様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２９　年　　６　月　２８　日現在

機関番号：　１２１０２
研究種目：　基盤研究(C)（一般）
研究期間：　2014～2016
課題番号：　２６３８０２８７

研究課題名（和文）Industrial Policy Implications of Encouraging Open Source Software in Commerce and Open Projects

研究課題名（英文）Industrial Policy Implications of Encouraging Open Source Software in Commerce and Open Projects

研究代表者

Ｓ．Ｊ　Ｔｕｒｎｂｕｌｌ（TURNBULL, Stephen）

筑波大学・システム情報系・准教授

研究者番号：９０２４０６２１

交付決定額（研究期間全体）：（直接経費）　　1,900,000円

研究成果の概要（和文）：パイソン言語の開発コミュニティの歴史について参加者のインタビュー調査とウェブ上の資料を調査し、結果をまとめた。
（１）パイソンコア開発者チームは小数人から数百人に拡大した。ほとんどがボランティアである。（２）フォウンダーが最終決定権を保留しても意図的に大きい権力を持つコア開発者を育てきた。（３）人数が多いに関わらず、協力的行動が染み込んで議論の結果がコンセンサスとなる場合がほとんど。（４）コンセンサスを促す公式的な課程設置、その記録に基づいた「文化」が以降の議論コンセンサスに導く。
（２から４）は日本人の価値観に近い「パイソンモデル」を日本のソフトウエア産業に導入することを検討すべきと考えられる。

研究成果の概要（英文）：An interview-based case study of the Python language development community, along with consideration of the web site and mailing list archives was conducted to create a history of the community.
Findings: (1) The Python development team grew from a handful to several hundred core developers, mostly volunteers, with authority to add new features and bugfixes. (2) The founder maintains a veto over design and implementation, but has deliberately encouraged the the inclusion of many new core members with great authority. (3) Despite the large number of developers, cooperative behavior is deeply rooted, and most discussions end in a working consensus. (4) Formal institutions that record discussions and their results support a "culture" of consensus, smoothing later discussions.
In view of the compatibility of findings (2-4) with Japanese values, it is recommended that introduction of the "Python model" into the Japanese software industry be considered.

研究分野：　management of technology

キーワード：　software development　management of technology　open source　Python language　case study

１．研究開始当初の背景

(1) Whether the open source software development model can be self-sustaining has long been controversial within the open source sector itself, and in the software development industry as a whole. The open source sector has thrived since Richard Stallman's GNU Manifesto in 1985. Industry leaders such as Oracle and Microsoft support some open source projects and including them in their own distributions, while "pure" open source firms such as Red Hat broke into the Fortune 500. Thus, survivability in the face of fierce competition from products funded by copyright and patent licensing fees is empirically demonstrated. But the theoretical foundation remained unclear.

(2) Still hotly debated in the open source sector are various models for governance and management of projects developed as open source software. Eric Raymond argued for an anarchic "bazaar" model of developers who coordinate around a "coincidence of interests," which might fracture the community if interests of the developers diverged. However, many successful projects have a more forward-looking attitude toward cooperation. Even senior leaders accept "good enough" for their own purposes in the face of a consensus of other developers.

(3) In 2010, Minister for Government Revitalisation Renhou (Murata) famously asked, "Isn't #2 good enough?," questioning the need for extensive government support of R&D. In today's software industry, "Japan as No. 1" is not an option. U.S. firms are expected to continue to dominate the industry. The open source development style allows sharing of actual implementation as well as data structures and algorithms, which allows developers to concentrate on new features. It seems well-adapted to the mindset of Japanese engineers, who often share their projects with like-minded friends, suggesting the incorporation of open source development models in the Japanese software industry as a way to leverage both Japanese culture and the cost advantages of "free" software. This model seems particularly compatible with Japanese organizational values emphasizing harmony, consensus, and compromise.

２．研究の目的

(1) Determine theoretical conditions under which the overwhelmingly superior financial resources available to proprietary software is insufficient to enable it to crowd out open source software in the long run.

(2) Examine successful project governance structures which have been shown to thrive in the software industry, and consider whether they are adaptable to the Japanese context.

３．研究の方法

(1) Sustainability of the open source sector is addressed with a simple model of a two-sector software industry. It consists of an open source sector producing new software with volunteer labor and (free) spillovers from the existing stock of open source software, and a proprietary sector using paid labor, (free) spillovers from the open source sector, and paid licenses for a portion of the stock of proprietary software. This specification is intended to give the maximum advantage to the proprietary sector.

(2) As a case study of a governance structure suitable for promoting open source development projects in the Japanese software industry, we studied the history and current governance structures of the Python computer language development community. Extensive interviews with core developers and support staff, users (developers of application software) and community activists (such as conference organizers) were conducted, and the mailing lists and other web resources examined.

４．研究成果

(1) The economic growth model used to analyze the theoretical survivability of an open source sector is a simple modification of Solow's model with two sectors having a common production function. The reduced form (per capita) model is the system of

differential equations

$$\dot{k}_i = f(\alpha_i k_2 + k_1) - n_i k_i, \quad i = 1, 2 \qquad (1)$$

$$f(k) > 0, \quad f'(k) > 0, \quad f''(k) < 0 \qquad (2)$$

where k_i is the capital-labor ratio in sector i, n_i is the rate of labor force growth in sector i, $\alpha_1 = 0 < \alpha_2 \leq 1$ indicates that the open source sector 1 has no access to software from the proprietary sector 2, and the production function f is "neoclassical."

(2) Decreasing marginal returns ensure that the open source sector grows at least as fast as the proprietary sector as long as $n_1 \geq n_2$. This inequality is ensured by assumping that utility of developers is homothetic in consumption goods (purchased by working in the proprietary sector) and leisure (of which a constant proportion is devoted to volunteer work in the open source sector). Homothetic utility is justified by recent labor market practice, where leading companies their developers time off to work in volunteer projects. Thus it is plausible that the open source development model can thrive in competition with the proprietary development model, despite the latter's overwhelming advantage in financial resources.

(3) With sustainability established, the prospect for application of open source development methodology in Japan was examined by studying the governance of the Python computer language development process. Interviews were conducted with leading developers including the founder, Guido van Rossum, community iactivists such as the organizers of the annual North American Python Convention (PyCon US), downstream developers in "open" projects using Python, and commercial developers using Python, as well as Japanese developers in various enterprises.

(4) Trying to manage a community of volunteer software developers has often been compared to "herding cats," as they are notoriously self-centered and arrogant. In Raymond's bazaar model, developers exchange small contributions to satisfy a personal or business needs for improvement of the software product of the project. Cooperation is reduced to cases of "coincidence of interests" among the developers. However, the interviews make clear that in the case of Python, there is a more forward-looking attitude toward cooperation, with even the most senior leaders willing to accept "good enough" for their own purposes in the face of a consensus of other developers. Even van Rossum, who is known as the "Benevolent Dictator for Life" in the community and has an effective veto over changes to the Python language and the core set of extensions and libraries bundled with it, sometimes concedes his favorite design to the consensus of other developers that a different design is better for the community as a whole.

(5) The voluminous documentation and correspondence of the Python community gave insight into the core values of the developers. Discussions on the mailing list and documents such as "The Zen of Python" provide a clear statement of shared values, such as readability over performance and concise expression, and a focus on demonstrated usefulness of implementations rather than theoretical purity. Some values are believed to be universal, while others express the preferences of the developers. These values are frequently referenced in the language design documents, the Python Enhancement Proposals (PEPs), provide a focus for discussions, and give a guide to likely outcomes when deadlocked discussion results in an appeal to van Rossum or one of his lieutenants.

(6) The Python project hass an explicit focus on the development process and community development. Several PEPs describe not only highly technical aspects of the coding process, such as style guides, but also provide standards for distribution of libraries and applications, and development of infrastructure such as code repositories, web sites, online documentation, and issue tracking. The project's intellectual property is protected by the non-profit, independent Python Software Foundation, which

has more recently expanded into education and diversity efforts.

(7) This “rough consensus and working code” approach is reminiscent of other successful projects, including the industry-leading Apache webserver and the Internet Engineering Task Force itself. From the point of view of adaptability to the Japanese context, it is important to note that while the development process is highly systematized, only parts of it are defined by formal documents. Much of the process is documented only in the “oral tradition” of the core developers, but is quite public on the mailing lists and other channels. The whole system is highly evolutionary, supported by a common consciousness of values and processes, similar to Japanese “joushiki.”

(8) To business, proprietary software offers large benefits from control of technology and exclusion of rivals, and remains important for mission-critical products. But in this crucial field Japan is slow in catching up to the U.S., while low-cost “offshore” development organizations in nations such as India and China, and even Korea, Ireland, and Israel, are rapidly achieving technical parity with Japan while continuing to offer relatively low cost. They also possess a native ability to interact with customers in English, an additional advantage over Japan in the global market. In this environment, use of open source licenses and community development models for infrastructure products such as programming languages, accounting, MIS and customer relations management systems, and website content management can provide cost reductions to businesses. At the same time it opens up educational opportunities for both working professionals and students in STEM, increasing the pool of skilled developers, system administrators, and security professionals.

＜引用文献＞

①Raymond, E. S. The cathedral and the bazaar. http://www.catb.org/esr/writings/cathedral-bazaar/cathedral-bazaar/
②Solow, R. 1956. A contribution to the theory of economic growth. Quarterly Journal of Economics 70:65-94.
③Stallman, R. 1985. The GNU manifesto. https://www.gnu.org/gnu/manifesto.html

５．主な発表論文等

〔雑誌論文〕（計１件）

①TURNBULL, Stephen J.、YOSHIDA, Masatoshi、The Balanced Budget Multiplier and Labour Intensity in Home Production、International Journal of Economic Behavior and Organization、査読有、Vol.3、No.2-1、2015、pp.21-30

〔学会発表〕（計６件）

①TURNBULL, Stephen J.、Chasing Diagrams to Equilibrium: A Speculative Application of Category Theory to Games、Further Development of Dynamic Economic Research 2017 Spring Edition of Hayama Meeting、2月13日、2017
②TURNBULL, Stephen J.、You Can Help Develop Python – and You Should!、 PyCon Pune、 Pune India、 2月18日、2017
③TURNBULL, Stephen J.、Where We Came From, What’s Special About Now, and Where We Might Go from Here、 PyCon Canada、Toronto Canada、11月14日、2016
④TURNBULL, Stephen J.、Project Governance in Open Source Software: A Case Study of Python、Society Technology and Japanese Entrepreneurship Conference 2015、Stanford CA USA、 4月3日、2015.
⑤TURNBULL, Stephen J.、The Zen of (Python) Software Maintenance、FOSSAsia 2015、Singapore、3月14日、2015.
⑥TURNBULL, Stephen J.、How to Grow an Open Source Project: A Case Study of Python、Further Development of Dynamic Economic Research 2015 Spring Edition of Hayama Meeting、Hayama Kanagawa、3月7日、2015

６．研究組織

(1)研究代表者

S.J.ターンブル（TURNBULL, Stephen）
筑波大学・システム情報系・准教授
研究者番号：　９０２４０６２１