MIKE PINDER
Imperial College London Business School, Tanaka Building, London, UK

Robin Hood and his Merry Innovator's: Integrating Outlaw Innovation from Unsanctioned External Sources to Firms in the Smartphone Market

Introduction

This paper will discuss and relate the literature on innovation with outlaw (non-firm sponsored) innovation sources from lead-user's or user-innovator's and the relationship with firms specialising in designing, developing and manufacturing mobile phones. I will discuss literature in context with smartphone manufacturing and software 'ROM development communities' (henceforth RDC's), specifically Google's Android operating system and XDA Developers community as a source of outlaw innovation in parallel with established firms' (HTC, Samsung, LG et al.) innovation strategies and some of the challenges faced in harnessing external innovation with internal R&D.[1]

Much of the literature on end or lead-user innovations has focussed on the process of identifying a group of individual innovators and inviting them into the organisation in order to uncover unmet needs ahead of the average user's requirements for future product development. The literature does not however discuss unsanctioned (or outlaw), but legal, innovation occurring outside firm boundaries in parallel and separate to the firm's innovation strategy. This has implications for the extent to which current theories can explain how such unofficial innovation sources can be exploited and integrated into a firm's mainstream value ecosystem.

--

[1] ROM: Read-Only Memory – operating system software in mobile phones such as Google Android, Apple iOS, Microsoft Phone 7.

The foundations of innovation studies can be found in Karl Marx's Communist Manifesto (Marx 1848). One of Marx's key contributions in this field was his view on 'the restless nature of capitalism'. Influenced by Darwinian evolutionary theories, Marx commented on degradation of the material conditions of the working classes; alienation of workers being stripped of self-worth and social context; and becoming reduced to simple, replaceable units of labour and general 'withering away of the state'. Although a severe critic of capitalism, he could foresee the possibilities to unleash the productive capacity of innovation for the benefit of the masses. In short he observed that capitalism needs innovation from any source (firm or individual) in order to survive.

Later work by Schumpeter set the foundations for research on innovation and entrepreneurship, mainly his Mark I and Mark II theories (Schumpeter 1942). A key question was based on Karl Marx theories and was concerned whether or not Capitalism could survive. Mark I Schumpeter states that the lone individual entrepreneur is the primary driver of economic development who comes-up with ideas and makes innovations to sell for rents. These individuals 'carry' novelty and are seen as new born entities that provide the primary source of entrepreneurship and innovative ideas for the benefit of the economy.

Schumpeter later changed his views whilst Professor at Harvard (known as Mark II Schumpeter) and he saw that innovation was shifting away from the individual to automated machines (via routines and managers) as well as recognising a shift from individual's and small firms to larger firms with in-house research and development facilities. Schumpeter's definition of innovation was broadly stated in his 1942 seminal work and claimed that innovation was the successful exploitation of the idea as applied commercialisation of inventions. It was via the utility or use of an idea, commercially applied within a context (i.e. when an idea becomes implemented), that it becomes and innovation. Mark II Schumpeter recognised that novelty could now be found through the institutionalisation of R&D and the emergence of the organisation. He observed the primary source of innovation driving the economy was shifting and the means of production was becoming driven by economies of scale, increased firm size and internal R&D departments. Today there are instances where both the individual *and* firm

operate as sources of innovation, concurrently, independently as well as in hybrid collaboration such as in the smartphone market.

Industry types

The Pavitt Taxonomy categorises firms along trajectories based upon sources of technology, requirements of users and the appropriability regime including knowledge flows between groups where different industries have different characteristics, relating to R&D intensity and product types – supplier dominated, product intensive and science based (Pavitt 1984). The smartphone industry falls under two categories: science-based firms and product intensive (specialised suppliers). In science-based firms, prior development of basic science is depended upon, such as semiconductors and within this group there are high technical knowledge boundaries between disciplines (design, production, manufacture, software development, marketing) and rely upon appropriability from patents, secrecy and tacit know-how. In the product intensive (specialised suppliers), firms are smaller and produce technologies to be sold to other firms such as machinery and hi-tech instruments. The taxonomy broadly includes many key factors (organisational structure, sources, management preferences for example), but fails account for innovation from open source communities that do not rely upon high levels of appropriability from tacit knowledge and are in fact based the opposite.[2] Outlaw innovator's within smartphone RDC's fall under the *product intensive* (specialised suppliers) category whereas the smartphone manufacturers fall under the *science based* category and both operate simultaneously within the industry.

Discontinuities

The mobile telephone has emerged as a semi-technological discontinuity that created a new consumer market; untethered personal communication between users (Anderson and Tushman 1990). Anderson's evolutionary cyclical model of technological change denotes a period of intense technological ferment (variants of mobile phone design) over the past twenty years, followed by the emergence of a dominant design, now largely consisting a 4 to 5 inch portable handheld colour touchscreen, speaker, headphone jack and charger socket. A period of ferment has seen the rise and

[2] Open source was in its infancy when the Pavitt Taxonomy was developed, therefore not incorporated.

fall of initially dominant firms within the market (Nokia, Microsoft) as variations of the original breakthrough culminated in the selection of a single dominant configuration of the technology. The dominant design renders obsolete the expertise required to design and manufacture the technology that replaces it, known as competence destroying. Firms driving the emergent dominant hardware design (Apple, HTC, Samsung, LG et al.) reap the benefits of successful physical hardware designs, whilst the software component design continues developing until reaching a dominant configuration; firms that initially led the way can fall behind.

Organisational learning

Rosenberg's (1982) theories on innovation define two ways that firms can learn: learning by using and learning by doing. Learning by using is achieved at the manufacturing stage once R&D has been completed. The producer of machines learns from the experience of those using it and knowledge is created in the making of an innovation. Learning by using is split into two types: embodied and disembodied knowledge. Embodied knowledge is gained in the production phases of prototyping, when purchasing new equipment for example and is found in routines, habits and tasks without conscious thought. Disembodied knowledge comes from learning efficiencies as a result of extended use of an innovation in the form of research spillovers. An example of this is the way in which SMS text messaging was actually used in contrast to what it was originally designed to do. Original designers of the system did not expect usage to the extent it actually was and so adapted handsets with predictive text input for faster message composition. In this context, learning by using acts as an efficiency mechanism and can be used as a primary source of innovation development over time. The RDC for Android-based smartphones operates in a similar manner whereby learning by using over extended periods of usage is a primary mechanism in which efficiency improvements can be identified and worked upon. Handset manufacturers on the other hand are limited by the availability and deployment of resources (as well as commercial incentives) in order to operationalise extensive learning by using initiatives once a product is sold and has entered into a relatively short product life cycle. And if the firm does not sell more handsets as a result of its innovations, then it is outside the scope of the business model and strategy.

Use of innovations and social capital value

Innovations gain value when used by others (socially) and cannot create value when they are not used (Edgerton 1999). In order to get innovations used by consumers and users, they need to recognise some degree of novelty, whether a firm, a thing, an object or a process. Smartphones devices incorporate high degrees of novelty in the form of new product features (software, aesthetic design, hardware specifications) as well as social and cultural capital (Bourdieu 1986). However firms place varying degrees of emphasis upon areas of novelty and innovation based upon market segmentation and perceived consumer preferences. A brief content analysis of marketing artefacts within the smartphone industry demonstrates manufacturer focus upon software features, product aesthetics and cultural capital, rather than underlying hardware architecture and specifications. Competitive advantage is offered (to majority early and late consumers) based primarily upon the software component. The logic being more features, functionality and productivity gains provide differentiation factors for adoption and diffusion of the innovation (Rogers 1995). Logic would then follow that further software component innovations from RDC communities (with even more features, functionality, productivity and performance enhancements) would offer increased perceived relative competitive advantage in the eyes of the consumer over rival offerings. This however does not appear the case as incumbent handset manufacturers, to varying degrees, show signs of path dependent behaviour with regard to previous and current and potential sources of innovations and are held back by cognitive inertial forces from previous routines, driving current closed firm behaviour and action. The result is firms may not adopt or incorporate external sources of innovation due to their rigid inertial forces and regimes developed during previous experiences over time (Nelson and Winter 1982).

Rogers (1995) takes a sociological view on innovation and explains how, why and what rate new technologies spread through cultures. He states innovation can consist of new ideas, not just physical objects and that innovation diffusion is made through five key adoption processes: knowledge, persuasion (adopting or rejecting), decision, implementing and confirming. Roger's goes on to define five factors that influence an individual's adopt or reject decision of an innovation: relative advantage,

compatibility, complexity/simplicity, trialability and observability. Five further adopter categories are defined (innovators, early adopters, early majority, late majority and laggards) and forms the basis of the s-curve and foundations of marketing studies.

RDC's can provide influential social opinion leaders as members interact on a one-to-one and visible basis with lead developers on public forums, code developer sites and IRC channels. This provides influential (positive or negative) information that may impact upon the evaluation stage for all adopter categories, especially if integrated in conjunction with firms rather than peripherally as outlaw voices. Future research on how lead-user opinion leaders can influence innovation adoption rates of new technologies would be beneficial.[3]

Lead-User Theory

Lead-user theory provides an extension to Roger's diffusion model in advance of the s-curve concept. Von Hippel (1986) introduces the term 'lead users' and distinguishes them from ordinary users of products. There are two criteria for identifying lead users: (a) they have general needs before others and (b) they are positioned to benefit from those needs. Lead users are specified to have strong needs that will later proliferate into the market as general needs, months or years in the future. Lead users can become a source of future forecasting for marketing research, product conceptualisation and design specification information (particularly in the high-technology fields). Von Hippel explains how lead users can be identified and how their unanticipated future needs can be incorporated into the organisation for new product development, processes and services.

A theoretical framework is developed in order to extract data for future product enhancements via similarity and dissimilarity studies of existing wants versus new wants. Research methods such as focus groups and surveys are then employed in order to extract lead user data using von Hippel's theoretical framework:

[3] The Roger's diffusion s-curve estimates that 2.5% of total consumers account for actual innovators. Estimates of a popular RDC modified handset rooted and running customised operating system ROM's around 3-4% of total unit sales. This represents significant number of handsets running customised code (~50k handsets). Based on an estimated 3 million HTC Desire HD handsets sold to date and number of EasyS-OFF downloads (required to install customized ROM's).

1. Identify important trends
2. Identify lead users
3. Analyse lead user data
4. Project data onto marketplace

The premise behind the theoretical framework is that once identified, lead users are invited into the firm for structured marketing data extraction in order to develop future products that are more closely aligned with future mass-consumer needs and relies on the firm's ability to successfully identify the lead users with the correct needs for future product and service development. The framework does not account for product innovations that occur outside of the firm's external radar, being developed independently, in parallel or entirely unaware to the organisation. Furthermore, this method of identify lead users, before products are designed and manufactured, provides data a priori to the innovation process. This does not account for innovations that occur a posteriori once the product is in the hands of the lead user and can occur within product markets where multiple innovation components exist.

Smartphones comprise two separate components for innovation R&D: the hardware and software (Rogers 1995 p.36). Once handsets are constructed, further innovations by user-innovators to the physical hardware is more difficult (if not impossible) to reinvent as many of the components are expensive and consist of hard to reproduce micro-chips and processors, tightly integrated into circuit boards and require a high degree of expertise and tools to manipulate, change or otherwise improve upon. This is not the case for the software component (operating system) as the source code can be innovated upon by extracting, decompiling and manipulating line-by-line on a separate computer. In effect the software component can be subjected to further improvement and innovation, but the physicality, cost and finalised state of the hardware component prevents this from occurring.

Arora et al. (2001) describe markets for technology that reduce the relative importance of technology as a source of distinct advantage (as with the Android open source operating system), as other firms can freely access the same technology. As a result firms must focus on developing other sources of

competitive advantage through internal assets that provide distinctive advantages as well as downstream differentiation and client needs. Such assets can also be sourced externally via RDC's as they maintain the capabilities of existing internally developed code and maintain IP protection from other firms. This logic has formed the basis of the Open Innovation paradigm (Chesbrough 2006).[4] Google's Android operating system for smartphones is an open source software package developed by a community of programming engineers and is incorporated and developed by manufacturers such as HTC into handsets. As Android is an open software project, much of the knowledge and expertise invested in its development lies in community of developers external to the organisation. This gives rise to conflicting intellectual property viewpoints, as the hardware component remains proprietary, whilst the handset source code and software layer remains on open code.

By contrast Apple relies upon already existing internal knowledge and capabilities by porting over its Unix based consumer computer operating system OS X over to its iOS smartphone operating system as a derivative of thousands of engineering hours work, knowledge and resources to power its hardware. Other competing smartphone manufacturers do not have decades of code development to draw from and look to the open source community of developers to provide a solution to the number of engineering hours required to build and continually develop high quality and stable operating system.

As Android is an open source project, a variety of other manufacturers have access to the same kernel and code that can be incorporated into competing handsets. This removes a large proportion of competitive advantage for the firm in terms of differentiating one handset software component with another. Millions of handsets can use the same code, performing to the same standards on the same hardware specifications. In order to provide product differentiation and increase the value proposition to consumers, intellectual property in the form of proprietary code and graphical user interface (GUI) enhancements are developed by major manufacturers such as HTC's Sense software layer. User

[4] Markets for technology have been found to reduce barriers to entry, increase competition and compress the PLC, but in the case of smartphone life cycles, this is primarily moderated by consumer subscriber contract duration.

interfaces (UI's) provide a number of firm-specific innovations and additional sources of competitive advantage over other firms using the same operating system code. This closed source code is specific to individual manufacturers and is subjected to intellectual property law enforcement and infringement if duplicated, modified or otherwise used by a rival firm.

Pisano and Teece (2007) discuss the challenges of creating and capturing value from innovations and address how firms can affect the IP environment and industry architecture. Firms can disclose and 'go open' by publishing their IP or strengthen their IP by law enforcement of competitor infringements. Handset manufacturers could gain bargaining power against proprietary firms by adopting open source code and then enhance this position by further appropriating open IP from their consumers and users.

At the product offering level, the hardware (physical design and ergonomics) and software (firm specific UI enhancements) make up the product value proposition to consumers. At the point of sale, software code is in a relatively immature state, due to the nature of learning by using required to incrementally improve, bug test and fully-optimise software code over extended periods of time. However firms market handsets to consumers on this immature code, as new innovative software is vital for product differentiation.

Firms like HTC rely on both open and closed source software code in order to provide competitive and innovative offerings to consumers. However once the open source code is back in the hands of lead users and innovators –some of whom may have helped develop it- they find that access to the code is restricted by locked bootloaders installed by manufacturers.[5] Lead users must circumvent or 'hack' manufacturer bootloaders put in place (designed to prevent non-firm access to proprietary source code) in order to extract, decompile, edit and innovate on top of the manufacturer layer and

[5] A bootloader is a piece of code that runs before any operating system is running and contains several ways to boot the OS kernel and also contain commands for debugging and/or modifying the kernel environment.

underlying open code.[6] In the case of firms like HTC, this means gaining access to the proprietary Sense UI software in addition to the open source Android operating system.

The lead user framework does not account for innovations that occur in existing products once they are in the hands user innovators. The framework also assumes that the firm has some degree of control over the selection and invitation of willing lead users into the firm. As is the case with HTC, there are a number of issues and tensions amongst lead user communities with large multinational firms, relating to the development, use and application of Open Source code in conjunction with proprietary intellectual property software and hardware. In this context, not all open-minded lead users will be willing to participate and divulge their user needs, data and code to closed mind set profit-driven firms.

Furthermore, von Hippel does not account for user innovations that take place outside of firm walls, irrespective of copyright law and IP infringement and independently of the firm entirely. A wider framework is needed in which 'outlaw' innovations is included as a source of innovation that can be absorbed back into the firm as an additional and complementary resource of cumulative knowledge and learning on top of that initially developed and created by the firm.

[6] A number of manufacturers have now released handsets with open bootloaders allowing for the extraction, manipulation and re-loading of customised operating system (ROM) code; see: http://www.pcworld.com/article/228823/htc_ends_locked_bootloader_policy.html.

Product life cycles and software component from both firm and user-innovators

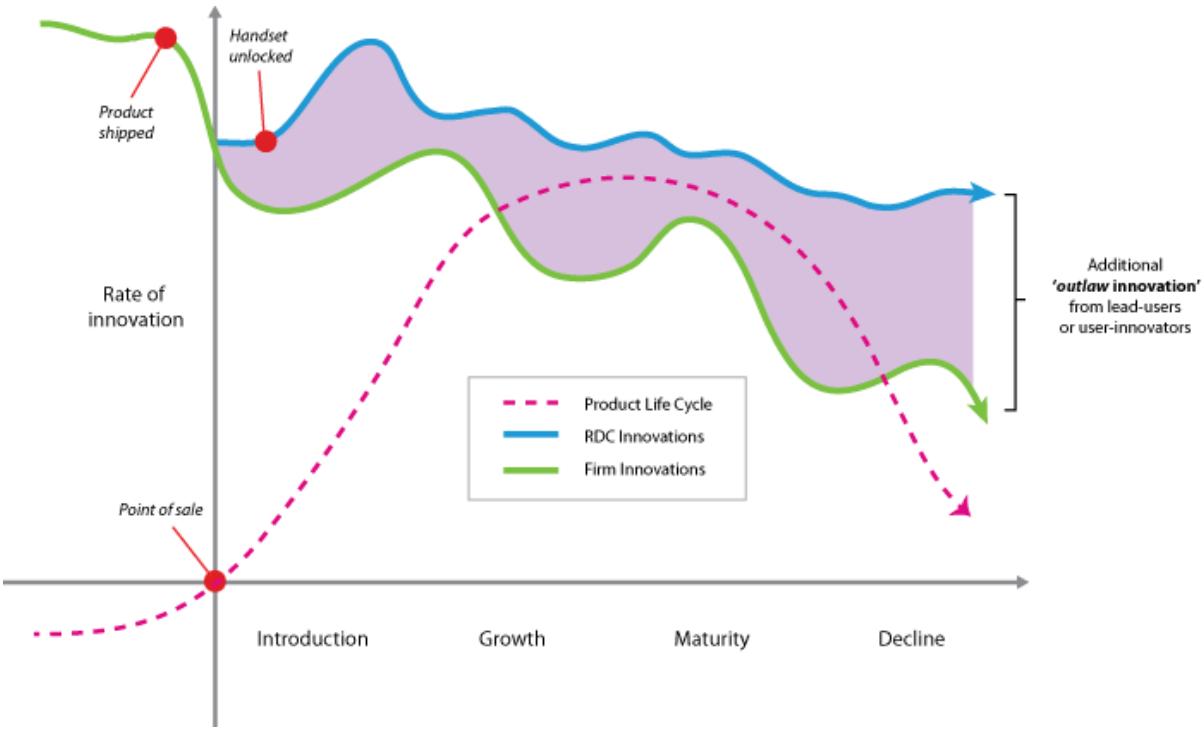

Figure 1. User & firm innovation rates with product life cycle

Figure 1 shows approximate innovation rates for both the firm and RDC innovation communities, representing complimentary and additional 'outlaw innovation' and knowledge available to the firm as unused and potential useful resources.

Smartphone handset manufacturers (HTC, Apple, Nokia et al.) innovate in both components simultaneously in order to offer incremental improvements over previous generations by adding new features, thereby enhancing the product value proposition to consumers. Software component innovations to the product are continuously updated, changed and improved, offering further performance and features, but the architecture remains largely the same (Henderson and Clark 1990). Once latest features and better performing components are assembled into a handset, in-house software developers add further component innovations to enable the user to interact with the hardware components via GUI's. The complete product is then sold to the consumer with an

expected product lifecycle of around 12-24 months, usually based on the subscriber contract duration (as handset prices are relatively high and subsidised to consumers on long-term contracts).

Firm innovation is high (at pre-product shipment) as handset components are new to market in both software and hardware components. At the point of sale, firm innovation levels significantly drop as resources are re-deployed to R&D activities for future product development. At this point, lead-user or user-innovators purchase high-end handsets and explore techniques for bootloader circumvention. Once manufacturer bootloaders have been disabled and replaced, customised code can be installed, modifying both open source and proprietary code. At this point innovation increases beyond the rate sustainable solely by firm resources for R&D (post product launch). Increased RDC's numbers emerge as newly optimised kernels, ROMS and mods are developed and distributed via ICT's. RDC software release cycles are consistently higher than those of the firm as many iterative and incremental innovations (learning by using) are continuously developed in RDC teams with lead developers and users testing and giving direct feedback using bug management software, IRC channels and forums. Both firm and RDC community innovation rates decline over time (due to decreasing firm support of products in the latter phase of the PLC and RDC teams moving to newer handsets), however RDC innovation rates remain above levels capable by the firm alone due to consistent update releases, system effects of handset performance improvements, the number of handsets rooted and expanse of RDC's in general (Freeman 1991). Figure 1 highlights in purple, additional innovation sources available to the firm over the duration of the PLC that may not be currently exploited by firms, as it exists independently of firm efforts entirely.

RDC innovation

As discussed smartphone innovation is split into two components consisting the overall product architecture: the hardware and software (Henderson and Clark 1990). RDC's do not physically modify and innovate the hardware component, but instead make significant modifications and innovations to the software (ROM, kernel and bootloaders). Innovations to software components do not constitute innovations in the Schumpeterian sense per se, although there is evidence that supports economic gain as a motivational driver to innovate. Regular donations are made through online

payment processors, of which links can be found on every developer forum post. Once a donation is made, members post verification codes to increase integration and social capital from other developer members and community as a whole. This challenges previous studies that focus on intrinsic motivations as the motivational factors for participation in firm hosted user communities rather than extrinsic monetary rewards that may also co-exist within non-firm hosted user communities such as XDA Developers for various platforms and handsets (Jeppessen and Frederiksen 2006).

Profiting from lead user innovation

Teece (1986) highlights two key factors in determining who profits from innovation: *complementary assets* (infrastructure, capabilities needed to support an innovation) and *imitability* (how easily a competitor can duplicate a technology or process needed for an innovation). Firms can deploy various barriers to prevent imitability such as tacit knowledge, IP regimes and complex internal knowledge and the model can be used to specifically understand innovation dynamics. The Teece two-dimensional matrix model allows firms to predict who will profit from rents generated by an innovation. In the case of HTC, the hardware component imitability is low (due to the level of technical expertise needed to build and develop hardware); and the complementary assets are tightly held by the firm (protected by IP regimes and tacit knowledge in physical objects). The hardware component holder (HTC) is therefore in a position to profit from its innovation. However for the software component, imitability is high because it is based on widely-available open source code; and the complementary assets are freely available as the operating system knowledge codified in a single object or ROM and can easily be downloaded, decompiled and distributed in RDC's. This means for the software component it is difficult to generate profits.

The Teece model may explain why hardware manufacturers have largely ignored and neglected external innovations from RDC's, where hardware components have traditionally generated rents. While the model is useful for single component firms, it has limitations for firms with multiple innovation components in software and hardware. In this instance the model does not describe or predict multi-dimensional dynamic situations, where innovations incorporate several components within a single value proposition, but rather treats them separately, making it hard to draw concrete

conclusions as to who will profit from innovations. Future research could develop a model that can deconstruct and predict dynamic multidimensional components for innovations such as smartphones.

Consumer lock-in mechanisms

The use of external sources of innovation from RDC's could provide a mechanism for smartphone manufacturers to avoid premature lock-in to older, more inefficient code in the PLC and allow for future software builds to innovate in greater increments from one PLC to the next. Including internal resources and knowledge that could be built upon if absorbed back in from RDC's as a virtuous cycle (Cohen and Levinthal 1990; Grant 1991; Kogut and Zander 1992; Teece, Pisano et al. 1997).

External dependence upon third party developers for the software component, means that future update cycles cannot be controlled by the firm alone, but must wait for the development community to release a new base for firms to build on. This creates code a lock-in at point of purchase and is dependent upon a third party for future improvements (David 1985). As handset code requires a high degree of iterative improvements and learning by using, RDC's would provide a mechanism to postpone, delay or even fully prevent inefficient code lock-in for the duration of the product's life cycle and would provide a continually improving product over time. Opposing viewpoints may argue that market inefficiencies within the product life cycle itself actually drive future consumer purchases, who seek better performing hardware and software components combined. But as the product life cycle is largely dictated by contract duration, this may not necessarily extend expected overall usage time (postponing the adoption of the next handset), but would provide a better overall performing product for the duration of the user's contract period, creating other forms of consumer lock-in and loyalty; and in addition, a stock of resources for the firm developed externally for free.

Freeman (1991) finds that internal and external networks are important drivers for innovation success, especially for new products and processes. One of his main contributions was to foresee computer networks and ICT as a way to rapidly deploy and diffuse new generic technologies such as programming and sharing code (the seeds of Open Source). Freeman saw ICT's as a 'techno-economic paradigm change', pervasive enough to affect the behaviour of the entire system as a hybrid

organisational form between markets and hierarchies. As handset manufacturing firms now rely on external network innovations for one of the two key components to their product innovations (i.e. operating system software), it seems counter productive to then prevent, obstruct and ignore this same external sources of innovation once smartphone handsets have been sold to these communities.[7] RDC developers can generate and contribute to a number of important consumer lock-in mechanisms from external and unofficial or unsanctioned outlaw sources that may not be fully understood in the literature.

Incremental vs. radical innovation

Henderson and Clark (1990) challenge the notion of incremental or radical innovation claiming they are misleading and do not account for changes in the architectures of a system and do not explain the type of company that would be in a better position to innovate and under what circumstances. A conceptual framework is introduced that highlights the implications of changed or unchanged linkages between architecture and components of an innovation. If linkages are unchanged, incremental and modular innovation takes place. If linkages do change, then architectural and radical innovations can take place and will destroy the usefulness of existing knowledge in firms and account for firm failures.

- Incremental innovation builds upon existing component and architectural knowledge.
- Modular innovation requires new knowledge for one or more components, but the architecture remains the same.
- Architectural innovation changes the linkages of components, but knowledge of individual components remains the same.
- Radical innovation changes both component and architectural knowledge.

In the case of smartphones and outlaw RDC's, innovation builds upon both modular and incremental innovations, as for example performance improvements are built upon existing software components

[7] (i) Samsung recently donated Galaxy S II handsets with unlocked bootloaders to ROM development team CyanogenMod in order to promote and encourage end user innovations: http://www.wired.com/gadgetlab/2011/06/samsung-cyanogen. (ii) HTC also recently announced code to unlock all handset bootloaders: http://www.reghardware.com/2011/07/11/htc_readies_sensation_bootloader_unlock

and modular components can be interchanged into new or unintended environments and require new knowledge to achieve. The linkages between hardware and software do not physically change but the knowledge required and generated to do so may be found within individual outlaw innovators contributing to RDC's. For the firm, new competencies and strategies will be required to exploit these innovation knowledge changes and sources effectively. Highly modularised manufacturer software components across all handsets ranges would allow knowledge to flow and be generated across many more RDC teams for example.

Absorptive Capacity

Absorptive capacity provides a framework in which to explore reasons for closed external scanning for innovation by lead or end-user innovators and is defined as 'the ability of a firm to recognise the value of new, external information, assimilate it and apply it to commercial ends and is critical to its innovative capabilities' (Cohen and Levinthal 1990). The theory essentially deals with the idea of innovation spillovers and the firm's ability to recognise, assimilate and apply this new knowledge and ideas to commercial, innovative ends in the Schumpeterian sense.

Key concepts look at the path dependent nature of prior knowledge, cognitive structures and the capacity to learn. It appears in the case of HTC that the development of absorptive capacity is somewhat absent due to lack of initial investment of resources in this area of organisational expertise. Prior software development for smartphone handsets was outsourced to Microsoft with its Windows CE/Windows Mobile operating system (a closed and propriety system) and as such the RDC was much smaller in terms of number of handsets using it, number software development cycles and releases and number of software developers versed in Microsoft's own programming languages: Visual Studio, Visual C++, and .NET Framework.

HTC's prior path dependent knowledge, relying on a single firm for its software innovation and not on a network of external innovators, trained in open software standards and programming languages provides some insights to explain this. Prior routines built around appropriating a closed operating system from a single organisation and from an in-house R&D department are very different from

appropriating it from external network of open source innovators who share vast amounts of codified and tacit knowledge among themselves in vastly geographically distributed ICT's.

In terms of cognitive structures, organisational memory acts as a self-reinforcing mechanism and tunes the firm into a certain type of knowledge; in this case sourcing and innovating internal UI enhancements on top of a closed (Windows) operating system. This routine can inhibit the capacity to learn new capabilities (learning to assimilate, problem solving and creating new knowledge) as organisational history matters and has an impact on future abilities to learn from external sources. The key concept here is dealing with innovation spillovers and how firms respond to them as unintended knowledge flows that have been the driving force of open source software development. Firms like HTC could learn and develop internal R&D structures to take advantage of external innovations from RDC's rather than attempt to restrict activities or be entirely disengaged.[8]

Incorporating outlaw innovations

Study of the non-spread of innovation, despite evidence has shown that social, cognitive and epistemological boundaries affect diffusion and adoption rates (Ferlie and Fitzgerald 2004). Evidence based diffusions studied in NHS healthcare have shown that diffusion is based around a dynamic processes that is affected by communities of practice and how groups make sense of information around them, contrary to Rogers (1995) linear model of diffusion. These communities affect perceptions of innovations and how likely members are to adopt. Innovations that span professional and organisational boundaries (complex innovations) have implications on diffusion because: communities are sealed-off from other communities and they develop different identities with members belonging to a specific profession (Greenhalgh, Robert et al. 2004). In this instance knowledge becomes 'stuck' and may be entirely ignored, unless external factors disrupt and diffuse knowledge such as social group practices designed to allow for knowledge transfer by incentivising members with rewards or policy enforcements.

[8] Google recently deleted tens of thousands of business accounts using its new social networking platform, Google+. Industry leaders can demonstrate strategic errors whereby users behave in unintended or unanticipated ways and reactions can be overly aggressive. Systems and managers need flexibility to adapt to such unanticipated uses, as was seen in SMS text messaging system usage.

RDC's are a community of practice very different to those of the firm (Brown and Duguid 1991). RDC's have clear common interests, outcomes and goals (to improve performance, optimise and extend functionality of handsets), whereas manufacturing firms are focussed on mass consumer appeal and average software innovation, optimisation and performance. As firm and development communities have differing but complementary goals (better performing handsets), some form of boundary-intermediary is needed in order to bridge both communities and incorporate both groups' outcomes and goals, rather for them to exist independently and in parallel with little or no knowledge crossover. Figure 2 demonstrates the cumulative potential of knowledge, resources and innovation from one product lifecycle to the next, from external outlaw development communities in relation to the firm and PLC.

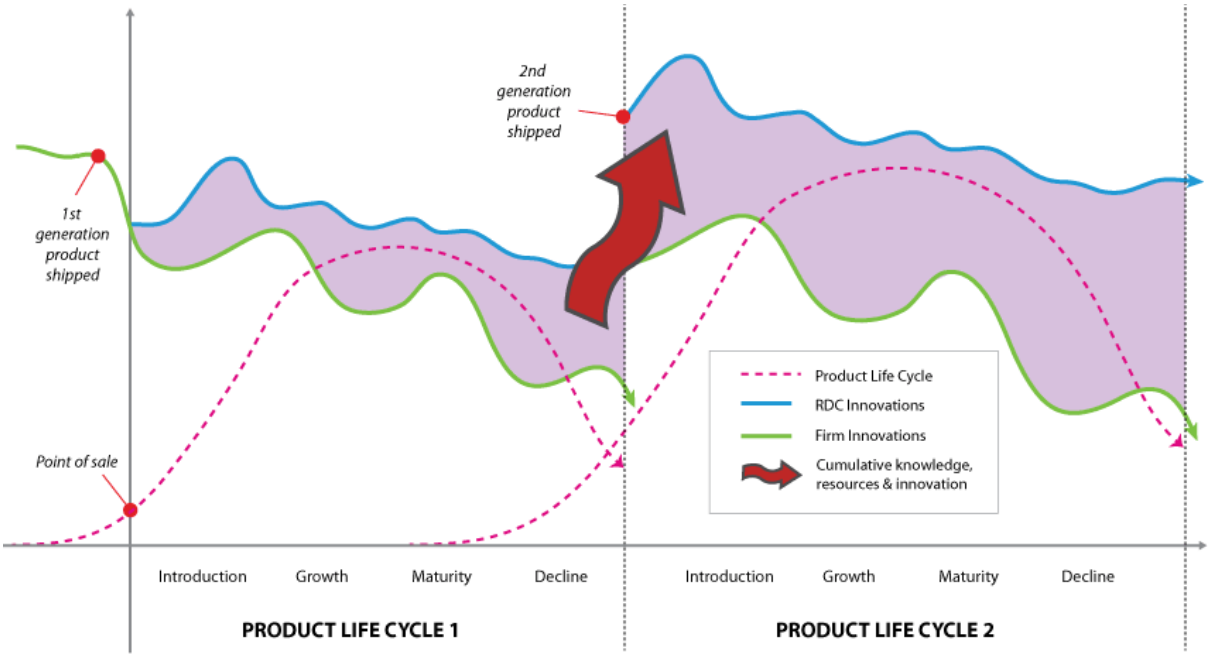

Figure 2. Cumulative knowledge spillovers, resources & innovation potential from one product life cycle to the next

Transaction cost economics

The hardware components of smartphone design is complex and is internalised by firms like HTC in order to achieve vertical integration (Coase 1937; Williamson 1979). This lowers transaction costs, synchronises supply and demand along the chain of products and allows the firm to monopolise the market. However the software component is also complex and requires thousands of engineering

hours, in order to develop an operating system with a competitive enough market offering. This is performed by strategic alliances between handset manufacturers and open source code from Google's Android operating system (Eisenhardt and Schoonhoven 1996). This alliance provides handset manufacturers with knowledge, expertise and intellectual property and provides Google with valuable end-user data; and aims to provide both parties with benefits greater than those of their individual efforts. The alliance allows for bi-directional technology transfer (software and hardware) in order to ensure new developments are accessible to a wider range of intra firm developers. However, firms at both alliance ends *as well as* RDC's can be integrated into an innovation network, creating value dynamically as an integrated-whole. Resulting in further developed, enhanced, optimised and exploited technology, new applications and services available to be absorbed back into the firm and be integrated into future innovations and product offerings (Cohen and Levinthal 1990).

Networks of innovations and collaboration

Research on collaboration and innovation provides insights into organisational forms (alliances and networks) that may be relevant for explaining RDC's in conjunction with smartphone manufacturers. Foundations of research lie in criticisms of transaction cost economics (TCE) and focus on knowledge based view (KBV) of the firm, open innovation (outside-in vs. inside-out innovation) as individuals within firms hold knowledge and know more than they can explain (Polanyi 1966; Williamson 1979; Kogut and Zander 1992; Chesbrough 2003). KBV addresses make or buy decisions and knowledge can be transmitted without loss of integrity, as in following a recipe in order to produce a desired outcome. This form of codified knowledge is then easy to produce and duplicate, as soon as the syntactic rules are known (or learned) in order to decipher and modify it. This is the case with highly codified external knowledge within open source projects such as Android O/S with a large user base and number of people with knowledge in how to decompile (decipher) proprietary and open code. However, the highly codified nature of smartphone code means relatively few RDC developers (vs. RDC users of code) actually exist with similar capabilities as firm developers.

Pfeffer and Salancik (2003) and the resource dependence view discuss the hazards of procuring knowledge and innovation from exterior sources outside the firm and the implications of external

power and control over organisations once a dependence is made. This is especially true for smartphone manufacturers as a large degree of knowledge is procured in the form of the software component from external sources, forcing such firms innovation output to be dependent upon software release cycles of a community of open source developers (Google Android).

Firms do not exist as a form of opportunism but for knowledge –transferring and sharing in teams, between people and firms- processing and spreading knowledge that in turn determines firm boundaries (Coase 1937; March and Simon 1958; Polanyi 1966). In this view sharing of knowledge in networks is seen as advantageous because new knowledge can be created by the increased combinative possibilities for radical and novel ideas spurring diverse innovative outcomes.

Thought worlds

Dougherty (1992) examines the idea of departmental thought worlds (what people know and where they fit in an organisation) and organisational routines (that bridge and reinforce or inhibit the innovation of products). She found that thought worlds existed between planning, design, technology, marketing and manufacturing departments and that routines contribute to the separation of these thought worlds; via knowledge silo's, pre-determined views of technology and sets of standards. Successful innovating firms were those that broke-past such strong routines by building on unique skills in thought worlds, joint working and collective action behaviours. Research into the existence of such routines and thought worlds would be particularly relevant in firm environments (HTC, Samsung, LG et al.) where strong prior routines are likely to exist within smartphone firms that have initially grown using proprietary (outsourced) operating system code from Microsoft where total control and integrity of codified knowledge can be maintained with closed sources not available in the public domain.

Research in this context would expect to find examples of innovation-enabling routines that bridge thought worlds, departments, firms with external RDC's, such as joint partnerships with lead external

developers and firm level policy changes to overcome barriers.[9] If past alliance activities lead to the likelihood of future alliance activities, then this could give insight into why handset manufacturers have so far not made any formal alliances and networks with RDC's (Gulati 1999).

IP protection does not function as an answer to knowledge leakiness within the Android operating system as it is based on open source software. Additional UI enhancements added on top of the operating system layer do however fall under IP protection law, consisting of proprietary code created by in-house R&D developers and coders, effectively making the entire software component a competitive asset that differentiates the common Android core from other competitors in the eyes of consumers. This hybrid (open and proprietary) software component is then offered to consumers (innovators through to laggards) who unpack and modify both open and closed code, as desired, with little regard of IP infringements or permissions to modify by manufacturers (Rogers 1995). This creates a grey area for copyright policy enforcement by firms, as it is not immediate competitors potentially infringing IP, but the community of highly influential lead-users and user-innovators and end-product consumers.

Implications for firm policy & strategy

Smartphone manufacturers find themselves in strong control of hardware component innovations using traditional IPR regimes, but in little control over the software innovation components either collaboration or partnerships with innovations developed in RDC communities. Coff (1999) views the firm as a nexus of contracts where people and relationships generate rents and bargaining power is low for stakeholders. If a firm lacking the time and resources, pursues collaboration with outlaw innovators, then appropriate reward mechanisms (rent splitting, rent appropriation) would need developing for both parties.

IP is strongly defended against competitor UI infringements such as Apple and at the same time outlaw innovators in RDC's have been largely ignored by HTC altogether.[10] Immediate firm policy

[9] A few examples recently reported include manufacturers donating smartphone handsets to RDC members and firm level commitments to open bootloaders.

implications would suggest a generally more permeable approach between firm and RDC, releasing SDK tools to RDC developers; non-rooting and unlocking of handset bootloaders; modular software architectures across handset ranges to increase knowledge spillovers and transfer; maintaining warranty integrity of non-manufacturer ROM flashing; easy to use manufacturer and RDC developed ROM loading tools; marketplace for integrating RDC with firm software innovations and avoiding pursuing outlaw innovator IP infringements.

Conclusion

Outlaw innovation can provide an important additional source of innovation for firms providing it can be appropriated in a manner that benefits all parties, especially as motivations for participants may vary significantly. If the software component provides less of a significant source of competitive advantage between firms using the same open core code, then RDC's could enable the firm developed software layer and component to become a key strategic and competitive asset in competition against closed system device manufacturers.

Outlaw innovation can provide challenges to existing business models, regulatory regimes and product development strategies that may not be fully understood. Firms have a number strategic options in order to react to outlaw innovations: monitoring, adopting, adapting, influencing, absorbing, exploiting or attacking external sources (Flowers 2007). As discussed, firms may take a combination of these approaches sending confusing signals to RDC communities that may further alienate rather than integrate. A combination of approaches may be needed (dependent upon for example –product type, scale and nature of innovation), but clear strategies need to be developed and communicated to all parties and could form a basis of future research.

Lead user innovations have largely been viewed as beneficial to firms within the literature, but parallel innovation from unofficial, unsanctioned sources, (potentially even in conflict with the firm's existing IP regime) from external, independent development communities, may provide additional sources for appropriation and incorporation into the product lifecycle, brand, consumer insight and

[10] A recent case in point Apple vs. HTC IPR lawsuit causing 6.5% drop in HTC share prices: http://www.ft.com/cms/s/2/30c3aaf2-b11c-11e0-a43e-00144feab49a.html#axzz1T8NvUxbW

future product development of both software and hardware architectures. Future research in this area may include; firm reactions to RDC innovation over time, conditions under which firms and outlaw innovators can mutually benefit from alliances and its impact on new product development and performance.

Anderson, P. and M. L. Tushman (1990). "Technological discontinuities and dominant designs: a cyclical model of technological change." Administrative Science Quarterly(35): 9-30.

Arora, A. and A. Fosfuri (2001). "Markets for technology and their implications for corporate strategy." Industrial and Corporate Change **10**(2): 419-451.

Bourdieu, P. (1986). The Forms of Capital. New York, Greenwood.

Brown, J. S. and P. Duguid (1991). "ORGANIZATIONAL LEARNING AND COMMUNITIES-OF-PRACTICE: TOWARD A UNIFIED VIEW OF WORKING, LEARNING, AND INNOVATION." Organization Science **2**(1): 40-57.

Chesbrough, H. (2006). "Open Innovation- The New Imperative for Creating and Profiting from Technology." Academy of Management Perspectives(3): 86-88.

Chesbrough, H. W. (2003). "The era of open innovation." Mit Sloan Management Review **44**(3): 35-41.

Coase, R. H. (1937). "The Nature of the Firm." Economica **4**(16): 386-405.

Coff, R. (1999). "When Competitive Advantage Doesn't Lead to Performance: The Resource Based View and Stakeholder Bargaining Power." Organization Science **10**(2): 119-133.

Cohen, W. M. and D. A. Levinthal (1990). "Absorptive Capacity: A New Perspective on Learning and Innovation." Administrative Science Quarterly(35): 182-152.

David, P. (1985). "Clio and the economics of QWERTY." American Economic Review(75): 332-337.

Dougherty, D. (1992). "Interpretive barriers to successful product innovation in large firms." Organization Science **3**(2): 179-202.

Edgerton, D. (1999). "From Innovation to Use: ten (eclectic) theses on the history of technology." History and Technology **16**: 1-26.

Eisenhardt, K. M. and C. B. Schoonhoven (1996). "Resource-based View of Strategic Alliance Formation: Strategic and Social Effects in Entrepreneurial Firms." Organization Science **7**(2): 136-150.

Ferlie, T. and L. Fitzgerald (2004). "The Nonspread of Innovations: The Mediating Role of Professionals." Academy of Management Journal **48**(1): 117-134.

Flowers, S. (2007). From Outlaws to Trusted Partners: Challenges in mobilising User-Centric Innovation in R&D projects. Centre for Research in Innovation Management (CENTRIM), University of Brighton. Brighton, University of Brighton (CENTRIM).

Freeman, C. (1991). "Networks of Innovators: A synthesis of research issues." Research Policy **20**(5): 499-514.

Grant, R. M. (1991). "The Resource-Based Theory of Competitive Advantage: Implications for Strategy Formulation." California Management Review **33**(3): 114-135.

Greenhalgh, T., G. Robert, et al. (2004). "Diffusion of Innovations in Service Organizations: Systematic Review and Recommendations." Milbank Quarterly **82**(4): 581-629.

Gulati, R. (1999). "Network location and learning: The influence of network resources and firm capabilities on alliance formation." Strategic Management Journal **20**(5): 397-420.

Henderson, R. and K. B. Clark (1990). "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms." Administrative Science Quarterly(35): 9-30.

Jeppessen, L. B. and L. Frederiksen (2006). "Why Users Contribute to firm hosted user communities." Organization Science **17**(1): 45-63.

Kogut, B. and U. Zander (1992). "Knowledge of the firm, combinative capabilities, and the replication of technology." Organization Science **3**(3).

March, H. and J. Simon (1958). Organizations. New York, Wiley.

Marx, K. (1848) "The Communist Manifesto."

Nelson, R. R. and S. G. Winter (1982). An Evolutionary Theory of Economic Change. Cambridge, Harvard University Press.

Pavitt, K. (1984). "Sectoral Patters of Technical Change: Towards a Taxonomy and a Theory." Research Policy(11): 147-162.

Pfeffer, J. and G. Salancik (2003). The external control of organizations: a resource dependence perspective, Stanford Business Classics.

Pisano, G. and D. J. Teece (2007). "How to Capture Value from Innovation: Shaping Intellectual Property and Industry Architectures." California Management Review **50**(1): 278-296.

Polanyi, M. (1966). The Tacit Dimension. New York, Doubleday & Company, Inc.

Rogers, E. M. (1995). Diffusion of Innovations. New York, The Free Press.

Rosenberg, N. (1982). Inside the Black Box: Technology & Economics. Cambridge, Cambridge University Press.

Schumpeter, J. A. (1942). Capitalism, Socialism and Democracy. London, Unwin.

Teece, D. J. (1986). "Profiting from Technological Innovation." Research Policy **15**(6): 285-306.

Teece, D. J., G. Pisano, et al. (1997). "Dynamic capabilities and strategic management." Strategic Management Journal **18**(7): 509-533.

von Hippel, E. (1986). "Lead Users: A Source of Novel Product Concepts." Management Science **32**(7): 791-805.

Williamson, O. E. (1979). "Transaction-Cost Economics: The Governance of Contractual Relations." Journal of Law and Economics(22): 233-262.