

An Empirical Study of Long Lived Bugs

Ripon K. Saha, Sarfraz Khurshid, Dewayne E. Perry

長命バグについての経験的研究

発表者：辻 裕真

IEEE CSMR-WCRE 2014

背景

ソフトウェアのバグの数はソフトウェア品質の重要な指標であり、開発者はバグの発見と修正に積極的ではあるが、長い間存在するバグは存在する。

ソフトウェアの品質を向上させるためには、それらの長命なバグへの対処が重要である。

Research Question

RQ1:長命なバグの割合はどの位か

RQ2:長命のバグはどれほど重要であるか

RQ3:バグ修正のどのプロセスに時間が費やされたか

RQ4:長命のバグの一般的な理由は何か

RQ5:長命のバグの修正の性質は何か

研究設定

- ・対象システム

JDT, CDT, PDE, Platformの4つのEclipseのOSSプロジェクト

プロジェクトの開始から2011年3月までのすべてのバグレポートをBugzillaから抽出

- ・長命なバグの定義

バグが報告されてから修正されるまでに1年以上かかったもの

研究結果-RQ1

RQ1:長命なバグの割合はどの位か

TABLE II BUG FIX TIME								
Time	JDT		CDT		PDE		Platform	
	# Bugs	[%]	# Bugs	[%]	# Bugs	[%]	# Bugs	[%]
<1 day	5,009	26.54	1,911	26.32	2,154	31.43	8,244	24.44
1-7 days	4,788	25.37	1,485	20.45	1,570	22.91	7,420	21.99
8-30 days	3,704	19.63	1,230	16.94	1,293	18.86	6,442	19.09
1-6 months	3,604	19.10	1,426	19.64	1,212	17.68	7,101	21.05
6-12 months	855	4.53	567	7.81	353	5.15	2,173	6.44
>1 year	913	4.84	641	8.83	272	3.97	2,358	6.99
Total	18,873	100.00	7,260	100.00	6,854	100.00	33,738	100.00

バグの5%～9%が修正までに1年以上かかった

研究結果-RQ2

RQ2:長命のバグはどれほど重要であるか

Bugzillaではバグの重要度をseverity(重大度)とpriority(優先度)の2つの項目で示す

簡単に修正できるバグなどはpriorityが高く設定されたりするので,今回の研究ではseverityを指標とする

研究結果-RQ2

TABLE III
IMPORTANCE OF LONG LIVED BUGS

System	Blocker	Critical	Major	Normal	Minor	Trivial
JDT	1	5	49	712	119	27
CDT	2	4	61	523	37	14
PDE	0	6	20	224	13	9
Platform	6	42	221	1856	170	63
Total	9	57	351	3315	339	113

長命のバグの約90%がNormal以上の重大度を設定されていた

研究結果-RQ3

RQ3: バグ修正のどのプロセスに時間が費やされたか

バグの修正プロセスは、割り当てフェーズ(AP), 修正フェーズ(FP), 検証フェーズ(VP)の3つのフェーズに分割できる

AP: Assignment Period

FP: Fixing Period

VP: Verification Period

研究結果-RQ3

TABLE VII
BUG ASSIGNMENT TIME VS. BUG FIXING TIME

System	Assignment Period (AP)			Fixing Period (FP)		
	Avg.	Med.	Max.	Avg.	Med.	Max
JDT	463	374	2745	407	376	2854
CDT	603	552	2035	330	97	1815
PDE	484	482	2728	437	393	1622
Platform	459	373	3326	407	409	2854

適切な開発者にバグを割り当てるのに平均1.5年かかっており,修正には平均1年かかっている

ほとんどのバグは検証されておらず,検証されていても1か月未満の期間であった

研究結果-RQ4

RQ4:長命のバグの一般的な理由は何か

JDT : 55(Major以上に設定されているすべてのバグレポート)

PDE, Platform : 50(Major以上に設定されているバグレポートから最近のものを25ずつ)

上記の計105のバグレポートを手動で分析

バグレポートの開発者のコメントを分析して遅延の理由を理解する

研究結果-RQ4

1) バグが理解しにくい (7)

ソフトウェア内のどこにバグが存在しているかを特定するのに時間がかかる

2) 修正方法が不確か (3)

そのバグを解決できてもソフトウェアの他の部分との整合性を確保しなければならない

3) 修正が難しい (14)

研究結果-RQ4

4) 修正のリスク (2)

リリースの直前にバグが発見された場合,コードを変更するのはリスクがあると考え次のリリースまで修正を延期

5) 不完全な修正 (4)

問題が完全に解決するまで何度か修正しなければならない

6) 同じバグが報告されるまで重要性が認識されなかった (14)

研究結果-RQ4

7) バグの再現に時間がかかる (9)

再現に1年4か月かかったが1日で修正できたバグも

8) スケジュールの問題 (7)

より重要なバグがあった場合それを優先

9) その他 (18)

開発者が休暇をとっているなど

研究結果-RQ4

様々な理由でバグの修正に時間がかかっていたが,特定の理由なしに遅延したバグは29件であり,最も多かった

研究結果-RQ5

RQ5:長命のバグの修正の性質は何か

変更されたファイルの数とコードの行数を調査

研究結果-RQ5

TABLE IX
ANALYSIS OF BUG FIXES

System	# Bugs	#Number of Files (NOF)						Med NOF	Max NOF
		1	2	3	4	5	>5		
JDT	223	80	41	37	10	15	40	2	47
CDT	185	50	32	19	18	8	58	3	237
PDE	105	34	13	13	12	4	29	3	211
Platform	740	349	114	72	47	36	122	2	91
Total	1253	513	200	141	87	63	249	-	-
(%)	-	40.94	15.96	11.25	6.94	5.03	19.87	-	-

修正の40%以上が1つのファイルの変更のみであった
4以上のファイルにまたがっているものは30%程だった

研究結果-RQ5

23%のバグで100行以上の変更がだったが,10行未満の変更が最も多く約40%であった

長命のバグは必ずしも大規模な修正が必要であるとは限らない

結論

バグが長命である理由は様々であるが,特定の理由なしに遅延したものが多数あった

長命のバグはバグの割り当てに平均1年以上,修正に平均で1年かかっており,そこを改善することで修正時間を短縮できるのではないか