Crashes Should I Fix First?: Predicting Top Crashes at an Early Stage to Prioritize Debugging Efforts, IEEE Trans. Softw. Eng., Vol. 37, No. 3, pp. 430-447 (2011).
- [9] 長本貴光, 亀井靖高, 伊原彰紀, 鶴林尚靖: クラッシュログを用いたソースコード不具合箇所の特定向け分析, 情報処理学会研究報告, ソフトウェア工学研究会, pp. 1-6 (2013).
- [10] Śliwerski, J., Zimmermann, T. and Zeller, A.: When Do Changes Induce Fixes?, Proc. Int'l Conf. on Mining Software Repositories (MSR'05), pp. 1-5 (2005).
- [11] Servant, F. and Jones, J. A.: WhoseFault: automatic developer-to-fault assignment through fault localization, Proc. Int'l Conf. on Softw. Eng. (ICSE'12), pp. 36-46 (2012).
- [12] Zhou, J., Zhang, H. and Lo, D.: Where should the bugs be fixed? - more accurate information retrieval-based bug localization based on bug reports, Proc. Int'l Conf. on Softw. Eng. (ICSE'12), pp. 14-24 (2012).

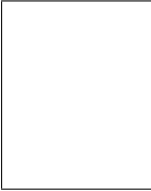
小須田光

2014年九州大学工学部卒業。現在、九州大学大学院システム情報科学府博士前期課程在籍。エンピリカルソフトウェア工学、特に、オープンソースソフトウェアのクラッシュレポートに関する研究に従事。IEEE 会員。

**亀井 靖高**

2005 年関西大学総合情報学部卒業。
2009 年奈良先端科学技術大学院大学
情報科学研究科博士後期課程修了。同
年日本学術振興会特別研究員 (PD)。

2010 年カナダ Queen's 大学博士研究員。2011 年九州大学大学院システム情報科学研究院助教。2015 年同大学同研究院准教授，現在に至る。博士 (工学)。ソフトウェアメトリクス，マイニングソフトウェアリポジトリの研究に従事。情報処理学会，電子情報通信学会，IEEE 各会員。

**鵜林 尚靖**

1982 年広島大・理・数学卒。1999 年
東京大学大学院総合文化研究科広域
科学専攻広域システム科学系博士課
程了。博士 (学術)。1982~2003 年

(株) 東芝に勤務。2002~2003 年芝浦工業大学システム工学部非常勤講師。2003 年九州工業大学情報工学部助教，2010 年九州大学大学院システム情報科学研究院教授，現在に至る。2003 年度情報処理学会山下記念研究賞受賞。ソフトウェア工学，プログラミング言語モデルに興味をもつ。電子情報通信学会，情報処理学会，ACM，IEEE-CS 各会員。