



Henri Korvela

FORUMS AND BLOGS AND  
WIKIS, OH MY! – AN  
ANALYSIS OF ON-LINE  
SOURCES OF SUPPORT FOR  
END-USER DEVELOPMENT

TURKU CENTRE *for* COMPUTER SCIENCE

TUCS Technical Report  
No 1032, December 2011





# FORUMS AND BLOGS AND WIKIS, OH MY! – AN ANALYSIS OF ON-LINE SOURCES OF SUPPORT FOR END-USER DEVELOPMENT

Henri Korvela

Åbo Akademi University, Department of Information Technologies  
Joukahaisenkatu 3-5 A, Turku 20520 Finland  
hkorvela@abo.fi

TUCS Technical Report  
No 1032, December 2011

## **Abstract**

End-user developers are often left to their own devices when trying to support development efforts. One place it would seem natural to look for support would be the Internet where several different sources can be found. On-line sources could alleviate several issues current support for end-user developers suffer from.

This paper presents, analyses and evaluates the main on-line sources, such as, manuals, wikis, blogs, software libraries and virtual communities with a focus on the needs of end-user developers. The analysis is based on four factors derived from existent literature on end-user development, context, cooperation, interactivity and immediacy that seem important for end-user developers.

The on-line sources have strong and weak support for different factors and as such we suggest some sources will fit the needs of end-user developers better than others. Of the sources explored in this paper, the virtual communities seem best with many benefits for the end-user developer, e.g. the strong support for contextual information and interactive functions. As such virtual communities could be instrumental in providing support for end-user developers currently and in the future.

**Keywords:** end-user development, on-line support

**TUCS Laboratory**

Laboratory for Data Mining and Knowledge Management

# 1. Introduction

End-user development (EUD) means that the software tools' principle users, i.e. the end-users, have created or designed the software tools themselves. The definition of an end-user developer varies somewhat in the field, but it usually is a variation on the theme that the user doing the development is not formally trained to do so, or the development is not the main focus of their efforts. Lieberman et al. (2006) formulate it thus: "EUD can be defined as a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact". As such end-user development is often done by people not trained as developers doing activities that are secondary to their main goals. End-user development seems primarily intended to support other work or activities and does not form the main focal point for the effort (Ko and Myers 2005; Nardi 1993; Sutcliff, et al. 2003). They are a tool, not a means.

Despite its common status of "not quite important" end-user development is a very common practice found in most organisations, whether they range from small one person companies to large multinational corporations. Perhaps the most common instance is spreadsheet and database applications (Scaffidi, et al. 2005). Furthermore, if using a broader definition of end-user development the estimate in Scaffidi et al. suggests that there would be around 55 million, potential and actual, end-user developers (in the United States alone). This would mean that the population of end-user programmers and developers outnumber professional programmers many times over. In addition to the more traditional areas of end-user development, such as spreadsheets and databases, web design and related activities are relatively new areas where many more end-user developers can be found. Fischer (2009) mentions end-users as an important part of the so called "Web 2.0" world. In fact, it is entirely reliant on user-provided content though not always actual end-user software development. Increasingly regular software also allows for extensive customisation that is pushing the envelope of customisation and parameterisation more towards being out-right end-user development e.g. through the introduction of macros and scripting. In all of these cases users are gradually taking on the role of developers and consequently facing a growing need to support these new tasks.

At some point a developer will run into a problem they do not know how to solve and will then need support. Different support sources have differing properties and functions just as developers have different abilities and needs. However, it seems there is not a very good fit between traditional sources and the needs of end-user developers. The Internet provides access to some old and a few new sources that could help alleviate some of the lack of support end-users face. The lack of development training would likely mean they are more reliant on whatever sources of support are available to them. While the end-user developer can be of any skill level the novice users are those in most need of support. Therefore, this paper looks at the issue from the point of view of the end-user developer as a novice developer of primarily spreadsheets.

In this paper, we will present, analyse and evaluate the main support sources found on-line and their usefulness for the end-user developer. The analysis and evaluation will be based on four factors derived from literature and observation, which it is argued, would

be important to end-user developers. The goal is to provide a more comprehensive understanding of effective support for EUD activities.

The paper is structured as follows. Section 2 contains a literature review. Section 3 introduces the concept of support for end-user development; available sources and defines four factors that can determine the choice of what source is used. Section 4 explores on-line support, what sources exist and their uses for end-user development. The paper concludes with an evaluation on the suitability of the different sources for end-user development support.

## 2. Literature review

End-user development is a knowledge-intensive process combining user-domain knowledge and computer knowledge. For the developer this process of information seeking and problem solving has the goal of achieving *actionable knowledge* (Cross and Sproull 2004), which in this case is represented by an application to solve a problem or task. Accessing support further complicates the process as this is an additional process of knowledge seeking in the domain the developer is often weaker in, i.e. computers. The constructivist perspective as described by Cross and Sproull fits well the domain of end-user developers where a more holistic approach to development with strong focus on context (Repenning and Ioannidou 2006) and iterative development (Repenning and Ioannidou 2006; Brandt, et al. 2008) makes the developer very much like Clarke's "opportunistic developer" who:

- Writes code in an exploratory fashion.
- Develops a sufficient understanding of a technology to understand how it can solve a business problem.
- Prides themselves on solving business problems.

(Clarke 2007)

Previous research has focused on the use of computer support *in general*, (e.g. Carr 2008; Constant, et al. 1996; Govindarajulu, et al. 2000; Govindarajulu 2002; Govindarajulu 2003; Nilsen and Sein 2004; Seeley and Targett 1997; Shaw, et al. 2002; van Velsen, et al. 2007) often within one organisation. Studies on support sources often focus on the characteristics of one type of source in a general setting (e.g. Phang, et al. 2009; Purchase and Worril 2002) or on a certain tool (e.g. Stylos and Myers 2006).

This paper contributes by analysing and comparing support for end-user developers from their perspective and needs including sources that normally lie outside the bounds of a single organisation.

## 3. End-user developers' preferences for support

The need for support has existed for as long as end-user development itself. The support sources are often split into two categories namely, formal and informal sources. Some examples of formal sources are: helpdesk, manuals, application help function and

vendors. Formal sources could be considered as being structured, safe and official. Informal support usually means the user's social network of colleagues and friends or family and is often considered a less reliable source of support. Additionally many organisations have local IS/IT staff which are people from the IS/IT departments placed in other departments or business functions to provide local support. Sometimes other colloquialisms are used such as super user or power user. These power users are people who as part of their work tasks provide support for other people, as they are recognized experts on certain applications (Nardi 1993). As the practice of local IS/IT staff and power users can be formalised to various degrees they can be seen as a continuum. At one end of the spectrum are IS/IT staff originating from the IS/IT department, with a technical background and at the other power users who are domain experts on the tools of their function and business processes. There are therefore many potential sources of support available for an end-user developer.

In this section, based on the literature, we investigate 1) whether end-users prefer one form of support over another, 2) which factors impact the choice of support and 3) the reasons why users would use on-line sources of support.

### **3.1. Do end-users prefer one form of support over another?**

End-users generally have several types of support available, but what do they use and do they prefer one type of support over another? Normally this is a question of whether they use the *formal* sources or turn to *informal* ones. Literature indicates some ambiguity to the question of what support users prefer, or at least that they eventually decide to turn to. End-users place a high importance on peer support. Govindarajulu (2003) notes that all user groups ranked friends the most preferred support method. Furthermore, in (Govindarajulu 2002) friends and local IT support were preferred over helpdesks. While in (Seeley and Targett 1997) executives mentioned contacting colleagues for assistance in most cases. However, even though many studies mention the importance of informal support, middle level managers preferred local IS/IT staff and information centres over informal support (Govindarajulu, et al. 2000). Cross and Sproull (2004) observe that most managers mention people as important sources of information rather than computerized knowledge repositories that were promoted in the organisation studied and also note that managers almost never mentioned these knowledge repositories or other impersonal sources. Segal (2007) notes developers making personal contacts their first port of call.

Instead of asking whether some type of support is more or less preferable it would seem more fruitful to look at why users choose a certain source of support. Govindarajulu et al. (2000) suggest that user attitudes toward a source of support will influence their choice more than subjective norms. If formal support is found to be useful, responsive and knowledgeable it will be used. If not, users will find other sources. But how are these attitudes formed? The requirements for support seem to depend on the characteristics of the users themselves. E.g. age, gender, computer self-efficacy and computer skills have potential to influence the choices of support source.

To be able to use Internet sources people would need to be comfortable with computers and using the Internet and search engines (Liaw 2002). Liaw also shows that skill is related to usage, higher computer skills increases confidence in using computers. Gender could also be a factor, either directly or indirectly. Nilsen and Sein (2004) mention gender and computer self efficacy as likely to influence the choices of support source. Males are more comfortable with computers and the web (Liaw 2002) and gender impact areas of end-user development such as debugging (Beckwith, et al. 2006) and self-efficacy in end-user developers (Beckwith, et al. 2007). Thus, previous research suggests that gender, self-efficacy and computer skill all impact the use of computers and Internet (Beckwith, et al. 2006, 2007; Grigoreanu, et al. 2006, 2008; Ko, et al. 2011; Liaw 2002). Age can have an impact as people who have grown up with technology are more familiar and comfortable with its use (Brown 2002). Proximity (both mentally and physically) to the user can also be an important factor (Govindarajulu, et al. 2000; Nilsen and Sein 2004).

Informal support in many cases seems the default source, probably because it can be more accessible to the user. Other support sources likely need to show a benefit to be considered by users. Thus, what the preferred support source is and the sources used will vary according to the end-user's characteristics and expectations, as well as the properties of the support sources themselves. There is unfortunately no "one size fits all" solution, which is important when considering different types of sources. Different users can and will prefer to use different sources. As Klann et al. (2006) notes, end-user developers are a very heterogeneous group. Some factors are prevalent for all types of information seeking such as gender, job type and relation to the source (Cross and Sproull 2004). Like Cross and Sproull we attempt to model characteristics of knowledge seeker, source and relationship simultaneously. Likewise the sources have very different characteristics and concepts such as "ease of use" and "ease of access" have widely different meanings for an electronic version of a paper manual versus a virtual community. With the great many factors potentially impacting choice of source we asked: *are there any factors unique to or particularly interesting with regards to the end-user developer?* Reviewing the literature it would seem that there are indeed a set of factors particularly suited for the end-user developer's pursuit of actionable knowledge. The next section will describe these factors that seem to be important in supporting end-user development.

### **3.2. Factors impacting choice of support**

As we have seen end-users differ greatly in their properties, abilities and thus likely in their needs. End-user developers are a very eclectic group with little in common beyond their need to personally develop their software. However, there are some themes that seem to be shared across most end-user developers when discussed in the literature. Following is an attempt to summarize these themes in to the four factors of *context, cooperation, interaction* and *immediacy*.

**Context** represents the domain the developer works within, the language by which the developer understands his or her field and the motivation to make the development effort. It represents the reason the end-user puts down the effort. The application has no value in itself only whatever value can be gleaned from its use. An end-user developer

will need to be able to recognize a future benefit so as to motivate the development effort (Blackwell 2002; Blackwell and Green 1999; Sutcliff, et al. 2003). Costabile et al. (2004) discuss the importance of context, e.g. the dialect a domain experts use in their work. Syntonicity (Papert and Harel 1993), being able to put oneself into the context, is important in end-user development (Repenning and Ioannidou 2006). End-user developers often know what they want to do, but are constrained by the how. The development effort is often a secondary activity to the end-user and attention is not focused on the implementation used in the environment (Ko and Myers 2005; Nardi 1993; Segal 2007; Sutcliff, et al. 2003).

It should also be noted that context here implies information easily available or visible to the user. A large database, while containing the information sought, will not support context if the end-user is unable to find it or sift it from all the other information. This also relates to immediacy as the speed at which information can be digested could be considered part of providing a timely answer.

**Cooperation** can play a part in successful development. It allows users to pool their skills and developing spreadsheets is often a collaborative work effort rather than an individual effort. Nardi, in conjunction with other researchers, presents the importance of cooperation for successful development in spreadsheets and other EUD. These findings show that developing spreadsheets is often a collaborative work effort rather than individual effort. (Gantt and Nardi, 1992; Nardi and Miller, 1990; Nardi, 1993)

Repenning and Ioannidou (2006) suggest that building community tools is beneficial to end-user development and Ko and Myer (2005) describe programmers contacting more expert users forming “informal apprenticeships”. Cooperation could also be considered involving a touch of human interaction and so simply appeals to us as social beings, meaning help involving other people are preferable simply because it involves interaction with other humans. As mentioned above there is certainly a strong bias towards accessing personal sources.

**Interactivity** in end-user development means that the developer can more or less directly see the cause and effect between code and action performed. The end-user’s somewhat tenuous grasp of the development environment means they are not apt at predicting behaviour of code. It also indicates a close relationship with development and execution. (Ko and Myers 2005; Nardi 1993; Segal 2007)

A recent example the author experienced in practice was an end-user developer running an Excel macro in a slower visual execution mode as he put it: “so I can see what happens”. Brandt et al. (2008) discuss the short edit-debug cycles used by subjects. Repenning and Ioannidou (2006) suggest end-user development tools should support incremental development and allow for immersion.

**Immediacy** is the ability to act, exactly when the developer wants to. Repenning and Ioannidou (2006) suggest the developer experiences a flow (Csikszentmihalyi 1990) state of mind, immediacy represent the need to get back into the flow. Arias et al. (2000) note the importance of the ability to act and Ko and Myers (2005) mention the users’ wish to act or react to the situation at hand. The end-user seems to be very much “in the moment” when developing. This can to some degree be observed on discussion forums where people often mention the urgency of the issue or frequently reply to their

own posts with short intervals (to draw attention to it by being “at the top”). As mentioned above about context immediacy is also being able to make use of the information. If the user is overloaded with information then the feel of being able to act rapidly will diminish. People also solicit multiple sources simultaneously to increase the audience and hence likelihood of getting a timely reply.

### 3.3. Why do end-users use on-line sources?

In many cases the end-user developer has nowhere to turn to for support. The helpdesks and other IT support may not be able or willing to provide development support. Govindarajulu et al. (2000) mention that support for development was not part of the responsibilities of the local IS/IT support function. Furthermore, in (Govindarajulu 2002) the helpdesks appear to provide only limited support for end-user development activities. Alternatively support is completely lacking. Such may be the case for many small organisations that may not have the resources for a formal support structure, such as a helpdesk. Many end-users are not part of any organisation at all. As a result many developers are left with informal support from their social network as their only option.

Friends, colleagues or local IS/IT support has had some success from a satisfaction perspective, in a wider context, but they are not always able to provide adequate and reliable advice in all situations. It is fairly likely that the user’s social network is on par with the user in terms of knowledge.

Compared to earlier decades today’s developers have a virtual (figurative and literal) treasure trove of information in the form of the Internet. Using the Internet as a channel allows an end-user developer to access support sources that overcome many of the obstacles that often make traditional sources inadequate for many end-users. Most if not all the traditional support sources are available on-line in one form or another.

| Off-line source                               | On-line source                      |
|---|-------------------------------------|
| Magazine articles, advice from social network | Blogs and “tips & tricks” web-pages |
| Software libraries                            | Software libraries                  |
| Personal contacts                             | Virtual communities                 |
|   | Internet search                     |
| Trial and error                               |                                     |
| Manuals, books                                | Manuals, knowledge bases, wikis     |
| Application help function                     | Knowledge bases, wikis              |

Table 1. On-line and corresponding off-line support sources.

Manuals and technical information as well as software updates (if applicable) are increasingly found on-line, sometimes exclusively so. Many organisations have an on-line version of their helpdesk where you can post questions or access other available support resources. It is also possible to reach colleagues, friends and family through

various methods such as e-mail, chats and forums. It should be noted that most of the on-line sources have off-line counterparts (see Table 1).

What makes the on-line sources special is usually that they can use features brought in by the medium that enhance their function.

## 4. Analysis of the different on-line support sources

The support sources are often split into two categories namely, *formal* and *informal* sources. In addition to this the on-line sources can be further categories based on their primary function: *sources that act as knowledge repositories* and *sources that facilitate communication* between people.

The **knowledge repositories'** main function is that they provide information for other users, though some of these have a communicative function as well. The support sources in this category have information content as the main focus. They contain a wealth of information of various levels, ranging from quick answers to simple questions to advanced technical information and source code. Some examples of knowledge repositories are:

- Manuals, knowledge bases and wikis
- Blogs and “tips & tricks” web-pages
- Software libraries

**Communicative support sources** primarily function by enabling communication, but they can also have some informative function when the record of the communications are stored and made available. These sources facilitate communication between people allowing them to engage in support activities. Some examples of communicative sources are:

- Discussion forums
- E-mail discussion lists
- Social networks

### 4.1. Formal knowledge repositories – manuals, knowledge bases and wikis

There are several forms of knowledge repositories online, e.g. manuals, knowledge bases, wikis or just plain information. It is common that vendors provide on-line manuals for products in electronic form, often a pdf version of the paper manual (if such exists). In this basic form there is little difference to the paper version, except for the ability to keep it updated with corrections and other modifications more easily.

A form of manual that uses the capabilities of the medium more directly is an electronic manual in the form of a living (hypertext) document or wiki. This uses the hypertext capabilities to link parts of the document together, and it can be seamlessly updated

with new information. Lists of frequently asked questions (FAQs) along with answers are also commonly found on-line.

Another common instance is a knowledge base. This is a database that has information on known problems and workarounds or answers to question users may have, in essence an advanced FAQ system. You can do text searches in the knowledge base to try and find solutions to problems. Some software programs can link directly to knowledge bases based on an error code or use the code as a search parameter. A knowledge base has many of the same limits that the application help function has with regards to finding and extracting information.

Most of these sources are formal support from the software provider due to the nature of the source (technical information), but a wiki can easily be semi or completely unofficial. Using these on-line sources makes it easy to distribute information and to keep it current for all potential users to benefit. Whereas distributed help (manuals, application help function) is current to the situation when it was written. It is also possible to use the medium to further expand the available support by linking to other forms of help, e.g. on-line tutorials.

Context, cooperation, interactivity and immediacy are all weakly supported by information resource sites. Context will not be focused on the end-user developers interest, but instead be as general and broad as possible to appeal to the largest audience. Manuals and knowledge bases are not inherently cooperative, though wikis are. However, this cooperation in wikis may not exist in way that supports the end-user developer, particularly if it is an official wiki, which likely limits the editing permissions. While these sources have some interactivity and can be changed and expanded easily they do not change in response to a developer's specific problem. The large amount of information contained and the associated issues with finding the right information means immediacy is usually weakly supported. They do provide a wide range of information for most users, though it may be too technical or not specific enough for end-user developers.

## **4.2. Formal knowledge repositories - software libraries**

Some programming languages have many thousands of built-in or externally produced objects, classes or functions. These extensions can be utilized by a developer in their own applications. For this purpose professional programmers can use (on-line) software libraries. A software library is structured repository of information containing code (usually in modules) that can be used by developers as a part of a user developed application. A software library will have a wide range of options (i.e. different code modules) available and can thus provide extensive support to developers providing solutions to a wide range of problems.

The code needs to be fitted into the developer's work, so the developer faces the problem of adapting code found in the library to her needs. The developer needs some knowledge to be able to do this, something end-user developers often lack. Ko and Myers (2005) mention how end-users had difficulties understanding what and how the code worked, even code they themselves had previously written. They also note that

end-users will tend to learn just enough of programming to do what they want instead of a whole process as the programming is not the end-users primary concern.

Another problem is finding the code in the database. Mili et al. (1998) note that despite considerable effort problems associated with finding code in the library remain. Belkin (2000) discusses the general problem of finding information in large information systems, something that is relevant for software libraries. Users of the software library cannot formulate queries correctly if they do not know the correct syntax used. The correct syntax is dependent on what kind of structure the software library has (Mili, et al. 1998). This relates to the so called vocabulary problem (Furnas, et al. 1987). These two problems are often interconnected as you need to know the exact terms used (vocabulary problem) and in what order to use them (syntax problem) to be able to fully utilize a software library. A developer would usually need to know or be able to guess what a function is called in the development environment to be able to find it in a software library.

A software library can provide a wide range of solutions and can be a valuable tool for those able to make use of them. However, context is weakly supported as the developer is faced with a significant attention investment. Similarly immediacy is weakly supported as syntax and vocabulary needs to be understood and searching strategies developed based on them disengaging the user further from the development process. A software library supports cooperation and interactivity only weakly if at all.

### **4.3. Informal knowledge repositories – “tips & tricks” web-pages and blogs**

A “tips and tricks” site is a place that contains a number of clever and simple solutions that people are likely to be interested in (or that the author thinks are interesting). Similarly a blog can contain pieces of code or ideas the author thought was interesting. Often these will consist of short pieces of code that can be copied and used by other developers. They will be simple but clever tricks, e.g. there are (or at least used to be) many places that present HTML and/or Java Script “tips and tricks” to help budding web developers. Common advice on such sites is how to create different effects when the cursor is scrolled over a link or an image. Such things are often simple to create if you only know the correct syntax to use. These sites can also contain more exotic code that is not extensively covered in manuals or the application help function or have interesting workarounds that the author has discovered.

In essence these are akin to software libraries (they contain code for you to copy and use) except that they are not as extensive and likely not organised as rigorously. The code will generally work with little modification and is usually self-encompassing and straightforward to use, literally a question of copy and paste. In most cases the user only needs to replace the “your text here” part.

The weaknesses with “tips and tricks” sites and blogs are that they lack advanced code, the amount of code available is small (so only a limited set of problems can be solved) and the capacity for interactive communication is limited. Thus, any support will mostly consist of whatever problems the author has thought of. This means that while the

site/blog often fits the basic end-user's skill level and goals their usefulness tend to be rather limited and cannot be relied upon to consistently provide support.

As knowledge repositories the EUD factors are mostly weakly supported. It is possible to consider context moderately supported as the information is presented more easily and will have some contextual meaning. Immediacy could also be considered moderately supported as any answers should be immediately obvious. Interactivity and cooperation is possible as these sites can support commenting and discussion. However, the nature of these sites lends itself more to discussion of the information already posted rather than new topics. As such we can conclude that, though technically possible, for our purposes interaction and cooperation are not adequately supported.

#### **4.4. Communicative sources – virtual communities**

Internet forums are places where people can post messages creating a platform for asynchronous communication and discussion. A forum is a collection of topics that members can read and post a reply, which can then in turn be comment upon by others, creating a chain of discussion. Email discussion lists are similar to forums and provide essentially the same function. The main difference is the separate channel (email client versus web browser) it is accessed by and that emails are pushed to the user while the user has to pull information from the forum. A social network allows communication like a forum, but usually includes an assortment of other facilities to allow people to connect, share information and interact in different ways. All of these can be considered as different types of virtual communities.

In addition to allowing communication between members a virtual community normally saves the messages that have been posted. Old messages and discussions can be read or even revived with new replies. This means they will also function as a knowledge repository, though due to their communicative focus this function is often less effective compared to regular knowledge repositories and in large virtual communities it can be quite difficult to find information.

As a communicative source a virtual community comes with the benefit of being able to interact with the end-user developer in need of support allowing for a negotiated or mediated approach. Both sides of the process has the ability to interpret what the other is saying and if required ask for clarifications. Neither is bound to a frame of reference fixed at some previous point and can adapt. One example would be negotiating a common vocabulary if one did not previously exist. Using a virtual community is very much an interactive experience. When a problem has been posted the members of the community (the “helpers”) can then either correct the code, provide information on how to solve the problem or where to find the information to do so or even provide alternate solutions.

The interactivity also means the developer can go through several iterations to solve the problem or refine the solution as well as explaining the exact circumstances with real data if needed. In this way context is very much present and this task specific help will likely be very useful for end-user developers. It also makes the supporting activity collaborative. The helpers can explain a difficult concept to the developer or correct code that would have had a different behaviour from what was expected. Thus, a virtual

community is potentially very good at providing support in context for the end-user developer and it can also alleviate problems related to syntax and vocabulary.

A virtual community strongly supports context, collaboration and interactivity. Immediacy can be anything from weakly or strongly supported as the timeliness of any answers will depend on the activity of other users. A hypothetical ideal community can provide quick yet personal support at anytime of the day.

#### **4.5. Other (information) sources**

There are several additional support sources found on-line. In many cases they are not unique for the medium though again their usefulness is greater as an on-line form as they can more easily be updated and linked together with other sources. Some examples include, but are not limited to: *on-line training, tutorials, video/animated demonstrations*. Often these are found as a part of another knowledge repository or communicative source.

In some senses the search engine could be considered a source of support. For the end-user “searching the Internet” is a perfectly valid source of support, even though the search engine itself merely reflects other sources. In this capacity it will be an information source, a knowledge repository with dynamic content reflecting the keywords/search term used. Like most other information sources context, interactivity and cooperation is weakly supported while immediacy will be moderate. While responses will essentially be immediate the large amount of information likely provided by the search engine will require considerable effort to sift through. This will be compounded by the impact of the used terms, i.e. the vocabulary used. However, unlike a software library the user can provide their own keywords and still likely get some relevant responses due to the wider range of sources accessed by the search engine.

The search engine will likely form the initial contact point for most information seekers, providing a way to find the other support sources. It is in fact unlikely that the end-user can find any other source without the search engine (unless directed to them somehow, e.g. a colleague or some other referral). For most intents and purposes the search engine equals the information sources it can find even though it is technically speaking a tool and not a source.

#### **4.6. Hybrid and mixed sources**

While the different types of sources are presented here as separate entities it is common to find several sources combined, or at least co-existing, together in the same place i.e. a website. Modern web technologies makes it is fairly easy, one could even say encourage, the integration of the different functions into a complete package. Most Internet forums have FAQs and lists of information resources available or use techniques such as wikis to share knowledge and code. A good example of combining sources is the website for the PHP programming language at “[www.php.net](http://www.php.net)” that combines a technical manual with the ability to post comments and code related to a chosen topic (normally a PHP function).

As we have noted earlier some of the knowledge repositories can have communicative functions, for example a feature for posting comments enabling discussion. Likewise a communicative source that archives the communications will form a rudimentary knowledge repository. In this way most on-line sources will probably combine features and provide communicative and information resources as well as including several different sources on the same site.

It is perhaps not a question of what type of support one picks but more which site one uses. Such a package will likely create a sum total value higher than its constituent parts. Users will still have different underlying motivators and characteristics that will lead them to those features or sites they feel fit best.

| Sources of support   | Context     | Cooperation | Interactivity | Immediacy     |
|--|-------------|-------------|---------------|---------------|
| Formal knowledge repositories *                              | Weak        | Weak        | Weak          | Weak          |
| Informal knowledge repositories ‡                            | Moderate    | Weak        | Weak          | Moderate      |
| Software libraries   | Weak        |             |               | Weak          |
| Virtual communities†   | Strong      | Strong      | Strong        | Weak→Strong   |
| Personal contacts  | Strong      | Strong      | Strong        | Strong        |
| Internet search  | Weak→Strong |             |               | Weak→Moderate |
| Trial and error  | Moderate    |             | Strong        | Strong        |
| Books, manuals   | Weak        |             |               | Weak→Moderate |
| Application help function                                    | Weak        |             | Weak          | Weak→Moderate |
| *Manuals, knowledge bases, wikis                             |             |             |               |               |
| ‡Blogs and “tips & tricks” web-pages                         |             |             |               |               |
| †Discussion forums, social networks, e-mail discussion lists |             |             |               |               |

Table 2. Degree of support for end-user development factors in sources

## 5. Evaluation of EUD on-line support sources

End-user developers are a very diverse group who are likely to pick those sources that seem to fit their personal preferences. If we can understand those preferences better it is possible to provide better support for people, or at least steer them the right way. It also allows us to some degree to explain why some methods of support are not used very much by users. Table 2 summarizes the properties of on-line sources based on the analysis in section 4. Furthermore, it includes some off-line sources for comparison.

**Software libraries** seem the least appropriate for the end-user developer. Not only does it require understanding of the development language it also requires knowledge about the library itself. This requires a much larger investment of attention than the average end-user developer is likely to feel is justified. There’s also the issue with adapting generic code to the specific problem at hand. On the plus side the software library has solutions for a wide range of issues, but the developer has to find them and be able to adapt them for their own problem.

**Formal knowledge repositories:** *manuals, knowledge bases* and *wikis* similarly contain a wide range of solutions, but suffer from the same issues with formalism as software

libraries. They are geared toward as many people as possible and will therefore provide less contextually relevant support. To fully benefit from these sources the developer needs to have a larger degree of skill to find and apply the information they need.

Considering the similarity in function to the traditional formal knowledge repositories: books, manuals and the application help function, and how unpopular they seem to be there is every reason to expect users to treat these sources similarly to their off-line counterparts and avoid using them.

**Informal knowledge repositories:** *blogs* and “*tips & tricks*” sites have less extensive information, but the information they contain would mostly be easier to absorb for the end-user developer. The knowledge needed to apply them is less as examples tend to be presented in context. Additionally the facility for communication allows for some interaction and thus an option for even more specifically contextual support. This source has some benefits for the less experienced user as the content can be geared more towards the less sophisticated user. While it would seem end-users could take to these sources more readily than formal knowledge repositories the limited availability of information will eventually let the developers down.

**The virtual community** seems to be the most appropriate source for the end-user developer in general. It allows its members to interact and discuss a problem, so the specific context of the developer can be addressed and support can be had for any problem the developer has. Developers can post their problem in their own words and provide sample data or troubleshoot their existing code by posting it. The information to solve even the most complex problems can usually be found. Additionally even a very novice user can make use of the answers as through interaction they can have the solution explained to them. In an extreme case the end-user need to even understand the solution as it can be served to them ready to use based on the data they provided.

As such a virtual community not only has a wide range of solutions, but it allows any type of user to benefit from it. Much of the interesting features of a virtual community come from the interaction of its members of differing abilities and knowledge. This interaction with novice and other advanced users is part of the pay-off for participating in a virtual community. Participating in a community rewards all participants, not only as one would surmise the people being helped. Virtual communities are interesting as they share the same characteristics with the so widely popular personal contacts and as such should be acceptable to most developers. Yet the virtual community provides access to a much broader range of expertise than most persons would have available through their social network. For example for a spreadsheet developer this is as close to having one’s own personal spreadsheet guru by one’s side as most people are going to get.

## 6. Conclusions

In this paper we have presented and analysed/evaluated several on-line support sources for end-user developers. The analysis/evaluation is based on four factors that it is argued would be important for users in determining the suitability and/or effectiveness of each source. As noted earlier virtual communities seem to be a very suitable source of support for end-user developers. With the similarity in properties to personal contacts

(a very popular source) there is some hope that users can be steered to virtual communities and thus benefit from a much broader range of expertise than is normally available from a user's social network.

Research indicates that users in many cases are disappointed by traditional sources and as such do not utilise them extensively. Sometimes the sources are completely absent. Understanding the properties of sources and users and why users use a source is important to furthering our understanding of the relationship between users and the sources they use. This means we can design support to fit the users better. If our users are novice developers then a massive manual or application help-system might not be helpful. Alternatively we can endorse and steer the user to certain types of support. E.g. local IT support may not possess the knowledge to support development in a particular application, but could refer the user to a good virtual community instead. In this way on-line sources can supplement the off-line sources and can also take advantage of the medium to leverage the supporting function to better help the users.

Future research will attempt to validate the end-user development factors among real end-user developers. Furthermore, the relative importance of the factors and how that could impact the use could be explored. It is likely that some factors may be more important than others. A more extensive comparison of on-line versus off-line sources could also be interesting. Studies of both users and providers of information in virtual communities would give valuable insight into the use of on-line support. Many of the on-line support sources rely upon the users providing their time and effort toward the goal of supporting the community. Yet why would users do so? It is something that has somewhat perplexed economists and been a focus of research especially in the context of open source software. Research in other fields suggests that intrinsic motivators play an important part. Future research will thus also focus on motivational questions like why helpers help and what actions an information seeker can take to improve their chances of receiving support.

## References

- Arias, E., Eden, H., Fischer, G., Gorman, A. and Scharff, E. (2000). Transcending the individual human mind - creating shared understanding through collaborative design. *ACM Trans. Comput.-Hum. Interact.*, 7(1), 84–113.
- Beckwith, L., Inman, D., Rector, K. and Burnett, M. (2007). On to the Real World: Gender and Self-Efficacy in Excel. In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, September 23-27, 2007, 119-126.
- Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A. and Cook, C. (2006). Tinkering and gender in end-user programmers' debugging. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 231–240, ACM Press, New York.
- Belkin, N. J. (2000). Helping people find what they don't know. *Commun. ACM*, 43(8), 58–61.

- Blackwell, A. and Green, T. (1999). Investment of attention as an analytic approach to cognitive dimensions. In: *Collected Papers of the 11th Annual Workshop of the Psychology of Programming Interest Group (PPIG-11)*, 24–35.
- Blackwell, A. (2002). First steps in programming: a rationale for attention investment models. In: *IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 2-10.
- Brandt, J., Guo, P., Lewenstein, J. and Klemmer, S. (2008). Opportunistic programming: how rapid ideation and prototyping occur in practice. In: *Proceedings of the 4th international workshop on End-user software engineering*, 1-5, ACM Press, New York.
- Brown, J. (2002). GROWING UP DIGITAL how the web changes work, education, and the ways people learn. *USDLA Journal*, 16(2).
- Carr, C. L. (2006). Reciprocity: the golden rule of IS-user service relationship quality and cooperation. *Commun. ACM*, 49, 77-83.
- Clarke, S. (2007). What is an end user software engineer? In Burnett, M. , Engels, G., Myers, B. and Rothermel, G., editors, *End-User Software Engineering*, number 07081 in *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany.
- Constant, D., Sproull, L., and Kiesler, S. (1996). The kindness of strangers: The usefulness of electronic weak ties for technical advice. *Organization Science*, 7(2), 119–135.
- Costabile, M. F., Fogli, D., Mussio, P., and Piccinno, A. (2006). End-User development: The software shaping workshop approach. In: *End User Development*, chpt 9, Springer Netherlands, Dordrecht
- Cross, R. and Sproull, L. (2004). More than an answer: Information relationships for actionable knowledge. *Organization Science*, 15(4), 446-462.
- Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper Collins, NewYork.
- Fischer, G. (2009). End-User development and meta-design: Foundations for cultures of participation. In Pipek, V., Rosson, M., de Ruyter, B., and Wulf, V., editors, *End-User Development*, volume 5435 of *Lecture Notes in Computer Science*, chapter 1, 3–14. Springer.
- Furnas, G., Landauer, T., Gomez, L., and Dumais, S. (1987). The vocabulary problem in human-system communication. *Commun. ACM*, 30(11), 964–971.
- Govindarajulu, C. (2002). The status of helpdesk support. *Commun. ACM*, 45, 97–100.
- Govindarajulu, C. (2003). End users: who are they? *Commun. ACM*, 46(9), 152–159.
- Govindarajulu, C., Reithel, B. J. and Sethi, V. (2000). A model of end user attitudes and intentions toward alternative sources of support. *Inf. Manage.*, 37(2), 77–86.
- Ko, A., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M., Rothermel, G., Shaw, M. and Wiedenbeck, S. (2011). The state of the art in end-user software engineering. *ACM Comput. Surv.*, 43.

- Ko, A. and Myers, B. (2005). Human factors affecting dependability in end-user programming. In: Proceedings of the first workshop on End-user software engineering, 1–4, ACM Press, New York.
- Klann, M., Paternò, F., Wulf, V. (2006). Future Perspectives in End-User Development. In: End User Development, chpt 21, Springer Netherlands, Dordrecht.
- Liaw, S. (2002). An internet survey for perceptions of computers and the world wide web: relationship, prediction, and difference. *Computers in Human Behavior*, 18(1), 17–35.
- Lieberman, H., Paternò, F., Klann, M., and Wulf, V. (2006). End-User development: An emerging paradigm. In: End User Development, chpt 1, Springer Netherlands, Dordrecht.
- Mili, A., Mili, R., and Mittermeir, R. (1998). A survey of software reuse libraries. *Annals of Software Engineering*, 5, 349–414.
- Nardi, B. (1993). *A Small Matter of Programming*. MIT Press, Cambridge, MA, USA.
- Nilsen, H. and Sein, M. (2004). What is really important in supporting end-users? In: Proceedings of the 2004 SIGMIS conference on Computer personnel research: Careers, culture, and ethics in a networked environment, 48-54, ACM Press, New York.
- Papert, S. and Harel, I. (Eds.). (1993). *Constructionism*. Norwood, NJ, Ablex Publishing Corporation.
- Purchase, H. and Worrill, J. (2002) An empirical study of online help design: features and principles. *International Journal of Human-Computer Studies*, 56 (5), 539–567.
- Repenning, A. and Ioannidou, A. (2006). What Makes End-User Development Tick? 13 Design Guidelines. In: End User Development, chpt 4, Springer Netherlands, Dordrecht
- Scaffidi, C., Shaw, M. and Myers, B. (2005). Estimating the numbers of end users and end user programmers. In: Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, 207–214.
- Seeley, M. and Targett, D. (1997). A senior executive end-user framework. *Information Systems Journal*, 7(4), 289–308.
- Segal, Judith (2007). Some problems of professional end user developers. In: IEEE Symposium on Visual Languages and Human-Centric Computing, 23-27 September 2007, Idaho, USA.
- Shaw, N., DeLone, W. and Niederman, F. (2002). Sources of dissatisfaction in end-user support: an empirical study. *SIGMIS Database*, 33, 41–56.
- Sutcliffe, A., Lee, D. and Mehandjiev, N. (2003). Contributions, costs and prospects for End-User development. In: *HCI International 2003*, volume 2. Lawrence Erlbaum.
- van Velsen, L., Steehouder, M., and de Jong, M. (2007). Evaluation of user support: Factors that affect user satisfaction with helpdesks and helplines. *IEEE Transactions on Professional Communication*, 50(3), 219–231.



TURKU  
CENTRE *for*  
COMPUTER  
SCIENCE

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | [www.tucs.fi](http://www.tucs.fi)



**University of Turku**

- Department of Information Technology
- Department of Mathematics



**Åbo Akademi University**

- Department of Information Technologies



**Turku School of Economics**

- Institute of Information Systems Sciences

ISBN 978-952-12-2695-3  
ISSN 1239-1891