The term business ecosystem was originally established more than 20 years ago. In the seminal article “Predators and prey: a new ecology of competition” [1], Moore proposes an ecological approach to explain the co-evolution of companies within a business ecosystem by adopting a complex interplay of cooperative and competitive strategies. According to Moore, business ecosystems evolve in four stages: birth, expansion, leadership, self-renewal (or eventual death). The success of a company increasingly depends on the healthy co-evolution of other participants of the ecosystem. Thus, the concept of business ecosystem suggests that companies are engaged in interdependent relationships and strongly rely on each other for their mutual survival. Iansiti and Levien [2] also propose biological concepts to explain the health of business ecosystems by means of productivity, robustness and niche creation measures. The authors describe three types of actors within a business ecosystem: dominator, keystone and niche creation. The keystone organization plays a critical role in the ecosystem by creating and sharing value with other participants in the ecosystem.

Software ecosystems can be considered a particular type of business ecosystems, where interactions among participants are performed around a shared technological platform. The software ecosystem research community has adopted several concepts and analogies from business and natural ecosystems fields. Different definitions of Software Ecosystem have been proposed. One of the oldest definitions is presented by Messerschmitt and Szyaperski [3]: “a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships.” Jansen et al. [4] have defined a software ecosystem as: “a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them.” Another definition by Bosch [5] is: “the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions.” A common characteristic of all definitions is the relevance of relationships among participants in a software ecosystem. These relationships frequently have different nature and strength. The diversity of relationships supports the ecosystem health and enables the creation of innovative business models.

Early work on software ecosystems was focused on the platform company and the models it could use to entice third party developers to contribute to its platform. However, software ecosystems are much broader. Modern software intensive systems companies depend on dozens and sometimes hundreds of suppliers, complementors and integrators in their surroundings. As these companies have successfully adopted agile work practices in their own R&D organizations, the next challenge becomes the relationship with all the parties in their ecosystem. In order to fully reap the benefits of agile, customer-oriented development, these companies need to accomplish the same transformation with their ecosystem partners and hence the perspective is changing from intra-organizational to inter-organizational. This results in a much stronger focus on software ecosystems as the implications of the aforementioned are predominantly concerned with the organization of software R&D.

Successful examples of effective collaboration around software R&D between organizations exist today, such as the open-source software communities and the apps in the mobile ecosystems. However, the vast majority of cases require different IP and business models, not provided by open source, and much deeper integration between the software solutions provided by ecosystem parties than what can be provided in a mobile app model. Consequently, software ecosystem research is still in its early stages and requires significant attention and investment from the research community before the software intensive systems industry will be able to fully benefit from the advantages of an effective software ecosystem.

The seven papers in this special issue cover a great variability in topics related to software ecosystems, from analyzing and understanding how they function to how software ecosystems contrast and develop our traditional ideas on how software is being developed and maintained. Empiricism from various domains such as embedded systems, telemedicine systems and open-source projects are used to mature and illustrate important concepts. In total, these seven papers add to a growing body of research and our understanding of an important concept that will influence the development of the software engineering discipline.

The first paper by Koch and Kerschbaum investigates independent developers’ motivations for selecting and joining smartphone ecosystems. Differences are found between ecosystems, e.g. iOS developers are more motivated by opportunities of financial gain. However, the main motivations for developers are intellectual stimulation and the ability to learn new skills. The authors suggest that the perceptions of application developers and their motivation to contribute are important factors to consider when managing large and diverse ecosystems.

Dittrich discusses how the emergence of software ecosystems challenges our traditional view of the project form for developing and maintaining software products. Based on an investigation of four product ecosystems, the author proposes directions on how design is distributed and needs to be coordinated across heterogeneous design constituencies.

Focusing on embedded systems, Axelsson, Papatheocharous and Andersson describes how development of embedded systems transforms into, what they define, as federated embedded systems,
benefiting from characteristics of software ecosystems that enable third-party actors to take a role in development of products, which traditionally has been closed to external roles.

Christensen, Hansen, Kyng and Manikas propose the concept of software ecosystem architectures as a tool to analyze existing software ecosystems and to design new ones. The proposed concepts are demonstrated through a case study of a telemedicine ecosystem.

In order to understand motives for involvement in ecosystems, Wnuk, Runeson, Lantz and Weijden discusses bridges and barriers to participate in ecosystems with substantial hardware dependencies. Findings from industry interviews suggest that ecosystem leaders benefit from sharing sales channels and that they should strongly emphasis rich communication. An inefficient business model can be a considerable barrier for participation in an ecosystem.

Jansen develops the concept of software ecosystem health as a means to support decisions about taking a role in an ecosystem or not. An analysis of four open source ecosystems is used to demonstrate and develop the concept.

The last paper in the special issue, by Berger, Pfeiffer, Tartler, Dienst, Czarnecki, Wasowski and She develops the concept of variability mechanisms – well known from product line development – to software ecosystems. The authors develop a conceptual framework based on an analysis of five open-source ecosystems.

Software ecosystems as a research field have been gaining increasing interest over the last decade. Several studies have investigated software ecosystems from multiple perspectives, such as business, technology and social [6]. In this special issue we see a growing understanding of how software systems function and how this concept challenge the established view of the software engineering practice. Based on recent work, including this special issue, we see a need for continuous research that even more seeks to learn from industrial experience and from thorough longitudinal studies to understand the complexities and opportunities of software ecosystems. Eventually, the goal for the research community should be to provide industry with empirically grounded knowledge and guidelines on how to reap the benefits of participating in ecosystems with a variety of actors.

References

Geir K. Hanssen
SINTEF ICT, Norway
E-mail address: ghanssen@sintef.no

Carina Frota Alves
Federal University of Pernambuco, Brazil
E-mail address: carinafrota@gmail.com

Jan Bosch
Chalmers University of Technology, Sweden
E-mail address: jan@janbosch.com

Available online 7 July 2014