

Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem

オープンソースプロジェクトにおける 持続的活動のエコシステムレベルでの決定要因: PyPIエコシステムのケーススタディ

オープンソースプロジェクト存続の要因をプロジェクトの依存関係から調べる

著者

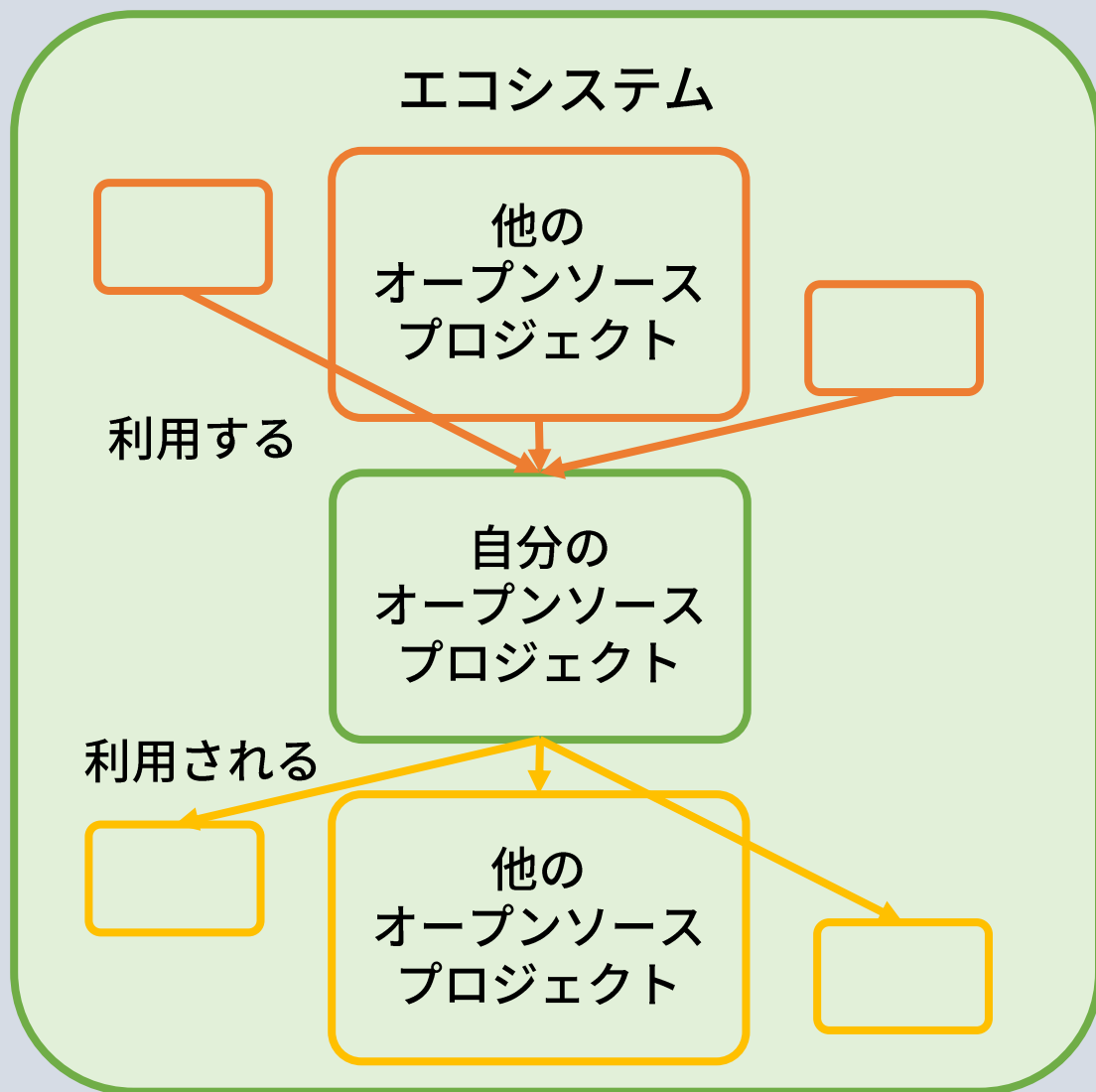
Marat Valiev, Bogdan Vasilescu, James Herbsleb
Carnegie Mellon University

論文情報

ESEC/FSE '18, November 4-9, 2018, Lake Buena Vista, FL, USA

紹介者: 堀口日向

概要



エコシステムがあるプロジェクトが
存続するための要因

プロジェクトレベルでの要因

+

エコシステムレベルでの要因



調べるために・・・

**プロジェクトが休止状態に入る
見込みをモデル化**

背景

オープンソースプロジェクトを利用してアプリケーションを作ることの経済的価値は年間数十億ドルに上る
しかし、多くのユーザはオープンソースプロジェクトのリスクに気付いていない

オープンソースプロジェクトのリスク

- ・知識の喪失
- ・品質の低下
- ・プロジェクトの消失

⇒ オープンソースプロジェクトの存続のリスク

オープンソースプロジェクトの存続を予測するための要因として、
プロジェクトレベル（issueの投稿数やコントリビュータの数など）での研究は盛んにおこなわれてきた

エコシステムレベルでの研究はない

PythonパッケージのエコシステムPyPIでケーススタディを行い
プロジェクトの存続を予測するモデルを作る

仮説の設定

仮説1: 自分のプロジェクトが使っている他のプロジェクトの数が少ないほどプロジェクトの持続可能性が高い

仮説2: 自分のプロジェクトを使っている他のプロジェクトの数が多いほどプロジェクトの持続可能性が高い

仮説3: カッツの中心性（※1）が高いほど、プロジェクトの持続可能性が高い

仮説4: バックポート（※2）を作るプロジェクトは、プロジェクトの持続可能性が高い

仮説5: 大規模な組織によってサポートされるプロジェクトはプロジェクトの持続可能性が高い

※1 カッツの中心性

自分のプロジェクトを使っている他のプロジェクトが持つ中心性を考慮した中心性

固有ベクトル中心性では、エッジが張られていないノードからしかエッジを貼られていない

ノードの中心性が0になるので、それを解決した中心性の指標

※2 バックポート

新版のソフトウェアでしか使えない機能を旧版でも使えるようにすること

方法論 | データセット

PyPIのAPIから登録されているパッケージ全てのメタデータを取得
パッケージ総数：125,699個

このうち、
GitHubでホストされてる2012年以降に登録されたパッケージを使う
パッケージ数：46,547個

パッケージのソースコードを調べたり
サンドボックス上でパッケージをインストールしてみても
要求される他のパッケージを調べることで
対象のパッケージと他のパッケージへの依存関係を抽出

方法論 | モデル(1/2)

生存時間分析とCox比例ハザードを使ってモデル化

- **生存時間分析**

説明変数が生存時間に与える影響を調べるための分析手法。**ハザード関数（次の瞬間死ぬ確率）**を推定。

- **Cox比例ハザード**

時間 t における説明変数 $x_1, \dots, x_n$ のときの**ハザード関数 λ** を以下の式で定義したもの

$$\lambda(t|x_1, \dots, x_n) = \lambda_0(t) \exp(\beta_1 x_1 + \dots + \beta_n x_n)$$

λ_0 ：ある基準となる説明変数セットにおけるハザード

係数 $\beta_1, \dots, \beta_n$ （説明変数の生存時間への影響）を推定することでモデルを求める（Rを使用）

生存時間 S は、ハザード関数を時間 t まで積分することで求める

$$S(t|x_1, \dots, x_n) = \exp\left(-\int_0^t \lambda(t|x_1, \dots, x_n) dt\right)$$

※前提として、説明変数の影響以外では生存時間は変化しない（比例ハザード性）という条件を満たすときに使えるモデル

ハザード関数（次の瞬間死ぬ確率）

$$\lambda(t|x_1, \dots, x_n) = \lambda_0(t) \exp(\beta_1 x_1 + \dots + \beta_n x_n)$$

係数 β_i （説明変数の生存時間への影響）の値の解釈

係数はネイピア数の指数部なので、生存時間について

係数が1より大きいと生存時間が短くなり
係数が1より小さいと生存時間が長くなる

方法論 | エコシステムレベルでの説明変数

- 自分のプロジェクトが使っている他のプロジェクトの数
- 自分のプロジェクトを使っている他のプロジェクトの数
- 自分のプロジェクトが使っている他のプロジェクトのうち休止しているプロジェクトの数
- カッツの中心性
（自分のプロジェクトを使っている他のプロジェクトが持つ中心性を考慮した中心性）
- バックポートが作成されたか/されていないか
- GitHubのアカウントが組織アカウントであるか/でないか
- 大学が関与しているか/していないか
- 企業が関与しているか/していないか

モデルの係数 (β_i) の推定の結果

Table 2: Regression models for early-stage survival and later-stage survival.

	Early-stage survival response: <i>dormant</i> = TRUE Pseudo R^2 = 43.6%		Later-stage survival response: <i>dormant</i> = TRUE R^2 = 17% R^2 = 17.2%			
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
	3.96 (0.05)***					
	24 (0.02)***	7675.27***	1.77 (0.01)***	3317.06***	1.84 (0.01)***	3326.17***
	00 (0.08)***	8276.50***	0.9 (0.05)***	1374.33***		
	91 (0.07)***	79.22***	0.65 (0.04)***	222.92***		
	12 (0.03)***	16.42***	1.09 (0.02)***	17.58***		
	60 (0.04)***	178.37***	0.69 (0.02)***	68.05***		
	25 (0.01)***	380.27***	0.95 (0.01)***	68.25***		
	69 (0.05)***	50.39***	1.11 (0.03)***	13.23***		
	12 (0.02)***	35.29***	1.27 (0.02)***	221.55***		
	08 (0.06)	1.32	0.75 (0.05)***	30.88***		
	51 (0.04)***	125.22***	1.15 (0.03)***	24.51***	1.15 (0.03)***	23.47***
	0.61 (0.12)***	16.63***	0.7 (0.07)	0.21	0.97 (0.07)	0.17
	0.62 (0.06)***	78.42***	0.63 (0.04)***	35.83***	0.84 (0.04)***	34.54***
	0.69 (0.10)***		0.75 (0.07)***		0.75 (0.07)***	
	0.82 (0.04)***		0.98 (0.03)		0.98 (0.03)	
	0.55 (0.04)***	259.95***	0.77 (0.03)***	84.62***	0.78 (0.03)***	79.91***
					1.00 (0.00)	0.08
					0.81 (0.03)***	43.60***
					0.91 (0.11)	0.75

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

初期段階（プロジェクト開始から6か月以内）のデータを使っての推定

初期段階以降のデータを使っての推定

仮説の検証とともに説明していきます

仮説1: 自分のプロジェクトが使っている他のプロジェクトの数が少ないほど、プロジェクトの持続可能性が低い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
Number of upstream dependencies	1.25 (0.01)***	380.27***	0.95 (0.01)***	68.25***	0.95 (0.01)***	19.05***
Some upstreams are dormant	0.69 (0.05)***	50.39***	1.11 (0.03)***	13.23***	1.11 (0.03)***	13.00***

- 自分のプロジェクトが使っている他のプロジェクトの数（表1行目）
初期段階では、係数は1.25なので、この数が多いほど、持続可能性が低くなる
それ以降では、係数は0.95なので、この数が多いほど、持続可能性が高くなる
- 自分のプロジェクトが使っている他のプロジェクトのうち、休止しているプロジェクトの数（表2行目）
初期段階では、係数が0.69なので、この数が多いほど、持続可能性が高くなる
それ以降では、係数が1.11なので、この数が多いほど、持続可能性が低くなる

休止状態の他のプロジェクトは、完成してメンテナンスが必要ないプロジェクトも含まれる
こういったプロジェクトを多く使っている場合は、むしろ持続可能性が高くなる

仮説2: 自分のプロジェクトを使っている他のプロジェクトの数が多いほど、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
Number of downstream projects	1.60 (0.04)***	178.37***	0.89 (0.02)***	68.05***	0.89 (0.02)***	68.08***

- 自分のプロジェクトを使っている他のプロジェクトの数
初期段階では、係数が1.60なので、この数が多いほど、持続可能性が低くなる
それ以降では、係数が0.89なので、この数が多いほど、持続可能性が高くなる

初期段階では効果がないが、それ以降は効果あり

仮説3: カッツの中心性が高いほど、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
Katz centrality	1.12 (0.02)***	35.29***	1.27 (0.02)***	221.55***	1.26 (0.02)***	171.36***

- カッツの中心性
初期段階では、係数が1.12なので、中心性が高いほど、持続可能性が低くなる
それ以降では、係数が1.27なので、中心性が高いほど、持続可能性が低くなる

仮説4: バックポートを作るプロジェクトは、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
Had a backporting release in the last 12 months	0.61 (0.12)***	16.63***	0.97 (0.07)	0.21	0.97 (0.07)	0.17

- バックポートが作成されたか/されていないか
初期段階では、係数は0.61なので、作成されていると、持続可能性が高くなる
それ以降では、係数は0.97なので、有意差なし

しかし、リリースがバックポートかどうかの判定方法の誤りにより
この説明変数が正しく観測できていない可能性がある

とはいえ、初期段階で複数のリリースをすることは、持続可能性を上げる効果があるといえる

仮説5: 大規模な組織によってサポートされるプロジェクトは、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
Hosted under organizational account on GitHub	0.55 (0.04)***	259.95***	0.77 (0.03)***	84.62***	0.78 (0.03)***	79.91***

- GitHubのアカウントが組織アカウントであるか/でないか
初期段階では、係数は0.55なので、組織アカウントであると、持続可能性が高くなる
それ以降では、係数は0.77なので、組織アカウントであると、持続可能性が高くなる

仮説5: 大規模な組織によってサポートされるプロジェクトは、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
High university involvement (>10%)	1.08 (0.06)	1.32	0.75 (0.05)***	30.88***	0.76 (0.05)***	30.35***

- 大学が関与しているか/していないか
初期段階では、係数は1.08なので、大学が関与していると、持続可能性が低くなる
それ以降では、係数は0.75なので、大学が関与していると、持続可能性が高くなる

仮説5: 大規模な組織によってサポートされるプロジェクトは、プロジェクトの持続可能性が高い

	Early-stage survival		Later-stage survival			
	response: <i>dormant</i> = TRUE		response: <i>dormant</i> = TRUE			
	Pseudo R^2 = 43.6%		R^2 = 17%		R^2 = 17.2%	
	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq	Coeffs (Err.)	LR Chisq
High commercial involvement (>10%)	1.51 (0.04)***	125.22***	1.15 (0.03)***	24.51***	1.15 (0.03)***	23.47***

- 企業が関与しているか/していないか
初期段階では、係数は1.51なので、企業が関与していると、持続可能性が低くなる
それ以降では、係数は1.15なので、企業が関与していると、持続可能性が低くなる

まとめ

この研究では、生存時間分析を使用し
オープンソースプロジェクトのライフサイクルの初期段階とそれ以降での
プロジェクトの持続可能性をモデル化した

近年、オープンソースプロジェクトが
ソフトウェアサプライチェーンに組み込まれることが増えているため
開発者は自分が他のオープンソースプロジェクトを利用することの
リスクを評価し、それを軽減するための戦略を学ぶ能力を
向上させる必要がある