

# The Smartphone Manufacturers' New Product Development Performance in the Generation Transition of Android Ecosystem: Exploiting Old Technologies to Enhance the Effects of Exploring New Technologies

## 安卓生態系的世代轉移中智慧型手機廠商的新產品開發表現：深化舊技術更能有效探索新技術

Jing-Ming Shiu, Department of Business Administration, National Cheng Kung University  
許經明 / 國立成功大學企業管理學系

*Received 2021/4, Final revision received 2022/1*

### Abstract

This research analyzes the new product development performance of 11 smartphone manufacturers (i.e. complementors) in a total of 10 generation transitions from Éclair Android 2.1 to Android 10.0 in the Android ecosystem (i.e. platform-based ecosystem) of Google (also known as platform vendors). My analysis results show that both the exploitative learning in the old generation and the explorative learning in the new generation of Android operating systems have effectively improved smartphone manufacturers' new product developments. In addition, the higher the degree of exploitative learning in the old generation, the more effective it is to increase the positive impact of explorative learning in the new generation on the smartphone manufacturers' new product development performance. This research supplements earlier studies in the discussions about the ambidexterity of complementors and the performance of new product development in the platform-based ecosystem theory.

**【 Keywords 】** Android ecosystem, ambidexterity, explorative learning, exploitative learning, smartphone

## 摘要

本研究分析在谷歌 (Google) (係一平台廠商) 主導的安卓生態系 (Android Ecosystem) (係一平台基礎生態系, Platform-based Ecosystem) 中, 11 家智慧型手機廠商 (互補財廠商) 從 Éclair Android 2.1 至 Android 10.0 之間, 共 10 次安卓版本的世代轉換過程中之新產品開發表現。本研究的分析結果顯示, 智慧型手機廠商對舊世代安卓作業系統所進行的深化性學習 (Exploitative Learning), 以及對新世代安卓作業系統進行的探索性學習 (Explorative Learning), 皆有效地提升了智慧型手機廠商在世代轉換中新產品開發的表現。除此之外, 智慧型手機廠商對舊世代安卓作業系統所進行的深化性學習之程度越高, 對新世代作業系統進行探索性學習以提高新產品開發的表現程度將會越好。本研究補足了平台基礎生態系理論中互補財廠商學習的雙面性 (Ambidexterity) 與新產品開發表現等相關討論, 並且補足了開放源軟體的研究中對於企業貢獻與新產品開發表現的討論, 提供了新觀點進一步釐清企業在使用開放源軟體的策略與相關表現。

**【關鍵字】** 安卓生態系、雙面性、探索性學習、深化性學習、智慧型手機

## 壹、緒論

在一個以技術進步為基礎的生態系中，平台廠商主要扮演持續創造新世代技術的角色，而互補財廠商則是要迅速使用平台廠商的新世代技術推出新產品，這樣的生態系被稱為平台基礎生態系 (Platform-based Ecosystem) (Kapoor and Agarwal, 2017)。過去研究大多從平台廠商的觀點探討如何藉由公開與更新平台的介面 (Interface) (Boudreau, 2010; Yoo, Boland, Lyytinen, and Majchrzak, 2012)，以及訂定與分配智慧財產權 (Intellectual Property Rights) (Tiwana, Konsynski, and Bush, 2010) 等提高互補財廠商的創新動機，相較之下，從互補財廠商的觀點來探究平台基礎生態系的研究則尚未充分 (Kapoor and Agarwal, 2017; Rietveld, Ploog, and Nieborg, 2020)。

詳言之，過去研究僅提及互補財廠商在平台基礎生態系中學習的重要性。例如，互補財廠商需要學習如何因應平台技術所產生的變化進行資產的規劃與投資 (Jacobides, Cennamo, and Gawer, 2018; Jacobides, Knudsen, and Augier, 2006; Teece, 2018)，互補財廠商需要學習如何在平台與互補財之間形成的技術相互依存性之下進行互補財的開發 (Kapoor and Agarwal, 2017)，以及互補財廠商應該學習如何有效地利用平台廠商提供的開發工具或者參考設計輔助互補財的開發 (Piller and Walcher, 2006; Shiu, 2017; Von Hippel and Katz, 2002)。不過，這些研究卻未針對互補財廠商如何因應平台的技術變化而採用不同的學習策略加以探討。

過去研究指出，互補財廠商在平台上會進行世代性產品創新 (Generational Product Innovation) (Chen, Zhang, Li, and Turner, 2021)，並且在平台上與其它互補財廠商競相提高創新的專屬性 (Appropriability) (Miric, Boudreau, and Jeppesen, 2019)。互補財廠商的創新策略會受到消費者需求端的影響 (Wang, Aggarwal, and Wu, 2020)，而需要不斷地進行測試與微調產品設計才能提高新產品開發的成功率 (Chen, Wang, Cui, and Li, 2021)。不過，互補財廠商在平台的世代轉換 (Generation Transition) (Ansari and Garud, 2009; Cennamo, 2018; Kapoor and Agarwal, 2017; Kretschmer and Claussen, 2016) 中，更會面臨平台本身的技術的複雜度與不確定性 (Cennamo, Ozalp, and Kretschmer, 2018; Kapoor and Agarwal, 2017)。互補財廠商在平台的世代轉換之中，若是無法採用適當的新產品開發方式，就將會失去自身在產品市場優勢。

本研究認為，當平台技術是透過世代轉換而持續進步的時候，互補財廠商需要針對平台的技術變化進行學習，藉此才能提高自身在世代轉換過程中的表現。本研究進一步將組織學習理論 (Organizational Learning Theory) 導入生態系研究領域，並且探討互補財廠商如何因應平台的世代轉移而採用深化性學習 (Exploitative Learning) 與探索性學習 (Explorative Learning) 這兩項策略。過去的組織學習理論

將新產品開發視為一個問題解決 (Problem-solving) 的過程 (Nickerson and Zenger, 2004)，探索性學習與深化性學習則皆可提升新產品開發等企業競爭力的表現 (Lee and Huang, 2012)。因此，我們認為採用這樣的觀點可以釐清與分析互補財廠商如何因應不同世代平台的技術變化，而採用不同的學習策略來提高新產品開發表現。具體而言，本研究聚焦分析在安卓 (Android) 智慧型手機生態系 (Koch and Kerschbaum, 2014; Liu, Liu, Zhu, Wang, and Zhang, 2019) 中，智慧型手機廠商（亦即互補財廠商）如何針對安卓作業系統（亦即平台，而 Google 則是平台廠商）的世代轉換來進行深化性與探索性學習，又如何據此提升智慧型手機開發的表現。

本研究補足了互補財廠商如何學習適應平台技術變化等之探討，特別是補足了組織雙面性 (Ambidexterity) 對於新產品開發等表現之研究不足之處 (Raisch, Birkinshaw, Probst, and Tushman, 2009)。本研究發現，當平台的世代轉移是具有技術連續性的時候，智慧型手機廠商對舊世代安卓作業系統所進行的深化性學習，以及對新世代安卓作業系統進行的探索性學習，皆有效地提升了智慧型手機廠商在世代轉換中新產品開發的表現。除此之外，智慧型手機廠商對舊世代安卓作業系統所進行的深化性學習之程度越高，越能提升在新世代作業系統進行探索性學習對於提高新產品開發表現的效果。本研究章節安排如下：第貳部分回顧組織學習的理論背景並據以推導出本研究的假說；第參部分敘述研究方法，第肆部分闡述本研究的分析結果。第伍部分歸納本研究的理論貢獻並提出未來研究課題。

## 貳、理論背景暨假說

### 一、組織學習的雙面性

組織長期的成功取決於是否有能力去深化學習自身現有能力的同時，同時探索學習新的基礎性技能 (Levinthal and March, 1993; March, 1991)。深化性學習與探索性學習對於新產品開發至關重要 (Atuahene-Gima and Murray, 2007)，但這兩種學習方式是既具有互斥也具有互補的邏輯 (Kuo and Lee, 2019; Wei, Yi, and Guo, 2014)。過去研究認為組織的資源稀少性與組織的慣性 (Organizational Inertia) 會讓深化性學習與探索性學習之間產生互斥的邏輯思考 (Lavie, Stettner, and Tushman, 2010)。互補的邏輯思考則是強調企業需要同時追求探索性學習與深化性學習的最佳組合，也就是基於組織的雙面性來實現卓越的創新績效 (Andriopoulos and Lewis, 2009; Cao, Gedajlovic, and Zhang, 2009; Gibson and Birkinshaw, 2004; He and Wong, 2004; Tushman and O'Reilly, 1996; Yalcinkaya, Calantone, and Griffith, 2007)。

組織的雙面性屬於動態能力 (Lin, Peng, Peng, and Teng, 2020; O'Reilly and Tushman, 2008; Siggelkow and Levinthal, 2003; Westerman, McFarlan, and Iansiti,

2006)，也是能夠讓企業因應環境的變化而獲取持續競爭優勢的重要條件 (Hannan and Freeman, 1984)。當環境變化的速度較慢的時候，企業需要採用同步型之組織的雙面性，也就是同時進行深化性學習與探索性學習；而當環境變化的速度較快的時候，企業可以採用順序型之組織的雙面性 (Sequential Ambidexterity) (Boumgarden, Nickerson, and Zenger, 2012)，也就是基於深化性學習（或者探索性學習）發展探索性學習（或者深化性學習）。

此外，企業的探索性學習與與深化性學習會因為在創新活動的不同階段中有著不同的效果 (Puranam, Singh, and Zollo, 2006)。通常企業在進行新產品開發初期，探索性學習將可使企業透過搜尋、創造、發現、新穎、創新與實驗創建多樣性；在新產品開發後期，則是經由深化性學習來提高可靠性與建立慣例 (Routines)。因此，探索與深化之間存在著一個自然循環 (Rothaermel and Deeds, 2004)，企業可以據此規劃，按照探索性學習與深化性學習的順序性持續進行新產品開發。例如，Brown and Eisenhardt (1997) 針對 1993 年至 1995 年電腦產業中六家企業的個案研究即指出，成功的新產品開發是以固定的週期轉移到新世代的產品開發專案，也就是在深化既有的新產品開發之後，緊接著進行探索未來的產品開發專案。

## 二、平台世代轉換中互補財廠商的雙面性

平台基礎生態系中的平台代表著產品的核心概念與定義了結構設計，亦是產品世代進化的技術性瓶頸 (Technological Bottlenecks) (Jacobides et al., 2006; Rochet and Tirole, 2006)。因此，平台的世代轉換意味著平台內部的瓶頸技術或者核心零組件產生技術變化，在平台中的既有功能有著明顯的改善或者增加新的功能 (Ansari and Garud, 2009; Kapoor and Agarwal, 2017; Venkatraman and Lee, 2004)，也改變了生態系中零組件之間的互動關係 (Adner and Kapoor, 2010; Ansari and Garud, 2009; Venkatraman and Lee, 2004)。Kapoor and Agarwal (2017) 指出一個平台的世代轉換代表著 Henderson and Clark (1990) 所稱之結構變化，也就是平台本身的核心設計並沒有突破式變化，但是對於平台和互補財廠商各自的技術之間的相互依存性會造成影響，使得前一代平台的互補財廠商在邁入到新世代平台的時候遇到挑戰。

基於上述原因，當互補財廠商面臨平台的世代轉換，需要重新學習平台的技術知識以及平台與自身產品的技術相互依存性，才能有效地把平台整合到新產品開發 (Kapoor and Agarwal, 2017)。在產品生命週期非常短且產品消費市場的變化非常快速的情況下 (Eisenhardt and Martin, 2000)，互補財廠商必須要在平台的世代轉換期間內規劃深化性與探索性學習 (Gupta, Smith, and Shalley, 2006)，並且在新世代平台問世之後及時推出互補財；因為平台本身具有生命週期的特徵，若是推出互補財的時間過長，恐將失去市場先機 (Rietveld and Eggers, 2018)。



過去研究指出，在一個技術相互依存性的環境之下，企業需要採用探索性學習來尋找問題的最適解。因此，互補財廠商在面臨技術與市場的壓力之下，需要在新世代平台正式推出之前針對新世代平台的技術特性等來進行探索性學習，互補財廠商若是能夠先行地進行探索性學習，就將能夠享有先發者優勢。這是因為後發互補財廠商在尋找方案的過程中，儘管進行了創新與識別了較高的性能組合，但是這個組合可能只限定在尋找方案中的一個局部最佳化，並非在整體尋找方案中的最優化 (Rivkin, 2000)。當平台的世代轉換讓平台與互補財之間的互動變得更為複雜的時候，後發的互補財廠商在提升表現就越有可能需要承擔因為些許選擇錯誤而導致全盤失敗的風險 (Levinthal, 1997)。因此，互補財廠商在新世代平台正式推出之前，針對新世代平台的技術特性等進行探索性學習，就將可以比其它廠商更有效率地使用新世代平台開發互補財。基於上述，我們提出本研究第一個假說。

假說一：

互補財廠商對於新世代平台的探索性學習的程度越高，其越容易貢獻於該互補財廠商在新產品開發的表現。

互補財廠商在平台的世代轉換中，不僅需要對新世代平台來進行探索性學習，亦是需要對舊世代平台進行深化性學習。深化性學習是一種進行局部尋找既有知識周邊的新知識 (Helfat and Raubitschek, 2000) 之學習策略，因此深化性學習的知識具有累積性 (Cohen and Levinthal, 1990) 也能創造出組織學習方式的路徑依存性 (Helfat, 1994; Nelson and Winter, 1982)。深化性學習與產品知識有關 (Gomory, 1987; Gomory and Schmitt, 1988)，並且會有聚焦於特定產品的工程與顧客導向之情況 (Rosenberg, 1982)。此外，學習經驗的積累取決於工作的新穎性，當工作的新穎性較低的時候，企業應該採用深化性學習的方式累積經驗。例如舊世代平台的新穎性較低，互補財廠商對於舊世代平台的深化性學習將可以提高新產品開發的效率，以及可以形成在新產品開發過程中解決問題的慣例。

通常平台廠商在每個世代的平台之後仍然會繼續釋出可修正問題與改善產品性能的平台版本。互補財廠商對於舊世代的平台進行持續學習，將可以持續解決互補財的問題與提高互補財的性能。過去研究亦指出，企業在新產品開發結束之後會進行事後檢討 (Kumar and Terpstra, 2004)，以學習與進行專案與專案 (Project-to-project) 之間有效的知識移轉 (Knowledge Transfer)，藉此預估新專案將會發生的問題，以及先行降低問題可能發生的數量 (Thomke and Fujimoto, 2000)。因此，互補財廠商使用舊世代平台進行互補財的開發經驗亦可以集結為所謂的專案事後剖析 (Post Mortem) (Cusumano and Selby, 1995; Kumar and Terpstra, 2004; Thomke and Fujimoto, 2000)，這亦有助於互補財廠商先行參考剖析，減少在使用新世代平台的時候所產生的問題

數目。基於上述，我們提出本研究第二個假說。

假說二：

互補財廠商對於舊世代平台的深化性學習的程度越高，其越容易貢獻於該互補財廠商在新世代產品開發的表現。

深化性學習與探索性學習之間存在著互補性關係 (Cao et al., 2009; Gupta et al., 2006)，並且深化性學習與探索性學習對於創新活動是有著相乘效果 (Interaction Effects) (Cao et al., 2009; Gibson and Birkinshaw, 2004; Tushman and O'Reilly, 1996; Wei et al., 2014)。Katila and Ahuja (2002) 發現新產品開發的知識搜尋廣度與深度之間具有著正向交互作用 (Positive Interaction Effect)，Auh and Menguc (2005) 與 He and Wong (2004) 亦是採用相乘交互作用進行探索性學習與深化性學習的統計模型之驗證。互補財廠商進行舊世代平台的深化性學習可以降低互補財的開發成本以及提升性能，對於新世代平台所採用的探索性學習則可以創造出新的產品市場。因此，互補財廠商在平台的世代轉移之中採用互補性邏輯就可以有效連結深化性學習與探索性。基於此觀點，本研究提出第三個假說。

假說三：

互補財廠商對舊平台深化性學習的程度越高，越可提升在新平台進行的探索性學習對於新產品開發表現的貢獻。

## 參、分析方法

### 一、資料收集

本研究的分析對象是諾基亞 (Nokia)、索尼愛立信 (SonyEricsson)、摩托羅拉 (Motorola)、三星 (Samsung)、樂金 (LG)、華為 (Huawei)、小米 (Xiaomi)、中興 (ZTE)、宏達電 (HTC)、華碩 (ASUS)、宏碁 (Acer) 共 11 家智慧型手機廠商。2019 年這 11 家廠商擁有全球 76.7% 的安卓智慧型手機市場份額<sup>1</sup>。本研究將研究範圍限定在 Google 推出 Nexus 或 Pixel 手機中所搭載的 Éclair Android 2.1、Gingerbread Android 2.3、Ice Cream Sandwich Android 4.0、Jelly Bean Android 4.2、KitKat Android 4.4、Lollipop Android 5.0、Marshmallow Android 6.0、Nougat Android 7.1、Oreo Android 8.0、Pie Android 9.0、Android 10.0，共 11 個版本的安卓作業系統，期間橫跨從 2010 年 1 月 6 日至 2019 年 10 月 15 日。

1 <https://gs.statcounter.com/vendor-market-share/mobile/worldwide/2019>

Nagle (2018) 強調企業在開放源軟體社群中的貢獻將可以獲取其它貢獻者的回饋，藉此提升企業內部軟體開發的生產性與增加軟體的附加價值，而這也是一種邊貢獻邊學習 (Learning by Contributing) 的方式。同樣地，Sowe, Stamelos, and Angelis (2008) 也將貢獻視為是企業學習如何提升軟體除錯的效率、學習開放源軟體的最新技術發展資訊，以及學習如何推廣自身產品或軟體的新功能。基於此，我們於 2020 年 7 月 3 日在存放安卓開放源軟體程式碼更改的歷史紀錄網站 (<https://android.googlesource.com>)，使用 Python 的 Beautiful Soup 程式擷取企業在安卓開放源軟體社群中對於軟體新增、修改、除錯等提交 (Commit) 的資料，並將提交資料作為觀察企業如何進行探索性與深化性學習的代理變數。具體而言，我們以電子郵件信箱分析該提交紀錄是屬於哪些特定的智慧型手機廠商。例如：xxxxx@nokia.com 就視作為是 Nokia 的提交紀錄、xxxxx@samsung 即為 Samsung 的提交紀錄。

接著，本研究從 PhoneDB 的網站<sup>2</sup>下載與分析前述各智慧型手機廠商在研究期間內推出的安卓智慧型手機的概貌。本研究共分析了 11 家智慧型手機廠商從 2010 年 1 月 6 日 Éclair Android 2.1 的公開釋出之後，截至 2019 年 10 月 15 日 Android 10.0 的公開釋出之間，跨越了 10 次安卓版本的轉換過程，總共共 96 個觀察樣本，以釐清互補財廠商在平台的世代轉換中所進行之組織學習的雙面性效果。

## 二、變數定義

在 Information Communication Technologies (ICT) 產業中，快速上市已經成為企業普遍認知的新產品開發表現。企業能夠擁有快速上市的新產品開發能力，代表著該企業可以用更有效率的方式節省產品開發的成本 (Clark and Fujimoto, 1991; Stalk and Hout, 1990)。同樣地，本研究認為智慧型手機企業的產品快速上市亦可以帶給企業在經營成效上的貢獻。這是因為在安卓智慧型手機開始普及於消費者市場時，消費者會被每一代安卓作業系統的新功能所吸引而產生更高的購買意願 (Raju and Kumar, 2015)。基於此，我們從 PhoneDB 的網站下載與分析計算在每一款新版本安卓作業系統釋出之後，智慧型手機廠商最早推出該新版本安卓智慧型手機的日期與 Google 的 Nexus 或者 Pixel 之間的時間差距，並且將之反向處理作為快速上市 (Time to Market; *TTM*)，也就是本研究的主要的依變數。我們也將最大值與最小值進行對調。例如，最大值 11 變為 1，10 變為 2，因此數值越高代表快速上市表現越好。

---

2 <http://phonedb.net>。PhoneDB 收集了從 1989 年至今全世界眾多的行動裝置、手機、平板電腦、智慧型手錶的詳細資料，包含裝置的企業名稱、操作系統、軟硬體規格、釋出市場時間等。過去也曾有學者使用 PhoneDB 的網路資料研究手機企業之間的交易關係 (Alcacer and Oxley, 2014)，顯示這個網路資料是具有可信度。



新產品的釋出數量亦可以視作為是企業的競爭力表現 (Eisenhardt and Tabrizi, 1995; Puranam et al., 2006)，因此本研究亦從 PhoneDB 資料庫中收集作為分析對象之 11 家廠商在每一個安卓版本中所釋出之智慧型手機的款式數目，作為本研究的第二個依變數：產品款式 (Model)。

接著，從舊世代安卓作業系統進展到新世代安卓作業系統的時間約為一年。當舊世代安卓作業系統公開版本之後，Google 會因為解決了軟體問題或者提升產品性能等而推出「舊世代安卓作業系統維護版本」，而在新世代安卓作業系統公開日之前的六個月推出「新世代安卓作業系統開發者版本」(Brown and Westenberg, 2019)。智慧型手機廠商在這一年中開發新款智慧型手機時，需要對「舊世代安卓作業系統維護版本」與「新世代安卓作業系統開發者版本」分別進行深化性學習與探索性學習 (圖 1 中的 A 與 B)，才能有效地從舊世代安卓作業系統跨入到整合新世代安卓作業系統至智慧型手機的開發。基於此，我們從企業的提交資料中區分出智慧型手機廠商在安卓的世代轉換中，所進行的深化性學習與探索性學習的期間 (圖一)。

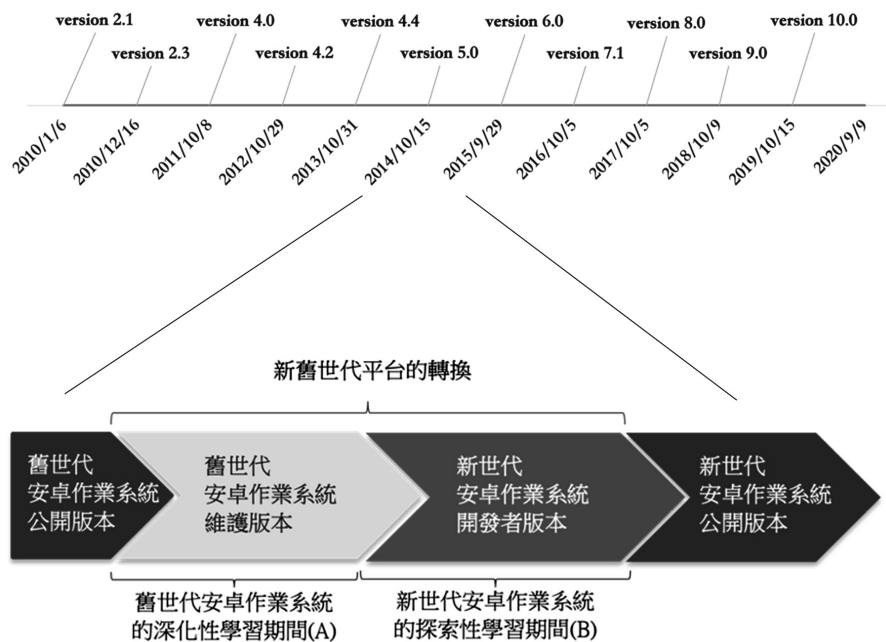


圖 1 新舊世代平台轉換中的深化性與探索性學習的期間

企業在整合開放源軟體至新產品開發的時候，除了需要深化學習每個檔案夾的功能性之外，更需要對檔案夾之間的相互關係進行學習，才能有效地提升新產品開發的表現 (Zhu, Zhou, and Mockus, 2014)。因此，探索性學習就會代表著企業在不同種類的檔

案夾中所進行軟體程式的貢獻，而深化性學習則是指企業在特定的檔案夾中所持續貢獻軟體程式的活動。

圖 2 是安卓作業系統檔案夾結構，其中灰色部分是指作為本研究分析對象之 11 家智慧型手機廠商，其主要貢獻軟體程式之處的檔案夾。透過於 2020 年 7 月至 2020 年 12 月和台灣谷歌負責管理開放源碼軟體的高階工程師與產品經理共 10 小時左右的訪談中得知，安卓作業系統檔案夾結構中與新產品開發較為關聯的是 Toolchain、Kernel 與 Platform 三個根目錄檔案夾，而這個根目錄之下又涵蓋著許多子目錄檔案夾。此外，這 11 家智慧型手機廠商在 Toolchain、Kernel 與 Platform 三個根目錄檔案夾的總累積貢獻次數（總累積提交次數）分別為 2,444、18,398、118,717，由於 Toolchain 與 Kernel 根目錄檔案夾中的貢獻次數遠低於 Platform 根目錄檔案夾，本研究只採用 Platform 根目錄檔案夾加以分析。

本研究使用辛普森指標 (Simpson Index-base Diversity) (Page, 2010) 衡量企業在 Platform 根目錄檔案夾以下的子檔案夾進行貢獻的分佈程度，藉此代表企業的探索性學習之程度。具體而言，在每一次安卓的世代轉換屬於新世代安卓作業系統開發者版本的期間內，我們將 Platform 根目錄檔案夾以下所有廠商貢獻過的子檔案夾（圖 2 中灰色的部分）視作企業應該會進行探索性學習的範圍 ( $M$ )， $P_i$  則是企業在特定  $i$  子檔案夾中的貢獻量比例。例如，企業 A 在兩個子檔案夾中分別貢獻 3 筆，企業 B 在三個子檔案夾中分別貢獻 6 筆，兩家企業的辛普森指標分別為 0.5 與 0.33。此時，我們可以基於下列公式 (1)，也就採用 1 減去辛普森指標的方式表示企業 B 的貢獻多樣性大過於企業 A。

$$\text{探索性學習}_{\text{Platform 檔案夾}} = 1 - \sum_i^M P_i^2. \quad (1)$$

不過，由於每家企業貢獻的子檔案夾的種類數量有所不同，就將會讓辛普森指標產生無法客觀地考慮貢獻的均度 (Evenness)，也就是辛普森指標存在著非線性的問題。為了解決這個問題，我們將辛普森指標轉換成有效數量等價 (Effective Number Equivalent) (Daly, Baetens, and De Baets, 2018; Jost, 2006) 加以重新表示。我們可以重新定義  $P_i = 1 / D$ ， $D$  用來表示在不同子檔案夾中皆有同等貢獻量的分佈情況，而  $P_i$  則是在這樣同等貢獻量中所佔的比例。此時，計算企業 A 的情況就會是， $0.5 = \sum_i^2 P_i^2 = \sum_i^2 (1 / D)^2 = 2 / D^2$ ， $D$  就等於 2，而計算企業 B 的情況就會是， $0.33 = \sum_i^3 P_i^2 = \sum_i^3 (1 / D)^2 = 3 / D^2$ ， $D$  就等於 3.02。此計算方式可幫助解決每家企業貢獻的子檔案夾的種類數量有所不同，以及前述辛普森指標特有的非線性問題。亦即，可以更為客觀地說明企業 B 的探索性學習的程度約是企業 A 的 1.5 倍。本研究以這樣的方式計算出智慧型手機廠商在圖 1 中「新世代安卓作業系統開發者版本」進行探索性學習的程度，也就是假說一中的自變數 (Explorative Learning;  $EXL_{new}$ )。

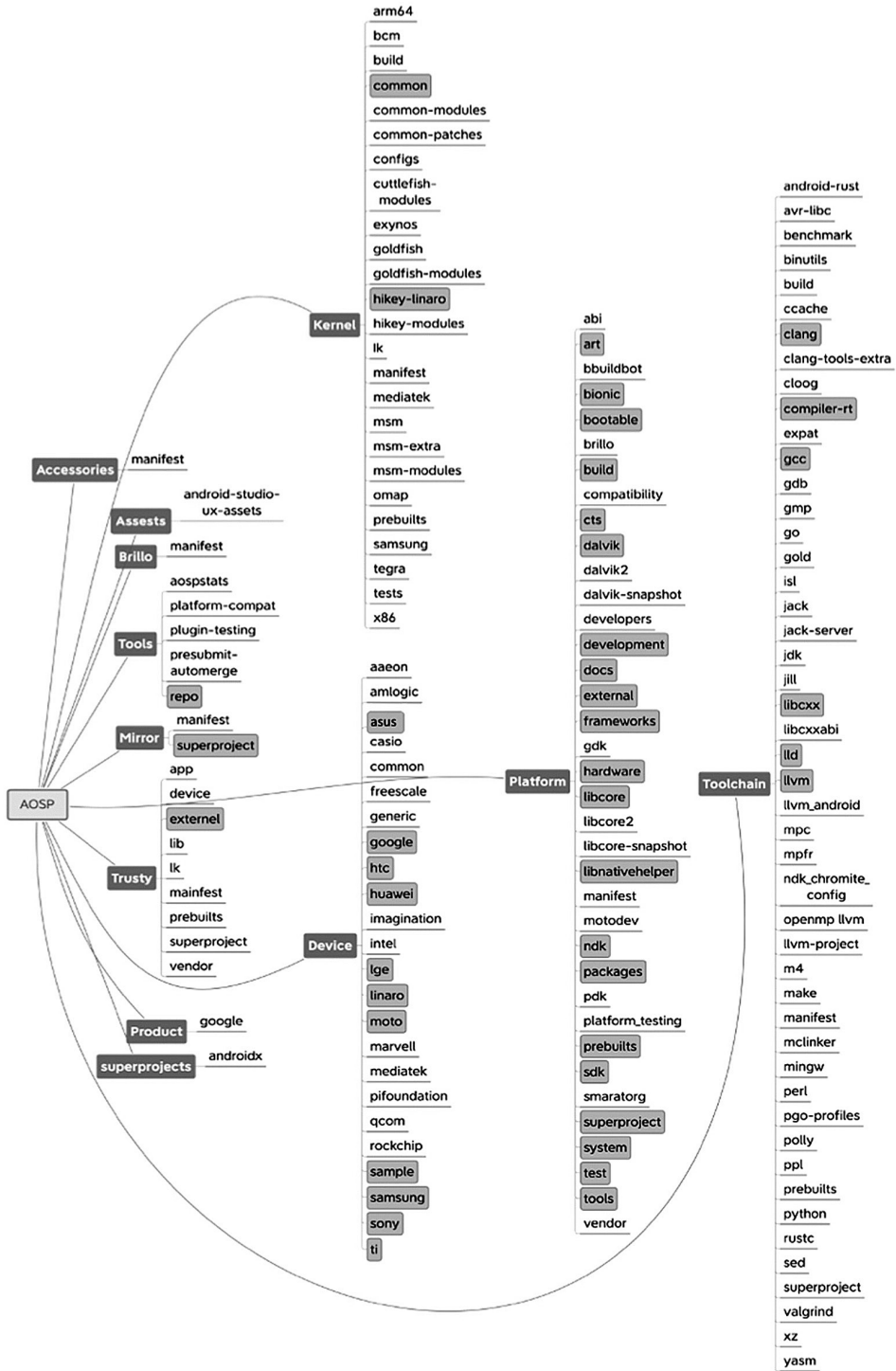


圖 2 安卓作業系統的檔案夾結構與智慧型手機廠商的主要貢獻之處 (灰色部分)

接下來，根據我們和台灣谷歌負責管理開放源碼軟體的高階工程師與產品經理的訪談，當安卓作業系統公開之後會進行追加新功能、優化與問題除錯的步驟，而這些步驟大部分都會反映在 Platform 根目錄檔案夾中的 Framework 檔案夾、Hardware 檔案夾以及 External 檔案夾。另外，因智慧型手機廠商在 Hardware 檔案夾中所做貢獻，其目的是為了要調配自家的硬體零組件之技術特性，因此這樣的貢獻會累積成為智慧型手機廠商固有的經驗與知識。本研究分析對象的 11 家智慧型手機廠商在 Framework 檔案夾、Hardware 檔案夾以及 External 檔案夾的總累積貢獻次數（總累積提交次數）分別為 5,478、82,610、23,917，由於 Framework 與 External 檔案夾中的貢獻次數低於 Hardware 檔案夾，本研究只採用企業在 Hardware 檔案夾中的貢獻次數，並且採用對數計算，其結果就成為企業在圖 1 的「舊世代安卓作業系統維護版本」中的深化性學習 (*EIL*)。而後，我們將深化性學習 (*EIL*) 與新世代的探索性學習 (*EXL\_new*) 兩個自變數採取中心化，並且計算出交互作用項 (*EXL-EIL\_Interaction Effects; Interaction*)。

接續我們控制智慧型手機廠商本身的異質性對於兩個依變數—快速上市 (*Time to Market; TTM*) 與產品款式 (*Model*) — 可能會產生的影響。首先，智慧型手機廠商可以使用「舊世代安卓作業系統維護版本」來搭配最新開發的硬體設計，藉此進行新產品開發的可行性分析等實驗。這時候，因為最新開發的硬體設計可能會反映出軟體設計不充分之處，因此智慧型手機廠商在「舊世代安卓作業系統維護版本」中若是持續進行貢獻的話亦可以達到探索性學習的效果。依照上述的「新世代安卓作業系統維護版本」中的探索性學習之計算方式 (*EXL\_new*)，我們亦衡量出「舊世代安卓作業系統維護版本」中的探索性學習，並且作為本研究的控制變數 (*Explorative Learning; EXL\_old*)。

另外，Cecere, Corrocher, and Battaglia (2015) 提出產品硬體零組件的組成會影響企業的競爭優勢。在本研究中我們將智慧型手機的 CPU Clock (*CPU\_Clock*)、CPU 廠商 (*CPU\_Firm*)、螢幕解析度 (*Display Resolution; DIS*) 分別作為本研究的控制變數。本研究亦加入智慧型手機廠商的國籍 (*Nationality; NAL*)、營業額收入 (*Revenue; RVE*) 以及智慧型手機廠商是否在當期有與 Google 合作開發 Nexus 或 Pixel 參考手機 (*Cooperation; COP*) 作為虛擬變數。圖 3 是本研究的舊世代的深化性學習 (*EIL*)、新世代的探索性學習 (*EXL\_new*) 與舊世代的探索性學習 (*EXL\_old*) 的概況；表 1 則是本研究的變數定義與編成方式。

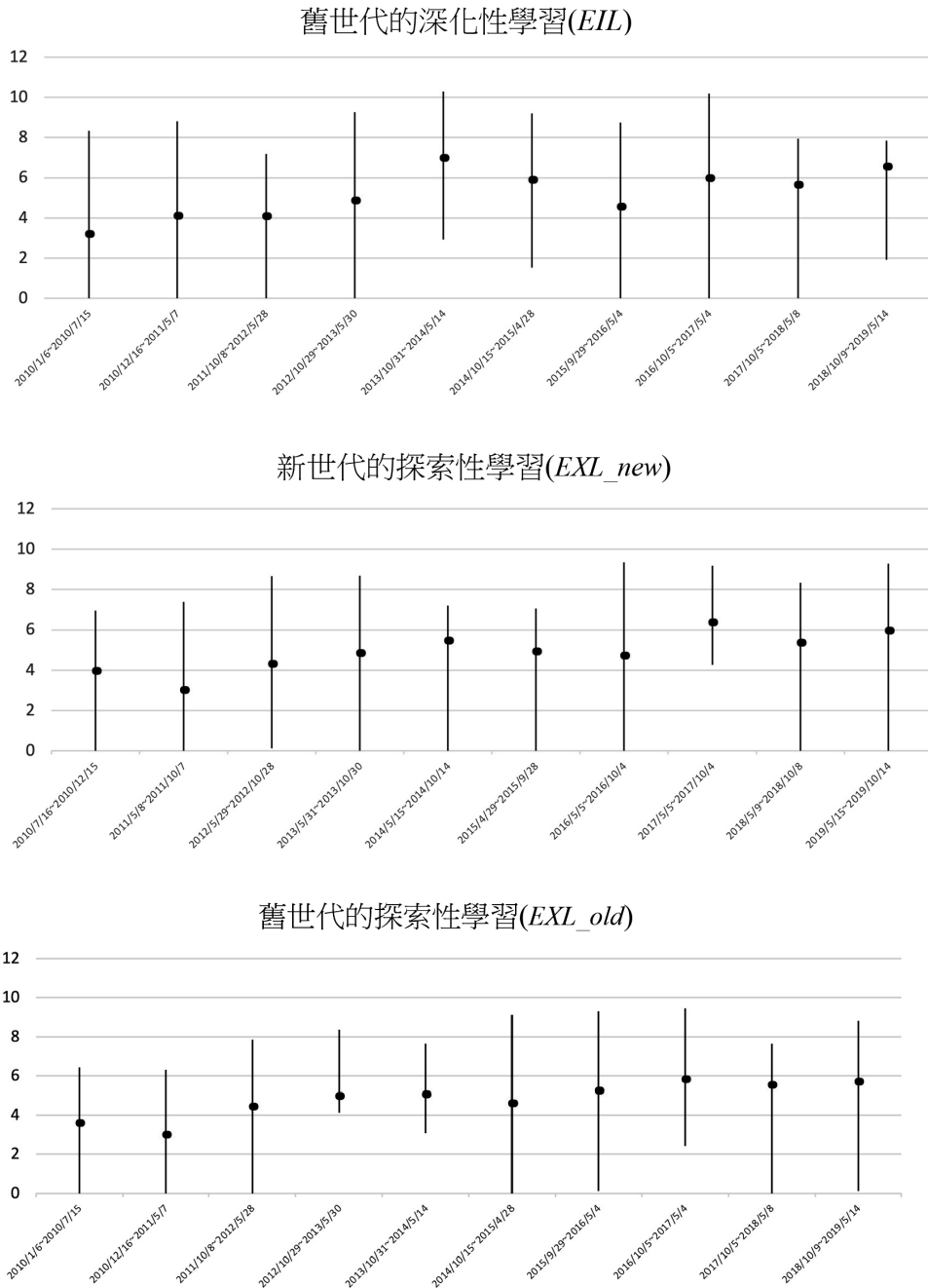


圖 3 智慧型手機廠商的在世代轉移中的學習方式與程度<sup>3</sup>

3 縱線表示在該版本之中的最大值與最小值，其中圓圈小點為平均值。



表 1 變數定義與編成方式

變數名稱		編成方式
依變數	快速上市 ( <i>TTM</i> )	在世代轉移的過程中，智慧型手機廠商最早推出該新版本安卓智慧型手機的日期與 Google 的 Nexus 或者 Pixel 之間的時間差距（以月份計算），並且將之以反向處理。
	產品款式 ( <i>Model</i> )	在世代轉移的過程中，智慧型手機廠商推出在該新版本安卓智慧型手機的所有款式數量。
自變數	舊世代的深化性學習 ( <i>EIL</i> )	在世代轉移的過程中舊世代平台的期間內（圖 1 的 A），智慧型手機廠商在安卓軟體程式中“Hardware”檔案夾中累積之貢獻量，並且取對數處理。
	新世代的探索性學習 ( <i>EXL_new</i> )	在世代轉移的過程中新世代平台的期間內（圖 1 的 B），以智慧型手機廠商在安卓軟體程式中“Platform”檔案夾中對於次檔案夾 (Sub-folders) 貢獻分佈的情況加以計算。
	交互作用效果 ( <i>Interaction</i> )	上述兩個變數的交互作用項，用以表示智慧型手機廠商在世代轉移的過程中，舊世代的深化性學習 ( <i>EIL</i> ) 與新世代的探索性學習 ( <i>EXL_new</i> ) 的互補性邏輯。
控制變數	舊世代的探索性學習 ( <i>EXL_old</i> )	在世代轉移的過程中舊世代平台的期間內（圖 1 的 A），智慧型手機廠商在安卓軟體程式中“Platform”檔案夾中對於次檔案夾 (Sub-folders) 貢獻分佈的情況加以計算。
	<i>CPU_Clock</i>	在快速上市的模型檢驗裏（款式的模型檢驗裏），我們採取在世代轉移的過程中，智慧型手機廠商最早推出該新版本安卓智慧型手機的 <i>CPU Clock</i> 赫茲數（所有手機之平均 <i>CPU Clock</i> 赫茲數），並且以對數處理。
	螢幕解析度 ( <i>DIS</i> )	在快速上市的模型檢驗裏（款式的模型檢驗裏），我們採取在世代轉移的過程中，智慧型手機廠商最早推出該新版本安卓智慧型手機的顯示分辨率（所有手機之平均顯示分辨率），並以對數處理。
	廠商收入 ( <i>REV</i> )	在款式的模型檢驗裡，我們計算在世代轉移的過程中智慧型手機廠商於該版本推出前一年之收入，並將各公司之收入轉換為美元以對數處理。
虛擬變數	國籍 ( <i>NAL</i> )	將各智慧型手機廠商之國籍轉換為虛擬變數。
	<i>CPU</i> 廠商	在快速上市以及款式的模型檢驗裏，我們採取在世代轉移的過程中，智慧型手機廠商最早以及所有使用之 <i>CPU</i> 廠商，將之轉為虛擬變數。
	與 Google 合作關係 ( <i>COP</i> )	在世代轉移的過程中，智慧型手機廠商是否與 Google 合作開發 Nexus 或者 Pixel。

表 2 變數的敘述性統計

變數種類	變數名稱	Mean	SD	Min	Max
依變數	快速上市 ( <i>TTM</i> )	6.080	3.134	0	12
	產品款式 ( <i>Model</i> )	11.830	8.979	1	41.000
自變數	舊世代的深化性學習 ( <i>EIL</i> ) <sup>1</sup>	4.246	2.838	0.000	9.300
	新世代的探索性學習 ( <i>EXL_new</i> )	4.888	2.748	0.000	9.355
控制變數	舊世代的探索性學習 ( <i>EXL_old</i> )	4.800	2.485	0.000	9.448
	<i>CPU_Clock</i> <sup>1,2</sup>	3.228	0.155	2.779	3.490
	螢幕解析度 ( <i>DIS</i> ) <sup>1,2</sup>	6.048	0.383	4.885	6.641
	平均 <i>CPU_Clock</i> <sup>1,3</sup>	3.240	0.113	2.964	3.472
	平均螢幕解析度 ( <i>DIS</i> ) <sup>1,3</sup>	6.097	0.326	5.061	6.498
	廠商收入 ( <i>REV</i> ) <sup>1,3</sup>	4.393	0.580	2.899	7.282
	虛擬變數	台灣企業 ( <i>NAL_TW</i> )	0.250	0.435	0
美國企業 ( <i>NAL_US</i> )		0.110	0.320	0	1
韓國企業 ( <i>NAL_KR</i> )		0.200	0.401	0	1
芬蘭企業 ( <i>NAL_FI</i> )		0.040	0.201	0	1
日本企業 ( <i>NAL_JP</i> )		0.100	0.307	0	1
中國企業 ( <i>NAL_CN</i> )		0.290	0.457	0	1
<i>CPU_MediaTek</i> <sup>2</sup>		0.140	0.344	0	1
<i>CPU_Qualcomm</i> <sup>2</sup>		0.660	0.477	0	1
<i>CPU_Samsung</i> <sup>2</sup>		0.060	0.243	0	1
<i>CPU_TI</i> <sup>2</sup>		0.040	0.201	0	1
<i>Model_CPU_MediaTek</i> <sup>3</sup>		0.570	0.497	0	1
<i>Model_CPU_Qualcomm</i> <sup>3</sup>		0.950	0.223	0	1
<i>Model_CPU_Samsung</i> <sup>3</sup>		0.120	0.332	0	1
<i>Model_CPU_TI</i> <sup>3</sup>		0.070	0.261	0	1
<i>Model_CPU_Marvell</i> <sup>3</sup>		0.050	0.223	0	1
<i>Model_CPU_HiSilicon</i> <sup>3</sup>		0.080	0.278	0	1
與 Google 合作關係 ( <i>COP</i> )		0.080	0.278	0	1

註：<sup>1</sup> 取對數之變數。<sup>2</sup> 僅用於快速上市多元分析之變數。<sup>3</sup> 僅用於產品款式的多元回歸分析之變數。

表 3 變數之間的皮爾森相關

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	
(1) 快速上市 (TTM) <sup>1</sup>	1																		
(2) 舊世代的深化性學習 (EIL) <sup>1</sup>	.343**	1																	
(3) 新世代的探索性學習 (EXL_new)	.325**	.309**	1																
(4) 舊世代的探索性學習 (EXL_old)	.388**	.334**	.609**	1															
(5) 台灣企業 (NAL_TW)	-.224*	-.342**	-.265**	-.354**	1														
(6) 美國企業 (NAL_US)	.011	.104	.089	.147	-.208*	1													
(7) 韓國企業 (NAL_KR)	.263**	.258*	.236*	.186	-.287**	-.179	1												
(8) 芬蘭企業 (NAL_FI)	-.022	-.111	.094	-.021	-.12	-.075	-.104	1											
(9) 日本企業 (NAL_JP)	.046	.232*	.301**	.331**	-.197	-.123	-.169	-.071	1										
(10) 中國企業 (NAL_CN)	-.047	-.081	-.260**	-.142	-.370**	-.231*	-.319**	-.134	-.219*	1									
(11) CPU_MediaTek	-.069	-.003	.04	-.064	-.018	-.142	.033	-.083	.064	.081	1								
(12) CPU_Qualcomm	-.065	-.154	-.02	.001	.114	.054	-.136	.151	.103	-.163	-.547**	1							
(13) CPU_Samsung	.159	.098	.068	-.021	-.149	.042	.412**	-.054	-.088	-.102	-.102	-.357**	1						
(14) CPU_Ti	.095	.144	-.079	.008	0	.089	.027	-.043	-.071	-.083	-.288**	-.288**	-.054	1					
(15) CPU_Marvell	.104	.194	-.009	.144	-.149	-.093	-.128	-.054	-.088	.402**	-.102	-.357**	-.067	-.054	1				
(16) CPU_Clock <sup>1</sup>	.317*	-.006	.16	.198	-.046	-.155	.069	.004	.007	.086	-.19	.137	.08	.228*	.181	1			
(17) 與 Google 合作關係 (COP)	.367**	.201*	.154	.168	.087	.128	.229*	-.063	-.103	-.111	-.009	-.02	.078	-.126	-.078	-.067	1		
(18) 螢幕解析度 (DIS) <sup>1</sup>	.322*	-.171	-.284**	-.242*	.006	.115	-.066	.217*	-.558**	.251*	-.057	.025	-.039	.057	.093	-.155	-.008	1	
Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(20)
(1) 產品款式 (Model)	1																		
(2) 舊世代的深化性學習 (EIL) <sup>1</sup>	.179	1																	
(3) 新世代的探索性學習 (EXL_new)	.244*	.309**	1																
(4) 舊世代的探索性學習 (EXL_old)	-.323**	-.342**	-.265**	-.354**	1														
(5) 台灣企業 (NAL_TW)	.003	.104	.089	.147	-.208*	1													
(6) 美國企業 (NAL_US)	.334**	.258*	.236*	.186	-.287**	-.179	1												
(7) 韓國企業 (NAL_KR)	-.113	-.111	.094	-.021	-.12	-.075	-.104	1											
(8) 芬蘭企業 (NAL_FI)	-.059	.232*	.301**	.331**	-.197	-.123	-.169	-.071	1										
(9) 日本企業 (NAL_JP)	.102	-.081	-.260**	-.142	-.370**	-.231*	-.319**	-.134	-.219*	1									
(10) 中國企業 (NAL_CN)	.281**	-.039	.136	.163	-.036	-.152	.006	.075	.019	.091	1								
(11) Model_CPU_MediaTek	-.104	-.14	.081	.018	.135	.084	-.001	.049	.08	-.262**	-.013	1							
(12) Model_CPU_Qualcomm	.413**	.249*	.072	-.014	-.218*	.062	.524**	-.079	-.129	-.173	-.056	-.053	1						
(13) Model_CPU_Samsung	.230*	.116	-.039	.032	-.069	.276**	.062	-.058	-.096	-.092	-.163	.066	.136	1					
(14) Model_CPU_Ti	.251*	.167	-.117	-.275**	-.027	.063	.119	-.049	-.08	-.047	-.082	.055	.337**	.115	1				
(15) Model_CPU_Marvell	.271**	.222*	-.067	-.182	-.074	-.108	-.15	-.063	-.103	.470**	.108	-.438**	-.114	.06	-.071	1			
(16) Model_CPU_HiSilicon	.355	.013	.228	.265**	-.064	.131	.003	.082	.118	.034	.009	-.155	.026	-.444**	-.312**	.061	1		
(17) CPU_Clock <sup>1</sup>	.065	.201*	.154	.168	-.087	.128	.229*	-.063	-.103	-.111	-.044	.071	.114	.205*	.099	.045	-.057	1	
(18) 與 Google 合作關係 (COP)	.31*	-.208*	-.276**	-.246**	.022	.068	-.015	.235*	-.754**	.348**	.122	-.055	-.009	.131	.124	.161	-.240*	.098	1
(19) 螢幕解析度 (DIS) <sup>1</sup>	.371**	.248*	.259*	.272**	-.303**	-.217*	.494**	-.035	.259*	-.151	.113	-.156	.412**	.074	.181	.17	-.089	-.234*	1

註：\* p < .05, \*\* p < .01, \*\*\* p < .001。1 取對數之變數。

## 肆、分析結果

表 2 是本研究變數的敘述性統計，表 3 則是變數間的皮爾森相關係數 (Pearson Correlation Coefficient)。從表 3 我們可以知道：舊世代的探索性學習 (*EXL\_old*) 對於快速上市 (*TTM*) 的相關係數為 0.609，*p* value 小於 0.01 有顯著性；舊世代的探索性學習 (*EXL\_old*) 對於產品款式 (*Model*) 的相關係數亦是 0.609，*p* value 小於 0.01 有顯著性，而其它變數之間的相關程度都是低於 0.6，並且本研究回歸模型中的變異膨脹因子 (Variance Inflation Factor; *VIF*) 皆小於 10，因此分析模型中並不存在著共線性的問題。接下來，我們對快速上市 (*TTM*) 進行 Kolmogorov-Smirnov 常態檢定，其顯著性為 0.054 大於 0.05，表示該快速上市 (*TTM*) 呈現常態分佈，因此我們採用普通最小平方法 (Ordinary Least Squares Method; OLS Method) 進行多元回歸分析 (Multiple Regression Analysis)。

表 4 的 Model 1 與 Model 2 是針對控制變數與虛擬變數的分析模型，其中智慧型手機廠商的國籍是以中國為參考組，我們發現智慧型手機廠商的國籍並不會影響到快速上市 (*TTM*)；CPU 晶片廠商則是以 Marvel 作為參考組，這些 CPU 晶片廠商對快速上市 (*TTM*) 的影響也不顯著。Model 1 至 Model 5 之中的螢幕解析度 (*DIS*) 對於快速上市 (*TTM*) 的影響不顯著，*CPU\_Clock* 則對快速上市 (*TTM*) 有著正向的顯著關係。*CPU\_Clock* 代表著 CPU 處理資料運算等能力。通常 *CPU\_Clock* 越高代表著 CPU 可以協助或者吸收週邊零組件的工作 (Task)，因此智慧型手機所需要的開發時間應該也就越短。值得一提的是，智慧型手機廠商與 Google 合作關係 (*COP*) 在所有模型中皆有著正向顯著關係。這是因為智慧型手機廠商跟 Google 合作開發 Nexus 或 Pixel 的時候，將可以進一步獲取新世代作業系統的技術資訊，而可加快自家搭載新世代作業系統推出智慧型手機的速度。

其次，表四的 Model 2 至 Model 4 中的舊世代的探索性學習 (*EXL\_old*) 對於快速上市 (*TTM*) 有著正向的顯著性影響，但是在 Model 5 中則是對快速上市 (*TTM*) 沒有顯著性影響。Model 3 的新世代的探索性學習 (*EXL\_new*) 對於快速上市 (*TTM*) 的  $\beta$  係數為 0.127，*p* value 小於 0.05 有顯著性；Model 4 的舊世代的深化性學習 (*EIL*) 對於快速上市的 (*TTM*) 的  $\beta$  係數為 0.193，*p* value 小於 0.05 有顯著性。基於此，本研究的假說一與假說二得以支持。Model 5 則是對新世代的探索性學習 (*EXL\_new*) 與舊世代的深化性學習 (*EIL*) 之交互作用 (*Interaction*) 加以檢定，我們發現對於快速上市的 (*TTM*) 的  $\beta$  係數為 0.119，*p* value 小於 0.05 有顯著性。基於此，本研究的假說三得以支持。值得一提的是，新世代的探索性學習 (*EXL\_new*) 與舊世代的深化性學習 (*EIL*) 在 Model 5 中對於快速上市 (*TTM*) 依然有著正向顯著的效果，而舊世代的探索性學習 (*EXL\_old*) 對於快速上市 (*TTM*) 則是變成沒有正向顯著的效果。這樣

表 4 快速上市的多元回歸分析<sup>3</sup>

	Model 1	Model 2	Model 3	Model 4	Model 5
常數 (Constant)	-6.378(10.524)*	-5.04 (10.499)*	-5.345(10.525)*	-5.99 (10.358)*	-7.828(10.114)***
舊世代的深化性學習 (EL) <sup>1</sup>				0.193( 0.152)*	0.836( 0.28 )**
新世代的探索性學習 (EXL_new)			0.127( 0.151)*		0.484( 0.215)*
上述兩個變數的交互作用 (Interaction) <sup>2</sup>					0.119( 0.049)*
舊世代的探索性學習 (EXL_old)		0.22 ( 0.153)*	0.153( 0.173)*	0.232( 0.125)*	0.061( 0.169)
台灣企業 (NAL_TW)	-1.323( 0.878)	-1.183( 0.877)	-1.231( 0.881)	-0.957( 0.873)	-1.037( 0.853)
美國企業 (NAL_US)	-0.032( 1.132)	-0.474( 1.166)	-0.621( 1.181)	-0.694( 1.155)	-0.779( 1.136)
韓國企業 (NAL_KR)	-0.242( 1.038)	-0.68( 1.076)	-0.861( 1.099)	-1.087( 1.082)	-0.793( 1.084)
芬蘭企業 (NAL_FI)	-0.114( 1.65 )	-0.284( 1.644)	-0.61 ( 1.693)	-0.121( 1.622)	0.01 ( 1.634)
日本企業 (NAL_JP)	1.127( 1.42 )	0.383( 1.504)	0.198( 1.523)	-0.184( 1.513)	0.367( 1.511)
CPU_MediaTek	-1.306( 1.322)	-0.988( 1.332)	-1.053( 1.337)	-0.7 ( 1.322)	-0.836( 1.291)
CPU_Qualcomm	-0.399( 1.086)	-0.092( 1.1 )	-0.059( 1.103)	0.236( 1.098)	0.157( 1.07 )
CPU_Samsung	1.422( 1.739)	2.072( 1.786)	2.081( 1.79 )	2.384( 1.768)	1.438( 1.765)
CPU_TI	-1.477( 1.864)	-1.295( 1.857)	-1.164( 1.867)	-1.427( 1.831)	-1.912( 1.807)
CPU_Clock <sup>1</sup>	3.304( 2.175)*	2.518( 2.229)*	2.377( 2.24 )*	2.53 ( 2.197)*	2.567( 2.146)*
與 Google 合作關係 (COP)	3.072( 1.164)**	2.801( 1.172)**	2.732( 1.177)**	2.512( 1.166)**	2.344( 1.138)*
螢幕解析度 (DIS) <sup>1</sup>	0.305( 1.048)	0.315( 1.042)	0.406( 1.049)	0.301( 1.026)	0.34 ( 1.005)
Adjusted R <sup>2</sup>	0.117	0.180	0.178	0.200	0.222
F value	2.050	2.601	2.470	2.697	2.699
df	12	13	14	14	16

註：\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$ ; N = 96。<sup>1</sup>取對數之變數。<sup>2</sup>取中心化之變數。<sup>3</sup>本模型對快速上市 (TTM) 採用反向處理。括號中的數值為標準誤差。



的結果亦更加顯示，智慧型手機廠商在跨越世代的時候，需要注重新世代的探索性學習 (*EXL\_new*) 與舊世代的深化性學習 (*EIL*) 對於快速上市 (*TTM*) 所帶來的效果。

接著，我們將舊世代的深化性學習 (*EIL*) 的數值中高過於平均值一個標準差的分成高分組，也就是圖 4 中的虛線（舊世代的深化性學習 (*EIL*)\_高），而將舊世代的深化性學習 (*EIL*) 的數值中低過於平均值一個標準差的分成低分組，也就是圖 4 中的實線（舊世代的深化性學習 (*EIL*)\_低）。經由單純斜率 (Simple Slope) 檢定顯示，舊世代的深化性學習 (*EIL*)\_高對快速上市 (*TTM*) 有顯著解釋力 ( $\beta$  為 2.744， $p$  value = 0.003)，而舊世代的深化性學習 (*EIL*)\_低對快速上市 (*TTM*) 則無明顯解釋力 ( $\beta$  為 0.933， $p$  value = 0.22)。從圖 4 中我們也可以明確地觀察出來，在舊世代的深化性學習 (*EIL*)\_高的情況下，新世代的探索性學習 (*EXL\_new*) 對於快速上市 (*TTM*) 有著非常明顯的貢獻。

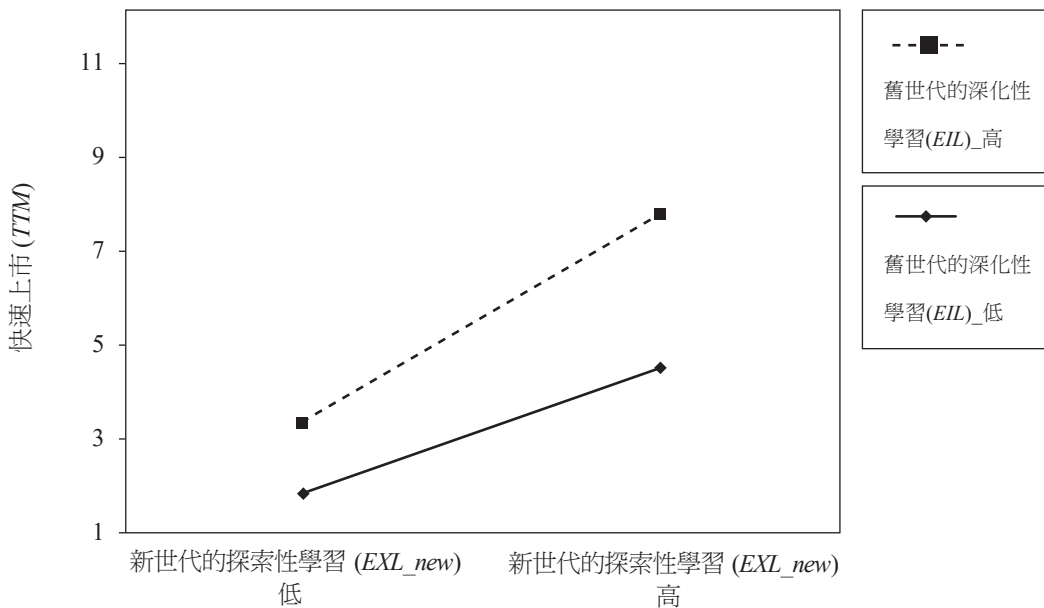


圖 4 交互作用與快速上市

為了進一步確認智慧型手機廠商的跨世代學習對於新產品開發的表現，我們從 PhoneDB 網站下載的資料庫中整理出 11 家智慧型手機廠商在每一個安卓作業系統版本中推出的手機款式數量。由於我們的三個自變數分別著重於 Platform 根目錄檔案夾中智慧型手機廠商的軟體貢獻與學習，而 Platform 的根目錄檔案夾中亦包含了 Hardware 檔案夾，這也意味著智慧型手機廠商對於該檔案夾的軟體貢獻，將可以學習到在搭配不同硬體零組件的時候將需要解決的可行性 (Feasibility)、相容性 (Com-

patibility) 以及該硬體零組件對於智慧型手機整體表現的貢獻程度等知識，其結果有助於開發出更多產品款式 (*Model*)。

產品款式 (*Model*) 是一個非負整數 (Nonnegative Integer Values)，且是卜瓦松 (Poisson) 或者負二項 (Negative Binomial Specification) 的分佈。在卜瓦松分佈中，平均值 (Mean) 與變異數 (Variance) 相等同一個  $\lambda$ ，也就是幾個自變數的函數， $E[Y] = \text{var}[Y] = \lambda$ 。然而，在我們的數據中，產品款式 (*Model*) 有著過度散布 (Overdispersion)，也就是變異數 64 大於平均值 11 的條件；我們採用負二項式回歸與固定效果 (Fixed Effects) 驗證本研究的三項假說。

首先，表 5 負二項回歸分析是以智慧型手機廠商的中國國籍作為參考組，我們發現智慧型手機廠商的國籍並不會影響到產品款式 (*Model*)；CPU 晶片廠商則是以 HiSilicon 作為參考組，我們發現除了 Qualcomm 之外，使用 MediaTek、TI、Samsung 與 Marvell 皆對產品款式 (*Model*) 有著正向的影響，這是因為智慧型手機廠商使用 Qualcomm 的 CPU 開發智慧型手機所需花費的工程時間較多，其它 CPU 廠商則是提供了較為接近於實體的參考設計，因此廠商得以較易開發出多款智慧型手機。接著，*CPU\_Clock* 對於產品款式 (*Model*) 並沒有直接性的影響，而螢幕解析度 (*DIS*) 則是對產品款式 (*Model*) 有著顯著性的影響，這意味著智慧型手機廠商推出多款產品的考量並非全然都是為了 CPU 的處理速度，更重要的反而是螢幕解析度。另外，企業收入 (Revenue) 則是對產品款式 (*Model*) 有著正向的影響，這說明著企業的營收表現越好，越有可能趁勝追擊推出更多產品款式 (*Model*)。不過，智慧型手機廠商與 Google 合作關係 (*COP*) 則是對產品款式 (*Model*) 沒有明顯的助益。這可能是因為智慧型手機廠商雖然可以藉由與 Google 合作開發 Nexus 或者 Pixel 獲取相關的技術資訊，但這樣的技術資訊具有時效性，也就是僅對快速上市 (*TTM*) 有顯著的效果，最終這些技術資訊會隨著 Nexus 或者 Pixel 的上市以及公開在安卓開放源軟體社群之後，其時效性逐漸下降而對智慧型手機廠商開發產品款式 (*Model*) 沒有幫助。

其次，表 5 Model 7 至 Model 10 中的舊世代的探索性學習 (*EXL\_old*) 對於產品款式 (*Model*) 有著正向的顯著性影響。Model 8 的新世代的探索性學習 (*EXL\_new*) 對於產品款式 (*Model*) 的  $\beta$  係數為 0.285， $p$  value 小於 0.05 有顯著性；Model 9 的舊世代的深化性學習 (*EIL*) 對於產品款式 (*Model*) 的  $\beta$  係數為 0.183， $p$  value 小於 0.05 有顯著性。基於此，本研究的假說一與假說二得以支持。Model 10 則是對新世代的探索性學習 (*EXL\_new*) 與舊世代的深化性學習 (*EIL*) 之交互作用 (*Interaction*) 加以檢定，我們發現對於產品款式 (*Model*) 的  $\beta$  係數為 0.625， $p$  value 小於 0.05 有顯著性。基於此，本研究的假說三得以支持。

表 5 產品款式的負二項回歸分析

	Model 6	Model 7	Model 8	Model 9	Model 10
常數 (Constant)	1.327(1.905)	4.12 (2.0246)*	4.148(2.0161)*	4.669(2.0723)*	4.815(2.0599)*
舊世代的深化性學習 (EIL) <sup>2</sup>				0.183(0.0129)*	0.481(0.063 )*
新世代的探索性學習 (EXL_new)		0.285(0.0158)*			0.27 (0.114 )**
上述兩個變數的交互作用 (Interaction) <sup>1</sup>					0.625(0.156 )*
舊世代的探索性學習 (EXL_old)		0.089(0.0186)***	0.077(0.0197)***	0.095(0.0192)***	0.073(0.0208)***
台灣企業 (NAL_TW)	-0.578(0.1025)	-0.51 (0.104 )	-0.508(0.1043)	-0.533(0.1052)	-0.521(0.106 )
美國企業 (NAL_US)	-0.195(0.1187)	-0.383(0.1249)	-0.414(0.1263)	-0.384(0.1251)	-0.387(0.1261)
韓國企業 (NAL_KR)	-0.001(0.1046)	-0.096(0.1049)	-0.135(0.1072)	-0.082(0.1057)	-0.081(0.1096)
芬蘭企業 (NAL_FI)	-0.579(0.2017)	-0.502(0.2023)	-0.554(0.2043)	-0.515(0.2026)	-0.517(0.206 )
日本企業 (NAL_JP)	-0.548(0.2083)	-0.672(0.2113)	-0.689(0.2102)	-0.638(0.214 )	-0.611(0.2149)
CPU_MediaTek	0.585(0.0694)***	0.529(0.0701)***	0.517(0.0706)***	0.523(0.0702)***	0.501(0.071 )***
CPU_Qualcomm	-0.085(0.1311)	-0.169(0.1336)	-0.2 (0.1347)	-0.222(0.1389)	-0.239(0.1407)
CPU_Samsung	0.222(0.1091)	0.389(0.1133)**	0.404(0.1135)***	0.408(0.1145)***	0.367(0.1184)**
CPU_TI	0.731(0.127 )***	0.631(0.1291)***	0.65 (0.1304)***	0.621(0.1295)***	0.587(0.1324)***
CPU_Marvell	0.601(0.1303)***	0.759(0.1346)***	0.75 (0.135 )***	0.796(0.1374)***	0.717(0.1413)***
CPU_Clock <sup>2</sup>	0.595(0.3585)	-0.212(0.4046)	-0.275(0.4051)	-0.318(0.4135)	-0.478(0.4175)
與 Google 合作關係 (COP)	-0.139(0.1112)	-0.222(0.112 )	-0.234(0.112 )	-0.193(0.114 )	-0.209(0.1137)
螢幕解析度 (DIS) <sup>2</sup>	0.306(0.1874)*	0.303(0.1895)*	-0.275(0.1898)*	0.311(0.1909)*	0.287(0.1908)**
企業收入 (Revenue)	0.193(0.0756)**	0.079(0.0809)*	0.072(0.0807)*	0.063(0.0813)*	0.064(0.081 )*
Log-likelihood	-299.343	-294.276	-293.502	-294.276	-292.42
$\chi^2$	321.646	351.334	353.847	328.847	355.847
df	14	15	16	16	18

註：\*  $p < .05$  , \*\*  $p < .01$  , \*\*\*  $p < .001$  ; N = 96 。 <sup>1</sup> 中心化之變數。 <sup>2</sup> 取對數之變數。括號中的數值為標準誤差。

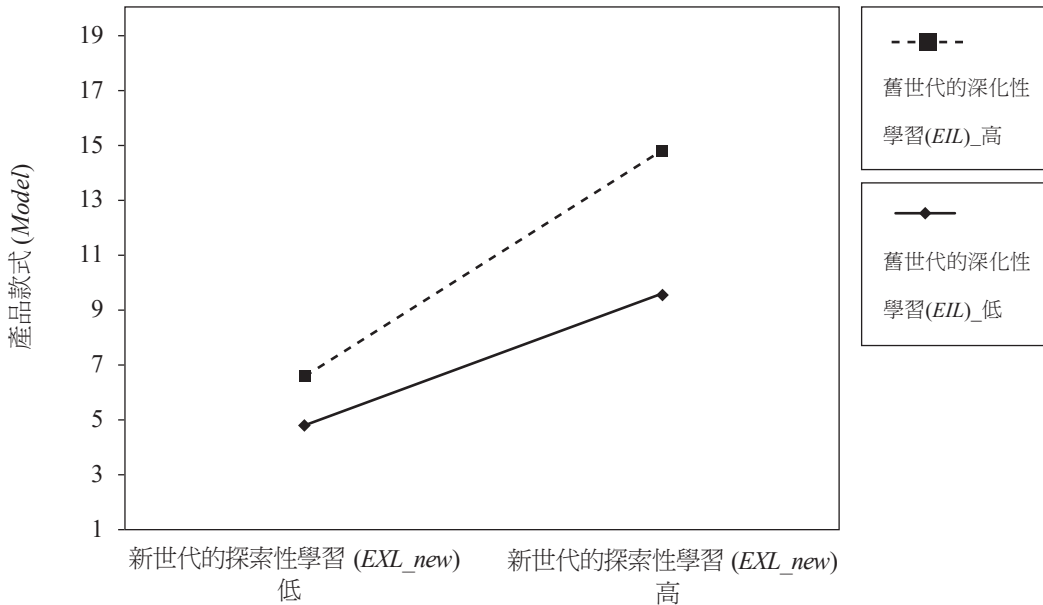


圖 5 交互作用與產品款式

最後，我們將舊世代的深化性學習 (*EIL*) 的數值中高過於平均值一個標準差的分成高分組，也就是圖 5 中的虛線（舊世代的深化性學習 (*EIL*)\_高），而將舊世代的深化性學習 (*EIL*) 的數值中低過於平均值一個標準差的分成低分組，也就是圖 5 中的實線（舊世代的深化性學習 (*EIL*)\_低）。經由單純斜率檢定顯示，舊世代的深化性學習 (*EIL*)\_高對產品款式 (*Model*) 有顯著解釋力 ( $\beta$  為 1.498,  $p$  value = 0.048)，而舊世代的深化性學習 (*EIL*)\_低對產品款式 (*Model*) 則無明顯解釋力 ( $\beta$  為 0.873,  $p$  value = 0.243)。從圖 5 中我們也可以明確地觀察出來，在舊世代的深化性學習 (*EIL*)\_高的情況下，新世代的探索性學習 (*EXL\_new*) 對於產品款式 (*Model*) 有著非常明顯的貢獻。

## 伍、討論

### 一、理論貢獻

本研究的理論貢獻主要有三點。首先，本研究釐清了智慧型手機廠商在安卓生態系中採用深化性學習與探索性學習的方式，並且證明了這兩種學習方式皆可提升新產品開發的表現。並進一步證明互補財廠商從事順序雙元性是具有著互補性邏輯，基於此，我們提出組織學習的雙元性可以導入於平台基礎生態系的研究中，並且藉此來補充說明互補財廠商在平台的世代轉換中應同時兼具兩種學習策略的能

力。另外，在安卓生態系的世代轉換過程中，智慧型手機廠商採用的組織雙元性並不屬於同時進行深化性學習與探索性學習的方式，而是屬於對舊世代平台技術採用深化性學習，對新世代平台技術則是採用探索性學習。

第二，我們的研究分析顯示，在一個連續性技術變化的平台基礎生態系中，互補財廠商對舊世代平台進行深化性學習將有助於對新世代平台進行探索性學習的效果。基於此，我們提出舊世代平台的知識將有助於學習新世代平台，並且我們認為平台廠商不應該只是單純地協助互補財廠商進行新世代平台的探索性學習，平台廠商亦可以促使互補財廠商共同來針對舊世代平台來進行深化性學習，藉此延伸與活用既有知識於平台的世代轉移。

過去研究認為組織內部不同單位同時從事新舊產品的開發與製造會產生互相侵蝕效應，因而造成組織內部負責新舊產品的單位之間的緊張關係 (O'Reilly and Tushman, 2004; Raisch et al., 2009; Tushman and O'Reilly, 1996)。同樣地，在平台基礎生態系中，平台廠商若是推出新世代平台的頻率越快的話，就越容易造成新舊互補財之間的侵蝕效應。平台廠商推出新世代平台雖然可以提升平台基礎生態系的進化性，但是卻有可能降低平台基礎生態系的穩定性。我們的研究分析顯示，在一個連續性技術變化的平台基礎生態系中，互補財廠商對舊世代平台進行深化性學習將有助於對新世代平台進行探索性學習的效果。基於此，我們提出平台廠商不應該只是單純地協助互補財廠商進行新世代平台的探索性學習，平台廠商亦可以促使互補財廠商共同針對舊世代平台進行深化性學習，藉此延伸與活用既有知識於平台的世代轉移。

因此，平台基礎生態系的進化機制中，互補財廠商如何能夠順利地對平台的世代轉換來進行深化性學習與探索性學習，以及降低平台的世代轉換所產生的緊張關係，就可以納入平台廠商在制定策略方面的考量 (Tilson, Lyytinen, and Sørensen, 2010; Tiwana et al., 2010; Wareham, Fox, and Cano Giner, 2014; Yoo, Henfridsson, and Lyytinen, 2010)。我們認為平台廠商不應該單純地只是制定平台相關標準、公開與更新平台的介面，更重要的是要如何協助互補財廠商在平台世代轉換的過程中來加以學習，亦或者透過提供參考設計與開發工具，以及對互補財廠商推廣與進行教育訓練，才能夠有效提升互補財廠商學習的雙元性，藉此提高平台基礎生態系整體的進化能力

第三，本研究補足了開放源軟體的研究中，企業貢獻與公開自家軟體程式可以獲取的優勢。儘管開放源軟體已經成為開放式創新、平台與生態系研究領域中爭相探討的議題，我們認為仍需要更多研究來針對企業策略與行為加以闡明，才得以真正知道企業使用開放源軟體的競爭優勢。基於此，本研究的分析結果提供了我們在進行開放源軟體研究的一個基礎，也就是今後的研究可以從貢獻即是學習的觀點來



進一步釐清企業在使用開放源軟體的策略與相關表現。具體而言，開放源軟體可以為企業降低軟體開發成本與增加自家軟體開發的生產性 (Nagle, 2018)，我們的分析結果則是更進一步顯示在開放源軟體社群中，企業會採取不同的深化性學習與探索性學習的學習策略，藉此來獲取開放源軟體的相關知識。

## 二、研究限制與今後研究課題

本研究只針對安卓作業系統中，屬於 Platform 檔案夾之下的次檔案夾以及 Hardware 檔案夾中企業的貢獻量加以分析，今後我們可以更加廣泛地對企業在安卓作業系統中其它的檔案夾中的貢獻進一步研究。例如，Kernel 檔案夾是屬於安卓作業系統中較為核心層的部分，企業貢獻於 Kernel 檔案夾的動機與目的，理當會有別於其它檔案夾中的貢獻動機與目的。此外，企業貢獻於不同檔案夾的差異性，亦是顯示出各家企業的學習策略，當安卓作業系統產生技術變化，理當也會產生不同的學習策略。因此，釐清企業在安卓作業系統的進化過程中，對於不同檔案夾的貢獻方式與變化，將可以進一步了解各家企業又是如何因應平台的世代轉換採取不同的學習策略，而這也可以加深我們理解組織學習的雙面性會如何受到技術變化而有所影響。

另外，今後研究可以基於 Hippel and Krogh (2003) 所提出的私有合作性創新模型 (Private-collective Innovation Model) 的觀點討論企業在安卓生態系中，組織學習的雙面性又會如何受到搭便車 (Free-rider) 的現象而有所影響。還有，企業在安卓開放源軟體社群中又是如何進行合作，其合作的方式又會如何提升企業的新產品開發表現。例如，Grewal, Lilien, and Mallapragada (2006) 就曾採用網絡分析 (Network Analysis) 的分析方式，建構出不同專案與專案管理者在相同檔案夾所形成的創新網絡，並且採用度中心性 (Degree Centrality)、中介中心性 (Betweenness Centrality)，與特徵向量中心性 (Eigenvector Centrality) 分別衡量專案與專案管理者的結構性嵌入、連結性嵌入與位置性嵌入，以及如何影響開放源軟體的技術性成功（以提交數目：Commits 衡量）與商業性成功（以下載次數加以衡量）。Grewal et al. (2006) 的分析說明了專案與專案管理者在開放源軟體中的合作性關係具有嵌入性特徵，也就是企業透過合作性獲取知識的方式是需要從社會網絡的嵌入性觀點加以說明。像這樣，今後研究亦可以基於嵌入性觀點與網絡分析之方式探討組織學習的雙面性與成效。

# **The Smartphone Manufacturers' New Product Development Performance in the Generation Transition of Android Ecosystem: Exploiting Old Technologies to Enhance the Effects of Exploring New Technologies**

---

Jing-Ming Shiu, Department of Business Administration, National Cheng Kung University

## **1. Purpose and Objective**

Platform firms continuously create new-generation platform technologies in a platform-based ecosystem, and complementors also constantly adopt platform technologies to develop new complements. However, past studies mainly focus on platform firms and mostly discuss how to encourage complementors' innovative activities from the perspective of these firms. Meanwhile, less studies concern the perspective of complementors and seldom consider how complementors can use platform technologies to develop new products during generational transitions. Therefore, in this study, I explore how complementors (smartphone manufactures) navigate the strategies of exploitative and exploratory learning in response to the generational transition of platform technologies by applying organizational learning theory to platform-based ecosystem research.

## **2. Defining Concepts and Proposing Hypotheses**

The platform in the platform-based ecosystem represents the core concept of the product and defines the architectural design. The platform is also the technical bottleneck during the generational transition of products. The generational transition of a platform means that the existing functions are significantly improved, or new functions are added. The transitions will also change the interaction among components in the ecosystem. Specifically, before the official release of the new-generation platform, complementors can conduct exploratory learning on its technical characteristics and adopt this new-generation platform to develop new complements more efficiently. Moreover, after the official release, usually platform firms will consistently release updated platform technologies that fix bugs and improve product performance. Complementors can thus continuously learn from the

old-generation platform technologies which help solve the existing problems and enhance product performance.

To sum up, this study considers that in the platform-based ecosystem, exploitative and exploratory learnings are complementary and synergistically affect innovative activities and new product developments. Accordingly, this study proposes the following hypotheses.

Hypothesis 1:

The higher the degree of the complementors' exploratory learning of the new-generation platform technology is, the better their new product development performance will be.

Hypothesis 2:

The higher the degree of the complementors' exploitative learning of the old-generation platform technology is, the better their new product development performance will be.

Hypothesis 3:

The higher the degree of the complementors' exploitative learning of the old-generation platform is, the greater the contribution of the exploratory learning on the new-generation platform to the performance of new product development will be.

### **3. Research Design**

The objects of analysis in this study are 11 smartphone manufacturers: Nokia, Sony Ericsson, Motorola, Samsung, LG, Huawei, Xiaomi, ZTE, HTC, ASUS, and Acer. In 2019, these 11 manufacturers accounted for 63.9% of the global Android smartphone market share. In this study, I focus on Éclair Android 2.1, Gingerbread Android 2.3, Ice Cream Sandwich Android 4.0, Jelly Bean Android 4.2, KitKat Android 4.4, Lollipop Android 5.0, Marshmallow Android 6.0, Nougat Android on Google's Nexus or Pixel phones 7.1, Oreo Android 8.0, Pie Android 9.0, and Android 10.0, a total of 11 versions of the Android OS, with the releasing time frame spanning from January 6, 2010, to October 15, 2019.

This study uses Python's Beautiful Soup program to download the firms' commits (also known as contribution) in the Android open-source software community on the website (<https://android.googlesource.com>). I use the commits data as a proxy variable to

observe how firms conduct exploitative and exploratory learning. Specifically, I identify which smartphone manufacturers the commits belonged to by email. For example, xxxxx@nokia.com is regarded as Nokia's commit record, and xxxxx@samsung is Samsung's commit record.

Second, the period from the old- to the new-generation Android OS is about a year. Upon releasing the old-generation Android OS to the public, Google will continue to release the old-generation Android OS as the maintenance version because it solves software problems or improves product performance. Moreover, six months before the official release date, the new-generation Android OS is released as the developer version. Based on above-mentioned release condition, I distinguish the period of exploitative and exploratory learning carried out by smartphone manufacturers in the generational transition of the Android OS (See Figure 1). In addition, I calculate firms' contribution in the hardware folder under the platform folder in the maintenance version, and in the sub-folder under the platform folder in the developer version as two independent variables for the two types of learning.

Third, this study calculates the elapsed time between the dates of a smartphone manufacturer first releases a new version of an Android smartphone, and the releasing dates of Google's Nexus or Pixel after each new version of the Android OS is released. This elapsed time is regarded as time-to-market, the primary dependent variable. In addition, this study counts the number of models launched by the 11 smartphone manufacturers in each Android version from the PhoneDB database as the second dependent variable. This study also controls for the firms' heterogeneity in their new product developments, such as hardware components, nationality, and relationship with Google.

## 4. Findings

I use the ordinary least squares method for multiple regression analysis. I find that the origin of the smartphone manufacturer, the CPU chip manufacturer, and display resolution all do not affect the time-to-market. Meanwhile, cooperation between smartphone manufacturers and Google has a significant positive relationship with time-to-market. Moreover, the exploratory learning of the new generation platform technologies and the exploitative learning of the old-generation ones significantly impact the time-to-market.

The interaction between exploratory learning and exploitative learning also positively impacts time-to-market.

Besides, model, the second dependent variable of this study, is a non-negative integer, and the variance is greater than the mean. Thus, I use negative binomial regression and fixed effects to test the three hypotheses. I find that the nationality or origin of smartphone manufacturers does not affect models, whereas CPU chip manufacturers positively impact the models. Furthermore, the screen resolution significantly impacts models, and corporate revenue positively impacts models. However, the cooperative partnership between smartphone manufacturers and Google has no significant impact on the models. Finally, exploratory learning of the new-generation platform technologies and the exploitative learning of the old-generation ones have significant impacts on models. The interaction between exploratory learning of the new generation and exploitative learning of the old generation also has a positive impact on the models.

## **5. Conclusions, Contribution and Future Research**

Our analysis results show that both the exploitative learning in the old generation and the explorative learning in the new generation of Android Operating System (OS) have effectively improved smartphone manufacturers' new product developments. In addition, the higher the degree of exploitative learning in the old generation of Android OS, the more it increases the positive impacts of explorative learning in the new generation of Android OS on new product development performance of smartphone manufacturers.

This study has three theoretical contributions. First, I supplement the discussions of past research on the governance mechanism of the platform-based ecosystem. This research proposes that platform firms should not just simply provide standards and disclose the platform's interface. What more importantly is to help complementors learn from the platform and successfully transit from the old platform to the new one. Second, my research analysis shows that in a platform-based ecosystem with continuous technological changes, the exploitative learning of complementors' old-generation platform technologies may contribute to the effect of exploratory learning on new generation platforms. Therefore, platform firms should not only encourage complementors to conduct exploratory learning, but should inspire the exploitative learning as well. Third,



the results of this study provide us with learning to clarify further the strategies and related performance of firms in the use of open-source software from the dynamic viewpoint of platform generational transition and contributing perspective.

Finally, future research can reveal firms' contributions in different folders in the evolution of the Android OS. Moreover, future research can also discuss how the ambidexterity of organizational learning in the Android ecosystem can surpass the free-rider problem. Finally, future research can use network analysis to measure firms' structural embeddedness in the Android open-source software community and examine whether it can significantly impact new product development performance.

## References

- Adner, R., and Kapoor, R. 2010. Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. *Strategic Management Journal*, 31 (3): 306-333.
- Alcacer, J., and Oxley, J. 2014. Learning by supplying. *Strategic Management Journal*, 35 (2): 204-223.
- Andriopoulos, C., and Lewis, M. W. 2009. Exploitation-exploration tensions and organizational ambidexterity: Managing paradoxes of innovation. *Organization Science*, 20 (4): 696-717.
- Ansari, S., and Garud, R. 2009. Inter-generational transitions in sociotechnical systems: The case of mobile communications. *Research Policy*, 38 (2): 382-392.
- Atuahene-Gima, K., and Murray, J. Y. 2007. Exploratory and exploitative learning in new product development: A social capital perspective on new technology ventures in China. *Journal of International Marketing*, 15 (2): 1-29.
- Auh, S., and Menguc, B. 2005. Balancing exploration and exploitation: The moderating role of competitive intensity. *Journal of Business Research*, 58 (12): 1652-1661.
- Boudreau, K. 2010. Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, 56 (10): 1849-1872.
- Boumgarden, P., Nickerson, J., and Zenger, T. R. 2012. Sailing into the wind: Exploring the relationships among ambidexterity, vacillation, and organizational performance. *Strategic Management Journal*, 33 (6): 587-610.
- Brown, C. S., and Westenberg, J. 2019. Android 10 release history: The Android Q beta releases and when they landed. *Android Authority*. <https://www.androidauthority.com/android-q-release-date-965486/>
- Brown, S. L., and Eisenhardt, K. M. 1997. The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations. *Administrative Science Quarterly*, 42 (1): 1-34.
- Cao, Q., Gedajlovic, E., and Zhang, H. 2009. Unpacking organizational ambidexterity: Dimensions, contingencies, and synergistic effects. *Organization Science*, 20 (4): 781-796.
- Cecere, G., Corrocher, N., and Battaglia, R. D. 2015. Innovation and competition in the smartphone industry: Is there a dominant design?. *Telecommunications Policy*, 39 (3-4): 162-175.

- Cennamo, C. 2018. Building the value of next-generation platforms: The paradox of diminishing returns. *Journal of Management*, 44 (8): 3038-3069.
- Cennamo, C., Ozalp, H., and Kretschmer, T. 2018. Platform architecture and quality trade-offs of multihoming complements. *Information Systems Research*, 29 (2): 461-478.
- Chen, L., Wang, M., Cui, L., and Li, S. 2021. Experience base, strategy-by-doing and new product performance. *Strategic Management Journal*, 42 (7): 1379-1398.
- Chen, L., Zhang, P., Li, S., and Turner, S. F. 2021. Growing pains: The effect of generational product innovation on mobile games performance. *Strategic Management Journal*, 43 (4): 792-821.
- Clark, K. B., and Fujimoto, T. 1991. *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*. Boston, MA: Harvard Business Review Press.
- Cohen, W. M., and Levinthal, D. A. 1990. Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, 35 (1): 128-152.
- Cusumano, M. A., and Selby, R. W. 1995. *Microsoft Secrets*. New York, NY: The Free Press.
- Daly, A. J., Baetens, J. M., and De Baets, B. 2018. Ecological diversity: Measuring the unmeasurable. *Mathematics*, 6 (7): 119.
- Eisenhardt, K. M., and Martin, J. A. 2000. Dynamic capabilities: What are they?. *Strategic Management Journal*, 21 (10-11): 1105-1121.
- Eisenhardt, K. M., and Tabrizi, B. N. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly*, 40 (1): 84-110.
- Gibson, C., and Birkinshaw, J. 2004. The antecedents, consequences, and mediating role of organizational ambidexterity. *Academy of Management Journal*, 47 (2): 209-226.
- Gomory, R. E. 1987. Dominant science does not mean dominant product. *Research and Development*, 29 (11): 72-74.
- Gomory, R. E., and Schmitt, R. W. 1988. Science and product. *Science*, 240 (4856): 1131-1132, 1203-1204.
- Grewal, R., Lilien, G. L., and Mallapragada, G. 2006. Location, location, location: How network embeddedness affects project success in open source systems.

- Management Science*, 52 (7): 1043-1056.
- Gupta, A. K., Smith, K. G., and Shalley, C. E. 2006. The interplay between exploration and exploitation. *Academy of Management Journal*, 49 (4): 693-706.
- Hannan, M. T., and Freeman, J. 1984. Structural inertia and organizational change. *American Sociological Review*, 49 (2): 149-164.
- He, Z. L., and Wong, P. K. 2004. Exploration vs. exploitation: An empirical test of the ambidexterity hypothesis. *Organization Science*, 15 (4): 481-494.
- Helfat, C. E. 1994. Evolutionary trajectories in petroleum firm R&D. *Management Science*, 40 (12): 1720-1747.
- Helfat, C. E., and Raubitschek, R. S. 2000. Product sequencing: Co-evolution of knowledge, capabilities, and products. *Strategic Management Journal*, 21 (10-11): 961-979.
- Henderson, R. M., and Clark, K. B. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35 (1): 9-30.
- Hippel, E. V., and Krogh, G. V. 2003. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, 14 (2): 209-223.
- Jacobides, M. G., Cennamo, C., and Gawer, A. 2018. Towards a theory of ecosystems. *Strategic Management Journal*, 39 (8): 2255-2276.
- Jacobides, M. G., Knudsen, T., and Augier, M. 2006. Benefiting from innovation: Value creation, value appropriation and the role of industry architectures. *Research Policy*, 35 (8): 1200-1221.
- Jost, L. 2006. Entropy and diversity. *Oikos*, 113 (2): 363-375.
- Kapoor, R., and Agarwal, S. 2017. Sustaining superior performance in business ecosystems: Evidence from application software developers in the iOS and Android smartphone ecosystems. *Organization Science*, 28 (3): 531-551.
- Katila, R., and Ahuja, G. 2002. Something old, something new: A longitudinal study of search behavior and new product introduction. *Academy of Management Journal*, 45 (6): 1183-1194.
- Koch, S., and Kerschbaum, M. 2014. Joining a smartphone ecosystem: Application developers' motivations and decision criteria. *Information and Software Technology*, 56 (11): 1423-1435.

- Kretschmer, T., and Claussen, J. 2016. Generational transitions in platform markets—The role of backward compatibility. *Strategy Science*, 1 (2): 90-104.
- Kumar, S., and Terpstra, D. 2004. The post mortem of a complex product development — lessons learned. *Technovation*, 24 (10): 805-818.
- Kuo, C. S., and Lee, J. R. 2019. Exploring dual business model choice of brand and OEM businesses. *NTU Management Review*, 29 (1): 51-94.
- Lavie, D., Stettner, U., and Tushman, M. L. 2010. Exploration and exploitation within and across organizations. *Academy of Management Annals*, 4 (1): 109-155.
- Lee, C. Y., and Huang, Y. C. 2012. Knowledge stock, ambidextrous learning, and firm performance. *Management Decision*, 50 (6): 1096-1116.
- Levinthal, D. A. 1997. Adaptation on rugged landscapes. *Management Science*, 43 (7): 934-950.
- Levinthal, D. A., and March, J. G. 1993. The myopia of learning. *Strategic Management Journal*, 14 (supplement 2): 95-112.
- Lin, K. H., Peng, Y. P., Peng, Y. C., and Teng, T. C. 2020. The application of social capital and organizational ambidexterity to the value creation process of dynamic capability: Antecedents and consequences. *NTU Management Review*, 30 (1): 1-36.
- Liu, X., Liu, J., Zhu, S., Wang, W., and Zhang, X. 2019. Privacy risk analysis and mitigation of analytics libraries in the Android ecosystem. *IEEE Transactions on Mobile Computing*, 19 (5): 1184-1199.
- March, J. G. 1991. Exploration and exploitation in organizational learning. *Organization Science*, 2 (1): 71-87.
- Miric, M., Boudreau, K. J., and Jeppesen, L. B. 2019. Protecting their digital assets: The use of formal & informal appropriability strategies by App developers. *Research Policy*, 48 (8), Article 103738.
- Nagle, F. 2018. Learning by contributing: Gaining competitive advantage through contribution to crowdsourced public goods. *Organization Science*, 29 (4): 569-587.
- Nelson, R. R., and Winter, S. G. 1982. *An Evolutionary Theory of Economic Change*. Cambridge, MA: Belknap Press of Harvard University Press.
- Nickerson, J. A., and Zenger, T. 2004. A knowledge-based theory of the firm—The problem-solving perspective. *Organization Science*, 15 (6): 617-632.

- O'Reilly, C. A., and Tushman, M. L. 2004. The ambidextrous organization. *Harvard Business Review*, 82 (4): 74-83.
- O'Reilly, C. A., III, and Tushman, M. L. 2008. Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. *Research in Organizational Behavior*, 28: 185-206.
- Page, S. E. 2010. *Diversity and Complexity*. Princeton, NJ: Princeton University Press.
- Piller, F. T., and Walcher, D. 2006. Toolkits for idea competitions: A novel method to integrate users in new product development. *R&D Management*, 36 (3): 307-318.
- Puranam, P., Singh, H., and Zollo, M. 2006. Organizing for innovation: Managing the coordination-autonomy dilemma in technology acquisitions. *Academy of Management Journal*, 49 (2): 263-280.
- Raisch, S., Birkinshaw, J., Probst, G., and Tushman, M. L. 2009. Organizational ambidexterity: Balancing exploitation and exploration for sustained performance. *Organization Science*, 20 (4): 685-695.
- Raju, K. V., and Kumar, D. P. 2015. Customer perceptions about android OS smartphones in Hyderabad. *Journal of Marketing Vistas*, 5 (1): 43.
- Rietveld, J., and Eggers, J. P. 2018. Demand heterogeneity in platform markets: Implications for complementors. *Organization Science*, 29 (2): 304-322.
- Rietveld, J., Ploog, J. N., and Nieborg, D. B. 2020. The coevolution of platform dominance and governance strategies: Effects on complementor performance outcomes. *Academy of Management Discoveries*, 6 (3): 488-513.
- Rivkin, J. W. 2000. Imitation of complex strategies. *Management Science*, 46 (6): 824-844.
- Rochet, J. C., and Tirole, J. 2006. Two-sided markets: A progress report. *The RAND Journal of Economics*, 37 (3): 645-667.
- Rosenberg, N. 1982. *Inside the Black Box: Technology and Economy*. Cambridge, UK: Cambridge University Press.
- Rothaermel, F. T., and Deeds, D. L. 2004. Exploration and exploitation alliances in biotechnology: A system of new product development. *Strategic Management Journal*, 25 (3): 201-221.
- Shiu, J. M. 2017. The scope of support of toolkits in the smartphone industry. *Annals of Business Administrative Science*, 16 (2): 55-65.



- Siggelkow, N., and Levinthal, D. A. 2003. Temporarily divide to conquer: Centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation. *Organization Science*, 14 (6): 650-669.
- Sowe, S. K., Stamelos, I., and Angelis, L. 2008. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems Software*, 81 (3): 431-446.
- Stalk, G., Jr, and Hout, T. M. 1990. Competing against time. *Research-Technology Management*, 33 (2): 19-24.
- Teece, D. J. 2018. Profiting from innovation in the digital economy: Enabling technologies, standards, and licensing models in the wireless world. *Research Policy*, 47 (8): 1367-1387.
- Thomke, S., and Fujimoto, T. 2000. The effect of “front-loading” problem-solving on product development performance. *Journal of Product Innovation Management*, 17 (2): 128-142.
- Tilson, D., Lyytinen, K., and Sørensen, C. 2010. Research commentary—Digital infrastructures: The missing IS research agenda. *Information Systems Research*, 21 (4): 748-759.
- Tiwana, A., Konsynski, B., and Bush, A. A. 2010. Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, 21 (4): 675-687.
- Tushman, M. L., and O’Reilly, C. A., III. 1996. The ambidextrous organization: Managing evolutionary and revolutionary change. *California Management Review*, 38 (4): 8-29.
- Venkatraman, N., and Lee, C. H. 2004. Preferential linkage and network evolution: A conceptual model and empirical test in the US video game sector. *Academy of Management Journal*, 47 (6): 876-892.
- Von Hippel, E., and Katz, R. 2002. Shifting innovation to users via toolkits. *Management Science*, 48 (7): 821-833.
- Wang, T., Aggarwal, V. A., and Wu, B. 2020. Capability interactions and adaptation to demand-side change. *Strategic Management Journal*, 41 (9): 1595-1627.
- Wareham, J., Fox, P. B., and Cano Giner, J. L. 2014. Technology ecosystem governance. *Organization Science*, 25 (4): 1195-1215.
- Wei, Z., Yi, Y., and Guo, H. 2014. Organizational learning ambidexterity, strategic

flexibility, and new product development. *Journal of Product Innovation Management*, 31 (4): 832-847.

Westerman, G., McFarlan, F. W., and Iansiti, M. 2006. Organization design and effectiveness over the innovation life cycle. *Organization Science*, 17 (2): 230-238.

Yalcinkaya, G., Calantone, R. J., and Griffith, D. A. 2007. An examination of exploration and exploitation capabilities: Implications for product innovation and market performance. *Journal of International Marketing*, 15 (4): 63-93.

Yoo, Y., Boland, R. J., Jr, Lyytinen, K., and Majchrzak, A. 2012. Organizing for innovation in the digitized world. *Organization Science*, 23 (5): 1398-1408.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. Research commentary the new organizing logic of digital innovation: An agenda for information systems research. *Information Systems Research*, 21 (4): 724-735.

Zhu, J., Zhou, M., and Mockus, A. 2014. *Patterns of folder use and project popularity: A case study of GitHub repositories*. Conference on ESEM '14: Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement, Torino, Italy.

## Author Biography

### \*Jing-Ming Shiu

Jing-Ming Shiu is currently working as an Assistant Professor, Department of Business Administration, National Cheng Kung University, Taiwan. He holds a Ph.D. in Economics from The University of Tokyo, Japan. His papers have been published in journals such as *Annals of Business Administrative Science*, *Sun Yat-sen Management Review*, and *Journal of Management and Business Research*.

---

\*E-mail: [jingmingshiu@mail.ncku.edu.tw](mailto:jingmingshiu@mail.ncku.edu.tw)

The author is grateful to two anonymous reviewers for their insightful comments and suggestions. I would also like to thank for the funding support from the Ministry of Science and Technology (109-2410-H-006-054-MT3).

The Smartphone Manufacturers' New Product Development Performance in the Generation Transition of Android Ecosystem: Exploiting Old Technologies to Enhance the Effects of Exploring New Technologies