



Is Your Software Ecosystem in Danger? Preventing Ecosystem Death Through Lessons in Ecosystem Health

Robert Evertse, Abel Lencz, Tea Šinik, Slinger Jansen^(✉), and Lamia Soussi

Department of Computer Science, Utrecht University, Utrecht, The Netherlands
s.jansen@uu.nl

Abstract. The health of an ecosystem is by definition the most basic requirement for its survival. This paper aims to examine the driving forces behind the health of software ecosystems, in a comparative manner between four different ecosystems which have experienced a major downfall. We examine these ecosystems for similarities, from which demise principles are derived. Consequently, countermeasures are proposed in an attempt to combat these demise principles. The findings show that the main demise principles are *Underestimation of competitors*, *Lack of innovation*, and *Incorrect management of the ecosystem*. The proposed countermeasures to address these demise principles are to *Increase market awareness to increase competitive advantage*, *Increase product quality*, *Increase platform quality*, *Adjust value propositions* and *Formulate a partner-oriented strategy*.

Keywords: Software ecosystem(s) · End-of-life · Software ecosystem demise · Demise principles · Demise countermeasures

1 Introduction

Within a networked environment, the actions of actors cannot be viewed in isolation. The actions have an effect on other parts of the network. A network of organisations can be found in software ecosystems (SECOs). The following definition that will be used throughout this paper: “A software ecosystem is a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts” [9]. A SECO is beneficial to all its stakeholders; as customers gain new functionalities, software developers generate revenue and enable lucrative network effects for the SECO’s owner. Hence, with the rise and fall of SECOs as more companies adapt their businesses, understanding the mechanism behind the health of such ecosystems has never been so crucial.

Success is not self-evident, as the health of SECOs is volatile to change. There are examples of multiple SECOs that have not managed to survive. Soussi [14]

defines software ecosystem death as: “A permanent termination of an entity due to a disturbance in the dynamic between actors where collaborations and links are no longer occurring”. Soussi then proposes that the health and death of an ecosystem are influenced by the actors themselves and collaborations between actors. SECO health has been researched earlier [5,6,8,10–12]. E.g., three critical measures for assessing an ecosystem’s health are its productivity, robustness, and niche creation [6]. In addition, the main components in the Software Ecosystem Health Framework are actors, software, and orchestration [12]. While current research is mostly focused on measuring SECO health, the academic forum has not yet been able to generalise the flaws in the foundation on which deceased ecosystems were built. The question arises as to which extent the lack of these factors contributes to the demise of SECOS. The goal of this research is to identify elements that negatively affect the health of SECOS. These elements are derived and generalised from examples of discontinued ecosystems, leading to the following research question: ***How can the demise of software ecosystems be prevented?*** The relevance of this research lies foremost in finding explanations of how several SECOS did not manage to survive. In addition, insights are provided on how to address the identified demise principles.

In Sect. 2, the focus lies on four SECOS that have all ceased to exist: BlackBerry OS, Windows Phone, Symbian and Palm OS. For each of the ecosystems, a general outline is created to seek which contributing factors have led them to their demise. The outline is constructed based on the obituaries and eulogies of the SECOS. Year-end reports, blog posts, and other non-academic sources have been used to substantiate the narrative. Consequently, the main contributing factors of the demise is listed per SECO. In Sect. 3, the contributing factors of the demise are specified as well as the countermeasures which SECOS can incorporate as guidelines to not face the same fate. Lastly, in Sect. 4 the main research question will be answered whilst paying attention to the validity of this research and suggestions for future research.

2 Four Case Studies of Demised Software Ecosystems

This section is dedicated to BlackBerry, Windows Phone, Symbian and Palm OS. These cases have been specifically picked due to the similarities shared by the SECOS and to fit a case selection criteria. Firstly, the SECOS were handheld operating systems to enable a simpler comparison of demise factors later on. Secondly, the SECOS have had to achieve a reasonable level of success in the handheld market before experiencing a major decay in their ecosystem health. This is to ensure the availability of data. Lastly, the SECOS have to be end-of-life through official discontinuation.

2.1 Blackberry

BlackBerry Limited, formerly Research in Motion (RIM) used to produce pagers, smartphones and tablets. In 2013, they changed their name to BlackBerry

Limited, hereinafter referred to as BlackBerry. Currently, BlackBerry is a software company specialised in enterprise software.

Background - BlackBerry OS was first used for the BlackBerry 850 two-way pager in 1999. RIM managed to sell multiple phones with a physical keyboard, used mainly for business purposes. The keyboard was considered to be useful for sending e-mails and staying connected. RIM's peak worldwide market share was around 20%.

BlackBerry World reached 60.000 available apps in February 2012. This increased to 90.000 in August of 2012. The estimation is that BlackBerry World reached more than 200.000 available apps. The question arises how such a leading manufacturer did not maintain its market position. According to their respective year-end reports¹, from 2010 onwards the share of handheld revenue of RIM's total revenue started declining. Although the sales show an increasing amount until 2011, RIM has put more focus to their other businesses. Currently, there are several BlackBerry smartphones being sold with Android as their operating system.

Inadequacy 1: Competition with Apple - There are indicators that the rise of Apple and their iPhones was a main contributor to the demise of BlackBerry [13]. Being screen-only devices, they were distinctive from other mobile phones. *"They all have these keyboards, and they are there whether you need them or not to be there"*, said Steve Jobs explaining that about 40% of the size of the device was always occupied by the physical keyboard during the iPhone announcement in 2007. BlackBerry deemed Apple yet another competitor into the smartphone market, and was not considered a threat to RIM's core business [13]. We can conclude that one of the inadequacies of the SECO lies in its attitude towards the market. Its underestimation of Apple has played a major role in BlackBerry's demise.

Inadequacy 2: Innovation - BlackBerry adapted the use of a touch screen fairly late. They considered the touch screen to be inferior to a capacitive keyboard and stuck to their familiar design. As years passed by, Apple grew bigger and bigger and even surpassed BlackBerry's sales with the release after the iPhone 4 in 2010. BlackBerry did not timely compete with innovations such as the front-facing camera. By disregarding innovative features for their devices, we can state that innovation was one of the factors why customers switched to different operating systems.

2.2 Windows Phone

The Windows Phone can be seen as the successor of Windows Mobile, and was released on November 8, 2010. It can be considered as a reboot of the Windows Mobile, which used the Windows CE kernel, dedicated to devices with minimal memory. Windows Mobile was first branded as Pocket PC, but from 2003 on, Windows Mobile was the denominator. It was possible to install messaging services, streaming capabilities and even versions of the Office Suite.

¹ <https://www.annualreports.com/Company/BlackBerry-Ltd>.

Background - Whereas Windows Mobile was more focused to the enterprise market, the Windows Phone had consumers as a target audience. Four versions have been released, namely Windows Phone 7, 8, 8.1 and Windows 10 Mobile. Samsung, Sony Ericsson and LG have produced devices running on Windows Phone, among many other manufacturers. The most notable manufacturer was Nokia, which has led to a partnership between Nokia and Microsoft in 2011. Windows Phone became the primary OS for Nokia devices, which were mostly running on Symbian. The goal of this partnership was to compete with Android and iOS, which were the major mobile SECOs. In 2013, Microsoft acquired Nokia's mobile phone division, aimed to accelerate the growth of Microsoft's share in the smartphone market. However, Microsoft gave up on Windows Mobile in 2017 due to its low market share.

The application store of Microsoft, Windows Phone Store, reached the 100.000 available applications after 20 months in June 2012. It had already surpassed BlackBerry, but was still way behind iOS and Android. They reached the 300.000 milestone in June 2014, whilst Apple's App Store had surpassed the 1.2 million apps. Still, Windows Phone held a mere market share of 3%, which showed how far Microsoft was behind Google and Apple.

Inadequacy 1: Market Entry - Microsoft entered the smartphone market two years after Google's Android was released, and three years after Apple released their first iPhone. The market was starting to grow towards a duopoly, with BlackBerry and Symbian rapidly losing their market share. Customers and developers started committing to either iOS or Android, and their commitment was not as flexible as Microsoft assumed it would be.

Inadequacy 2: Ecosystem Configuration - Microsoft's late release caused developers to be already invested in iOS and Android. The developer interest was relatively low, and the revenue from advertisements was less compared to iOS, Android, BlackBerry and even Symbian, which did harm the enthusiasm of potential developers. This resulted in a lower amount of apps being developed. The appeal of Windows Mobile was not high, leading to a small user base. This also contributes to a lower developer appeal, which results in a vicious cycle. Subsequently, popular applications such as YouTube were not available.

Inadequacy 3: Dependencies - Microsoft adopted Apple's idea of a closed system, trying to maximise control over the developers. However, in the early stages they were dependent on other manufacturers for the hardware. Google's Android, which was released 2 years before, was also dependent on others to produce the hardware, but was a less closed platform. Microsoft's acquisition of Nokia's mobile devices division in an attempt to gain market share and break the duopoly, but this was without success. After writing off the assets, Microsoft discontinued Windows Mobile in 2017.

2.3 Symbian

Background - Symbian was established in 1998, as a partnership between Nokia, Ericsson, Sony, Motorola and Psion. Symbian hoped to achieve similar results as Android by opening up the source code, however this did not save the

platform. With the rise of Apple and Android, all (potential) sponsors stopped funding the Symbian Foundation and discontinued their partnership, leaving only Nokia to support the Symbian Foundation and its R&D department. In February 2010, the Symbian Foundation released their code, and shortly after that Nokia announced that it would completely phase out Symbian in February 2011.

Inadequacy 1: Incorrect Management of the Ecosystem - Symbian faced challenges whilst building their ecosystem; they attracted a number of potential ecosystem members who all had their own motives for joining. Symbian transferred its knowledge to partners in three ways: (1) personalized technical support, (2) codified documentation, and (3) Symbian's source code. Most partners only had access to a subset of the source code, because 76.67% of the ecosystem members were part of a competing (mobile phone) ecosystem. To battle these divided loyalties, Symbian introduced the 'refrigeration period'. During a six-month period, the ecosystem partners were prohibited from working for competing platforms. Brusoni and Prencipe [3] mentioned that due to loose coupling the entire ecosystem suffered from poor execution by one or multiple key partners.

Inadequacy 2: Complexity of the Architecture and Code - According to app developers, the problem lies within the user interface. The application developers were faced with several challenge². The first challenge was caused by the customisable design of the OS, the different available versions led to fragmentation, causing some software and apps to be incompatible. The second challenge was that the programming model used by Symbian (in particular memory management) was unlike the models used by iPhone OS, Android or Windows Mobile [9]. Lastly, the third challenge was a result of divided platform control between Symbian and the UI companies. This resulted in developers encountering in hick-ups whilst trying to find documentation.

Inadequacy 3: Weak Defensibility Due to Lack of Resources - In 2008, Symbian and its platform had attracted approximately 9,300 third-party software applications and had been shipped in almost 200 million phones. Even though Symbian managed to acquire a 49.3% of the worldwide smartphone market, it still suffered from great losses. This was caused by the enormous investments into R&D, which were estimated to be more than 200 million from 1998 to 2004. Symbian managed to achieve its first profit in 2005.

Symbian faced difficulties whilst trying to penetrate the North American market they only managed to acquire 3% of the smartphone sales. The entry was predominantly blocked by three local platforms: iPhone (2007), Android (2008) and BlackBerry (2002). Furthermore, the lack of financial resources limited the continuation of projects. An example is that Symbian decided not to proceed in creating its own app store in 2005, which is three years before Apple released the iPhone App Store.

² The complexity of the code, increased the average time spent on writing code and thus increased the costs. Development took Nokia approximately 22 months compared to the 12 months it took Windows Phone.

2.4 Palm OS

Background - Out of the four different operating systems examined in this research, Palm OS is the first one to be highly commercially successful in the handheld market [1]. The initial success in the PDA market of Palm OS can be attributed to its user-friendliness and ease of synchronisation with desktop computer. The company created a device and OS that complimented PCs instead of attempting to replace them [4].

Inadequacy 1: Management - Palm Inc. spent \$30 million to purchase the Palm trademark and another \$44 million for the keystone Palm software. Whilst trying to ease the split between the product's hardware and software components, the company was not able to remain innovative next to the shifting market trends and the increasing pressure from competitors such as Blackberry or Windows Mobile.

After the failed product launch of HP Touchpad (running webOS), the Palm hardware division was discontinued. As a last effort, the webOS software was made open source but it did not halt the mass departure of key engineers working for Palm. Without a strong backbone of a software team, webOS lost its strategic heading.

Inadequacy 2: Innovation - PDA market sank quicker than Palm anticipated as the first smartphones were already sold in the early 2000s. These advanced mobile phones incorporated the same features that PDAs had. By the time Palm reacted to the changing market trends by adding voice and improved data capability to their new devices, the damage could be seen in its market share, dropping from 10% in 2005 to 5% in 2006. Palm was unable to transfer their success from PDAs to smartphones. Palm OS 5 was released in 2002 and was shipped with Palm devices until 2007. During these five years, the OS was drained, receiving no major updates and rapidly aged in terms of design and functionality compared to other operating systems. In addition to being outdated, the Palm OS 5 was suffering from software issues. Palm was in a vicious circle, dropping sales resulted in less revenue and hence lower budget for R&D, however without an up-to-date OS it could not claim a major position in the growing smartphone market.

Inadequacy 3: Ecosystem - Several OEMs (Original Equipment Manufacturer) sold Palm OS based devices however Palm itself viewed the other companies as direct competitors who will reduce their revenue, and hence did not manage to establish long-term and healthy relationships. Furthermore, Android devices exploded in the market in 2009 with 50 different devices sold the first year, made popular among OEMs due to its open-source license. In comparison, Palm OS with its proprietary license was virtually non-existent among OEMs. Palm Pre looked promising however HP was only able to sell the phone initially through Sprint. This hurt sales of the phone as Sprint was third in the phone carrier market and was financially struggling at that time. HP branded Palm Pre Plus also had issues with carriers, namely Verizon refusing a shipment and the negative advertising the device received. The applications for webOS were

extremely limited in comparison to Apple or Android. A lack of applications led to fewer users adopting the platform, and fewer users meant that fewer developers produced applications.

Table 1 shows an overview of the SECO characteristics and the applicable demise principles derived from the narrative. Notably, the demise principles do not stand on their own, seeing as their presence enforces one another.

Table 1. SECO overview

	BlackBerry	Windows Phone	Symbian	PalmOS
OS Birth year	1999	2010	1998	1996
OS Latest update	2018	2020	2012	2007
Estimated peak worldwide smartphone market share	22% (2009)	4% (2012)	73% (2006)	10% (2005)
Competitor underestimation	✓	✓		✓
Late market entry		✓		
Lack of innovation	✓			✓
Incorrect management of the ecosystem		✓	✓	✓
Complexity of the architecture and code			✓	
Lack of resources			✓	
Unsupportive partnerships				✓

3 Interpretation: Demise Principles and Countermeasures

Section 2 consists of a detailed research of the four SECOs regarding their background, weak points and the contributing factors which had led to their demise. For each SECO the contributing factors of the demise are specified as well as the countermeasures which SECOs can utilize as a guideline to not face the same fate.

As illustrated earlier, there are some commonalities between the different SECOs and how they did not manage to survive which is listed in Table 2. The commonalities are phrased as demise principles, which are derived from the sources of undesired situations which these SECOs have encountered. Similar sources have been deduced to general SECO characteristics.

We conclude that the different demise principles are not standing on their own, as the demise cannot be attributed to one sole factor. Some factors are tied to each other or can be consequential of one another, as competitor underestimation is some sort of market underestimation, which may result in lower sales and therefore may cause financial issues.

Table 2. Contributing factors to the demise of the observed SECOs.

Demise principles	Countermeasures
Competitor underestimation	Increase market awareness to increase competitive advantage
Late market entry	Increase market awareness to increase competitive advantage
Lack of innovation	Increase product quality & Adjust value proposition
Incorrect management of the ecosystem (including incorrect ecosystem configuration)	Increase product quality & Increase platform quality & Formulate a partner-oriented strategy
Complexity of the architecture and code	Increase product quality & Formulate a partner-oriented strategy
Lack of resources (resulting in a weak defensibility)	Formulate a partner-oriented strategy
Unsupportive partnerships	Formulate a partner-oriented strategy

Formulate a Partner-oriented Strategy - The partner ecosystems are responsible for a significant share of value creation within the ecosystem. There are several ways to examine the partner ecosystem. Lessons can be learned from Avila and Terzidis [2] and Jansen [7] among others. It is important to carefully assess partners, their health and network.

Increase Product Quality - This research illustrated that some SECOs have designed their OS to be too customisable, which led to fragmentation and resulted in incompatible software and apps. Secondly, due to the complexity of the code, the SECOs would spend too much of their resources such as manpower (time) and finances, resulting in a lack of available resources to be allocated for innovation. Possible actions could be to (1) initiate code checks, (2) highlight the bugs and crashes of the code after release updates, (3) evaluations of the accessibility, compatibility and maintainability of the code and User Interface and (4) make use of the available internal and external developer community. By making use of websites such as GitHub and Stack Overflow, the ecosystem can easily monitor which questions are being raised. To increase the community feeling within the developers, SECOs could organize workshops, hackathons and seminars. This will not only promote your product, but it may also lead to innovation which may decrease the development time.

Increase Market Awareness to Increase Competitive Advantages - During the “birth” and whilst growing the ecosystem, it is of importance to perform a market analysis in which the competition is researched. The ecosystem should be constantly aware of its competitors and of their innovations, in order for them to be able to respond promptly and to not lose its customer base or market share. Business and market oriented research has provided several methods to perform market analysis.

Increase Platform Quality - Developers can either make or break a platform. This should force the platforms to create an attractive environment for the developers. There are several ways for platforms to build their developer program. Organisations should pay attention to their API, SDK, documentation among other features. Developer Economics, the largest developer research program across the globe states that platforms could improve the quality of their platform by opting for: access to rich APIs and features, community support, ease of coding and prototyping, low cost development, familiar development environment, revenue potential, good documentation and tech support and large installed base of devices.

Adjust Value Proposition - To address customer needs, one should carefully look at the value ones product portfolio has. USPs (unique selling propositions) are ways of a product to differentiate from other substantiating and competing products. Customer needs are flexible, and change over time. Ideally, an organisation should therefore adapt to the changing customer needs and ensure that the products are continuously developing and adapting to contemporary wishes.

4 Conclusion and Future Work

By analysing four SECOs involved in handheld operating systems which have experienced a downfall, demise principles have been identified. These demise principles with their respective countermeasures are displayed in Table 2. Ideally, by correctly applying these countermeasures, SECOs should be able to thrive.

The four investigated SECOs were similar in nature and market, being active handheld operating systems with a reasonable level of (potential) success. The question arises whether the handheld operating systems market can be compared to other markets targeted by SECOs, and whether the different demise principles and countermeasures are applicable to those. Whether the identified demise principles and countermeasures are correct remains uncertain. They may not be an accurate representation on their own, as they reinforce one another. In addition, other markets may show different demise principles, which could require different countermeasures to correctly solve them.

The authors' recommendation lies foremost in expanding this research, both in breadth as in depth. Including other categories of SECOs could further discover the drive behind SECO demise, and in-depth interviews could help substantiate the claims made in this research. Additionally, a study of a healthy ecosystem where the absence of these demises is observed, could further enforce our hypothesis. Lastly, the proposed countermeasures are hypothetical, and it is important to identify whether the proposed countermeasures are effective. The countermeasures could be applied in unhealthy ecosystems to turn the tide.

References

1. Allen, J.P.: Redefining the network: enrollment strategies in the PDA industry. *Inf. Technol. People* **17**(2), 171–185 (2004)

2. Avila, A., Terzidis, O.: Management of partner ecosystems in the enterprise software industry. In: International Workshop on Software Ecosystems (IWSECO), pp. 39–55 (2016)
3. Brusoni, S., Prencipe, A.: The organization of innovation in ecosystems: problem framing, problem solving, and patterns of coupling. In: Collaboration and Competition in Business Ecosystems. Emerald Group Publishing Limited (2013)
4. Butter, A., Pogue, D.: Piloting Palm: The Inside Story of Palm, Handspring, and the Birth of the Billion-Dollar Handheld Industry. Wiley, Hoboken (2002)
5. Iansiti, M., Levien, R.: The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability. Harvard Business Press, Cambridge (2004)
6. Iansiti, M., Levien, R.: Strategy as ecology. *Harvard Bus. Rev.* **82**(3), 68–78 (2004)
7. Jansen, S.: A focus area maturity model for software ecosystem governance. *Inf. Softw. Technol.* **118**, 106219 (2020)
8. Jansen, S., Cusumano, M.A.: Defining software ecosystems: a survey of software platforms and business network governance. In: Software Ecosystems. Edward Elgar Publishing (2013)
9. Jansen, S., Cusumano, M.A., Brinkkemper, S.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishing, Cheltenham (2013)
10. Mageau, M.T.: The development and initial testing of a quantitative assessment of ecosystem health. *Ecosyst. Health* **1**, 201–213 (1995)
11. Manikas, K., Hansen, K.M.: Reviewing the health of software ecosystems—a conceptual framework proposal. In: Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO), pp. 33–44. Citeseer (2013)
12. Manikas, K., Hansen, K.M.: Software ecosystems—a systematic literature review. *J. Syst. Softw.* **86**(5), 1294–1306 (2013)
13. McNish, J.: *Losing the Signal : The Untold Story Behind the Extraordinary Rise and Spectacular Fall of Blackberry*, 1st edn. Flatiron Books, New York (2015)
14. Soussi, L.: Health vulnerabilities in software ecosystems: five cases of dying platforms. Master’s Thesis, Utrecht University (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

