

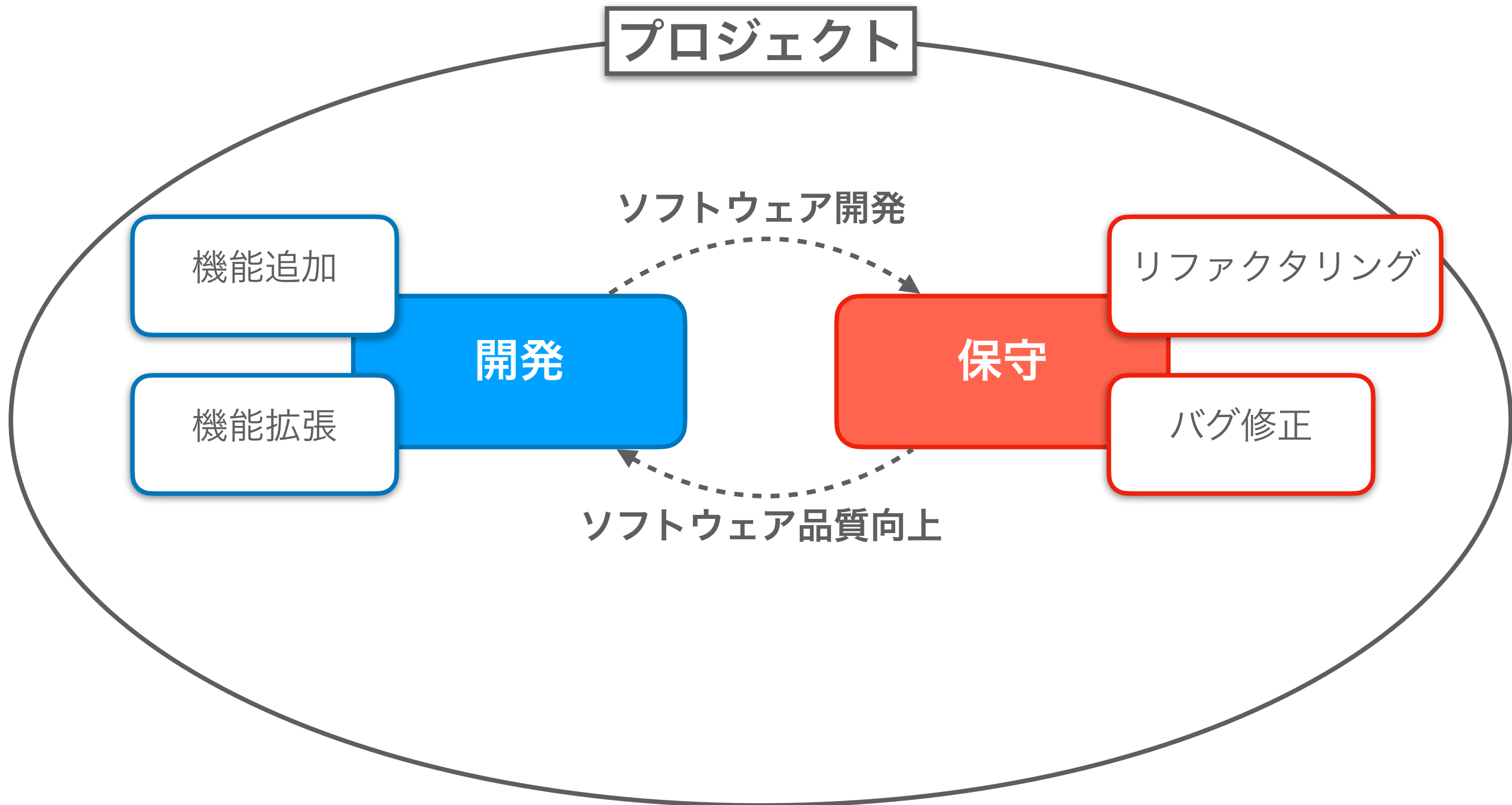
An Exploratory Study on Self-Admitted Technical Debt

Aniket Potdar

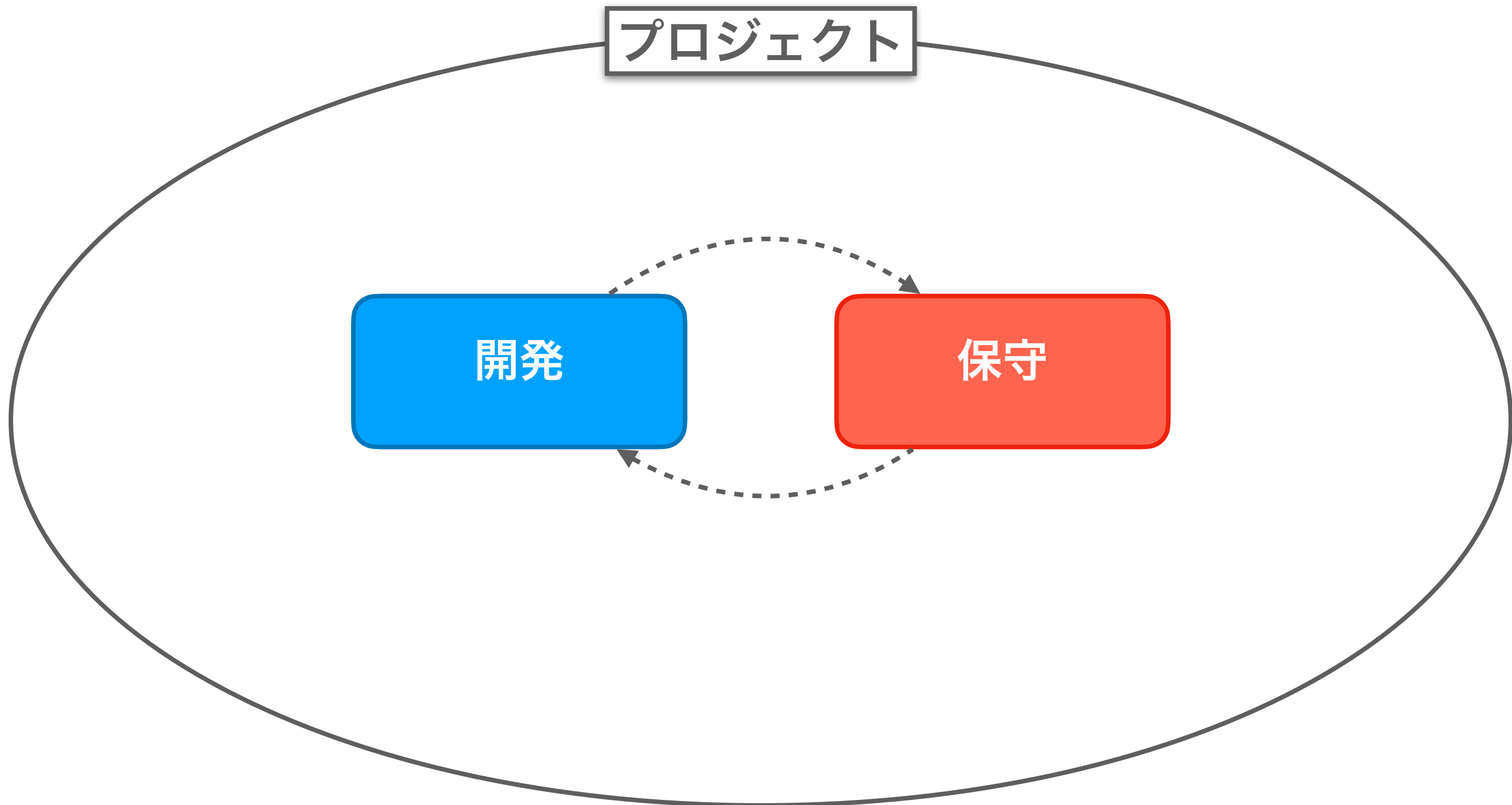
Emad Shihab

International Conference on Software Maintenance and Evolution (ICSME'14)

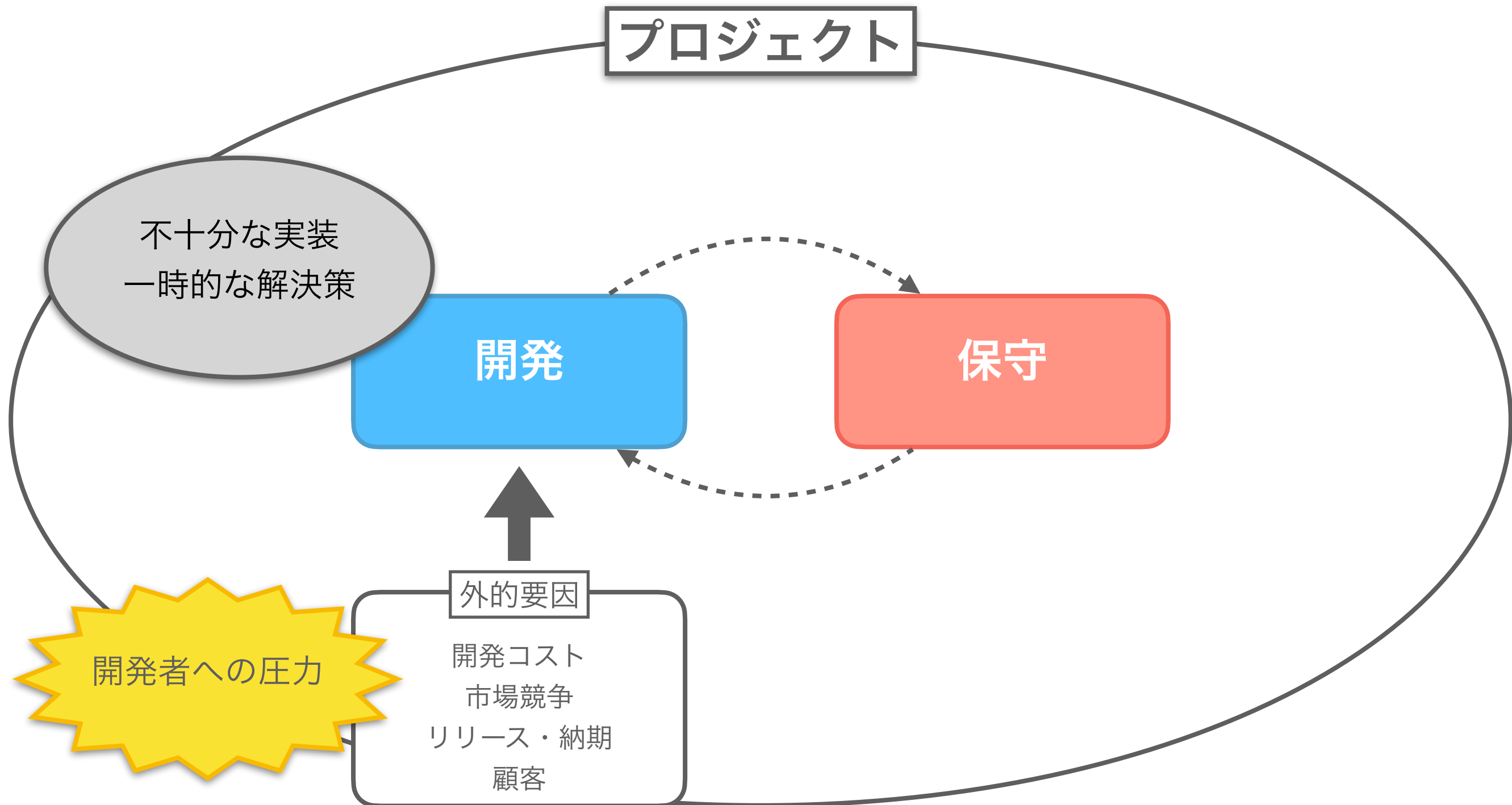
高品質なソフトウェアの提供を目指して



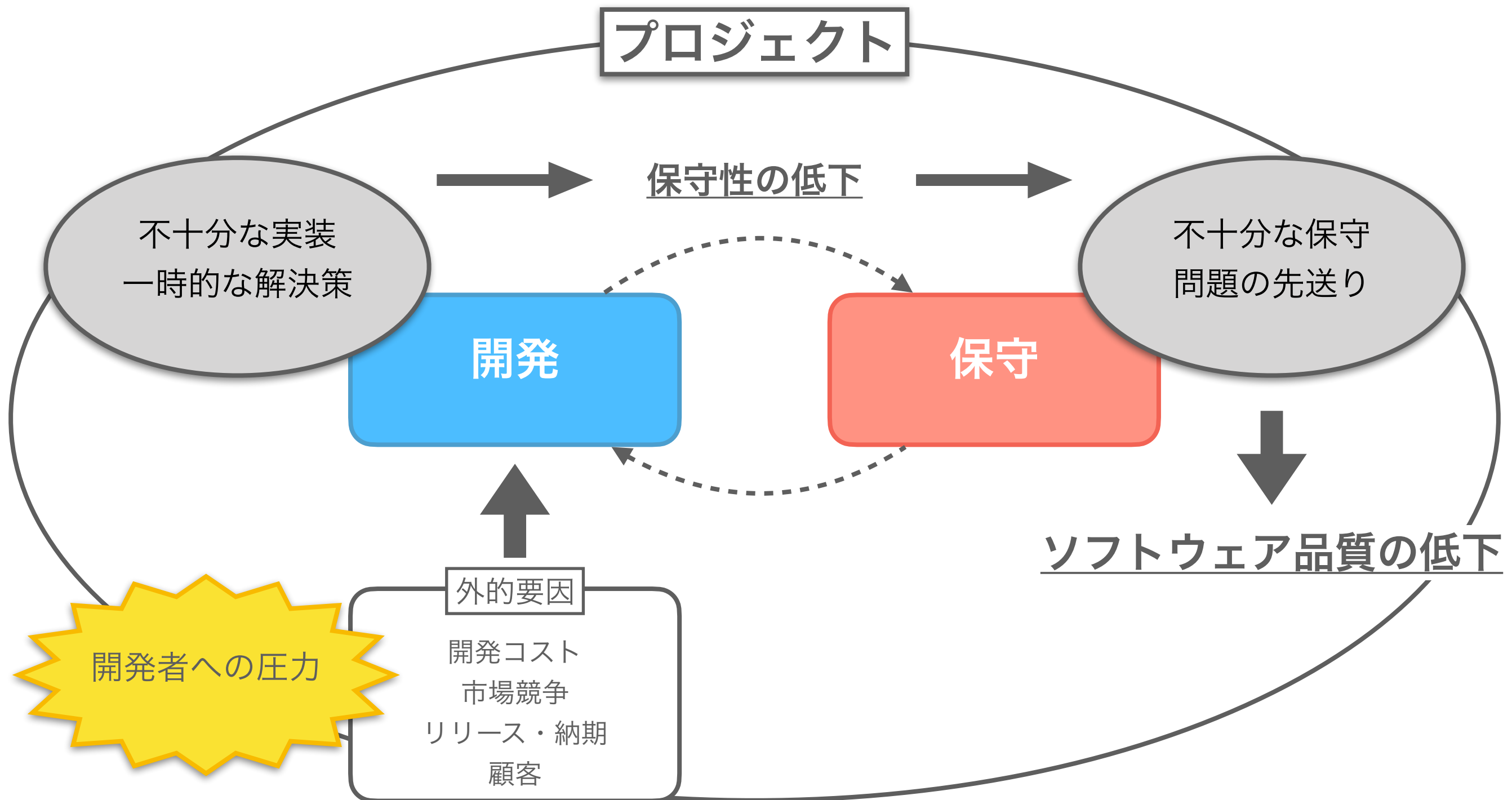
高品質なソフトウェアの提供を目指して



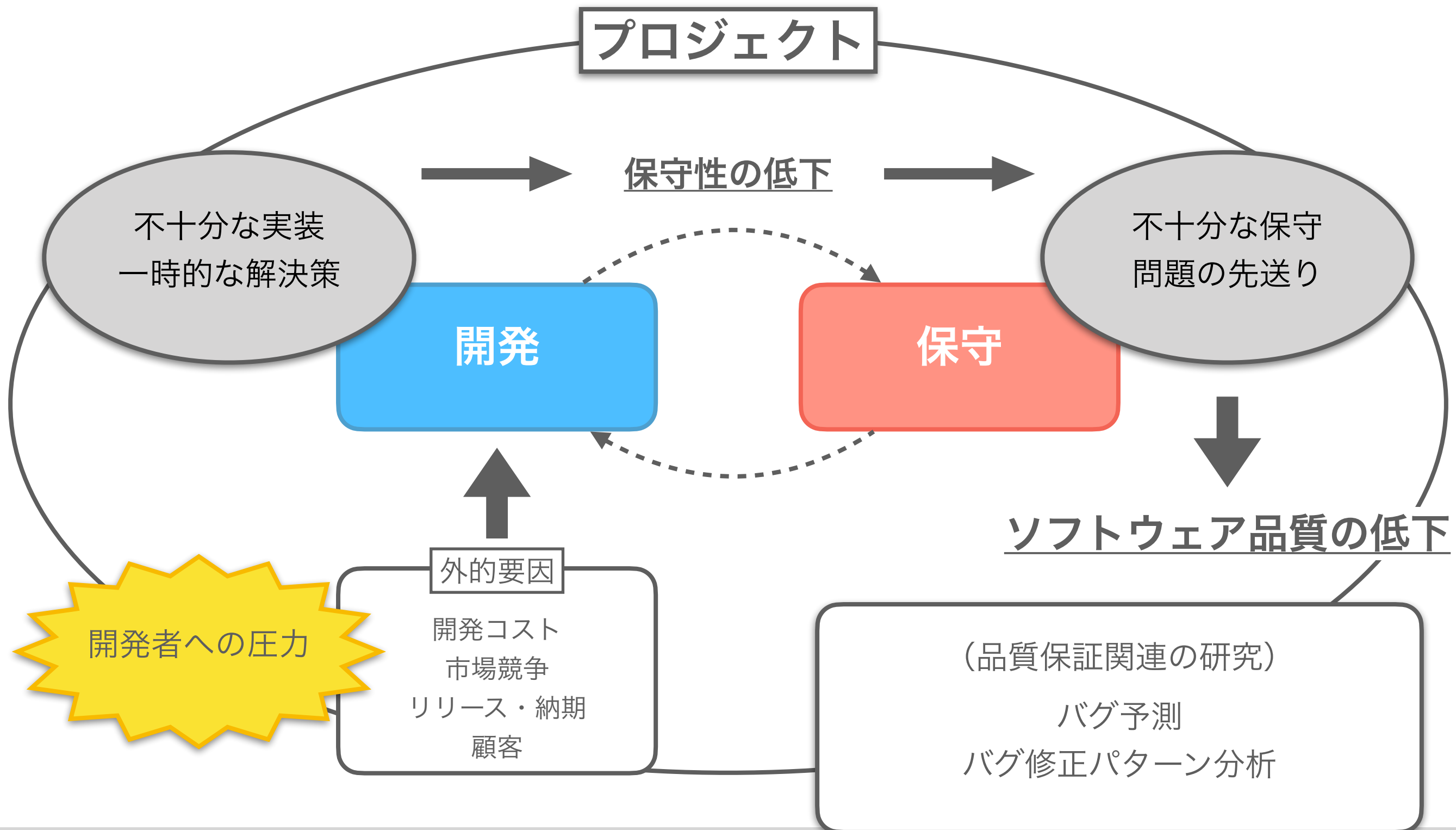
高品質なソフトウェアの提供を目指して



高品質なソフトウェアの提供を目指して



高品質なソフトウェアの提供を目指して



Self-Admitted Technical Debt

- その場しのぎの解決案
- 下表のようなコメントがソースコード内に記述

TABLE II
SAMPLE COMMENTS INDICATING SELF-ADMITTED TECHNICAL DEBT

Project	Comment
Eclipse	// TODO this is such a hack it is silly
Chromium OS	// Unsafe; should error.
ArgoUML	// FIXME: This is such a gross hack...
Apache	/* Ugly, but what else? */

目的

- SATDについてより理解を深めるための調査を実施
 - 4つの大規模OSSプロジェクトを対象
 - Eclipse, Chromium OS, ArgoUML, Apache httpd



アプローチ

TABLE I
PROJECT STATISTICS

Project	Release	Release Date	Code Lines	Comment Lines	Committers
Eclipse	4.3	June 2013	659,231	437,640	221
Chromium OS	30	November 2009	107,706	37,889	1,784
ArgoUML	0.34	December 2011	122,575	115,713	53
Apache httpd	2.4.6	July 2913	192,333	54,295	145

- 1.リポジトリの抽出
- 2.ソースコードからコメントを抽出
- 3.目視でSATDであるコメントを特定, 分類
- 4.各プロジェクトごとに分析

Research Questions

- **RQ1：プロジェクトにSATDはどのくらい含まれるのか**
 - SATDが直感的にも経験的にも良くないものとする
 - 目に見える形でSATDの問題理解をより良くする
- **RQ2：なぜSATDが混入するのか**
 - 開発者の経験，リリースまでの時間，コードの複雑さ
- **RQ3：どのくらいSATDは取り除かれるのか**
 - SATDはのちに変更，修正されることを意図しているが，実際にどの程度取り除かれているのか定かでない

RQ1

プロジェクトにSATDはどのくらい含まれるのか

Files

```
import ~  
# Comments  
class ~ :  
  
def ~ :
```

コメントの挿入位置で
SATDを分類

Classes

```
import ~  
  
class ~ :  
    # Comments  
  
def ~ :
```

Methods / Functions

```
import ~  
  
class ~ :  
    def ~ :  
        # Comments
```

File粒度のSATDが
一番多い

TABLE III
SELF-ADMITTED TECHNICAL-DEBT

Type	Project	Total	Self-Admitted Technical Debt	%
Files	Eclipse	6,389	152	2.4%
	Chromium OS	14,328	574	4.0%
	ArgoUML	1,846	77	4.2%
	Apache httpd	255	79	31.0%
Classes	Eclipse	11,993	151	1.3%
	Chromium OS	18,249	82	0.4%
	ArgoUML	2,304	76	3.3%
	Apache httpd	-	-	-%
Methods/ Functions	Eclipse	48,382	145	0.3%
	Chromium OS	153,514	626	0.4%
	ArgoUML	12,845	71	0.6%
	Apache httpd	4,361	112	2.6%
Constructors	Eclipse	5,561	151	0.2%
	Chromium OS	18,988	82	0.1%
	ArgoUML	1,920	76	0.5%
	Apache httpd	-	-	-%

SATDの粒度

We find that 2.4 - 31.0% of files, 0.4 - 3.3% of classes and 0.3 - 2.6% of methods/functions contain self-admitted technical debt.

RQ2

なぜSATDが混入するのか

- 開発者の経験

TABLE IV
SELF-ADMITTED DEBT-COMMITS BY DEVELOPERS

Project	Developer	Total Commits	Self-Admitted Technical Debt	Experience Rank	Total Committers
Eclipse	E1	1,231	21	8	221
	E2	2,513	20	2	
	E3	1,508	12	5	
	E4	4,436	10	1	
	E5	1,190	7	9	
Chromium OS	C1	4,853	28	1	1,784
	C2	410	24	219	
	C3	1,034	21	64	
	C4	1,930	21	26	
	C5	388	21	232	
ArgoUML	AU1	2,544	17	4	53
	AU2	4,879	7	1	
	AU3	3,088	6	3	
	AU4	3,192	4	2	
	AU5	480	3	8	
Apache	AH1	4,908	14	1	145
	AH2	3,214	13	3	
	AH3	721	11	26	
	AH4	1,219	11	13	
	AH5	1,239	10	11	

開発経験が長い開発者の方が
SATDを混入させる傾向にある

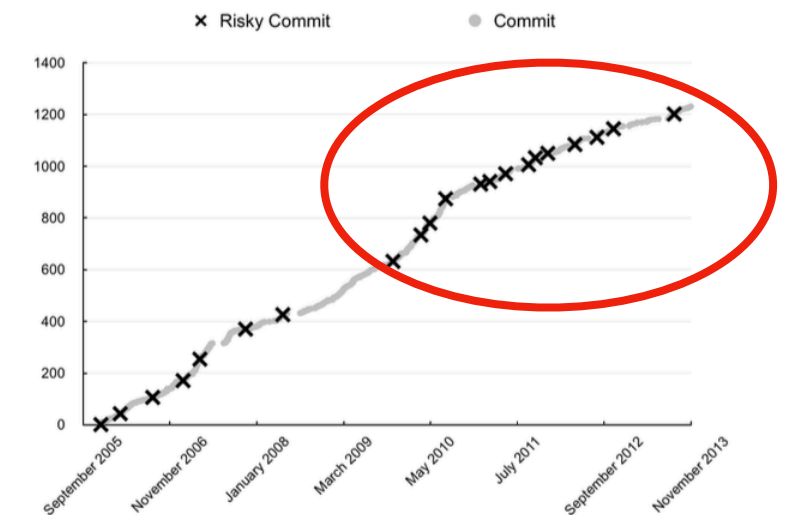


Fig. 2. Eclipse: Commits by E1

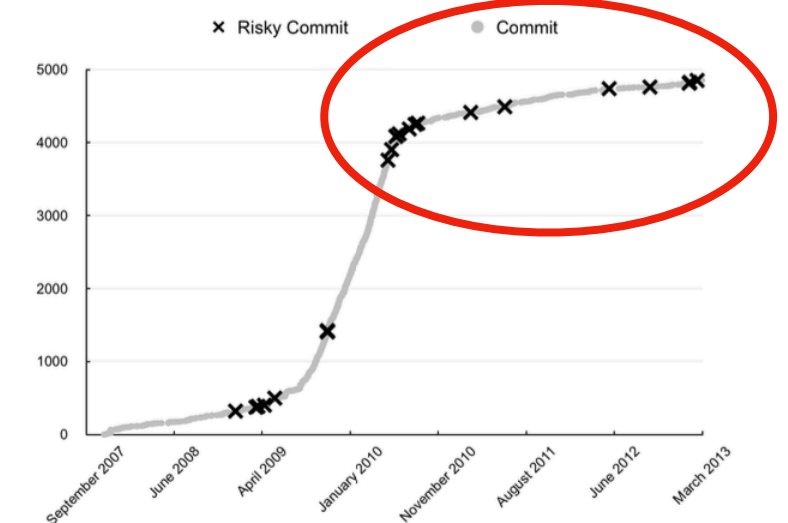


Fig. 3. Chromium OS: Commits by C1

RQ2

なぜSATDが混入するのか

- リリースまでの時間

TABLE V
DEBT COMMITS - TIME TO RELEASE WINDOWS

Project	Time to Release	Self-Admitted Technical Debt	Total Changes	% Over Total Changes	% Self-Admitted Technical Debt
Eclipse	$\geq 6\text{month}$	55	16,682	0.33%	41.7%
	$\geq 3\text{month} \ \& \ \leq 6\text{month}$	48	7,452	0.64%	36.4%
	$\geq 1\text{month} \ \& \ \leq 3\text{month}$	22	5,591	0.39%	16.7%
	$\leq 1\text{month}$	7	1,777	0.39%	5.3%
ArgoUML	$\geq 6\text{month}$	16	7,973	0.20%	33.3%
	$\geq 3\text{month} \ \& \ \leq 6\text{month}$	20	5,917	0.34%	41.7%
	$\geq 1\text{month} \ \& \ \leq 3\text{month}$	8	4,315	0.19%	16.7%
	$\leq 1\text{month}$	4	1,708	0.23%	8.3%
Apache	$\geq 6\text{month}$	51	23,579	0.22%	37.5%
	$\geq 3\text{month} \ \& \ \leq 6\text{month}$	43	13,960	0.31%	31.6%
	$\geq 1\text{month} \ \& \ \leq 3\text{month}$	28	11,670	0.24%	20.6%
	$\leq 1\text{month}$	14	6,614	0.21%	10.3%

市場の圧力とSATDの混入に関係があるとは言えない

RQ2

なぜSATDが混入するのか

- プログラムの複雑さ

TABLE VI
PARTIAL SPEARMAN CORRELATIONS BETWEEN CYCLOMATIC
COMPLEXITY, FAN-IN, FAN-OUT AND THE NO. OF SELF-ADMITTED
TECHNICAL DEBT WHILE CONTROLLING FOR SIZE

Project	Cyclomatic Comp.	Fan-in	Fan-out
Eclipse	-0.1992618 **	0.3159485	-0.08568448
Chromium OS	0.1840126 **	-0.01470615 *	-0.2529673 *
ArgoUML	-0.03449873 *	-0.01459957 *	0.1686399
Apache	0.2386505 **	0.3333192 ***	0.1108314 *

(p < 0.01 ***; p < 0.1 **; p < 1 *)

コードメトリクスとファイル内のSATDの数はとても弱い相関関係
→ 複雑さはSATD混入の主な要因ではない

RQ3

どのくらいSATDは取り除かれるのか

TABLE VII
DEBT PATTERNS ACROSS ECLIPSE RELEASES

Version	Release Date	Removed Admitted Technical Debt	Self-	Change
3.0 - > 3.0	June 2004	-		-
3.0 - > 3.1	June 2005	26.3%		26.3%
3.0 - > 3.2	June 2006	27.9%		1.6%
3.0 - > 3.3	June 2007	29.9%		2%
3.0 - > 3.4	June 2008	35.6%		5.7%
3.0 - > 3.5	June 2009	35.6%		0%
3.0 - > 3.6	June 2010	36.9%		1.3%
3.0 - > 3.7	June 2011	36.9%		0%

TABLE VIII
DEBT PATTERNS ACROSS ARGOUML RELEASES

Version	Release Date	Removed Admitted Technical Debt	Self-	Change
0.20 - > 0.20	February 2006	-		-
0.20 - > 0.22	August 2006	37.3%		37.3%
0.20 - > 0.24	February 2007	41.3%		4.0%
0.20 - > 0.26	September 2008	58.7%		17.4%
0.20 - > 0.28	March 2009	62.7%		4.0%
0.20 - > 0.30	March 2010	65.3%		2.6%
0.20 - > 0.32	January 2011	65.3%		0%
0.20 - > 0.34	December 2011	65.3%		0%

TABLE IX
DEBT PATTERNS ACROSS APACHE RELEASES

Version	Release Date	Removed Admitted Technical Debt	Self-	Change
1.3.x - > 1.3.x	February 2002	-		-
1.3.x - > 2.0.x	January 2004	65.45%		65.45%
1.3.x - > 2.2.x	December 2005	73.64%		8.19%
1.3.x - > 2.4.x	February 2012	74.55%		0.91%

26.3%～65.45%のSATDが
次のリリース時には取り除かれる

しかし、複数のリリースにまたがって
存在するSATDがある

まとめ

- 4つの大規模なOSSプロジェクトを用いて、SATDについて調査を実施
 - SATDがどれくらい存在するのか
 - プロジェクト内のファイルの2.4～31.0%がSATDを含んでいる
 - なぜSATDが混入するのか
 - 経験の多い開発者ほどSATDを混入させる傾向にある
 - 時間的なプレッシャーや複雑さはSATDとは相関関係がない
- SATDが混入した後、どれくらい削除されるのか
 - 複数のリリース間でもSATDの26.3～63.5%しか除去されない