An Investigation into Inner Source Software Development: Preliminary Findings from a Systematic Literature Review

Henry Edison  
Lero, National University of Ireland Galway  
henry.edison@nuigalway.ie

Noel Carroll  
Lero, National University of Ireland Galway  
oiel.carroll@nuigalway.ie

Kieran Conboy  
Lero, National University of Ireland Galway  
kieran.conboy@nuigalway.ie

Lorraine Morgan  
Lero, National University of Ireland Galway  
lorraine.morgan@nuigalway.ie

ABSTRACT

Given the value and effectiveness of open source software development to date, practitioners are keen to replicate these practices inside their respective corporations. This application of open source practices inside the confines of a corporate entity has been coined inner source software development. However, while organisations have found ways to directly benefit from revenue streams as a result of leveraging open source practices internally, the current research on inner source is scattered among different areas. Thus gaining clarity on the state-of-the-art in inner source research is challenging. In particular, there is no systematic literature review of known research to date on inner source. We address this challenge by presenting a systematic literature review that identifies, critically evaluates and integrates the findings of 29 primary studies on inner source. Case study approach is the common research approach undertaken in the area. We also identified 8 frameworks/methods, models and tools proposed in the literature to support inner source, as well as a set of benefits and challenges associated with inner source. We envision future work to perform deeper analysis and synthesis on the empirical research on inner source software development.

ACM Classification Keywords


Author Keywords

inner source software development, inner source, systematic literature review, open source

INTRODUCTION

Open Source Software (OSS) has been highly prevalent in both practice and research. Given the value and effectiveness of open source development to date, practitioners are keen to replicate these practices inside their respective corporations. This application of open source practices inside the confines of a corporate entity has been coined inner source software (ISS) development [7, 50, 16, 42]. ISS development has been seen as a way in which organisations adopt OSS [17, 19]. Leading organisations such as Lucent Technologies, Nokia, Philips, IBM and HP etc, have provided example of inner source implementations [5], which has led to the emergence of studies on inner source in the scientific literature (e.g. [24, 25, 43, 42, 44]).

While the literature purports that ISS development presents many benefits for internal use, there have been growing research efforts to equally highlight the challenges associated with adopting and scaling inner source practices. However, understanding the state-of-the-art in inner source research is challenging. Capraro and Riehle [5] recently published a review on inner source literature. Although the findings are significant and shed light on the grey area of ISS development, the inner source research field remains dispersed among different research areas. Thus, the objective of this study is to understand the current research that has been carried out on the usage of ISS development or the application of OSS principles within existing organisations. This objective is broken down into the following research questions: (i) RQ1 - What research methods have been used in studies on ISS development? (ii) RQ2 - What types of contributions are provided by the studies on ISS development? and (iii) RQ3 - What are the reported benefits and challenges associated with ISS development?

The remainder of this article is structured as follows. Section 2 presents a brief discussion of related work in this area and followed by a presentation of our research approach in Section 3. In Section 4, we present an overview of the characteristics of the primary studies. The key findings of this study are presented in Section 5, and discussed in Section 6. Finally, the conclusion is presented in Section 7.
RELATED WORK
We identified four secondary studies [17, 19, 6, 5] on inner source that were considered relevant to this study. For example, Hauge et al. [17] and Höst and Oručević-Alagić [19] perform literature reviews by following the guidelines provided by Kitchenham and Charters [22]. The study by Crowston et al. [6] could be considered partly systematic, since the relevance can be seen in terms of search strategy, data sources, inclusion/exclusion criteria and data extraction. On the other hand, the study by Capraro and Riehle [5] presents an extensive literature review but show no evidence of following any systematic guidelines.

Both Hauge et al. [17] and Crowston et al. [6] conducted a search on specific journals or conferences on open source, while Höst and Oručević-Alagić [19] and Capraro and Riehle [5] only focused on software engineering related databases such as Inspec, Compendex, ACM Digital Library and IEEE Explore. Furthermore, Hauge et al. [17] did not report the used search string and Capraro and Riehle [5] did not reveal the timespan for the search. To overcome this, we organised our search strategy into three categories and included both software engineering (SE) and information systems (IS) related databases.

Neither Crowston et al. [6] nor Capraro and Riehle [5] used explicit criteria for quality assessment of the primary studies, which hinders interpretation. Contrary to these studies, Hauge et al. [17] and Höst and Oručević-Alagić [19] adopt the quality assessment criteria developed by Dybå and Dingsøyr [8]. For our review, we adopted a comprehensive set of evaluation guidelines based on scientific rigour and industrial relevance proposed by Ivarsson and Gorschek [20].

The aim of the review conducted by Hauge et al. [17] and Höst and Oručević-Alagić[19] was to assess the current research on how organisations adopt OSS development principles, while Crowston et al. [6] investigated the state of research on open source in general. Capraro and Riehle [5] develop a model of the elements that constitute inner source. The study also presents a classification framework for inner source programs and projects, and present a map of known inner source endeavours, as well as qualitative models summarising the benefits and challenges of inner source adoption. All studies identify that there is a trend towards organisational adoption of open source principles in their internal development processes.

In summary, while majority of the existing relevant literature reviews mainly focus on OSS, there is only one study specifically on inner source [5] that is closer in similarity to our study. Nevertheless, the study does not show adherence to any systematic guidelines and the scope is less broader than ours since it does not consider IS literature. Moreover, the absence of quality assessment of the primary studies hinders interpretation of the findings. In this study, we address these gaps by following an SLR guidelines [22] and we limit the review only to ISS development.

RESEARCH METHODOLOGY
To answer our research questions, we systematically assessed existing evidence related to ISS development using SLR guidelines [22]. An SLR facilitates in identifying and collecting key papers in a specific area of interest, and evaluating and interpreting the reporting discussions and findings [22]. A defined review protocol, search strategy, explicit inclusion and exclusion criteria, and specified information that will be retrieved from primary studies differentiates a systematic review from a conventional literature review [22].

Search Strategy
To help build the search terms, a set of key papers were identified by all authors. Keywords used in these papers were extracted and aggregated and used as the input for the search terms. From the key papers, we extracted a total of 76 keywords (30 unique). The top three keywords are “inner source” (10.53%), “open source” (23.68%) and “software development” (19.74%). These include the variations e.g. “open-source” or “inner-source”. The developed search terms were then searched through in EngineeringVillage and Scopus to view its effectiveness. Revisions to the search terms were made until we were able to retrieve all seed papers from these two databases.

The search terms were organised into three groups: intervention, control and population, and separated by AND-clauses. Table 1 describes the generic search strings. A search for relevant literature was conducted on the metadata, in particular on the title, abstract and keyword. We decided to use digital libraries that have good coverage, familiarity, reputation, advanced features and exportability [10]. The digital libraries used to search for relevant literature were relevant to (i) SE research: Compendex, ISI Web of Science and Scopus [9, 3], and (ii) IS research: AIS e-Library, ScienceDirect, Wiley Science, ACM, IEEEExplore databases returned similar search results as Compendex or ISI Web of Science [9]. Hence, in this study, we only used Compendex and ISI Web of Science. We also selected Scopus since it is considered as the largest abstract database for peer-reviewed papers [21].

<table>
<thead>
<tr>
<th>Group</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intervention (key concepts)</td>
<td>{inner?source} OR innersource OR {open?source} OR opensource software</td>
</tr>
<tr>
<td>Control (context)</td>
<td>OR organisation OR firm OR company OR corporat* OR enterprise OR industr* OR vendor OR institution* OR internal OR inside OR hybrid</td>
</tr>
</tbody>
</table>

Table 1. Search terms organisation

Selection Strategy
Studies were eligible for inclusion in this study if they are (i) peer-reviewed papers, (ii) written in English, (iii) full-text available, and (iv) focusing on ISS development or the adoption of OSS principles within an organisation. Both theoretical and empirical studies were included in the review process. Similarly studies conducted in both academic and industry settings were included.
The removal of duplicates across digital libraries and papers that were clearly irrelevant studies was conducted as a pre-screening process in the literature review. This was then followed by applying inclusion and exclusion criteria, which was done in two stages. In the first stage, the criteria were applied on the title and abstract level of the papers. In the second stage, the inclusion and exclusion criteria were applied on the full-text of the remaining papers. In every stage, two independent reviewers evaluated the same paper. To be included in the next phase, two reviewers had to be in agreement that a paper met the inclusion criteria. In the cases where both reviewers did not agree on the decision, a third reviewer was called. Full-text papers that met the inclusion and exclusion criteria were then assessed on their quality.

**Quality Assessment**

The quality of the primary studies was evaluated based on their scientific rigour and industrial relevance as presented in Ivarsson and Gorschek [20]. We also devised a similar rubric for assessing the rigour of the philosophical papers. Scientific rigour assess the following aspects: (i) study context: whether a reviewer can understand and compare it to another context, (ii) study design: whether a reviewer can understand how rigour the research method is applied in the study or to which the theoretical contribution used sound theoretical bases to guarantee the quality of the research, and (iii) validity discussion: to what extend the threats of the study or limitation of the theoretical approach are described and measures to limit them are detailed.

For the industrial relevance, we used the same rubric for both empirical and philosophical studies. Relevance was assessed using the following aspects: (i) subject: whether the subjects in the study were representative of inner source practitioners e.g. students or practitioners, (ii) context: whether the study was conducted in the representative industry setting, (iii) scale: whether the size of the study was realistic or based on a toy example, and (iv) research method: whether the research method employed in the study facilitates investigating real situations and relevant for practitioners.

**Data Extraction and Synthesis**

To help answer our research questions, data extraction was carried out guided by an extraction form implemented in MS Excel. The following aspects were extracted from the primary studies: (i) type of the studies, classified as (adapted from Wieringa et al. [51]): empirical research, experience report, philosophical, opinion, (ii) contributions of the studies: model, theory, framework or method, guidelines, lessons learned, advice or implication, and tools [35], and (iii) research method, including data collection and analysis method, and theoretical lens.

The data extracted from each primary study were integrated in categories representing the research topic addressed, the research method used, the contributions and the benefits and challenges reported. Frequencies of each component in the categories were recorded. The results were then presented and discussed with the other coauthors.

**Search Results**

From all digital libraries, we retrieved a total of 12,304 articles. By applying the inclusion/exclusion criteria, we accepted 29 articles as primary studies. The 29 primary studies are listed in Table 2 and referred using their IDs throughout the rest of the paper.

**Publication Sources and Years**

The distribution of the years and venues of the primary studies is shown in Fig 1. While the term inner source was introduced in the year 2000 [32], the first research paper in this topic was published two years later (PS2). 14 out of 29 papers (48%) are published in journals i.e. Research Policy, IST, JAIS, TOSEM, etc., whereas 15 papers (52%) are presented in various conferences in both SE and IS, i.e. OSS Symposium, ECIS, PROFES, etc.

**Quality of the Primary Studies**

In this study, the quality assessment was performed independently and the results were not used to decide whether to include or exclude a particular study. Based on the rigour and relevance scores, the primary studies can be considered to be of good quality. The percentile rankings of the quality scores are shown in Fig. 2. The maximum score that a paper could get was 7. Studies with scores below the lower quartile lacked clear information about the study design and threats to validity, as required in the rigor rubric. Typically, these studies were published in practitioners oriented journals e.g. IEEE Software (PS23, PS26, PS29) or Communication of the ACM (PS6). Studies with minimum scores are philosophical papers. Since, the identified or proposed framework or methods have not been studied empirically in industry settings, the relevance scores were zero. Moreover, most of the studies within the interquartile range did not discuss validity threats and how they
were mitigated, which negatively affected the trustworthiness of the reported findings [40].

**FINDINGS**
In this section, we present an overview of the body of inner source literature that originated from our literature review. We structure this section according to our research questions.

**RQ1 What research methods have been used in studies related to ISS development?**
Out of 29 primary studies, 23 papers are empirical research papers. Case study approach was the main common research approach undertaken to investigate inner source approach (13 studies), whilst 2 studies (PS1 and PS7) were employing mixed method, and one study used design science approach (PS19). Some experience report papers were submitted to a special industry experience track of particular conference (PS5, PS11, PS19).

Our primary studies shows that most research on inner source approach has been conducted in established and large multinational organisations, and only one study in an SMEs context (PS7). These organisations come from various business domains, e.g. engineering, software development, medical equipment or telecommunication. These organisations have a long history with proprietary oriented and industrial/commercial mode of software development and gradually incorporated open source software and methods into their internal development (PS30). The overview of the empirical research papers is illustrated in Table 3.

Interview and coding technique were the main data collection and analysis method. The total number of interviews in each study were ranged between 5 (PS7) and 32 (PS21). The participants involved in the interview were typically employees with engineering backgrounds and roles, i.e. software developer, architect, programmer and administrative role, e.g. manager. The working nature of these two roles are also highly affected by the decision of the organisation to adopt inner source approach (PS1).

Out of 13 case study papers, only 6 studies used a theory or conceptual framework as a theoretical lens. Conceptual framework is used to explain the primary objects of a research project e.g. key factors, constructs or variable and the relationship among them [29]. It is a sensitising and sense-making device that guides the data collection and analysis processes. Three of them were published in IS conferences (ECIS - PS10, PS13) and journal (JAIS - PS2). In contrast to SE research, the use of theory in IS research is critical [23]. For example, Orlikowski and Lacono [33] argued that IS research is under theorised. Hence, IS researchers aim for strong theoretical contributions and ground their work in theory [23].

**RQ2 What types of contributions are provided by the studies on ISS development?**
Fig. 3 shows the distribution of the contribution of primary studies. The main contribution of the primary studies was in the form of theory (15 studies) around an ISS approach. Nonetheless, 27% (8 studies) of the contributions provided concrete approaches that could be used to support an ISS approach. These approaches included framework or method for implementing an ISS approach (5 studies), e.g. progressive open source (PS2), corporate open source (PS6), models representing relevant concepts of inner source (2 studies) e.g. theoretical model to promote software reuse (PS28), and inner source business model (PS29) and tool supporting technical infrastructure of inner source (PS19). However, study PS8 concerned about the framework used in the study, which has not been evaluated by others than the authors. Thus, they called for more studies to evaluate the existing frameworks, methods and tools in a different context so that they can be generalised. Other studies provided guidelines (2 studies), lesson learned (3 studies) and advice (1 study) for implementing an ISS approach. The list of the frameworks/methods, models and tools are described in Table 4.

For a successful technology transfer, academic research results are required to be validated statically or dynamically in real setting [13]. Static validation involves presentation of the solution in industry and collecting feedback from practitioners, whilst dynamic validation includes piloting the solution in a real development setting. Out of 8 solutions, 3 studies that had been validated in industry (P19, PS21, PS25). In the study PS19, the proposed solution had been tested in a real implementation as part of the design science process. In the study P21, the framework was validated using qualitative ap-
<table>
<thead>
<tr>
<th>ID</th>
<th>Research Approach</th>
<th>Data Collection Method</th>
<th>Data Analysis Method</th>
<th>Theoretical Lens</th>
<th>Context of study</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS1</td>
<td>Mixed method</td>
<td>Interview &amp; survey</td>
<td>content analysis</td>
<td>–</td>
<td>a multinational engineering firm</td>
</tr>
<tr>
<td>PS2</td>
<td>Case study</td>
<td>not-specified</td>
<td>not-specified</td>
<td>–</td>
<td>HP</td>
</tr>
<tr>
<td>PS3</td>
<td>Interview</td>
<td>interview</td>
<td>coding technique</td>
<td>–</td>
<td>7 large multinational companies</td>
</tr>
<tr>
<td>PS4</td>
<td>Case study</td>
<td>data archival</td>
<td>log transformation</td>
<td>–</td>
<td>a commercial software vendor and Sourceforge.net</td>
</tr>
<tr>
<td>PS6</td>
<td>Case study</td>
<td>not-specified</td>
<td>not-specified</td>
<td>–</td>
<td>Lucent Technologies</td>
</tr>
<tr>
<td>PS7</td>
<td>Mixed method</td>
<td>Interview &amp; survey</td>
<td>not-specified</td>
<td>–</td>
<td>3 SMEs</td>
</tr>
<tr>
<td>PS8</td>
<td>Case study</td>
<td>Data archival &amp; interview</td>
<td>archival analysis</td>
<td>inner source framework [42]</td>
<td>an international software development firm</td>
</tr>
<tr>
<td>PS9</td>
<td>Case study</td>
<td>Interview</td>
<td>interview</td>
<td>–</td>
<td>Nokia</td>
</tr>
<tr>
<td>PS10</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>institutional entrepreneurship</td>
<td>Philips and Nokia</td>
</tr>
<tr>
<td>PS12</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>Actor-Network Theory (ANT)</td>
<td>HP</td>
</tr>
<tr>
<td>PS13</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>open innovation theory [11]</td>
<td>large medical equipment supplier</td>
</tr>
<tr>
<td>PS15</td>
<td>Case study</td>
<td>Focus group</td>
<td>coding technique</td>
<td>–</td>
<td>global market leader in software and hardware</td>
</tr>
<tr>
<td>PS16</td>
<td>Case study</td>
<td>Participatory observation</td>
<td>not-specified</td>
<td>knowledge life-cycle and human aspect of KM</td>
<td>Nokia</td>
</tr>
<tr>
<td>PS17</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>–</td>
<td>three international software companies</td>
</tr>
<tr>
<td>PS19</td>
<td>Design science</td>
<td>Prototyping</td>
<td>not-specified</td>
<td>–</td>
<td>Insta DefSec Ltd.</td>
</tr>
<tr>
<td>PS21</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>–</td>
<td>3 large multinational companies</td>
</tr>
<tr>
<td>PS22</td>
<td>Case study</td>
<td>Interview</td>
<td>coding technique</td>
<td>–</td>
<td>a large, globally distributed organisation</td>
</tr>
<tr>
<td>PS23</td>
<td>Case study</td>
<td>Interview</td>
<td>not-specified</td>
<td>–</td>
<td>Philips Healthcare, Lucent</td>
</tr>
<tr>
<td>PS25</td>
<td>Case study</td>
<td>Interview</td>
<td>thematic analysis</td>
<td>Avison and Fitzgerald [2] framework</td>
<td>a large telecommunication company</td>
</tr>
<tr>
<td>PS26</td>
<td>Case study</td>
<td>not-specified</td>
<td>not-specified</td>
<td>–</td>
<td>2 large European companies</td>
</tr>
<tr>
<td>PS28</td>
<td>Case study</td>
<td>Interview</td>
<td>not-specified</td>
<td>–</td>
<td>IBM</td>
</tr>
<tr>
<td>PS29</td>
<td>Case study</td>
<td>not-specified</td>
<td>not-specified</td>
<td>–</td>
<td>Philips Healthcare</td>
</tr>
</tbody>
</table>

Table 3. Overview of the empirical research papers
The adoption of an ISS approach enables organisations to meet the challenges of software development, process management, tool and technology and business units using the shared assets. Moreover, development costs are shared among the projects, number of projects that can be shared across the organisation, and the collaboration between various business units increases core team innovativeness in order to satisfy user needs (PS13). Several studies have also reported that an ISS approach reduces development effort as well as cost (PS6, PS7, PS10, PS20). Software reuse maximises the number of projects that can be shared across the organisation. Moreover, development costs are shared among the projects or business units using the shared assets.

On the other hand, the common challenge of an ISS approach is related to security (PS2, PS3, PS11, PS27). The openness sought after in an ISS approach makes security and access control to the internal software artefacts e.g. code or design more difficult. For example, project managers and developers are wary of random exposure to their internal product artefacts, e.g. source code and design. For developers, there is a fear of sharing or showing substandard or work. Moreover, in the case once various contractors that are employed within different areas of the organisation are given access, there is the perception that they have no control over what security the vendor is using. On the other side, the contractor might be reluctant to use an ISS approach due to the fear of loss of intellectual property.

**Process Management**

The role of contributors in an ISS approach might evolve and shift based on their personal interest. For example, they can shift from developer to maintainer or reviewer, before they will be the project leader or benevolent dictator. This predefined path allows new developer to familiarise themselves with the architecture, perform tasks with different difficulty levels. Similar to OSS project, the community has a great deal of freedom to choose processes, methods and tools (PS25) in their works.

The main challenge associated with process management however, is how to build an effective community within the organisation (PS13, PS21). The study by Stol et al. (PS21) found that there are several issues between the core team and business units in relation to their roles. The core team may be reluctant to adopt contributions from business units, due to the “not invented here” syndrome and non-generic contributions. On the other hand, business units treat the core team as a traditional component supplier. They are also often reluctant to contribute to the shared assets since they consider that development is the responsibility of the core team. The study by Morgan et al. (PS13) also found the challenges in achieving a common vision and aligning objective in an ISS environment. This study also found that even though knowledge sharing is perceived as important in facilitating value creation, it is difficult to get people to invest time or effort in sharing code or building skills and knowledge outside their own domain. The core team and business units only contribute to their areas, as there are often no incentives to contribute beyond them.

**Tools and Technology**

Our primary studies reveal that an ISS approach has increased information availability and visibility across organisations, defining an entry path for newcomers and new ways of working (PS25). A good software forge indexes information sources, including source code assets or components and allows for centralised searching. This is important for technical roles, insofar as they are able to contribute to further development.

Despite the benefits of software forges, several studies identified challenges related to the tools and technology used in an ISS approach. Once the software forge attains a certain size, searching and navigating through projects and components details can be time consuming (PS2, PS3, PS8, PS16). Thus, a proper search and navigation infrastructure is important for all contributors. Typically, each software project and group...
<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Description</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS2</td>
<td>Framework</td>
<td>Progressive Open Source (POS): consists of three-tier model: (i) Inner Source – refers to the application of the OS approach and benefits to developers within the corporate environment, (ii) Controlled Source - which is outside of the corporate firewall, but restricts access limited to specific corporate partners and (iii) Open Source - refers to the open use of Internet for development, and release of the software source code in a license approved by OSI.</td>
<td>–</td>
</tr>
<tr>
<td>PS6</td>
<td>Framework</td>
<td>Corporate Open Source (COS): evolves in four phases: (i) Initial Development, led by the author of the code, (ii) Ad-hoc Partners, distributes the binary to a wider audience inside the company, (iii) User-initiated Change Request, expanding the class of users within the company to get feedback or wishes for new features, (iv) Establishing a COS Project, as the request for product-specific changes began to accelerate, other within the company started to contribute code and ideas.</td>
<td>–</td>
</tr>
<tr>
<td>PS19</td>
<td>Tool</td>
<td>OpenCart-based platform that acts as a marketplace for promoting software reuse within an organisation. The platform also provides information about the name and version of components, the technical and functional descriptions, the locations and contact persons of the components and prices and licenses if a third party component was included.</td>
<td>Dynamic</td>
</tr>
<tr>
<td>PS20</td>
<td>Framework</td>
<td>The framework guides the creation and management of hybrid-OSS communities in organisations, consists of three major elements: (i) community building, (ii) community governance, and (iii) community infrastructure.</td>
<td>–</td>
</tr>
<tr>
<td>PS21</td>
<td>Framework</td>
<td>The framework identifies nine important factors that need to be considered when implementing Inner Source. The framework can be used as a probing instrument to assess an organisation on these nine factors so as to gain an understanding of whether or not Inner Source is suitable.</td>
<td>Static</td>
</tr>
<tr>
<td>PS25</td>
<td>Framework</td>
<td>The framework to describe a development methodology from its origin to its practical, and to compare two or more development methodologies.</td>
<td>Static and dynamic</td>
</tr>
<tr>
<td>PS28</td>
<td>Model</td>
<td>Theoretical model to promote reuse within the organisation.</td>
<td>–</td>
</tr>
<tr>
<td>PS29</td>
<td>Model</td>
<td>ISS business model.</td>
<td>–</td>
</tr>
</tbody>
</table>

**Table 4. Framework/Method, Model and Tools**

<table>
<thead>
<tr>
<th>Facet</th>
<th>Benefits</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Development</td>
<td>Better software quality (PS2, PS3, PS8, PS18, PS20, PS25)</td>
<td>Lack of documentation (PS21)</td>
</tr>
<tr>
<td></td>
<td>Reduced development time and time-to-market (PS2, PS20, PS21)</td>
<td>Missing functionality (PS21)</td>
</tr>
<tr>
<td></td>
<td>Shared community debugging (PS2)</td>
<td>Balancing between architectural refactoring and implementing new requirements (PS21)</td>
</tr>
<tr>
<td></td>
<td>Reduced development cost (PS6, PS7, PS10, PS20)</td>
<td>Complexity in using and configuring the shared asset (PS21)</td>
</tr>
<tr>
<td></td>
<td>Increase innovativeness (PS13, PS18)</td>
<td>Security (PS2, PS3, PS11, PS27)</td>
</tr>
<tr>
<td></td>
<td>Avoid duplicate work and promote the reuse of software (PS2, PS3, P10, PS12, PS13, PS17, PS19, PS28)</td>
<td></td>
</tr>
<tr>
<td>Process Management</td>
<td>Define an entry path for newcomers (PS25)</td>
<td>Building an effective community (PS13, PS21)</td>
</tr>
<tr>
<td>Tool and Technology</td>
<td>Increase information availability and visibility (PS25)</td>
<td>Migration from existing tools and infrastructure (PS2, PS5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maintain code tree, platform, version control and related software engineering tools (PS2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time consuming to search and navigate the software forge (PS2, PS3, PS8, PS16)</td>
</tr>
<tr>
<td>Managerial and</td>
<td>Rapid re-deployment of key developers (PS2, PS12, PS21)</td>
<td>Leadership and task assignment (PS2)</td>
</tr>
<tr>
<td>Organisation</td>
<td>Improve company’s image (PS26)</td>
<td>Achieving a high level of commitment (PS13)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cultural resistance to change (PS11, PS14, PS29)</td>
</tr>
</tbody>
</table>

**Table 5. Benefits and Challenges of ISS**
within the organisation has their own infrastructure e.g. version control, bug reports etc., that suits their needs. Therefore, the introduction of uniform toolset and infrastructure in the organisation is a challenge from a technology and user perspective (PS2, PS5). Once they all have migrated, the new IT support is critical to maintain both software e.g. the uptime, running schedule backups and recovery when necessary, and hardware.

Managerial and organisation
In the ISS approach, the community of contributors is the developers who are familiar with the OSS environment within the organisation e.g. source tree, bug reporting, source management tools, corporate specific coding, commenting and code review process. This also makes it easier to identify the talent across organisations (PS21). Hence, an ISS approach creates an opportunity for rapid re-deployment of developers from one project to another and from one product to another (PS2). It also increases the number of parallel development (PS12).

While most organisations today operate in a hierarchical organisational structure, the adoption of an ISS approach creates a virtual organisation (PS2). The contributors may come from different business units within the organisation. Hence it makes it more difficult to manage the talent and skill set at the corporate level. Additionally, it takes more time and effort for people to communicate (PS3), for example, the coding standard must be maintained at corporate level, rather than at a project or group level, and new developers must be trained for maximum use of an ISS approach.

Cultural resistance to change is one of the main challenges for ISS approach adoption (PS11, PS14, PS29). It requires a high level commitment from all stakeholders (PS13). ISS approach changes the relationship within the organisation from a one way dependency to a two way dependency (PS29). Those who are used to developing software, may become the users of the software, and vice versa.

DISCUSSION
Implication for Research and Practice
As far as we are aware of, this is the first attempt to review on inner source research by incorporating the relevant SE and IS literature in a systematic way. Our findings show a great concentration of empirical research on the study of how organisations adopt ISS development into their internal software development processes. Other research areas receive much less attention. Among the frameworks/methods, models and tools identified, none of them have been empirically validated in real industry settings. One of the implications of these findings for research and practice is the need for more empirical studies on engineering practices to support ISS development. Specifically, while ISS development is highly influenced by OSS development, there is a need to translate OSS practices to suit the organisational context to achieve the many benefits associated with OSS. Furthermore, future research is required to empirically validate the proposed frameworks/methods, models and tools. To advanced our understanding of the inner source phenomenon, researchers need to draw on theoretical foundations that have been used in prior research on OSS, as well as other theoretical lens that are considered relevant to ISS approach.

The implication for practice also lie in the evidence of the benefits and challenges of ISS development. The findings have shown that the adoption of ISS development helps organisations to improve better quality, time-to-market and innovativeness. However, as suggested by Brown et al. [4], newcomers should understand the reality of the method through an appropriate enculturation, so that they can recognise what works and what does not work, and thus be aware of changing working processes.

Threats to Validity
Our study is not impervious to threats to validity, which may affect the outcome of this study. In the following section, the threats to validity of this study will be identified and discussed. To mitigate selection bias, we had tested various versions of the search string. We did not use the variation of “open source” for the following reasons. We observed that papers use the term “libre” together with the term “open source”. In addition, papers that use the term OSS or FLOSS must open the abbreviation, which contain “open source”.

To limit subjective bias for an individual reviewer, each paper was reviewed by two reviewers when applying inclusion/exclusion criteria. Prior to the actual selection of the primary studies, all reviewers performed pilot runs with 50 papers. The aim was to see whether all reviewers had the same understanding and perspective on the inclusion/exclusion criteria. Any dissimilarity in assessment between reviewers was discussed in the presence of all reviewers.

CONCLUSION AND FUTURE WORK
Influenced by the success of OSS development, the area of ISS development is gaining more attention from both academic and practitioners. This approach allows organisations to create high quality products in a shorter timeframe by combining heterogeneous development. Unlike existing literature reviews, this review is performed systematically and focuses specifically on ISS development within organisations. Furthermore, our review is interdisciplinary, drawing on both software engineering and information system literature to provide an extensive overview of the ISS phenomenon.

Through our SLR, the study establishes a state of research on ISS development. We found that the case study approach is the common research approach undertaken in the area. We also identified existing frameworks/methods, models and tools proposed in the literature to support ISS development, as well as a set of benefits and challenges associated with ISS development.

We envision future work could perform a deeper analysis and synthesis on the empirical research on ISS development. Based on this analysis and synthesis, we will further investigate the limitations of the current research on ISS development and establish a research agenda on inner source. To enhance the findings of this review, we intend to conduct a comprehensive survey of practitioners to identify the key challenges.
involved in ISS development and propose some resolution strategies to overcome the challenges.

ACKNOWLEDGMENTS
This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie)

REFERENCES


