

# ネットワーク外部性を利用したオープンソースソフトウェアプロジェクトの持続可能性を高める戦略の提案

Proposed Strategies to Increase the Sustainability of Open Source Software Projects Using Network Externalities

濱田諭・ネットワーク分科会・情報セキュリティ大学院大学

**Abstract:** Open source software is developed by voluntary contributors from all over the world. The cost of development projects is largely dependent on contributions from sponsors and members, and there are no guarantees, so it is not uncommon for projects to suddenly close, collapse, or break up. While many related studies have analyzed internal factors of projects, this paper looks externally and proposes a strategy to increase the sustainability of projects using network externalities in order to survive in a competitive market. First, we verify that network externalities are at work in open source software. Second, we summarize related studies on network externalities that can be used as strategies. Third, we divided network externalities into three phases in terms of time, and proposed strategies for each phase in a form specific to open source software. By comparing the proposed strategy with the preceding case studies, it is possible to estimate the duration of the first and second phases.

## 1. 本研究の背景

- オープンソースソフトウェア(OSS)は現代において、利用していないものはないと言われるほど重要な位置づけである。
- 無償提供されるOSSは、商用製品と違いサポートは保証されず、有志によるボランティアによって開発される。
- セキュリティやバグを改善するためには、定期的なリリースサイクルによる更新が求められる。
- つまり、OSSプロジェクトの持続可能性を高めた、継続した開発が必要となる。

## 2. 本研究の目的

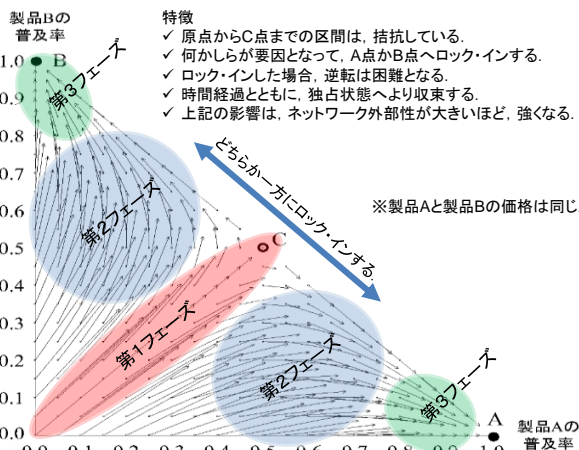
- 関連研究では、OSSプロジェクトの成功や失敗の要因を分析において、プロジェクト内部を調査したものが多く、
- 本研究では、外側からの視点でネットワーク外部性を利用した戦略を提案し、持続可能性を高めることを目的とする。

**ネットワーク外部性(NE)とは:**

- 利用する利用者の増加が、ソフトウェアの価値や魅力を高める「**直接的な効果**」
- 利用するソフトウェアの機能を拡張する外付けのサービスなどが増加することによる価値や魅力を高める「**間接的な効果**」をいう。

## 3. 本研究で提案方法

- ① NEの関連研究から、その特徴と評価結果を収集する。
- ② 収集したものを戦略へ利用できるものを選択しOSSの特徴を考慮して、さらに整理する。
- ③ NEの特徴から、戦略を時間的に3つのフェーズに分ける。  
第1フェーズ:競争段階  
第2フェーズ:ロック・イン状態から独占までの段階  
第3フェーズ:独占段階



出典:岡本, 隆, ネットワーク外部性に基づく製品普及と特性および普及策(2002)

## 4. 本研究で提案する戦略

**第1フェーズ:最も重要な段階, 時間的切迫性が高い**

- ① 早期にユーザー数の獲得およびいきい値を超える戦略をとる。
- ② スポンサーを獲得し、認知と信用を得る。
- ③ 新技術を生かした必要とされる技術を提供する。
- ④ 機能的、社会的、快樂的な向上を図る。
- ⑤ 競争の有利や不利に対して、計画を前倒したサービスの向上を図る。
- ⑥ 初期の配布を地理的に集中させて行う。
- ⑦ モジュール化の体制をとり、開発者やユーザーが貢献しやすい環境を作る。間接的な外部性の影響を受けやすくする。

**第2フェーズ:時間的切迫性が高い**

- ① 互換戦略をとる。なるべく早期に採用する。時間が経つほど、互換性の効果は薄まる。
- ② 固有の便益を大きく高めて、差別化や示差性をアピールする。ユーザーの評価にばらつきがある場合に、より優位となる可能性が高い。

**第2フェーズ(プラットフォーム型)**

- ① プラットフォーム間互換では、補完財となるサービスが増加すると互換の便益が薄くなる。ユーザーの選好によって、決まる。
- ② プラットフォーム内互換では、市場での優位性が高まる。

**第3フェーズ**

- ① 競争市場において、共存か撤退をする。
- ② 新しい市場へ新技術などをもって、参入する。
- ③ プロジェクトの解散も視野に入れる。

## 5. OSSの先行事例との比較検討

- 第1フェーズは、2015年から2017年の約1~2年
  - 第2フェーズで、Kubernetesとの統合する「プラットフォーム内互換戦略」をとる。
  - 2017年のMoby Projectによる「差別化戦略」
- (事例1) Docker

2013年

- Dockerがリリースされる。

2015年

- CNCFが発足し、Kubernetesをホストし、その開発主体となる。

2017年

- DockerがKubernetesと統合することを発表する。
- Docker社は、Moby Projectを発表する。

## 6. まとめ

- フェーズに分けた戦略をOSS先行事例とで比較検討もより、**第1フェーズの期間が約3年であることを示した**。
- 戦略の傾向として、**共存/互換戦略を早期に行い、差別化のため固有の特徴をアピールする戦略が多いことが示された**。