

WE ARE HAPPY to introduce to you the “Managing Software Platforms and Ecosystems” special issue of *IEEE Software*. This special issue is characterized by contributions from diverse theoretical fields, such as computer science, information science, and economics, with research applied in practical settings in software-intensive business. It is an overview of the current state of research and illustrates some of the largest innovations in the field as well as the gaps for future research.

Software Ecosystems

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. Software ecosystems are

other to enable faster adoption of new technologies supplied by new entrants in the market.

Society, organizations, and economies are experiencing and anticipating fundamental changes that are shaped, embedded, and influenced by ecosystems. Ecosystems are social, technical, and economic systems that are large, multilevel, complex, dynamic, adaptive, emergent, and global in nature, and they concern a wide range of stakeholders (managers, policy makers, and society), each with different perspectives and incentives. Ecosystems cannot be created; they must be cultivated and fostered.

The complex scale, scope, and dynamics make systematic modeling, analysis, engineering, and management

and we have had to be stricter than we would have liked. An interesting phenomenon is the wide range of backgrounds of the submitters. We received work about open source from software engineers, work concerning automotive platforms from management and information scientists, and work regarding visualizations from computer scientists. We reiterate that this domain is relevant from different perspectives and, thus, should be considered multidisciplinary.

During a recent meeting at Dagstuhl Castle in Germany, “Software Business, Platforms, and Ecosystems: Fundamentals of Software Production Research,” a research agenda was defined. The outlines of that research agenda frame the contributions from the community to this special issue.

Engineering of Ecosystems

As previously stated, ecosystems must be cultivated; that is, they must be grown to provide the power that keystones source from their ecosystems. We find that enabling technologies, such as plug-in architectures, App Store architectures, and application programming interface (API) architectures, create the infrastructure that enables partners to cocreate and innovate within an ecosystem. Our first set of challenges centers around enabling partners to engage in the ecosystem through technical infrastructure. There are many enabling technologies for ecosystems; the first that come to mind are App Stores, plug-in architectures, and blockchains, but there are many more. To mobilize developers, for instance, there is a need for code repositories, sandboxes, integrated development environment combinations of ecosystem resources, license protection mechanisms, testing platforms, and even full development stacks. There is a need for

In this special issue, we demonstrate that research in the domain of software ecosystems is very active.

pervasive, and software-producing organizations are realizing that it is the ecosystem that makes them and their technologies successful.¹

Digital business has become an essential pillar under most economies and has driven innovations for several decades. The introduction of new technologies and the convergence of the Internet of Things, cloud technologies, and artificial intelligence have led to myriad new possibilities, but they require an ecosystem approach for extensive adoption. These technologies are rapidly adopted, in large part due to the “ecosystemification” of the digital business; software-producing organizations depend on each

challenging. Multidisciplinary perspectives in research, such as views from computer science, economics, management, information systems, innovation sciences, engineering, and policy, are required. Given its economic and societal relevance, successful ecosystem research requires collaboration by scholars, practitioners, and individuals. The value and impact of engineered ecosystems are manifested through mobilization, participation, and facilitated collaboration, which enable growth, innovation, and welfare.

In this special issue, we demonstrate that research in the domain of software ecosystems is very active. Submissions were very competitive,

incentive systems, partner quality-monitoring systems, and API performance monitoring to observe and enable partners. These systems need to be studied more extensively to establish how they contribute to the ecosystem, developer satisfaction, and overall partner performance.

Second, there appear to be recipes for enticing and attracting new partners into an ecosystem and lowering their barriers to entry; however, these recipes have not been inventoried sufficiently. We need to study ways to keep thresholds low, create gravity, and use network effects for the incremental growth of the ecosystem. The precise interactions between these recipes and ecosystem health over time are currently not known and require more research.

Analysis of Ecosystem Data

When performing repository mining for the study of an ecosystem, it quickly becomes apparent that the study of ecosystems is effectively a big data exercise. Studying a repository, such as Github, is often compared to drinking water from a fire hose, especially when a research project is focused on the needle in the Github haystack. There are several challenges in software ecosystem research that remain to be tackled.

First, there is the challenge that most researchers typically limit themselves to a snapshot of one or several ecosystems. However, to answer some of the deeper questions on ecosystem health and attracting new developers and partners to an ecosystem, concrete recipes need to be evaluated in terms of metrics that can be gathered from repositories over time. A large challenge here is that, in software ecosystems, data are generally hard to collect. Data are of different types, hidden behind organizational barriers, and sometimes


simply overwritten and unavailable. Monitoring ecosystems over time has become a challenge that requires extensive and long-lasting efforts.

Second, the concept of ecosystem health, that is, the propensity for growth of an ecosystem, has been extensively studied. These frameworks have become increasingly elaborate and comprehensive, thereby making it challenging to apply them to a research project. Some even call for a customized set of health metrics for every (type of) ecosystem. More research is required into the main performance indicators of ecosystems, the recipes that aim to influence them, and their measurable influence on these performance indicators.

A second challenge is that the current languages do not scale upward easily. Ecosystems with up to five actors can still be modeled in goal-modeling languages and power models, but beyond those numbers, these models become too complex. Finally, even with such models, it becomes complex to monitor and model ecosystems over time.

Management of Developer Ecosystems and Platforms

Decisions to join ecosystems are made on both a strategic level and an operational level by senior software engineers. Some have called these software engineers *king makers*, because these



Monitoring ecosystems over time has become a challenge that requires extensive and long-lasting efforts.

Modeling of Ecosystem Structure and Behavior

Ecosystems are increasingly used as tools for reasoning about an organization's business model,² market position, opportunities, and threats. However, we have not been able to reason at the highest levels of fidelity. The development of modeling languages that provide insight and enable analysis at different levels of scale is needed. There are several modeling languages used in the field, such as social-network, goal, and supply-chain models. These models appear to have significant overlap, as they all aim at model actors, software structures, and relationships, yet each serves a different purpose.

decisions may lead to long-lasting relationships with the technical platform and the keystone organization that supports it.³

Software-producing organizations call the groups of software engineers in their software ecosystems *developer ecosystems*. Managing developer ecosystems can challenge software-producing organizations in four different ways. First, the platform that the developer ecosystem focuses on needs to be extensible, flexible, robust, and evolvable, and it must provide facilities for the rapid development of new solutions. Second, the developer community must be managed, by organizing events, coordinating feedback, helping developers support each other,

and so on. Third, the software-producing organization must be ready to accommodate developers, by readily providing them with easy access to the platform as well as to support, knowledge, and advice. Finally, the organization needs to keep track of other ecosystems and the role of open source in the platform, and it should invest in supporting platforms and ecosystems.

Even though the field of software ecosystem management and software ecosystem governance is rapidly maturing, many organizations are still reinventing tools and methods for becoming stronger in a software ecosystem. There is little usable knowledge about quickly implementable processes and practices for organizing an ecosystem. Research into the mecha-

of the future research areas. There are three contributions in the domain of software ecosystem engineering. First, a recurring theme in developer ecosystem research is what motivates and mobilizes developers to contribute to open source projects. The article “Are Donation Badges Appealing?” demonstrates how the use of a simple reward system (badges) can reduce the time developers take to respond to bug reports in an open source project. Second, “Software Development Analytics for Xen: Why and How” reports on how the use of a popular virtualization technology in code reviews can improve decision making and coordination among ecosystem participants.

Third, there are invisible yet existing supply chains among software compa-

ecosystems. Because there is no education targeted at training partner managers, companies hiring partner managers have a hard time determining which competencies besides experience are crucial success in this activity. In “Successfully Governing Software Ecosystems,” the authors identify two very different types of relationships between platform owners and complementors (arm’s length versus dyadic governance) and argue that it is the responsibility of platform owners to recruit and train partnership managers for one type of relationship management or the other.

Second, new ecosystems are emerging in traditional sectors, such as the automobile industry, which is experiencing an increasing reliance on software-controlled hardware systems as well as discrete software programs. Digital innovators in this sector face specific challenges in dealing with a small number of powerful manufacturers as well as government standards and regulations. “Strategies for Competing in the Automotive Industry’s Software Ecosystem” uses a case-study approach to discuss three strategies for dealing with potential obstacles to data collection and analysis as well as to software innovation itself. Software producers can actively cooperate with automakers, or they can use open standards to circumvent control by the manufacturers (as Robert Bosch has done) or produce cars themselves (as Tesla has done).

There are two contributions in the domains of software ecosystem modeling and analysis. A large body of research exists for the human mechanics of power and dependencies in different societies. Software ecosystems are gatherings of people in a certain, predefined setting, with defined roles and different levels of power as well as interdependencies. There is an obvious need to drill deeper into the

It is still unclear how ecosystems start, grow, develop, die out, and renew again.

nisms that entice, attract, keep, and lock in developers is needed.

These mechanisms range from tools for knowledge sharing, such as joint repositories and API documentation, to release coordination, where release candidates are released to partners early to enable them to do compatibility checking of their extensions. Another concept that is insufficiently studied is the role of dominant design and standardization in industries. Finally, ecosystem lifecycles have not yet been researched comprehensively, and it is still unclear how ecosystems start, grow, develop, die out, and renew again.

In This Issue

The seven articles in this special issue address at least one, and often more,

nisms that provide software components to each other. Although this allows a vendor to concentrate on differentiating parts of the created software and use third-party components for nondifferentiating parts, it also creates a dependency. “Securing Software Ecosystem Architectures” illustrates how an open architecture model covering software sourcing, integration, deployment, and evolution can improve security throughout the software supply chain.

There are two contributions in the domain of software ecosystem management. Although the focus of very large software ecosystems is often centered around technology and APIs, many smaller ecosystems still use partner managers to entice, engage, and guide partners to success in software



SLINGER JANSEN is an assistant professor in the Department of Information and Computer Science at Utrecht University. His research interests include software ecosystems, and he is a cofounder of the International Conference on Software Business and International Workshop on Software Ecosystems. He is the lead editor of *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry* and several other books. He actively supports new enterprises and sits on the boards of advisors of several start-ups. Contact him at slinger@slingerjansen.nl.



MICHAEL CUSUMANO is the Sloan Management Review Distinguished Professor of Management at the Massachusetts Institute of Technology's Sloan School of Management. His research interests include strategy, product development, and entrepreneurship in computer software as well as automobiles and consumer electronics. He received a Ph.D. from Harvard University in Japanese studies and completed a postdoctoral fellowship at Harvard Business School in production and operations management. He recently taught the courses Software and Internet Entrepreneurship and Advanced Strategic Management. He has published 14 books and more than 120 articles and columns. His latest book is *The Business of Platforms: Strategy in the Age of Digital Competition, Innovation, and Power*, with Annabelle Gawer and David Yoffie (Harper Business, 2019). He is a member of the ACM. Contact him at cusumano@mit.edu.



KARL MICHAEL POPP is a senior director in the Global Business Development and Ecosystem team at SAP SE. His research interests include strategic partnerships, mergers, acquisitions, and postmerger integration. Popp received a Ph.D. in information systems from the University of Bamberg. He serves on the program committee of the International Conference on the Software Business, the European Workshop on Software Ecosystems, and the European Platform Economy Summit. He is a senior member of the ACM and a Member of the IEEE and the German Computer Society. Contact him at karl.michael.popp@sap.com.

smaller or less influential participants, all of whom need each other to be successful. Second, “Analyzing and Managing Complex Software Ecosystems” presents a visual representation scheme to help ecosystem participants, such as in an open source software project, understand their shared goals, analyze relationships within the value network, and align their individual business strategies and contributions.

We hope that this special issue and the presented research agenda outline inspire you to include the ecosystems perspective in your daily work, whether you are a researcher, engineer, or both. We thank everyone who did a great job writing or reviewing the high-quality submissions for this theme issue. Also, we thank Paul Grünbacher, Jens Förderer, Zhi Jin, Samuel Fricker, and Rahul Basole for creating the research agenda with us. Finally, we thank *IEEE Software's* Editor in Chief Emeritus Diomidis Spinellis and magazine staff for their guidance and support. Happy reading! 📖

References

1. S. Jansen, M. Cusumano, and S. Brinkkemper, Eds., *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, U.K.: Edward Elgar Publishers, 2013.
2. R. Meyer and K. M. Popp, *Profit From Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. Norderstedt, Germany: Books on Demand, 2010.
3. S. O'Grady, *The New Kingmakers: How Developers Conquered the World*. Sebastopol, CA: O'Reilly Media, 2013.

roles of power and dependence in software ecosystems. The article “Exercising Power in Software Ecosystems” argues that participants need to

understand how power is distributed and used in ecosystems, where there often are a small number of large, powerful keystone players, along with