Hadoop and its evolving ecosystem

J. Yates Monteith, John D. McGregor, and John E. Ingram

School of Computing
Clemson University
{jymonte,johnmc,jei}@clemson.edu

Abstract. Socio-technical ecosystems are living organisms that grow and shrink, that change velocity, and that split from, or merge with, others. The ecosystems that surround producers of software-intensive products exhibit all of these behaviors. We report on the start of a longitudinal study of the evolution of the Hadoop ecosystem, take a look back over the history of the ecosystem, and describe how we will be observing this ecosystem over the next few months. Our initial observations of the early days of Hadoop's ecosystem showed rapid change. We present these observations and a method for taking and analyzing observations in the future. Our goal is to develop an ecosystem modeling technique that provides practical guidance to strategic decision makers.

1 Introduction

Socio-technical ecosystems are living organisms that grow and shrink, that change velocity, and that split from, or merge with, others. Recently researchers have found it useful to describe the environment surrounding certain software platform-based communities in ecosystem terms. Many of those descriptions focus on the mutual benefit derived from the platform. However, in trying to support strategic decision makers, the true predator-prey notion of an ecosystem, in which both collaborators and competitors interact, gives a comprehensive view of the ecosystem.

Business strategists and software architects both must balance opposing forces to achieve the best possible result for their organization. In an organization that builds software-intensive products the business and technical forces are closely related and interconnected. New business models, such as Platform as a Service (PaaS), require new architectures to accommodate collaborators and to separate that which is the basis for collaboration from that which is the basis for competition. Over time the line between these two shifts as more features become commoditized and organizations innovate to identify new proprietary features. These are the changes that motivate this work.

New algorithms and paradigms are often the basis for new communities and in the early days there is much activity as the forces of competition from established technologies clash with the enthusiasm for the new capabilities. This leads us to some interesting questions:

- How is the ecosystem surrounding a new technology different from that surrounding a mature, established technology?
- What influence does that difference have on the business decisions that must be made?
- Will the frequency and types of change show a different pattern as the technologies mature and the buzz words become accepted termnology?
- How do the linkages between the business and software aspects of the ecosystem respond to changes over time?

In 2004 researchers at Google published a new Map/Reduce algorithm for distributed computation. This algorithm has formed the nucleus of a new ecosystem for distributed computing, which is the focus of this paper.

As pointed out by Hannsen et al, there is a need for more detailed accounts of actual ecosystems and the changes they undergo over time [1]. A portion of our research time is spent tracking a few ecosystems and examining how they are changing. Some data is easy to identify, like major software changes indicated by version numbers; however, most useful data is difficult to identify and parse. Data including both motivations for and changes to code, business models, governance structures, collaborative and coopetitive alliances are all useful data points. By conducting longitudinal studies we have the opportunity to search for patterns in these changes and to anticipate their frequency and direction in the future.

Our current contribution is a baseline report on the Hadoop ecosystem. We apply STREAM, our ecosystem analysis method, to Hadoop distributions from the early releases to the present. We consider two dimensions. We describe portions of the value chain that relates suppliers to customers at the current point in time. We define data useful in evaluating where value is added. We also explore the evolutionary forces responsible for changing where value is accrued over time.

The remainder of this paper is organized as follows: Section 2 and Section 3 provides background information on STREAM; Section 4 describes the observations that were made; Section 5 presents the results collected from the observations; Section 6 provides a view into our future efforts, and Section 7 is a brief conclusion.

2 STREAM

The **STR**ategic **E**cosystem **A**nalysis **M**ethod (STREAM) [2] addresses the various facets of a socio-technical ecosystem which encompasses a community, usually associated through a common interest in a particular domain. STREAM presents the ecosystem through three types of views: business [3], software [4], and innovation [5], which correspond to the three types of ecosystems featured in the ecosystem literature. The business and software views represent the state of the ecosystem at any given moment. The innovation view shows the forces that will result in evolution.

Each application of STREAM is customized to answer specific questions. The exact data collected and the analysis methods applied will directly address those questions.

Each of the types of views has specific attributes, artifacts, and analysis techniques. We introduce each here and give more detail in the case study.

- Business view - The organizations in an ecosystem interact explicitly, e.g. trading partners, and implicitly, e.g. through pricing models. Michael Porter's Five Forces for Strategy Development model gives a structure to this view [6].
- Software view - The software architecture is the major structuring element for the software view. We do as detailed an analysis as possible with the level of architecture description that is available.
- Innovation view - The innovation view represents both business and software innovations. We organize those innovations according to Businessweek's categories of innovation: product, process, business model, and customer experience.

We use the structuring elements in each view to define appropriate abstractions and to guide collection of the data needed to instantiate them.

3 E-STREAM

STREAM as it originally was defined gives a snapshot of an ecosystem at a point in time [2] [7]. E-STREAM is an extension of STREAM that supports modeling the ecosystem's evolution by a combination of a series of snapshots, obtained through multiple applications of STREAM, with measures analyzing the changes in-between the snapshots. In section 4 we illustrate these extensions.

3.1 Ecosystem Evolution

The evolution of both organizations and software have been well studied but the evolution of ecosystems, particularly those that encompass software-intensive products, is not as well understood. Tiwana et al refer to evolution in an ecosystem as coevolution since both organizational and technical changes occur [8]. STREAM handles this coevolution naturally with its multiple views. Tiwana et al proposed a framework for studying evolution of a platform ecosystem that separates "internal" platform forces from "external" platform forces and separates the internal forces into platform governance and platform architecture. The dependency graphs we construct for both organizations and software modules represent this separation and, in fact, allow for multiple separations.

Hanssen et al have conducted a longitudinal study of the ecosystem surrounding CSoft [1] [9] [10] [11] [12]. They hypothesized a set of charactistics for software ecosystems which we will revisit in Section 5.

Evolution is essentially a time-based view of change. Since change often comes about as a result of innovation we have organized the rest of this section using the Businessweek categories of sources of change to discuss evolution. We use forward references into the case study in the next section to illustrate each category.

3.2 Product

Each new release of a software product offers a different value to customers. Some notable exceptions excluded, each release provides more value than the one before. A measure of this value can be seen by looking at the number of releases per year, number of downloads per year, or other measure of use. Our timeline shows releases per year, shown in Figure 1.

3.3 Process

A value chain is a model of the steps through which a product passes as it is created and the value that is added at each step. One paper has hypothesized a value chain for software in which the standard development life cycle phases are the steps in the value chain [13]. In the ecosystem a visible measure of change will be differences in mechanisms by which a product is assembled. For example, the project may begin to provide build processes or pre-configured distributions for targeted groups of developers or users, respectively. The Substitutes section of the Five Forces analysis in Section 4.1 list organizations providing users with improved processes for using Hadoop.

3.4 Business model

Business model changes usually occur as a result of a strategic decision to change directions. Open source projects may maintain repositories of minutes of the governing councils such as the architecture council or a project management committee. Commercial organizations often convert projects which have previously been proprietary to an open source project, e.g. Hadoop.

3.5 Customer experience

The customer experience is tied to the evolution of the business model and the product itself. Defect reports and change requests reflect customer issues and these can be tracked over time. Tools such as Jira allow all users to comment on issues. “Big Data” techniques can be used to mine information from the Jira logs.

4 Case study

Apache Hadoop is a scalable computing framework that abstracts away the issues of data distribution, scheduling, and fault tolerance from applications. Hadoop is a framework that is the core of a rapidly growing ecosystem in which a number of providers are building Platforms as a Service (PaaS) based on Hadoop.

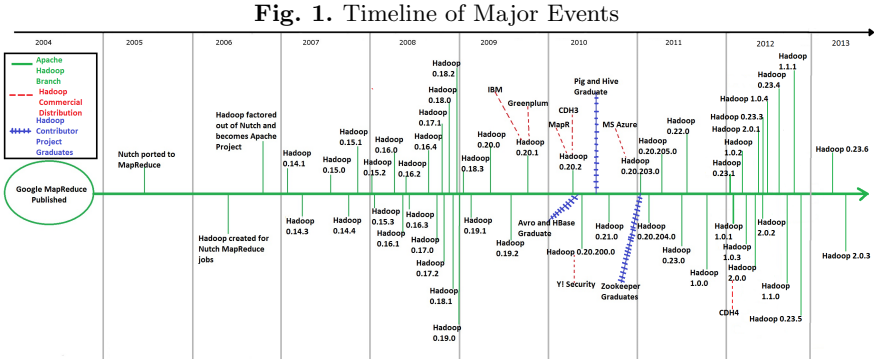
Hadoop utilizes an innovative approach called Map/Reduce intended for processing data collections that are so big that it is more efficient to move the computation to where the data is rather than vice versa. The user of the Map/Reduce

approach writes a Map program that divides the data and directs it to the set of computing nodes. The user then writes Reduce programs that accomplish the needed computation by first computing on each node and then taking the partial result from a node and combining it with the partial results from neighboring nodes to reduce iteratively down to a single answer.

We have developed a timeline, shown in Figure 1 (full color expandable figures can be found at http://www.cs.clemson.edu/sserg/iwseco/2013/), to capture some of the historical information we collected about the Hadoop ecosystem. In 2004 two formative papers were published by Google authors [14][15]. These papers defined the Google file system and Map/Reduce architecture, respectively. Hadoop was initially housed in the Nutch Apache project, but split off to become an independent Apache project in 2006. In January 2010, Google was granted a patent that covers the Map/Reduce algorithm. Three months later Google issued a license to the Apache Software Foundation. Since that time use of Hadoop has grown rapidly.

Over the last few years parts of the original Hadoop Apache project have matured and spun off to become independent Apache projects: Avro, HBase, Hive, Pig, Flume, Sqoop, Oozie, HCatalog and Zookeeper. These products are used with Hadoop depending upon the configuration and are focal parts of the ecosystem.

Fig. 1. Timeline of Major Events



4.1 Business view

At the core of the Hadoop ecosystem is the Apache Hadoop project which maintains the Map/Reduce framework and the Hadoop Distributed File System (HDFS). The project is governed by a project management committee (PMC) that is self-perpetuating and self-directing. Many of the members of the committee are from larger organizations that use the Hadoop distribution as part of strategic product offerings.

Figure 3 shows the network of organizations that contribute to the Hadoop Ecosystem and to which project they contribute. Triangular nodes represent or-

ganizations that contribute personnel to the PMC or committers. The nodes are sized by how many personnel are contributed by the organization to all projects combined. Circular nodes represent the "Hadoopified" projects, their sizes based on how many committers and PMC members they have. Edges between these nodes represent an organization contributing some number of personnel to the project. The edges' thickness is sized to reflect the number of personnel assigned from that organization to that project. In some cases, a single person from a single organization may contribute to multiple projects. This data was collected from the apache.org team-lists for the "Hadoopified" Projects.

Following Porter's Five Forces model we consider the five classes of organizations that influence the direction of Hadoop.

Suppliers Not surprisingly, Hadoop, as an Apache project, mainly pulls from Apache sources. Additional component requirements are satisified by integrating existing open-source and open-license components. Because of the open-source nature of the components, suppliers are unable to leverage profitability of the market into increased profitability; however, greater visibility is useful in convincing organizations to collaborate. Commercial organizations that are contributing code to the Hadoop project, which they have developed to facilitate their proprietary features, are both users and suppliers. We will discuss the supply network in section 4.2.

Substitutes A number of different substitutes for big data analysis are available that diverge from Map/Reduce. GridGrain[16] offers an alternative architecture that also uses a Map/Reduce approach. Rather than use a distributed file system, GridGrain uses an in-memory data grid concept. This architecture handles less data than what is often meant by "big data," hence its classification as a substitute rather than a competitor, but there are many applications for which it is sufficient. Additional substitutes include Spark, ScaleOut, and GraphLab, which offer alternatives to Map/Reduce as well.

Potential Entrants Our analysis did not identify any organizations that are publicly considering entering into this ecosystem.

Competitors The core of Hadoop includes the file system and the computation engine. There are several competitors to the Hadoop file system: Lustre, Orange File System, GIGA+, Google File System, Ceph, and NFile System. Additionally, several alternatives exist to both the Map/Reduce architecture and algorithm, including those offered by Sector/Sphere, Disco, HortonWorks, Cloudera and MapR.

Buyers An open source project has users rather than buyers. Hadoop is used by a large number of organizations. The web-based download makes it impossible to provide a comprehensive list of users. There are some organizations building on top of Hadoop and making their use of Hadoop as a feature. Amazon Elastic, Windows Azure, Google App Engine, and IBM SmartCloud are offering PaaS and IaaS solutions.

There are a number of collaborations being formed around Hadoop that bring organizations together to offer comprehensive configurations that isolate

users from the complexities of replication and fault tolerance. Hewlett Packard, NetApp, and Cisco are a few examples.

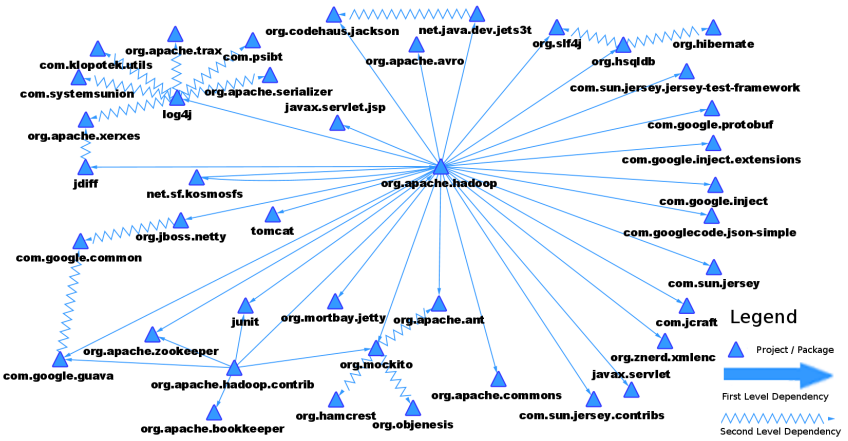
4.2 Software view

The software architecture of Hadoop provides a robust and scalable distributed computing infrastructure for unstructured data.

Although the supply network is made up of organizations, we will consider it from a software product perspective, but we will ignore other suppliers such as makers of development tools. As an open source project the source code is available and shows the references to imported software. The licensing conditions also generally force all imports to be open source and can be used to travel further down the supply network.

Figure 2 shows two levels of the software supply network of Apache Hadoop. The software supply network is modeled as a dependency graph where nodes represent packages of source code, named by their qualified Java package name, and edges represent a "uses" dependency between two source packages.

Fig. 2. Two levels of the Hadoop supply chain



The dependency graph represents the second level of software suppliers, i.e, our suppliers' suppliers. This graph was obtained through analysis of source code for library imports and build dependencies. Nodes connected by a solid line are suppliers obtained by analyzing Hadoop 2.0.3. Nodes connected via zigzag lines represent our suppliers' suppliers.

Due to the fact that many of the third party components included in Hadoop are libraries that facilitate utilities, such as testing, logging or I/O, it is not surprising, though no less interesting, that relationships exist among the software provided by Hadoop's suppliers, which are therefore related implicitly.

4.3 Innovation view

Using the Businessweek categories [17]:

Product The MapReduce architecture was an innovation when Hadoop was initiated. Besides the innovativeness of the architecture, the Hadoop framework uses a functional programming paradigm which is unusual if not innovative. Features such as abstraction of data replication, automatic handling of node failures/fault tolerance and the use of streaming to create a language agnostic interface are innovations in distributed computing products.

Process The concept of storing data and then bringing the computation to it is innovative.

Business model Hadoop itself does not present a new business model, but the other companies that are using Hadoop are implementing high performance computing versions of PaaS and IaaS solutions. These solutions put Hadoop at the core of many products.

Customer experience Hadoop is a fairly traditional open source organization. The evolving architecture that is increasingly modular has made it possible for a customer to replace portions of Hadoop with more hardware specific solutions such as a different file system.

4.4 Risks

The primary risk for the Hadoop project is its current intense popularity and status as a buzz word. The Apache Hadoop core project may have difficulty meeting the needs of the large and diverse number of users. The proliferation of distributions and the independence of the "Hadoopified" projects may lead to divergence. One approach to mitigating this risk would be to broaden the representation on the PMC or to create a separate advisory board that can represent the needs of the diverse user community.

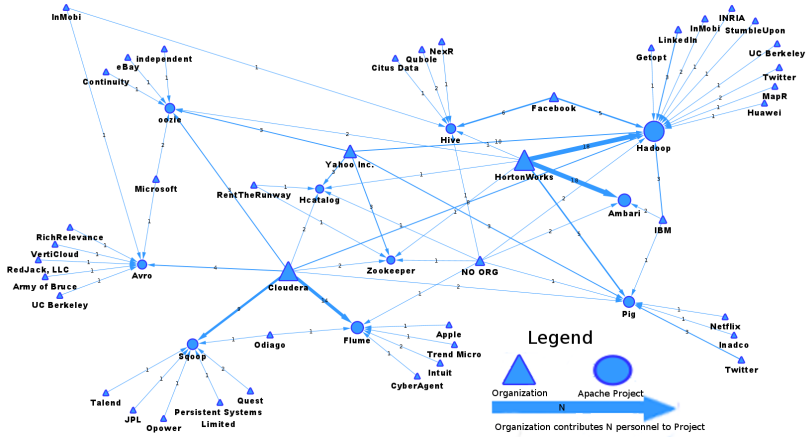
The splitting of several pieces into separate Apache projects potentially harms the architectural integrity of Hadoop. These separate projects are now independent and may make design decisions that will move them away from the trajectory of Hadoop. This is a risk, particularly due to the flat governance structure of Apache projects.

4.5 Ecosystem Health

STREAM uses the ecosystem health criteria described by den Hartigh et al [18].

Robustness In this early phase the ecosystem is very robust. The Project Management Committee (PMC) for Hadoop has representation from several organizations with major resources. The departure of any of these would not cause the project to fail; however, it could cause the project to change directions. Additionally, there are a few dominant players in the Hadoop ecosystem, such as Cloudera and HortonWorks, that support both the core Hadoop project and various "Hadoopified" projects. Figure 3 shows that each of these organizations contributes heavily to projects in the ecosystem.

Fig. 3. Network of Hadoop Contributers



Niche Creation Many organizations are attempting to create niches within the Hadoop ecosystem. "Hadoopified" projects are Apache projects that typically began as Hadoop Contrib projects but have split off into independent projects. These include Avro, Pig, Zookeeper, Flume, HBase, HCatalog, Oozie, Sqoop and Hive. Each of these products delivers a set of features, complementary to those offered by Hadoop, that meets the needs of particular stakeholders. For instance, Hive facilitates data summarization through a domain specific language, HiveQL, that closely mirrors SQL while providing functionality for ad-hoc queries, a useful feature for a database specialist wanting to use Hadoop. Other, mainly commercial, organizations are also working to differentiate themselves from others in the ecosystem. Many organizations are contributing some features to the core Hadoop project and the "Hadoopified" projects. Some organizations create a niche by providing distributions that add end user features making Hadoop easier to deploy, manage, and maintain. Others are providing services related to Hadoop including service and training.

Productivity The ecosystem continues to be very productive. The core Hadoop project maintains four release streams: legacy, stable, beta and alpha. Apache is fixing defects an dreleasing builds. In addition to the efforts that surround HDFS and MapReduce, the set of tools surrounding Hadoop, the Hadoopified projects, represent a significant amount of productivity, with at least nine new Apache projects started since 2008, the majority of which have evolved into top-level Apache projects.

4.6 Evolutionary Forces

In this baseline model we are considering the current state of Hadoop, but we briefly consider the evolutionary forces, both internal and external, at work in the Hadoop ecosystem. Rather than simply internal and external forces there are

layers of forces. At the core there is the Hadoop Apache project with the two fundamental components: MapReduce and the Hadoop file system. Then there are the Hadoopified products which augment, but have split from or formed independently of, Hadoop. Still further removed are those organizations like Cloudera, Hortonworks, and MapR that are adding features to the basic Hadoop distribution, in addition to developing workflow solutions and training and support materials for Hadoop. Further still there are organizations, such as HP and Microsoft, bundling basic Hadoop or a supplemented Hadoop to provided completely configured installations ready for end users who have big data but no systems expertise.

These organizations are addressing several markets segments which present different forces and which will most likely evolve at different rates, new levels may emerge, and organizations may move to different levels. Our longitudinal look will capture these changes.

5 Results

As part of developing this initial snapshot we have we have systematically covered the Apache Hadoop project documentation to identify relevant stakeholders and organizations, visited contributing organizations sites to identify their contributers and analyzed source code for supply-chain modeling in this early baseline model of the Hadoop ecosystem. The data is organized into diagrams which are the information managers use.

Gathering information and conducting analyses on an ecosystem surrounding an open source project has proven to be a difficult, but manageable process. Several observations can be made based on the techniques we have used and the information we have gathered:

- Upstream suppliers can be identified by analyzing build dependencies and library inclusions,
- Niche creation can be evaluated via downstream users who also exist as internal suppliers,
- Productivity can be measured in part by the number and frequency of releases determined from changelogs of projects within the ecosystem, and
- Evolutionary information can be related to a timeline created from the above data.

Hanssen et al hypothesized a set of characteristics for software ecosystems [1]. Our study of Hadoop supports several of those hypotheses:

- central organization - Apache Hadoop project and “Hadoopified” projects provide the basis for the commercial development
- adaptation - decomposing into niche projects to address user needs
- networked - dependencies among software elements
- use of technology - the Apache infrastructure

– shared values - the shared domain leads to certain shared values around distributed computing while there is diversity from a business perspective

While similar tactics might prove useful in ecosystems surrounding commercial or closed source offerings, the abundance of accessible data within an open-source ecosystem is what powers an analyst's ability to model such an ecosystem. It is necessary to continue to gather all available data concerning the emergence or splitting of new projects, the addition of new, or exclusion of existing, software dependencies, the governance structure of projects within the ecosystem and business models leveraging the ecosystem.

6 Future Work

In Section 1, we posed four questions to help guide our analysis of the ecosystem surrounding the Hadoop Map/Reduce framework and architecture. While we are closer to the answers for the questions, the continuing evolution of the ecosystem requires continued data collection and analysis. We will revisit the Hadoop ecosystem at quarterly intervals to add to and revise our models and analyses. Recurring analysis and revision is necessary due to the rapid rate of adoption by companies and the unkowable number of companies using Hadoop, both of which threaten the validity of this study. At these intervals, we will consider more quantified measures for evolution and ecosystem health metrics. Evolutionary metrics may include rates of change on ecosystem elements that were examined in this work: ecosystem size, roles of ecosystem members, external suppliers, release rates and niche offerings. Additionally, the work of the authors in [19] may be helpful in providing project analysis in the software view.

7 Conclusion

We have observed two major trends: a splitting of an initial project into projects that are more narrowly focused and a broadening of the ways in which organizations monetize their participation in the ecosystem. By providing an innovative architecture Hadoop has an advantage but other organizations are already following this approach and offering competing products. STREAM is providing us with a framework within which to add the tools needed to answer the questions in which we are interested. The subsequent snapshots and our analyses of the deltas will provide additional insights about Hadoop specifically and sociotechnical ecosystems in general.

References

1. Hanssen, G.: A longitudinal case study of an emerging software ecosystem: Implications for practice and theory (2012)

2. Chastek, G., McGregor, J.D.: It takes an ecosystem, SSTC (2012)
3. Iansiti, M., Levien, R.: Strategy as ecology strategy as ecology. Harvard Business Review **82**(3) (2004) 6881
4. Messerschmitt, D.G., Szyperski, C.: Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press, Cambridge, MA, USA (2003)
5. Adner, R.: Match your innovation strategy to your innovation ecosystem. Harvard Business Review **84**(4) (2006) 98–107; 148
6. Porter, M.E.: The five competitive forces that shape strategy. Harvard Business Review **86**(1) (2008) 78–93, 137
7. Monteith, J.Y., McGregor, J.D.: A three viewpoint model for software ecosystems. In: Proceedings of Software Engineering and Applictions 2012. (2012)
8. Tiwana, A., Konsynski, B., Bush, A.A.: Platform evolution: Coevolution of platform architecture, governance, and nvironmental dynamics (2010)
9. Hanssen, G.K., Faegri, T.E.: Agile customer engagement: a longitudinal qualitative case study. In: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering. ISESE '06, New York, NY, USA, ACM (2006) 164–173
10. Hanssen, G.K., Fígri, T.E.: Process fusion: An industrial case study on agile software product line engineering. J. Syst. Softw. **81**(6) (June 2008) 843–854
11. Hanssen, G., Yamashita, A.F., Conradi, R., Moonen, L.: Software entropy in agile product evolution. In: Proceedings of the 2010 43rd Hawaii International Conference on System Sciences. HICSS '10, Washington, DC, USA, IEEE Computer Society (2010) 1–10
12. Faegri, T.E., Hanssen, G.K.: Collaboration, process control, and fragility in evolutionary product development. IEEE Softw. **24**(3) (May 2007) 96–104
13. Insemble: Building bridges in the software value chain through enterprise architects "http://www.insemble.com/software-value-chain.html".
14. Ghemawat, S., Gobioff, H., Leung, S.T.: The google file system. In: Proceedings of the 19th ACM Symposium on Operating Systems Principles. (2003)
15. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: Proceedings of OSDI'04: Sixth Symposium on Operating System Design and Implementation. (2004)
16. GridGrain: "http://www.gridgain.com/features/".
17. Businessweek: Fifty most innovative companies. (2009) "http://bwnt.businessweek.com/interactive_reports/innovative_50_2009/".
18. den Hartigh, E., Tol, M., Visscher, W.: The health measurement of a business ecosystem. In: Proceeedings of ECCON 2006. (2006)
19. Bjarnason, E., Svensson, R., Regnell, B.: Evidence-based timelines for project retrospectives x2014; a method for assessing requirements engineering in context. In: Empirical Requirements Engineering (EmpiRE), 2012 IEEE Second International Workshop on. (2012) 17–24