An Evaluation Model for Social Development Environments

Hani Bani-Salameh¹, Sanaa Alwidian², Maen Hammad¹ and Fadi Wedyan¹

¹ Department of Software Engineering, The Hashemite University, Jordan
² Department of Computer Science, The Hashemite University, Jordan
¹ {hani,mhammad,fadi.wedyan}@hu.edu.jo, ² {sanaa}@hu.edu.jo

Abstract

Distributed software development is becoming a common practice among developers. Factors such as the development environments improvement, their extensibility, and the emergence of social networking software are leading factors. They lead the development process (both co-located and geographically distributed) to a practice that: 1) improves the team’s productivity, and 2) encourages and supports the social interaction among the teams’ members.

The above factors along with the distributed development emergence, Integrated Development Environments (IDEs) evolution, and the social media advances got the attention of the software development teams, and made them consider how to better assist the social nature of software developers, and the social aspects of software development, including activity awareness of team members’ progress, their presence, collaboration, communication, and coordination around shared artifacts.

IDEs are the most commonly used tools by developers and programmers. Integrating the most needed development tools inside the IDE, makes it a Collaborative Development Environment (CDE), and embedding a variety of social software features within a CDE helps support distributed development teams. It also creates a new genera of tools called Social Development Environments or “SDEs” as it is commonly called.

This article discusses the origins and development of “SDEs”. It reflects and presents the research into development environments, explaining their role in supporting the communication and coordination needs of development teams. Also, it presents the effect of embedding social network features in such environments. It presents some emerging SDEs and future work. Finally, it concludes by introducing a new model named 5-star that classifies, and helps to evaluate and compare the collaborative tools. 5-star presents a new way to compare and evaluate the SDE systems based on their supported features. It bases the comparison on five major categories including: Communication, Coordination, Collaboration, Awareness & Presence, and Social networking (SN) tools.

Keywords: Social Software; Social Networks (SNs); IDE; CDE; SDE

1. Introduction

Software engineering is a group effort. Software developers spend a majority of their time on programming activities using a development environment coupled with a variety of collaborative tools. To facilitate collaboration and coordination, developers always feel the need to maintain awareness of their peers’ activities, awareness of the project artifacts, and also social awareness of the interests and opinions of all stakeholders [1]. They use Integrated Development Environments (IDEs) in their everyday activities; in different editing and debugging activities while solving particular technical problems or developing different
software tools. These programming environments evolved from command-line tools, to text-based IDEs, graphical IDEs, and eventually to collaborative and social development environments.

There are many tools available for software developers and engineers to use (e.g., IDEs) that integrate many features to enhance communication, coordination, and collaboration. Such environments support features including project management, version control, bug tracking, and mailing lists. Also, they may provide other features, such as text and audio chat, and email to provide awareness (both formal and informal) of the developers’ activities.

Software developers work within a community and form teams. The existence of wikis, forums, emails, chat, user groups, and now social networks have always been an important factor of the way teams are formed, and developers and software engineers come together.

Software engineers frequently make use of social media to integrate tools, and adapt them in their development environments [1, 2]. They make use of a variety of social media tools to coordinate their work, communicate with their team members, maintain awareness of their activities, and be informed about new technologies and teams’ interests.

Social development environments (SDEs) cover several related disciplines of computer science and software engineering. To better understand the social development environment tools, knowledge of these fields is needed. These disciplines involve (1) Source code control which is a popular practice in almost all software development projects, where developers and programmers are able to check in/out their project artefacts of a shared repository; examples include sourceforge.net, Git, and Subversion; (2) CSCW (Computer Supported Collaborative Work) that studies the nature of the collaborative activities among groups and team members, and how it can be supported by means of development environments [3]; (3) Awareness and Presence which refers to the state or fact of feeling present or exist, and aware of the surrounding activities, project and artefact changes; (4) Social media and networking tools are media that supports interactions among people, and help them create virtual communities and networks; and (5) Integrated development environments. SDE forms an intersection of these overlapping fields (see Figure 1).

In this article, we propose a new model that classifies, and helps to evaluate and compare the collaborative tools. To our knowledge, no existing publication or model that evaluates and compares the so called social development environments the way presented in this article.

![Figure 1. Disciplines Associated with Social Development Environments](image-url)
The rest of this article is organized as follows. Section 2 identifies the research approach and research questions. Section 3 provides a description of the SDEs and presents a number of existing software systems that fall under the CSCW and social software umbrella. Section 4 provides a preliminary ranking model of these systems based on the features integration and support. Finally, this article sums up the current state of this work and discusses future directions in Section 5.

2. Identification of Research

The research and literature introduced here has been developed by following the guidelines presented in [4]. In this section, we give an overview of this process. In short, this process is conducted by three steps: (i) Defining the objectives, research questions, and planning; (ii) Conducting the search by defining and comparing the tools; (iii) Analysis and classification of the tools, and creating a novel evaluation model. These steps are explained in more details later in this article.

2.1. Goal

This study is undertaken to achieve some major goals. The main objectives of this study are as follows:

(i) To review and describe the mostly used communication and collaboration tools that integrate social awareness, and support the distributed software development teams.

(ii) To classify the tools in order to allow members of the development community to make use of such tools.

(iii) To produce an evaluation model that helps to evaluate and compare the SDE systems based on their supported features.

2.2. Definition of Research Question

This study was performed as a review through a search for existing investigating studies to answer the following research questions:

- What social development tools and environments (SDEs) are available and mostly used to support the software development community?
- What features are integrated in such tools?

The review study is expected to reveal the following information:

- Which social tools are available and used by the distributed developers.
- Which features are integrated in these tools, how the tools compare and are ranked to one another based on their features support.

3. Social Development Environments

While working individually or collaborating with their teams, software engineers need to coordinate with other members, to communicate and interact to learn from others, to be aware of others activities, and to become informed about new advances and technologies. To support such activities, software developers usually make use of a wide variety of tools ranging from code editors to development environments, project portals and forges, and social media tools (e.g., wikis, blogs, and SNs).
SDE is where developers find all these tools integrated in a single environment. Although having different objectives and diverse communities, SDE systems share a common set of tools. These tools were analyzed and classified according to our social-aware version of the 3C collaboration model. The new model updates the 3C collaboration model introduced by Fuks et al., [5]. The goal of this update is to show the importance of social media and SN tools in 1) coordinating, 2) easing the communication, and 3) supporting the collaboration between the developers (see Figure 2).

![Figure 2. A Social-aware 3C Collaboration Model](image)

The proposed model bases the SDE features classification on five elements criteria: 1) Communication, 2) Coordination, 3) Collaboration, 4) Social networking (SN) tools, and 5) Awareness. The domain context of these tools is pinpointed as “social development environments (SDEs)”.

Figure 3 represents a “features analysis” of the major supported features depending on the 3C collaboration model. The features model not only shows the structure of the SDE environment, the components an SDE consist of, and the supported features and tools, but also it shows that an SDE consists of: (i) a CDE; (ii) Community building tools and features; and (iii) Social media and social networking tools.

Section 3 presents SDEs as they form the platform where social media features are integrated and put into use by developers [6].

### 3.1. Defining SDEs

An SDE is a real-time collaborative programming tool with integrated social networking features. This emerging technology is important for distributed software developers, eLearning and technical communities. SDEs advance the state of the art for collaboration, coordination, and project management in software development.

Social Development Environments provide a wide range of facilities for synchronous and asynchronous collaboration and information sharing between team members.

According to Treude [6], SDEs are tools that provide support for development activities, ranging from requirements engineering and development to validation/testing and through evolution, using a variety of integrated social media mechanisms.

Researchers provided several definitions for Social Software. This section present some of these definitions.
According to Coates [7], “Social software can be loosely defined as software which supports, extends, or derives added value from human social behaviour—message boards, music taste-sharing, photo-sharing, instant messaging, mailing lists, social networking”

In his book “Social Software in Libraries: Building Collaboration, Communication, and Community Online”, Farkas [8] defines social software as:

“a tool that must meet at least two of the three following conditions:
- It allows people to communicate, collaborate, and build community online. - It can be syndicated, shared, reused, or remixed, or it facilitates syndication.
- It lets people learn easily from and capitalize on the behavior or knowledge of others.”
In this article the term social development refers to software development in collaborative online communities with social relationships. They are developers’ networks where developers do not usually interact merely to socialize, as in conventional social networks. In contrast, they interact and collaborate primarily through shared project artefacts [9].

3.2. Common Properties

SDEs are virtual spaces where developers of a project can meet, brainstorm, discuss, record, and go over ideas to solve design and coding problems, and create software products [9]. As same as CDEs, SDEs create a platform for environments that aims to reduce friction in collaborative processes, and improves both individual and teams’ productivity [12, 13].

As mentioned earlier, SDEs support both of the synchronous and asynchronous collaboration between team members. An SDE is not a single application, it is a harmony between many different development tools. Several applications support the primary collaborative infrastructures for complete SDEs in a single environment. SDEs include a spectrum of collaborative tools that can be of a big benefit to the development communities. Each one of these tools adds its own taste and value to the integrated environment, including eeb logs (blogs), mailing lists, walls, chat rooms, whiteboards, and wikis. This article provides an overview of SDEs and presents a comparison between the state of the art examples (see Section 3.4).

3.3. SDEs and Software Development Communities

Social networks build online communities of people loosely connected by their common interests or activities. Community is a vital aspect of software development, but software developers tend to focus their attention primarily on their programming environment tools. Software development projects usually include communities of users. Community members play vital roles that reflect the success of such projects, such as reporting bugs, helping other users, and analyzing problems. These observations lead to a new category of tools, the Social Development Environment (SDE).

Software development requires interaction between people involved in the development process. For this reason, social activities form a big part of their daily work. In software development, developer networks are an instance of object-centered sociality [10], where developers do not usually interact merely to socialize, as in conventional social networks. In contrast, they interact and collaborate primarily through shared project artifacts.

Communication is an important factor for software projects to succeed. There are many factors that make communication and collaboration among developers a difficult task. These factors involve: geographical, cultural, time-zone, or language barriers. As teams become more and more geographically distributed, collaboration becomes more critical and difficult to achieve. Team members usually use different tools to communicate and that may lead to difficulties to collaborate and finish their tasks.

There are hundreds of communication applications that are used by software development communities and these can be combined to support distributed development teams [10, 11]. SDEs integrate the development environment and the social network in a single environment, and support the interaction between the social networking features and the communication and collaboration tools. SDEs add value because they touch on the social and presence elements of software development. An SDE aims to help users maximize their productivity, and to help distributed team members maintain a level of social awareness regarding other team members’ roles, activity patterns and contributions to the project, as well as the resources of the project. In general, the purpose of the SDE is to provide a frictionless
environment for software development by eliminating the need to switch between different tools in order to perform their various solitary and collaborative tasks [9].

To summarize, SDEs add value to the development community by integrating features that help establish and support interaction among distributed developers, strengthen social communication and bonds, build trust, and have the ability to network. SDEs are a category of CDEs where the system provides a fully featured social network for distributed developer communities. In an SDE, developers can browse profiles, create development projects, and form professional teams. In addition, SDEs provide views for listing people and groups, project data, activities, etc. all from inside the environment. By using these views, users will be able to request help and assist others in their projects, and form strong social bonds among the community members. In general, an SDE is an environment that allows distributed developers to work hand in hand and also have the ability to network. It is a combination of social networking and collaborative software development [9].

3.4. SDE Systems

In this section we briefly examine a number of development environments, both commercial and research oriented, that have been the focus of attention in the CSCW and distributed development communities.

MydeveloperWorks: A professional network and unified set of community tools for connecting, sharing, and collaborating. It is introduced by IBM to facilitates a new way for software developers and students who are distributed all over the world to communicate, interact and connect with their fellow developers and to help them in developing skills, solving problems, collaborating with their peer and staying aware of the latest advances and trends in technology. It allows communicating and interacting with others using feeds, tags, bookmarks, blogs, groups, discussion forums, wikis, and other resources, all from their own profile [14].

Jazz is the most widely-publicized social development environment that aims to foster collaboration within the group by adding a set of collaboration tools and features to the Eclipse IDE to support coordination, communication, and awareness among team members [15-20].

SourceForge.net\(^1\): is an open-source SDE that focuses on development of open source software. It connected with Ohloh, a social network for open source software communities that collects data about software and developers from the source code. It allows developers create their own profiles, provides them with useful information about the software projects, and makes it easier for developers to find experts and request help [12, 21, 22]. SourceForge aims to unify disconnected teams and tools, and help foster collaboration while improving project efficiency and teams’ productivity [23].

GitHub\(^2\) is a web-based hosting environment for software projects that allows developers collaborate in their project artefacts. They share code, watch others activities, and search for others with common interests and expertise [24]. GitHub includes social networking features such as: followers, feeds and blogs, and shows developers’ activities on the repository versions [22, 25, 26, 27, 28].

CollabVS [24] is an SDE based on Microsoft Visual Studio that allows developers to work together whether planned or ad-hoc [22, 29, 30].

---

\(^1\) SF: http://sourceforge.net/

\(^2\) GitHub: http://github.com/
Zembly [31] is an online web-based development environment. It allows developers to edit, test, debug, and document their shared code and from within the browser. It is more than just an on-line editor, it is a development in a social context aimed at social tools, where users can do social programming, and develop applications with other people using social networking type features. Developers are able to invite team members and project partners for collaboration. Zembly supports the developers with different channels to communicate and be aware of others changes to their projects’ artifacts [32, 33].

SCI [9, 22, 34] is an SDE that supports developers’ collaboration, and interaction. It allows them to be aware of the actions of others in real time, aware of artifacts changes, avoiding coding errors, and potentially improves the collaboration during the software development process. It is an environment that combines: communication, collaboration, awareness, and social networking and online presence features from inside a single environment.

Codebook is a social networking systems that helps people to be “friends” not only with other people, but also with the work artifacts they share with them [35]. It helps software engineers to maintain communication, collaboration and connections with friends, and enables awareness, especially at a distance. Also, It helps to keep track of task dependencies, discover and maintain connections to other teams, and understand the history and rationale behind the code that they work on and use [36].

StackOverflow 3 is a community site where developers and programmers can ask and answer each others’ questions. Users can establish profiles and re-tag questions or voting to close a question.

The SDE tools presented previously in this section are categorized in Table 1.

4. A 5-Star Model

This section discusses SDEs focus, interests, and visions about such environments. It introduces a model named 5-Star, which describes the structure and components of the SDE tools. This model is designed to allow researchers, CSCW and groupware experts, and software engineers to evaluate, rate, and compare their collaborative tools.

The model reflects the categories and services nature of the SDEs environments. The services provided to developers are rated from one to five where each category represents a star. The name (5-Star) reflects our believe that SDEs are the environments where developers are looking for comfort, and it resembles their five stars hotel. Each edge of the star shows a

3 Stack Overflow: www.stackoverflow.com
category of the tools and features supported by the environment, and presents the developers’ expectations of such environment. These categories are: 1) Communication; 2) Coordination; 3) Collaboration; 4) Awareness & Presence; and 5) Social networking (SN) tools (See Figure 4).

The above categories combine a list of requirements for collaborative and group-supported environments suggested by some pioneers in this field, such as Booch and Brown [12], Fournier [37], Broday [38], and others. Also, it includes a list of awareness and presence types which researches believe that it is needed for a successful collaboration. Greenberg et al., [39] suggests that people work in teams and collaborate with their groups maintain the six types of awareness listed Table 1.

<table>
<thead>
<tr>
<th>Communication (α)</th>
<th>Jazz</th>
<th>OtherWorks</th>
<th>Collab/S</th>
<th>Scl</th>
<th>Zembly</th>
<th>CoEd</th>
<th>StockDesk</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video conferencing</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VoIP/Audio chat</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instant messaging</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coordination (β)</th>
<th>Jazz</th>
<th>OtherWorks</th>
<th>Collab/S</th>
<th>Scl</th>
<th>Zembly</th>
<th>CoEd</th>
<th>StockDesk</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task management</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code version</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calendering</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Collaboration (δ)</th>
<th>Jazz</th>
<th>OtherWorks</th>
<th>Collab/S</th>
<th>Scl</th>
<th>Zembly</th>
<th>CoEd</th>
<th>StockDesk</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous editor</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous debugger</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forums, Wikis, Blogs</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shared whiteboards</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documents co-browsing</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Awareness &amp; Presence (γ)</th>
<th>Jazz</th>
<th>OtherWorks</th>
<th>Collab/S</th>
<th>Scl</th>
<th>Zembly</th>
<th>CoEd</th>
<th>StockDesk</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informal</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group-Structural</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social/Presence</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Historical</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workspace/Artifacts</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SN Tools (σ)</th>
<th>Jazz</th>
<th>OtherWorks</th>
<th>Collab/S</th>
<th>Scl</th>
<th>Zembly</th>
<th>CoEd</th>
<th>StockDesk</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friends</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help/Support/Experts</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ∑ R_i                    | 23   | 20         | 18       | 23 | 17     | 19   | 17        | SED=10  |

Table 1. Categories and Features of Existing SDE systems. [✓: Not Supported, ~: Supported & Not Integrated, √: Supported & Integrated]

In this model, each category is given a particular rate relative to their importance to the development community, and their role in the environment. The final 5-star value of each tool is the sum of all weights times their category rates (see Equation 1). In this article, the category rates (α, β, δ, γ, σ) are assigned an equal value (0.2). The reason for this is because we believe that, in such a tool, all the categories complement each others, and they are with
the same value (significance) to the environment. This is relatively, and others might find that some categories are of more importance than the others, and assign them different category rates. Also, this depends on: 1) the targeted community; 2) the ways those tools are used and their purpose; and 3) what software development phases they support.

The universal model can be written as

\[
star\_value = \frac{\sum_{i=1}^{5} (w_i \times category\_rate)}{5}
\]  

(1)

For example a simple rating equation for a specific tool is as follows, where each category is multiplied with its associated weight \((w1...w5)\).

\[
star\_value = (\alpha \times w1 + \beta \times w2 + \delta \times w3 + \gamma \times w4 + \sigma \times w5)/5
\]  

(2)

**Example:**

Assume the tools A and B have the weights \((4, 4, 6, 6, 3)\) and \((3, 4, 2, 4, 4)\) respectively. By implementing the universal model the \(\text{star\_value (A)} = 0.92\) and \(\text{star\_value (B)} = 0.68\), which means that A is a better collaborative tool than B.

From the above, someone could argue that awareness, social network tools, communication, and collaboration are interleaved, can’t be separated, and share common characteristics. Also, we can conclude the following facts:

The 3Cs (communication \((\alpha)\), coordination \((\beta)\), and collaboration \((\delta)\)) foster awareness.

\[
\alpha + \beta + \delta \implies \gamma*
\]  

(3)

The awareness \((\gamma)\), social network tools \((\sigma)\), and communication \((\alpha)\) features support interactions between developers, enable better collaboration \((\delta*)\) among them, and improve their productivity.

\[
\gamma + \sigma + \alpha \implies \delta*
\]  

(4)

From the above we can conclude that any SDE can be represented using the following vector.

\[
\text{SDE} = (3C, \gamma, \sigma)
\]  

(5)

Where 3C is a term represents Communication, Coordination, and Collaboration.

5. Applying the Model

In this section, we selected the seven different tools discussed in Section 3.4. These tools are introduced in Table 1, and they represent the most popular and commonly used SDEs. These systems are both desktop \((e.g.,\ Jazz, \text{DeveloperWorks, CollabVS, and SCI})\) and web-based applications \((e.g.,\ Zembly, \text{CodeBook, and StackOverflow})\), and they are rated according to the suggested categories and their service support to the development community.
From Table 1, we conclude that a successful SDE (5-star environment) is the environment where:

(i) The IDE is fully integrated within the environment.

(ii) Major communication tools such as 1) Audio chat; 2) Email; and 3) Instant Messaging (IM) are supported and integrated.

(iii) Source code version and information management tools are integrated.

(iv) Social media and SN tools such as 1) Wikis; 2) Forums; and 3) Blogs are integrated. They help supporting the awareness of others, their activities, and encourage and supports the social side of their interactions.

(v) Users find it easy to collaborate in their coding problems using synchronous editors, compilers, and debuggers.

(vi) Integration within a virtual environment supports the online and social presence.

6. Discussion and Summary

The SDE’s discussed in this section illustrate just the beginnings of this genre of software tools. The cited and existing projects implement a subset of social networking. Few development environments aims to support the full social network features, but do not fully integrate social networking features within their IDE. The tools and literature in this article comes relate to the CSCW and CDE fields.

Distributed developers always feel the fact to interact, and require environments to ease, encourage, and support their collaboration. Conventional development environments inherent some limitations and are not able to provide the distributed development community of successful collaborative environment. On the other hand, SDEs support have the potential to raise the level of awareness, presence, social interaction, and communication between such developers [22].

There is great promise in exploring tool support for the social side of software development. Collaboration plays a crucial role in software development. For this reason, continuing to improve the collaborative development environments is of great potential benefit. Collaborative tools can be used alongside a non-collaborative IDE, but integration adds qualitative and quantitative awareness information and reduces the cost of collaboration during the development process, particularly for distributed teams, and reduces the friction that results from switching among different tools [9].

In addition to supporting distributed communities’ work effectively, we believe SDEs will add better benefit to the development communities, help them increase their productivity, and produce better software products.

There exist survey publications of existing tools in this area [48-50]. But, to our knowledge, the proposed evaluation model is the first and only proposal to evaluate, and compare the social development tools based on their fetures support to the development community. There are no existing related research the way introduced in this article. The model can be tailored and improved to fit all the tools that fall under that CSCW and groupware umbrella.

A future improvement direction will focus on updating the features and tools listed under the model five main categories, and use different category rates to each category depending on their importance to the compared tools.
References


Authors

**Hani Bani-Salameh** is an Assistant Professor in the Software Engineering Department at The Hashemite University, Jordan. He holds a B.Sc. in Computer Science, a M.Sc. in Computer Science from New Mexico State University (NMSU), and a Ph.D. in Computer Science from the University of Idaho (UI). His research interests include software engineering, computer supported cooperative work (CSCW), software development environments, collaborative software development in virtual environments, and social networking and social media. He studies social interactions in social networks and online environments.

**Sana’a Alwidian** is a full time lecturer at The Hashemite University, Jordan. She received her B.Sc. degree in Computer Information Systems from Jordan University of Science and Technology, in 2007 and her M.Sc. degree in Computer Science from Jordan University of Science and Technology, in 2009. Her research interests include Computer Networks, Natural Language Processing, Distributed Systems, Software Engineering and Information Security.

**Maen Hammad** is an Assistant Professor in Software Engineering Department at The Hashemite University, Jordan. He completed his Ph.D. in computer science at Kent State University, USA in May 2010. He received his Master in computer science from Al-Yarmouk University, Jordan and his B.S. in computer science from The Hashemite University/Jordan. His research interest is Software Engineering with focus on software evolution and maintenance, program comprehension and mining software repositories.

**Fadi Wedyan** is an Assistant Professor in Software Engineering Department at The Hashemite University, Jordan. He completed his Ph.D. in computer science at Colorado State University (2011). He received his Master in computer science from Colorado State University (2008), and his B.S. in computer science from the Yarmouk University/Jordan. His research interest is software testing, aspect-oriented testing and development, and static analysis.