

Open Borders? Immigration in Open Source Project

著者 : Christian Bird

Fourth International Workshop on Mining
Software Repositories(MSR 2007)

論文

▶ Open Borders? Immigration in Open Source Project

▶ 脱国境？ オープンソース内での“移民”(昇進)

▶ 著者:カリフォルニア大学

- ▶ Christian Bird
- ▶ Alex Gourley
- ▶ Prem Devanbu
- ▶ Ananda Swaminathan
- ▶ Freta Hsu



Christian Bird

▶ 出典:

- ▶ Fourth International Workshop on Mining Software Repositories(MSR '07) ©2007 IEEE



背景

- ▶ OSSを発展させるのに重要な要因の一つに新規開発者との関係がある。
 - ▶ OSSは全般的に開発者不足だと言われている。
 - ▶ OSSプロジェクトを育てるにはコアチームに入りたい人を常に獲得しなければならない。
- ▶ 権限を求めてコア開発者チームに入りたい人を誰がいつどのような方法で判断しているのかは不明確である。
 - ▶ OSSのプロジェクトにはそれぞれコア開発者チームがいて、リポジトリへのコミットの全権限を任されている。
 - ▶ コアチームはエリート集団であり、高い技術力、知識量が求められる。



概念的枠組み(本研究の視点)

1. 開発者はどれくらいコミットすればプロジェクトを成功させるのか
 - ▶ どれくらいの量をどれくらいの期間に？

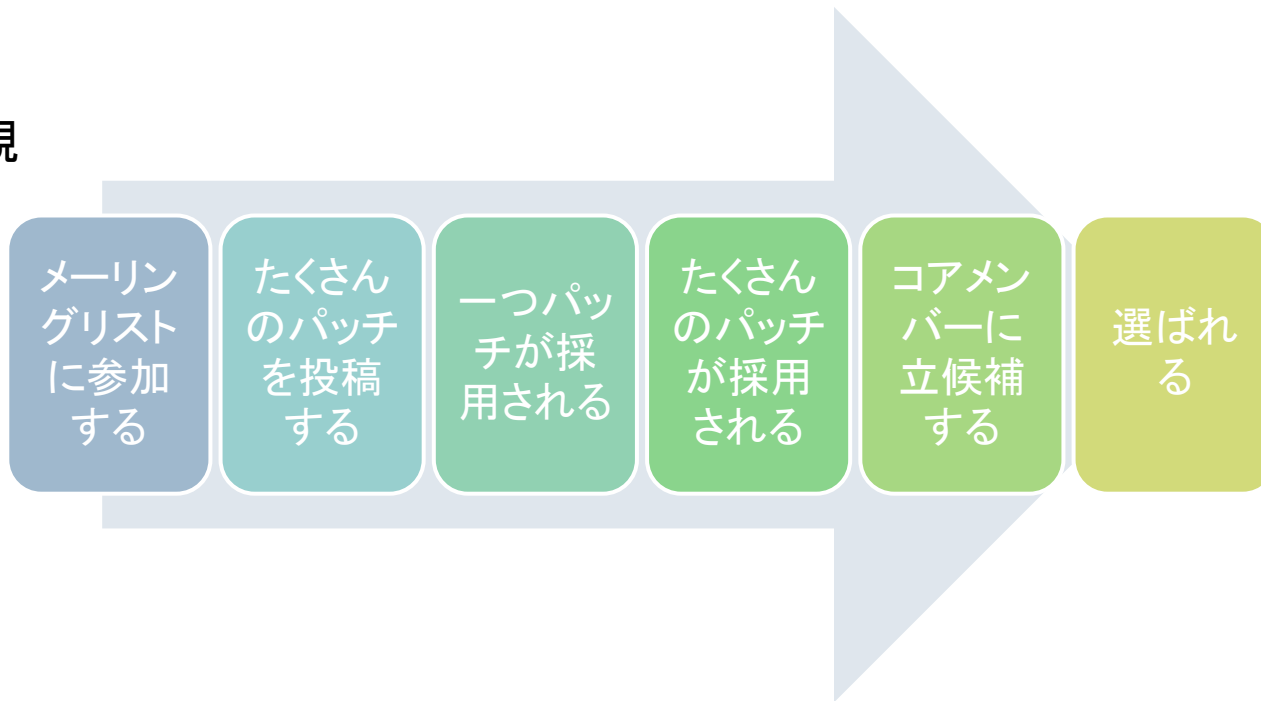
2. 開発者にそのプロジェクトにとって、どれくらいの技術と知識があれば妥当なのか

3. コミュニティでの評判とは何か？
 - ▶ コミュニティでの評判がコア開発者への昇進に関することは先行研究で分かっている。



コアチームに至るまでのプロセス

移民=新規
開発者



コアメン
バー

コアチームに入る際の障壁

▶ 知識量の壁(Knowledge-Barrier)

- ▶ 実際のプログラミングは知識がたくさん必要である。
- ▶ ある程度キャリアを積んできた人でさえ、ハードに勉強することを強いられる。
- ▶ その傾向は特に成熟したプロジェクトに多い。(先行研究)

▶ 知識の壁対策(先行研究)

- ▶ 成熟したプロジェクトにはメンター(新人教育者)が必要
 - ▶ 新規開発者の頃から教育が必要
- ▶ 開発者とプロジェクトの間での微妙な興味的一致
 - ▶ 持ちつ持たれつの関係。どちらかが一方的だと関係が破たんする。



先行研究の新規開発者定義の問題

- ▶ **新規開発者の定義が間違っているのではないか。**
 - ▶ パッチ投稿をしたことがある人や、その投稿が採用された人など
 - ▶ 今回は初めにメーリングリストに現れた人を新人開発者として定義している。
- ▶ **実はパッチ投稿が一番技術力が必要とされる仕事である。(先行研究)**
 - ▶ すでに投稿したパッチが採用されている人を新規開発者として対策を立ててしまっても無意味なのでは？

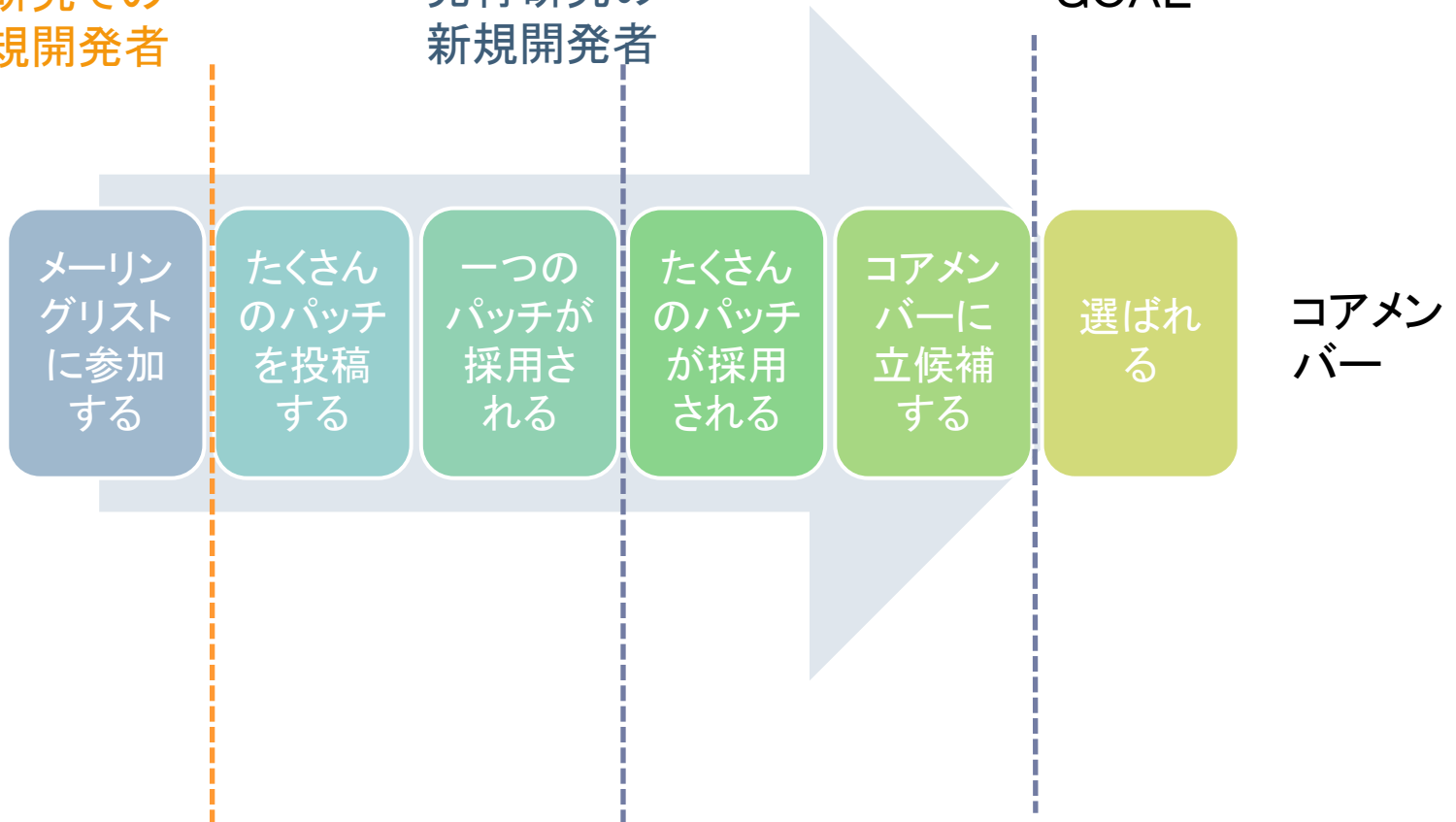


新規開発者の定義の違い＊

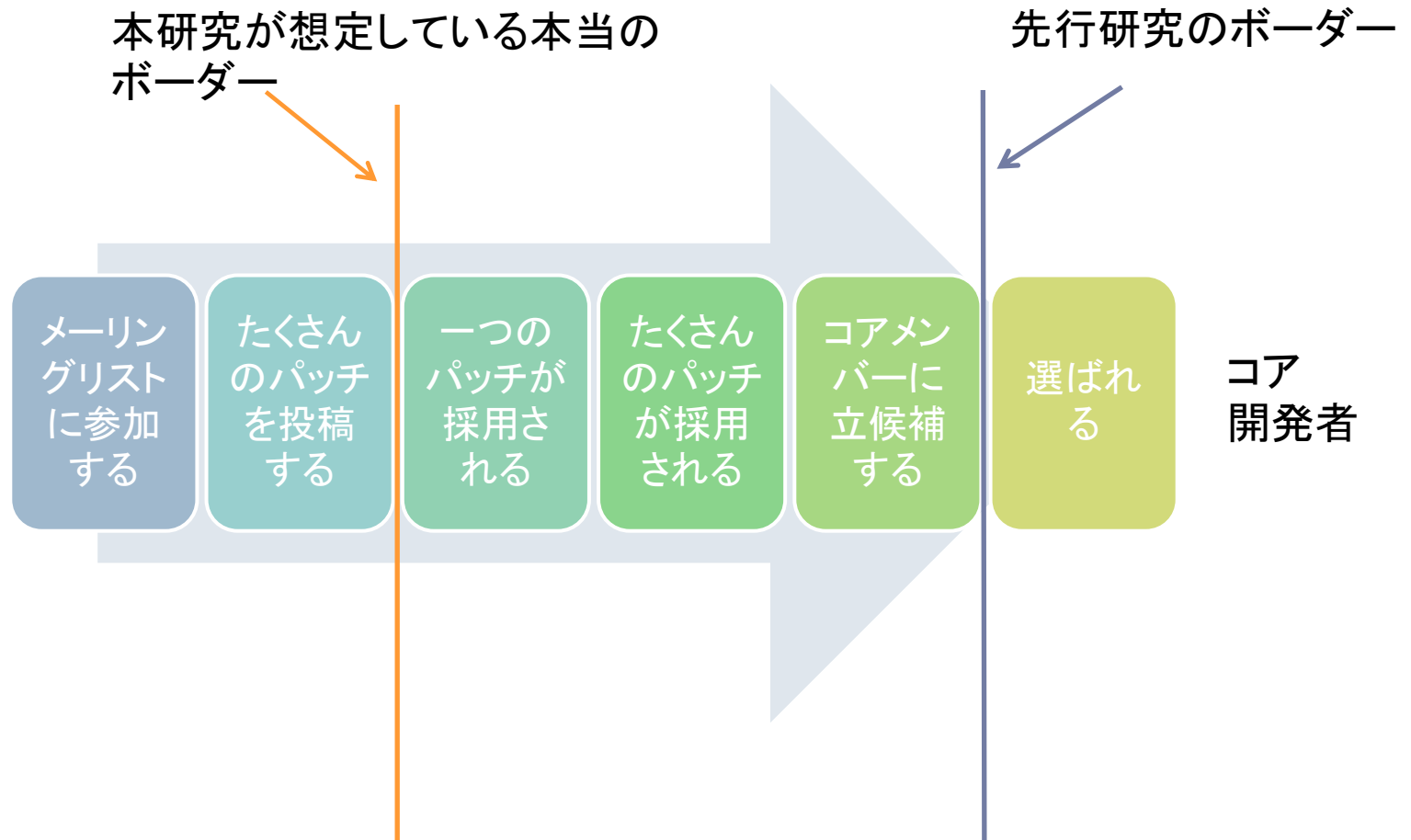
本研究での
新規開発者

先行研究の
新規開発者

GOAL

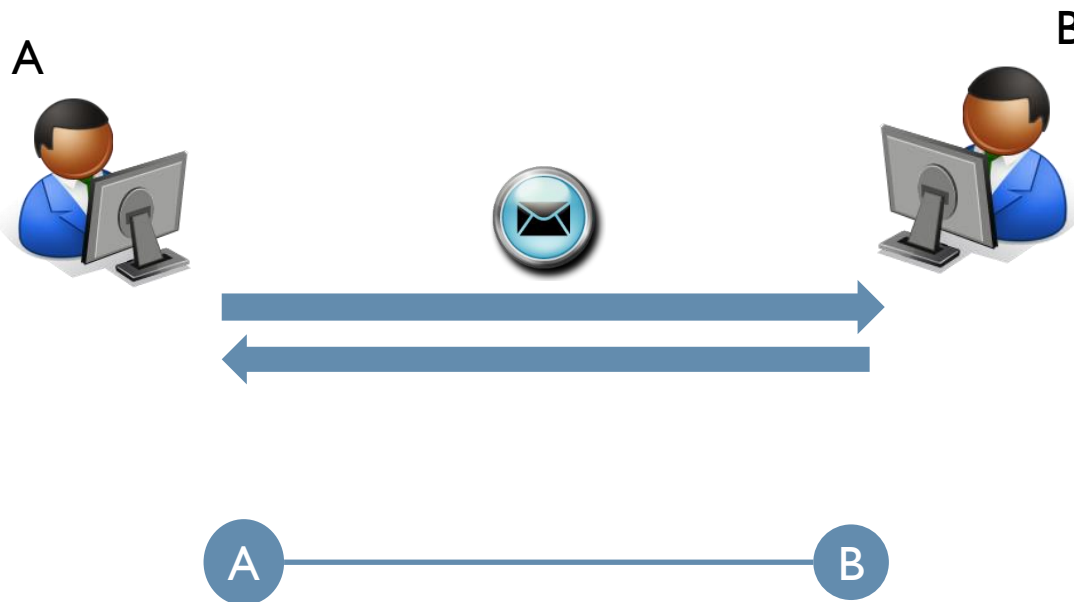


コア開発者になるまでのボーダーの違い＊



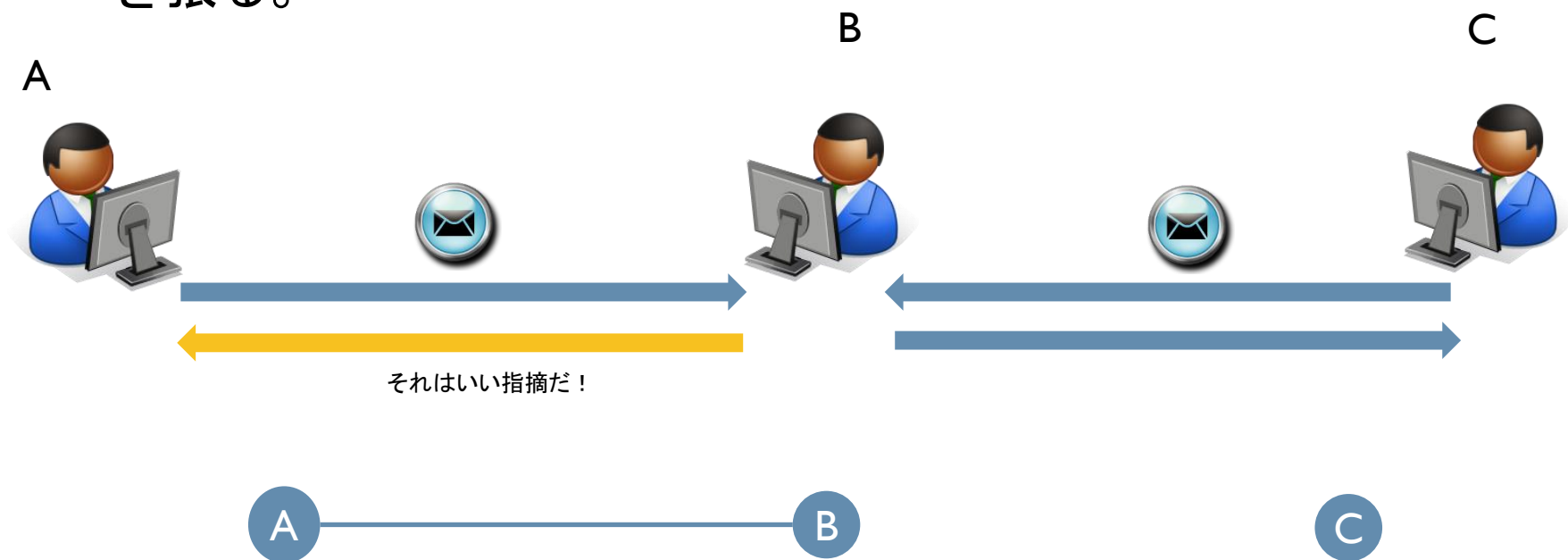
コミュニティ内での評判

- ▶ ネットワーク分析
- ▶ 先行研究ではメーリングリスト上で開発者Aと開発者Bとの間にやりとりがあった場合にノードを張っている。



コミュニティ内での評判

- ▶ 本研究ではメーリングリスト上で開発者Aのメールを開発者Bが興味深いと判断した場合のみ開発者Aと開発者Bにノードを張る。



開発者AのIndegree= $1/2=0.5$

コミュニティの評判とは

▶ 開発者Aの評判

▶ In-degree

- ▶ メールングリストに投稿した開発者Aのメールが興味深いと言われた割合
 - ▶ 開発者BがAに対して返信したメールに興味深いと思ったことを示す内容が書かれていた割合
 - ▶ Norm-Indegree: $\text{=興味を示した人} \div \text{メールングリストの人}$

▶ Out-degree(今回は登場しない)

- ▶ メールングリストに投稿された他の開発者のメールに対して興味深いと言った回数
 - ▶ 開発者Aが他の開発者のメールに興味深いと思ったことを示す内容を書いた割合



In-degree, パッチ投稿, 昇進の関係

- ▶ In-degreeの減少とパッチ投稿数は相関している。
- ▶ パッチ投稿/採用はコアチームに昇進することにおいて重要な要因である。
 - ▶ 上の2つは先行研究で既に判明している。
- ▶ Postgreプロジェクトでは、新規開発者は約3~4年でパッチ投稿を止めることが知られている。

- ▶ よってIn-degreeは約3~4年で低下する。

- ▶ コア開発者への昇進の確率も3~4年で低下するかもしれない？

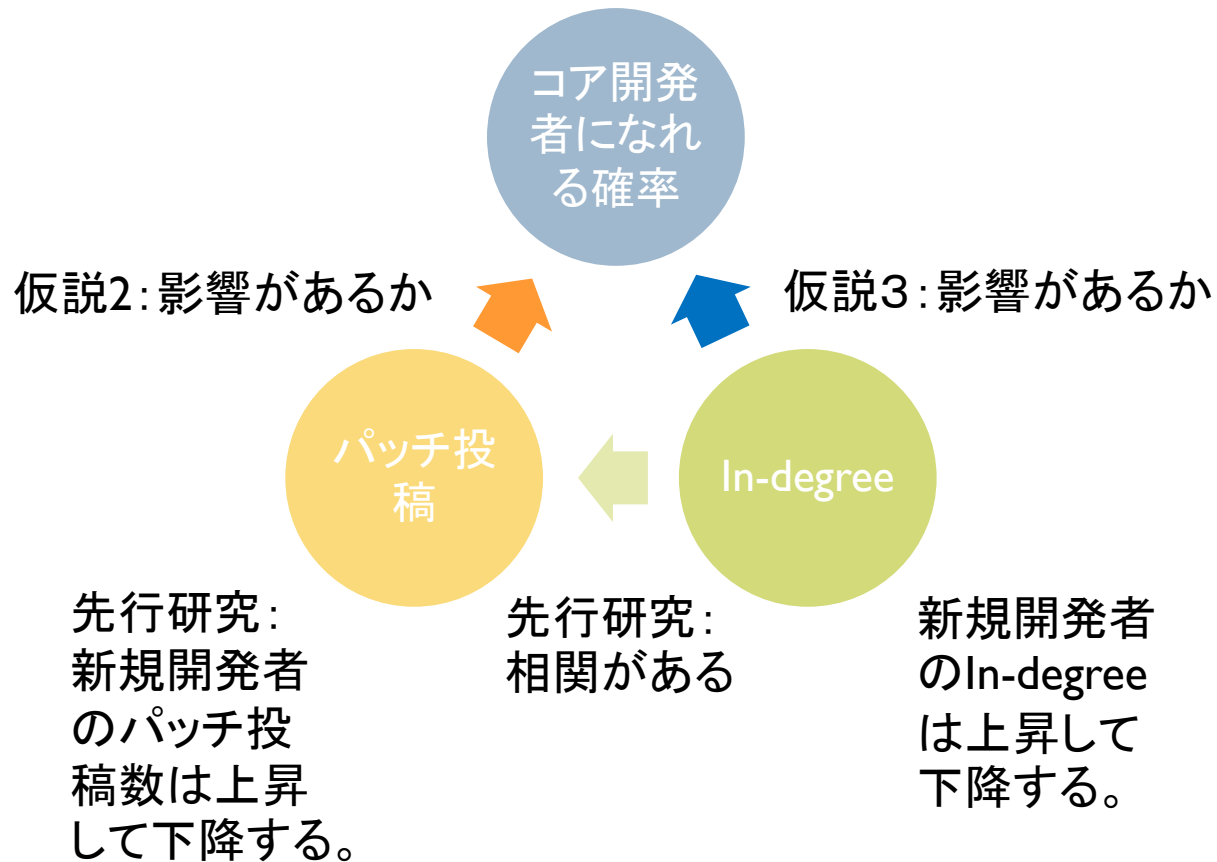
仮説

- ▶ 仮説1: プロジェクトのコア開発者への昇進の確率は常に一定ではない。
 - ▶ プロジェクトのコア開発者になれる確率は、ある一定の期間まで上昇し続け、それを過ぎると下降する。
- ▶ 仮説2: パッチ投稿でのスキルの実証はプロジェクトのコア開発者になれる確率を上げる要因となりうる。
- ▶ 仮説3: コミュニティ内での社会的地位がプロジェクトのコア開発者になりうる確率に大きな影響を与える。



仮説の図解＊

仮説1: コア開発者への昇進の確率は常に一定ではない。



各プロジェクトのコアチームに入るプロセス (Postgre, Apacheプロジェクト)

▶ Postgre

- ▶ コア開発者に2つの種類がある。
- ▶ 前者はコミット権限に加え、誰かを呼ぶことができる権限もある開発者
- ▶ 後者はコミット権限がある人
- ▶ 今回は後者を対象とする。

▶ Apache

- ▶ コア開発者になるための選挙が行われる。
- ▶ 最低6か月の間、アクティブに貢献したものがコアメンバーによってノミネートされ、満場一致で賛成の場合、コアメンバーに入ることができる。



各プロジェクトのコアチームに入るプロセス (Pythonプロジェクト)

▶ Python

- ▶ コア開発者になれるかどうかは“実社会”的に決定される。
- ▶ コア開発者は、以下のようなことで、アクティブに活躍し目立った人を、今後コア開発者になりそうな候補者に書き込む。
 - ▶ パッチ投稿/採用
 - ▶ コア開発者会議のお手伝いをする。
 - ▶ いいアイデアを出す。



分析の流れ＊

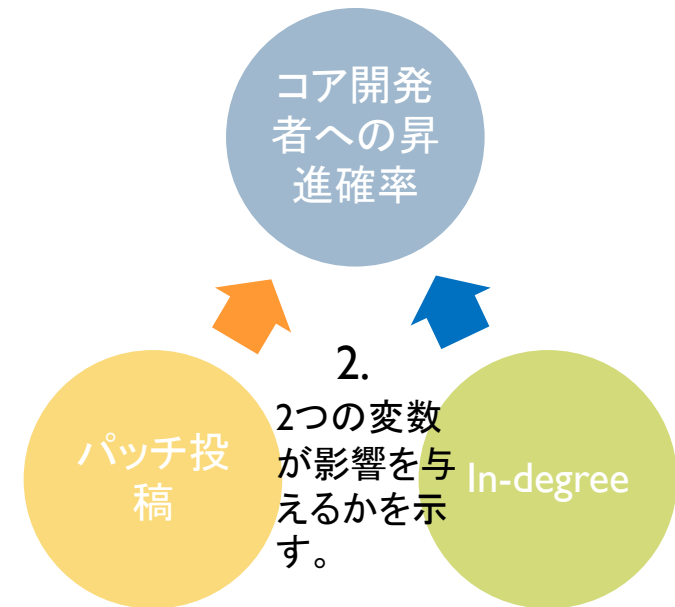
1. プロジェクトのコア開発者への年別昇進確率 データセット

- ▶ メール数(全メーリングリスト)
- ▶ 非コア開発者数
- ▶ その後コア開発者になった数

2. 影響を与える確率がある変数の抽出

- ▶ Patch投稿の有無(バイナリ)
 - ▶ 研究対象: 人ではなくプロジェクト
- ▶ パッチの採用
- ▶ コミュニティ内のコア開発者の数
- ▶ コア開発者以外のメール送信数(1カ月)
- ▶ In-degree

1. 生存曲線をプロット。
昇進確率は常に一定
ではないことを示す。



データセット

※データを取った期間の明確なものは書かれていなかった。

Postgre: 10年前

Apache: 1995年 1/5~2007年? (推定)

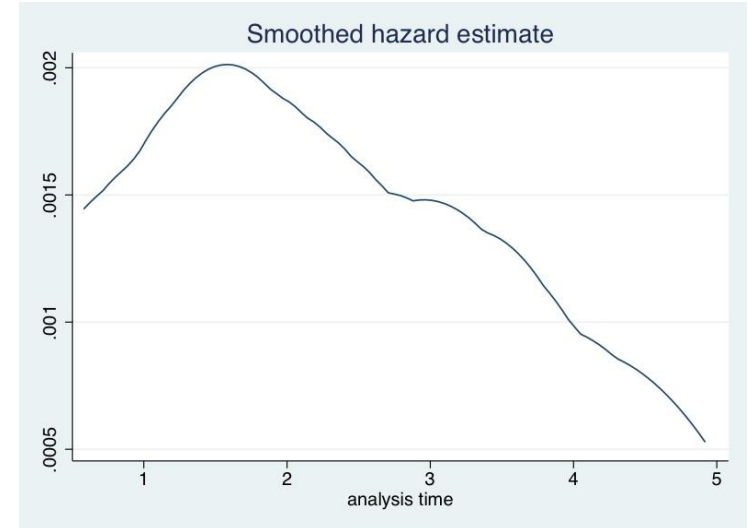
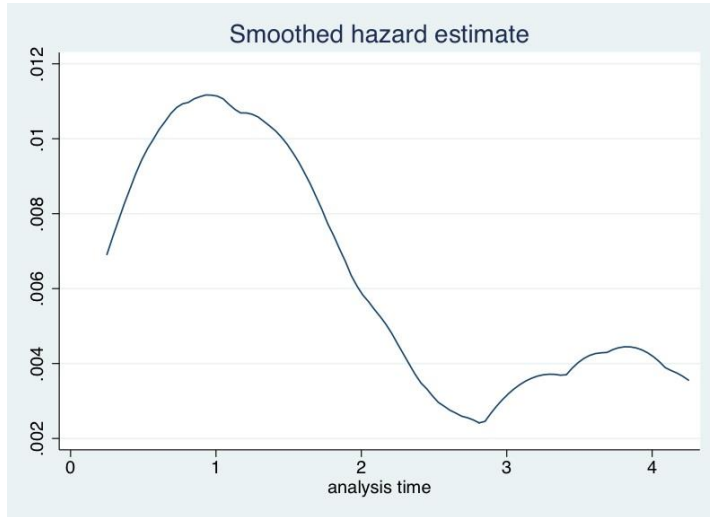
Python: 7年前

プロジェクト	メール数	非コア開発者数	コア開発者に昇進した人
Postgre	169,118	3,283	19
Apache	50,170	1,445	30
Python	45,216	1,320	62



生存曲線：プロジェクトのコア開発者に昇進する確率(Smoothed Hazard Estimate)

▶ Smoothed Hazard Estimate(生存分析した結果を曲線でプロット)

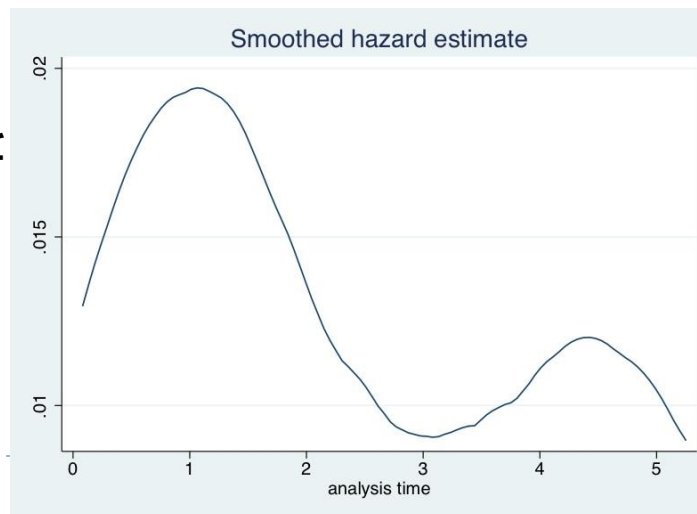


Apache

縦軸: コア開発者になることに成功した人の割合

横軸: プロジェクト経過年数

Python



Postgre

仮説 1 の検証

- ▶ 総合すると仮説1は弱くではあるが、支持された。
 - ▶ 仮説1:コア開発者への昇進の確率は一定ではない。
 - ▶ コア開発者の確率は初め上昇し、ピークを迎えて下降する。
- ▶ Apache,Postgreプロジェクトは支持された。
 - ▶ グラフが初めのうち上昇し、その後ピークを迎えて下降している。
- ▶ Pythonプロジェクトは支持されなかった。
 - ▶ 理由としてPythonプロジェクトはApache、Postgreに比べ4年ほど歴史が短く、新人に対しての反応がまだまだ進化中?



影響を及ぼす確率のある変数の抽出結果： Postgre, Apacheプロジェクト

Postgre	平均	標準偏差	最小値	最大値
Patch投稿の有無	0.04	0.2	0	1
Norm-indegree(最頻値)	0.0024	0.0065	0	0.18
パッチの採用(%)	1.27	9.78	0	100
コミュニティ内のコア開発者の数	13.92	4.14	1	20
コア以外のメール送信数(1カ月)	13.12	63.55	1	2370

Apache	平均	標準偏差	最小値	最大値
Patch投稿の有無	0.1	30	0	1
Norm-indegree(最頻値)	0.0026	0.0049	0	0.87
パッチの採用(%)	1.13	8.98	0	100
コミュニティ内のコア開発者の数	7.34	18.18	1	20
コア以外のメール送信数(1カ月)	46.24	9.76	3	2370



影響を及ぼす確率のある変数の抽出結果： Pythonプロジェクト

Python	平均	標準偏差	最小値	最大値
Patch投稿の有無	0.04	0.2	0	1
Norm-indegree(最頻値)	0.0098	0.032	0	0.70
パッチの採用(%)	0.62	6.98	0	100
コミュニティ内のコア開発者の数	58.96	14.21	0	79
コア以外のメール送信数(1カ月)	13.87	54.44	1	1648



詳細分析結果

Variable	Apache Model			Postgres Model			Python Model		
	Coef.	z	P > Z	Coef.	z	P > Z	Coef.	z	P > Z
tp1 (0-6 months)	-6.69	-2.97	0.003	-20.63	-10.62	0.000	1.10	0.67	0.5
tp2 (6-12)	-6.09	-2.66	0.008	-5.20	-2.32	0.021	1.18	0.69	0.492
tp3 (12-24)	-6.48	-2.7	0.007	-6.51	-2.9	0.004	0.098	0.06	0.955
tp4 (24-36)	-7.98	-2.93	0.003	-5.77	-3.00	0.003	-0.23	-0.13	0.9
tp5 (36-48)	-6.79	-2.53	0.011	-5.98	-2.88	0.004	0.10	0.06	0.96
tp6 (48-60)	-8.16	-2.60	0.009	-8.14	-2.88	0.004	0.91	0.47	0.64
indegree*	61.21	2.78	0.005	72.2	5.46	0.000	5.49	2.92	0.004
patch*	2.71	5.09	0.000	0.98	1.42	0.154	2.09	5.95	0.000
acceptance_rate*	0.15	2.03	0.043	0.0039	0.45	0.656	0.0062	0.10	0.918
sent_cum*	0.014	4.20	0.000	.00158	1.16	0.246	0.0041	1.86	0.062
devs_cum	0.14	1.84	0.066	1.09	2.17	0.03	0.14	2.81	0.005
time_trend	-0.91	-1.90	0.057	-2.06	-1.93	0.054	-1.46	-3.19	0.001

Table 3. Results of Hazard rate model fit. Coefficients represent log-proportional effect of the relevant variable on the hazard rate. For example, in the case of Apache, prior experience of *patch* acceptance (binary variable) increases the rate by $e^{2.71}$, or nearly 15 fold. *tp1*, *tp2*, etc. are time periods with ranges marked in months. Note that in Postgres and Apache the rate increases and then decreases as we move through from *tp1* to *tp4*. Since there are very few non-developers that stay past 4 years, the interpretation beyond this point is unclear.

•各変数が先ほどのhazard estimateにどれくらい影響を与えているかを表している。

•Coef.が影響を与えた値

-indegreeがとても高い！

Patch・・・パッチ投稿経験(バイナリ変数)

Acceptance_rate・・・パッチ採用の割合

Sent_cum・・・1カ月に送信されたメール数

devs_cum・・・コミュニティ内のコア開発者数

Time_trend・・・時間経過による低下



仮説 2 の検証

- ▶ 仮説2は弱くではあるが支持された。
 - ▶ 仮説2: パッチ投稿でのスキルの実証はプロジェクトのコア開発者になれる確率を上げる要因となりうる。
- ▶ Apache, Pythonプロジェクトは支持された。
 - ▶ 強い影響を与えていることが分かった。
- ▶ Postgreプロジェクトは統計的には支持されなかった。
 - ▶ 理由として、Postgreはコア開発者以外からのパッチ投稿が50%あり、他のプロジェクトと比較して知らない人が多い。
 - ▶ コア開発者以外からのパッチ投稿: Apache(31%), Python(23%)
 - ▶ 非コア開発者からのパッチ投稿を平等に扱っている。
 - ▶ よってパッチ投稿で、コアメンバー志望アピールをすることはPostgreプロジェクトでは難しいのでは？



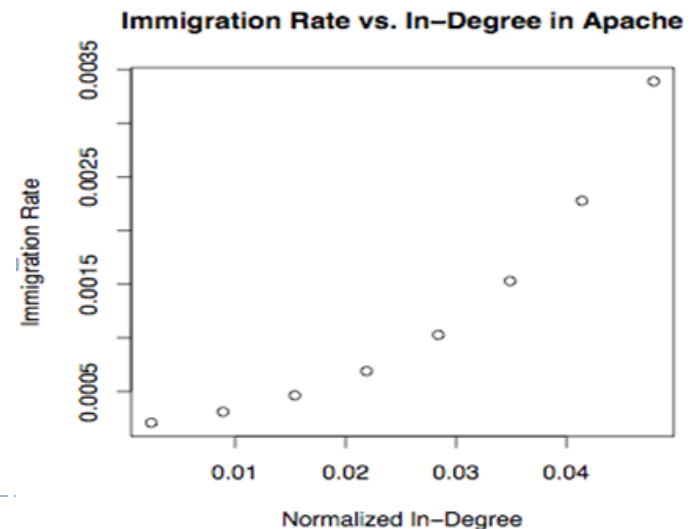
仮説 3 の検証

▶ 仮説3は支持された。

- ▶ 仮説3:コミュニティ内での社会的地位がプロジェクトのコア開発者になりうる確率に大きな影響を与える。(In-degree)

▶ 3つ全てのプロジェクトでの影響が認められた。

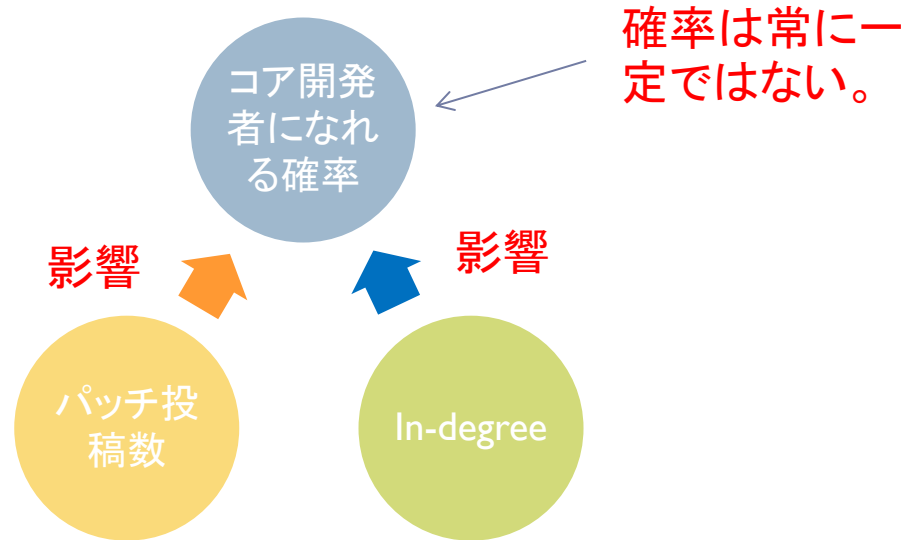
- ▶ In-degreeはとても強い影響を与えていることが分かった。
- ▶ 活発に活動しているという要因は、In-degreeに比べて、あまり影響はなかった。



※著者のスライドより引用

まとめ

- ▶ 著者が立てた3つの仮説は弱くではあるがすべて支持された。



- ▶ コア開発者に昇進するのに最も重要な要因は、パッチ投稿に限らず、**コア開発者の目を引くような投稿すること**ではないか。
- ▶ プロジェクトのコア開発者に昇進する確率は一定期間が過ぎると低下していく。