Computing Handbook
Information Systems and Information Technology
Heikki Topi, Allen Tucker

Information Systems as a Practical Discipline

Publication details
Juhani Iivari
Published online on: 14 May 2014

Accessed on: 04 Jan 2024

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
3

Information Systems as a Practical Discipline

3.1 Introduction

The purpose of this chapter is to discuss information systems (IS) as a practical discipline or science, “which is conceived in order to make possible, to improve, and to correct a definite kind of extra-scientific praxis” (Strasser 1985, p. 59). Referring to Hassan (2011) but contrary to Strasser (1985), this chapter prefers to speak about a discipline rather than a science in the context of IS.

The practical relevance of disciplines is often discussed as a dilemma between relevance and rigor (Keen 1991, Benbasat and Zmud 1999). A dilemma more specific to IS is how to ensure its practical relevance without condescending into commercial faddism. Much of the IS research agenda still continues to be determined by consultants and gurus, and various fads are part of our scientific language (Baskerville and Myers 2009). Following the commercial culture of the fashion setters, many IS researchers tend to introduce each hype as a radical or disruptive innovation, instead of seeking invariants and continuities between ideas and technologies.

Partly because of this commercial faddism, it is hard to see a greater cumulative tradition in IS research than 25 years ago when Banville and Landry (1989) characterized it as a “fragmented adhocracy.” Due to the technological development, IS research problems and topics have continued to proliferate, theoretical frameworks applied have become more numerous and diverse, and research methods more pluralistic. Precisely because of these factors, IS researchers should, however, resist the temptation to artificially reinforce and amplify fashion waves.

Benbasat and Zmud (2003) attempted to bring some order into this fragmentation. According to them, the core of IS is determined by the information technology (IT) artifact and its immediate nomological net. They discuss their proposal in terms of errors of exclusion and errors of inclusion in IS research. Errors of exclusion refer to studies that do not include the IT artifact or elements of its immediate nomological net. Errors of inclusion, on the other hand, refer to studies that include non-IT constructs—that is, those not included in the nomological net—especially if they are causally distant from those of the nomological net.
Their proposal has received exceptional attention and also severe criticism (see Agarwal and Lucas 2005, King and Lyytinen 2006), much from the viewpoint of the academic legitimacy of IS. The academic legitimacy of IS is hardly a serious problem, at least in Scandinavia or elsewhere in Europe, and according to Mason (2006) not even in North America. Thus, this point of criticism is not very relevant, although Benbasat and Zmud (2003) themselves raise the legitimacy issue as a major motivation for their proposal.

On the contrary, the thesis of the present chapter is that the intellectual core of IS is crucial for the practical relevance. Because of errors of exclusion, IS research has suffered from a weak attention to the IT artifact (Orlikowski and Iacono 2001). Theories are mainly borrowed from reference disciplines. These theories are often totally void of any IT-specific substance. All this has likely diminished the practical relevance of IS research.

Furthermore, the present chapter is based on the conviction that seeking invariants and continuities between ideas and technologies is a more effective way of ensuring the practical relevance of IS than adhering to commercial faddism. There are a number of reasons for this. First, there is no point competing with consultants and gurus by just emulating their behavior. Second, helping practitioners to understand that a new fad is a mutated version of an older idea or just old wine in a new bottle helps them to make better sense of the fad in question. Third, a better understanding of predecessors makes it possible to transfer knowledge and experience from the past and possibly to avoid repeating mistakes. All these can be expected to increase the credibility of IS research at least in the longer run.

The structure of this chapter is as follows. The next section defines the territory of IS among the disciplines of computing, suggesting that the special focus of IS lies on IT applications rather than on IT artifacts in general. After it, the external praxes or practices to be supported by IS research are discussed, identifying three major areas—IT management, IT application development, and IT application usage. Focusing on IT application development, the following section suggests a three-tier model that includes design by vendor, design by client collective, and design by users. Continuing to focus on design, the question of how IS research can effectively support the identified external practices is discussed next. The section suggests that IS research should focus on actionable and, more specifically, on designable qualities of IT applications so that the research outcomes can inform design at different stages of design in the three-tier model. Finally, the implications of the article are discussed.

### 3.2 IT Artifacts and IT Applications

All disciplines of computing—computer engineering, computer science, software engineering, IT, and IS (Shackelford et al. 2005)—are interested in IT artifacts. IT artifacts form the *raison d’être* of their existence. This common interest in IT artifacts makes disciplines of computing—especially the last three of them—sister disciplines. There is a considerable overlap between them especially at the level of base technologies such as data communication, software, databases, and user interfaces, as well as in the case of system development methods, techniques, and tools.

Iivari (2007) suggests that the distinctive focus of IS among disciplines of computing lies in IT applications rather than on IT artifacts in general. As a consequence, IS can be conceived as a discipline of computing that specifically focuses on IT applications, on their development, use, and impact at individual, group, organizational, community, society, and global levels.

IT applications are IT artifacts that have human beings as users and have functionality/capability to produce direct services to their users.* The concept of IT application is close to “application software,” but not exactly the same. First, an IT application does not comprise only software but also the physical device(s) required for the operation and use of the system. Second, it may also include content that is not software by nature (e.g., information content). Third, an IT application may be an instantiation

* The word “direct” attempts to exclude so-called systems software from the category of IT applications. The services provided by systems software are indirect in the sense that they are required for providing the services of IT applications.
of application software (e.g., an enterprise resource planning (ERP)-based information system is an instantiation of ERP software, but ERP software is not an application as itself).

Iivari (2007) also introduces an open-ended typology of IT applications that, in its current version, includes eight types depicted in Table 3.1. The proposed typology is not based on any reference theory. Instead it is a Type I theory in the framework of Gregor (2006)—“theory for analysing” that is specific to IT applications, based on the analysis of their major purposes.

The first four archetypes are close to “technology as a labor substitution tool,” “technology as a productivity tool,” “technology as a social relations tool,” and “technology as an information processing tool” in Orlikowski and Iacono (2001). So, essentially Table 3.1 extends Orlikowski and Iacono (2001) by introducing four additional types. Computer games illustrate the capability of IT applications to entertain. IT applications may also attempt to arouse artistic experience, and one can easily imagine a new sort of art that is essentially built on the dynamic and interactive character of computer technology. IT artifacts such as digital pets can accompany human users. Finally, computers allow users to build digital fantasy worlds and have fantasy-like experiences. Table 3.1 identifies some references to these non-traditional application types.

As ideal types, the IT applications of Table 3.1 may not occur in their pure forms in practice, but one can identify examples that are close to them. Yet, many real IT applications are combinations of the ideal types for several reasons. First, a single IT application may be deliberately designed to include several purposes in terms of Table 3.1. For example, an information system (proper) may be designed to include game features to motivate the use of the system and to support learning in the case of educational IS as exemplified by Age of Empires II: The Age of Kings (http://www.microsoft.com/games/age2/). Second, an IT application of one archetype may include auxiliary functions of another archetype to support of the use of the system. For example, e-mail with the original function of communicating messages includes mailboxes that allow one to build a directory to informate about previous communications. Thus, it can be developed into a fairly sophisticated information system about one’s electronically mediated social network and one’s electronic communication within that network. Third, Zuboff (1988) claimed that to automate also allows one to informate. This can obviously be extended to cover other uses of IT applications, so that computer games, for example, could at least in principle collect information about their use and about users’ actions and reactions during playing. A final point is that applications of one archetype can be used to implement applications of another archetype. For example, spreadsheet software as an augmenting tool can be used to implement a specific information system.

Referring to the proliferation of IT applications, one should note that a sound typology of ideal types also enables to address this increased variety of IT applications. The eight ideal types alone can be

<table>
<thead>
<tr>
<th>Intended Purpose</th>
<th>Examples</th>
</tr>
</thead>
</table>
| To automate      | Many embedded systems  
|                  | Many transaction processing systems |
| To augment       | Many personal productivity tools, computer-aided design tools |
| To mediate       | Computer-mediated communication (CMC): Internet phone calls, e-mail, instant messaging, chat rooms, etc. |
| To informate     | Information systems (proper) |
| To entertain     | Computer games (Aarseth 2001, Davis et al. 2009) |
| To accompany     | Digital (virtual and robotic) pets (Kusahara 2001, Friedman et al. 2003) |
| To fantasize     | Digital fantasy world applications (Davis et al. 2009) |

* The word “artisticize” is introduced to refer to subjecting a person to an artistic experience as a consumer or possibly also as an interactive creator of a piece of (computer) art.
combined in 255 ways. If one adds different application domains (e.g., business, government, healthcare, education, library, and geography), one could have thousands of different application types just in terms of the eight ideal types. It is this variety of IT applications that makes the ideal types significant. Recognizing that many practical IT applications are combinations of ideal types, research can focus on much fewer ideal types, assuming that a deeper understanding of ideal types—their similarities and differences within and across application domains—also informs about the hybrid applications in different application domains.

Related to fashion waves, the first question to be asked in the case of each new (type of) IT application is whether or not it really is so new that it does not have any predecessors and cannot be viewed as a subset or an instance of a more general type of IT applications of Table 3.1. Because IT applications represent artificial reality, it may well be that fundamentally new IT applications will be invented that cannot be interpreted in terms of the eight archetypes of Table 3.1. That is why the typology of Table 3.1 is open-ended. Yet, situations like this are likely quite rare events.

If one accepts that IS is a discipline of computing that specifically focuses on IT applications in Table 3.1, it is clear that the territory of IS goes beyond the workplace use of utilitarian IT artifacts. Although non-utilitarian use is occasionally recognized in the IS literature (e.g., Davis et al. 2009, Lin and Bhattacherjee 2010), much of IS research still continues to assume utilitarian use in the workplace (e.g., Benbasat and Zmud 2003, Benbasat and Zmud 2006, Burton-Jones and Straub 2006, Barki et al. 2007).

### 3.3 External Practices to Be Supported by IS Research

The idea of IS as a practical discipline leads to the need to identify those practices that IS attempts to enable, to improve, and to correct. One can distinguish three interacting focus areas with associated actor groups depicted in Figure 3.1.*

Among three subfields of IT management—IT project management, management of IT function, and IT governance—the last one is introduced to emphasize three points.† First, it can be practiced at the national, regional, and global levels in addition to the organizational level as normally assumed. Second, IT governance is exercised by actors external to IT such as senior management in organizations, top government officials, and political leaders at the national, regional, and global levels. Third, IT governance practiced by these external actors points out that the core of IS should not be interpreted as exclusive. If IS researchers are able to inform these decision-makers in the issues of IT governance, the relevance of their work is likely very high, even though may include some errors of inclusion.

The three subfields of IT application usage are inspired by Barki et al. (2007), who distinguish three behaviors—technology interaction behaviors, task-technology adaptation behaviors, and individual adaptation behaviors—in the case of individual usage. The idea of Figure 3.1 is, however, that IT application usage can be viewed at different levels such as individuals, groups, organizations, communities, and societies. Therefore, the three behaviors suggested by Barki et al. (2007) are renamed aiming at more generality, pointing out that in addition to IT application use, the application may be redesigned in the adopting unit while used, and also the adopting unit may adapt or be adapted to reflect the IT application and its use.

Figure 3.1 identifies three subfields of IT application development: design, implementation, and evaluation. Design is here interpreted to cover also analysis activities needed in design. IS implementation refers to the process of adoption, appropriation, acceptance, instantiation, and/or institutionalization of the designed IT application in the adopting unit in question. Adapting Ammenwerth et al. (2004),

---

*The actor groups in Figure 3.1 may be interested in several areas, but for simplicity they are associated with one.
†IT Governance Institute (ITGI 2003, p. 10) defines IT governance as “the responsibility of the board of directors and executive management. It is an integral part of enterprise governance and consists of the leadership and organizational structures and processes which ensure that the organization’s IT sustains and extends the organization’s strategy and objectives.”
evaluation is interpreted here as an act of measuring or exploring qualities (such as worth, value, benefits, cost, and impact) of an IT application and the related changes, the result of which informs a decision to be made concerning that application in a specific context. Evaluation studies in health informatics (Kaplan and Shaw 2004) illustrate the potential of evaluation studies to inform the practice. General IS research has also a similar potential, if IT constructs are incorporated more strongly in behavioral science research (BSR) models—as implied by the nomological net of Benbasat and Zmud (2003). In that case much of descriptive BSR would actually evaluate IT applications or types of IT application.

The dotted arrow in Figure 3.1 reminds that depending on the perspective redesign during usage may be viewed as a part of IT application usage and a part of design activity in IT application development. The following section elaborates this by distinguishing a three-tier model of IT application design.

### 3.4 A Three-Tier Model of IT Application Design

Benbasat and Zmud (2003, p. 191) claim that “our focus should be on how to best design IT artifacts and IS systems to increase their compatibility, usefulness, and ease of use or on how to best manage and support IT or IT-enabled business initiatives.” If one accepts this view, design of IT applications is a key...
activity to be supported by IS research. Still, research on systems analysis and design is underrepresented especially in mainstream IS research (Vessey et al. 2002, Bajaj et al. 2005).

Furthermore, IS research has largely focused on the in-house IS development (Light and Sawyer 2007). However, much of IS development has, almost from the beginning of the field, been based on prefabricated application software provided by vendors. This software covers application packages (e.g., Gross and Ginzberg 1984, Iivari 1986/1990, Lucas et al. 1988), ERP software within the previous category (e.g., Davenport 1998, Pereira 1999), business components in the component-based software development (e.g., Fellner and Turowski 2000, Vitharana et al. 2004), and software services at different levels of granularity. The units of software services may be complete IS applications as provided by software service bureaus at first and application service providers later (Tebboune 2003) or more granulated units of service to be assembled into applications (Elfatatry 2007, Papazoglou et al. 2007). In the case of other types of IT application, the vendor role tends to be even more prominent, since they are typically acquired by clients as packaged applications (such as text processing), as parts of software-intensive products (e.g., vehicular control software), or accessed as services (e.g., social networking websites).

As a consequence, the design of IT applications has become more distributed not only geographically but also temporarily in the sense that the initial design of the application is often done by a vendor and the design may be continued by a client collective and by individual users.* The term “collective” is used here to refer to a group, organization, or community, in association with which a user—as a member, as an employee, as a customer, etc.—makes usage of the IT application in question. The software provided by the vendor may be commercial (e.g., MS Office) or free (e.g., Open Office). Let’s explore an example of application packages such as ERP suites. The software provided by an ERP vendor is supposed to be configured by a specific organization. Although not normally recommended, the client may also customize the system. Finally, the resultant ERP-based IS application may be personalized by the user. This leads to the model of Figure 3.2.

* In each case there may be agents involved. The vendor may use contractors, the collective may have outsourced part of its software design, and the user may use technical experts to help. The present article does not delve into this complication.
Design by users in Figure 3.2 corresponds to “secondary design” in Germonprez et al. (2011) who discuss it at the functional layer and at the content layer. Figure 3.2 also includes tailoring by users as a third example, since “secondary design” does not necessarily concern the functionality of the system or its content, but may be limited to the redesign of the user interface, for example.

Figure 3.2 expands the distinction between primary design and secondary design in Germonprez et al. (2011) into a three-tier model. Design between users and design by client collective do not assume a predefined order, but can proceed either way or interactively. For example, a collective such as an organization may design a standard for web pages of its employees and after that employees are assumed to design their web pages so that they comply with the standard. On the other hand, a group or community may design its web pages bottom-up so that members design their web pages first and after that they are linked together to form the group’s website.

From the viewpoint of the practical relevance of IS the three-tier model is significant in two respects. First, if IS research wishes to influence the quality of IT applications in practice, it is pivotal to pay attention to the initial design of those applications by vendors. It is obvious that the quality of this initial design significantly determines the quality of the final IT application, after possible configuration, customization, and user tailoring. Second, all this implies that we should pay much more attention to the vendor side in addition to the client side and user side as is traditionally done, although a better understanding of the latter ones may inform the vendor side, too.

### 3.5 How Can Information Systems Support Practice?

Figure 3.1 as well as Figure 3.2 assume that IS research—whether following the BSR orientation or the design science research (DSR) orientation—may enable, improve, or correct the related practices by providing relevant knowledge (K). This is the case also in the context of DSR, since the essence of IT artifacts as DSR contributions is knowledge (Iivari 2007).

Although DSR in IS has a potential to increase the practical relevance of IS research (Iivari 2007, Sein et al. 2011), the present chapter does not specifically focus on this research orientation, since it is widely discussed in the recent IS literature (e.g., Hevner et al. 2004).

Instead, this section focuses on how to make descriptive nomothetic research more design-oriented. Much of recent nomothetic IS research has been based on the TAM (Davis et al. 1989; see also Chapter 38 by Venkatesh et al. in this volume). Benbasat and Barki (2007) criticize it in that it has been weakly linked with design. Indeed, it has focused on contextual aspects such as computer anxiety, end user support, experience, gender, personal innovativeness, and self-efficacy (Lee et al. 2003) rather than on designable qualities of IT applications. Perceived ease of use, which—when operationalized as in Davis (1989)—closely corresponds to the core usability (Preece et al. 1994), is the only fairly designable quality of IT applications that has been systematically analyzed in TAM research. Other designable qualities such as system quality, information quality (=output quality) are only occasionally included (Venkatesh and Davis 2000, Wixom and Todd 2005).

Benbasat and Zmud (2003) recommend that IS researchers should focus on IS aspects of the phenomena instead of treating IT artifacts as a “black box.” In line with their recommendation, this section suggests a set of designable qualities of IT artifacts to be considered in nomological empirical research into IT applications.

Each IT application type in Table 3.1 has essential designable qualities (see Table 3.2). Since automating applications often aim at high autonomy of the system execution, system quality—such as reliability, security, and safety—is very important in their case (e.g., embedded vehicular software systems). Augmenting IT applications, on the other hand, typically are highly interactive, underlining usability in the sense of ease of use and ease of learning. Mediating IT applications provide a medium for social

---

* One should note that perceived usefulness in TAM is a more dependent variable. In terms of Table 3.2, it can be expected to be dependent on functional quality, information quality, usability (=perceived ease of use), and so on.
interaction between human beings. Therefore, user-to-user interactivity—such as reciprocity, multimodality, and responsiveness—is particularly significant in their case. Because information content is an essential part of IS proper, information quality is an essential characteristic of IS.

It seems obvious that computer games should be funny in some sense (Davis and Carini 2005) so that they are attractive and engaging. Just as in the case of any piece of art, “artistic quality”—that is, the technical skills and originality exhibited by the piece of art in question—is very essential in the context of computer art. In the case of digital pets, users are assumed to build a personal attachment with the pet. The pet also provides companionship (Friedman et al. 2003). Therefore, emotional quality as the quality of emotions built into a digital pet and exhibited by it is very essential in the case of this type of application.*† Although one can claim that all communication includes identity construction

---

* Emotional quality as used in the present article is a topic of affective computing (Picard 1997), although affective computing is more interested in how computers can recognize, interpret, and process users’ emotions than in producing artificial emotions in computers.

† Emotional quality differs from affective quality as defined by Zhang and Li (2005). Affective quality in Zhang and Li (2005) refers to the ability of an IT artifact to cause a change in the (core) affect of the user. Obviously, emotional quality of an IT artifact may influence its affective quality, but to my knowledge this is an unexplored territory. Sun and Zhang (2006) review the role of affect in IS research, but do not specifically address artificial emotions built in the IT artifacts.

---

### TABLE 3.2 Designable Qualities of IT Applications

<table>
<thead>
<tr>
<th>Ideal Type</th>
<th>Designable Quality</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automating</td>
<td>System quality</td>
<td>System quality in the sense of DeLone and McLean (1992), excluding aspects of usability</td>
</tr>
<tr>
<td>Augmenting</td>
<td>Usability</td>
<td>Usability in the sense of ease of use and ease of learning (Preece et al. 1994)</td>
</tr>
<tr>
<td>Mediating</td>
<td>User-to-user interactivity</td>
<td>User-to-user interactivity supported by qualities such as reciprocity, multimodality, and responsiveness (McMillan 2002, Johnson et al. 2006)</td>
</tr>
<tr>
<td>Mediating</td>
<td>User identifiability</td>
<td>User identifiability covering user anonymity as a special case</td>
</tr>
<tr>
<td>Informating</td>
<td>Information quality</td>
<td>Information quality in the sense of DeLone and McLean (1992)</td>
</tr>
<tr>
<td>Entertaining</td>
<td>Funniness</td>
<td>Funniness of the characters, story, sound, graphics, interface, etc. (Davis and Carini 2005)</td>
</tr>
<tr>
<td>Entertaining</td>
<td>User-to-system interactivity</td>
<td>User-to-system interactivity supported by qualities such as user control of the interaction, multimodality, and system responsiveness of the user–system the interaction (McMillan 2002, Johnson et al. 2006).</td>
</tr>
<tr>
<td>Artisticizing</td>
<td>Artistic quality</td>
<td>Artistic quality comprising technical skill and originality (Kozbelt 2004)</td>
</tr>
<tr>
<td>Artisticizing</td>
<td>Aesthetic quality</td>
<td>Aesthetic quality referring to the beauty of IT artifacts, covering classical aesthetics and expressive aesthetics (Lavie and Tactinsky 2004)</td>
</tr>
<tr>
<td>Accompanying</td>
<td>Emotional quality</td>
<td>The quality of emotions built in the IT artifact (Bates 1994, Fujita 2001)</td>
</tr>
<tr>
<td>Fantasizing</td>
<td>Identity constructability</td>
<td>Identity construction by choosing nicknames, wearing different embodiments (e.g., Avatars), and talking, discussing, and negotiating about various identities (Talamo and Ligorio 2001).</td>
</tr>
<tr>
<td><strong>All</strong></td>
<td>Functional quality</td>
<td>Does the system provide the expected or desired functionality to clients and its users or user groups? Does the system exclude undesired or not allowed functionality from clients, users, or user groups?</td>
</tr>
</tbody>
</table>

© 2014 by Taylor & Francis Group, LLC
(Hermans 2004), it is the most conspicuous in the case of virtual fantasy world applications where a user
can typically choose and/or construct his/her identity on will or have a number of identities (Talamo
and Ligorio 2001, Yee and Bailenson 2007). Identity constructability includes self-presentability as a
special case, if the user wishes to present his/her real identity.

Table 3.2 summarizes the qualities presented earlier including a few additional ones. The final row
identifies “functional quality” as a generic quality characteristic that is common to all software-based
systems. ISO 9126 standard for software quality characteristics defines functionality by a question: “Are
the required functions available in the software.” Table 3.2 does not consider required functions or
functional requirements as a reference of functional quality, but expectations and desires of clients,
users, or user groups. Required functions or functional requirements as a baseline artifact in software
development may more or less correspond to these expectations and desires. Functional quality also
covers the question if the system excludes undesired or not allowed functionality from clients, users, or
user groups. This aspect is close to compliance that is one subcharacteristic of functionality in the ISO

Table 3.2 identifies an application type where the relevance of the designable quality in question is
most obvious and justifies its inclusion in that way. It does not, however, claim that the designable
quality is relevant only in the case of that application type. As discussed earlier, IT applications in real life
tend to be hybrids of several ideal types in Table 3.1. Therefore, when considering relevant designable
qualities of a real application, it is justified to consider all the 12 qualities—whether or not they are rel-
evant in the case of the IT application at hand.

Of course, one cannot claim that Table 3.2 is an exhaustive list of designable qualities of IT artifacts.
For example, one could consider complementing Table 3.2 with some innovation characteristics such
as complexity, compatibility, result demonstrability, and trialability (Rogers 1995). Although they can
partly be influenced by design, generally they are less designable than the qualities in Table 3.2, and
above all they are more general without any IT-specific substance.

The AIS website of theories used in IS research (http://istheory.byu.edu/wiki/Main_Page, accessed
March 15, 2012) identifies 14 theories that have originated in IS. Of these, DeLone and McLean (1992,
2003) models and Media Synchronicity Theory (Dennis et al. 2008) are strongest in addressing designable
qualities. Since the latter is very specific focusing CMC applications only, let us compare Table 3.2
with the DeLone and McLean models, which are more general. DeLone and McLean (1992) explicitly
include only system quality and information quality and implicitly usability as an aspect of system qual-
ity as designable qualities. One should note, however, that the core usability differs quite a lot from more
technical aspects of system quality such as reliability of the system, its maintainability (flexibility), and
efficiency (resource utilization). Therefore, it is justifiable to separate it—in particular since users are the
best experts to assess usability, while IT specialists are the best experts to assess the technical aspects
of system quality. DeLone and McLean (2003) additionally include service quality, acknowledging that
the latter may be more relevant in the case of IS functions than in the case of individual applications.
Table 3.2 incorporates functional quality that resembles the “technical quality” aspect of service quality
in Grönroos (1984).*

So, despite the fact that Table 3.2 is tentative, it provides a far more systematic and comprehen-
sive list of designable qualities to be included compared with the DeLone and McLean models. As a
consequence, one can imagine research inspired by Wixom and Todd (2005), for example, in which
one has the qualities of Table 3.2 as independent variables and TAM/UTAUT constructs (Davis et al.
1989, Venkatesh et al. 2003) as intervening and/or dependent variables. Inclusion of all or most of
the qualities of Table 3.2 as independent variables reduces the risk of specification errors. If they are

* Grönroos (1984) distinguishes two aspects in service quality: technical quality, which refers to what the customer is
actually receiving from the service; and functional quality, which involves the manner in which the service is delivered.
As a consequence, “technical quality” in the sense of Grönroos (1984) is close to functional quality in Table 3.2, whereas
systematically included, one can expect a more cumulative research tradition on designable qualities that significantly affect the success of IT applications.

3.6 Discussion and Conclusions

To summarize, the present chapter argues that IS is as a practical discipline of computing that specifically focuses on IT applications, on their development, use, and impact at individual, group, organizational, community, society, and global levels.

The major ideas of the chapter can be summarized in four points. First, the chapter encourages IS researchers to seek technological continuities and invariants in order to foster a cumulative research tradition in this world of technological change. The first question to be asked in the case of each new technology is whether or not it really is so new that it does not have any predecessors and cannot be viewed as an instance of a more general genre of technologies. If not, then one can ask how prior research into its possible predecessors and more general genre of technologies may inform the focal new technology, how the focal technology differs from existing comparable technologies, and what implications these differences may have.

To illustrate, although ERP-based IS are examples of application package-based IS, it is not a norm in the ERP literature to emphasize this continuity and to carefully review how earlier research on the implementation of application package-based IS could inform ERP implementation; “implementation” in this context covering design by the client organization (configuration and possible customization) and the final implementation. This has likely resulted in omission of relevant earlier research.*

Second, the chapter advises IS researchers to identify the extra-scientific practices (Strasser 1985) IS research aims at supporting and the associated actor groups to be informed. When suggesting practical or managerial implications of their research, IS researchers could be much more explicit in this regard. One reason is that what is actionable in practice depends on the actor groups. For example, top management support is actionable to managers but not to individual users, users’ age and gender are not actionable to users and actionable to managers through recruiting at most.

Third, the chapter proposes that design of IT applications should be a central topic of IS research, because it is the key practice to be supported by IS research and because design of IT applications essentially affects the quality of IT applications and their success. Recognizing that software for most IT applications is acquired as packaged software or as prefabricated components and services, we should also pay more attention to the whole chain of design from the initial design by vendors and further design by client collectives and individual users.

Finally, the chapter suggests that IT applications can be conceptualized in terms of designable qualities. Focusing on designable qualities helps to build more IS-specific theories for explaining and predicting (Gregor 2006) than theories adopted from reference disciplines, which tend to be void of any IT-specific substance. Also many theories, which have originated in IS, tend to be fairly weak in addressing IT. For example, it is almost paradoxical that the major theoretical achievement of IS—Technology Acceptance Model (Davis et al. 1989)—is so general that it is totally empty of any IT-specific substance.†

As a consequence, it can be applied in the case of almost any technology (such as farming technologies, Flett et al. 2004). One could expect that more IT-specific theories would be more informative to practitioners.

---

* As an example, Iivari (1986/1990) suggested a theoretical model that explains implementation challenges of application package-based information systems in terms of originality, complexity, radicalness, and divisibility of the systems. It seems obvious that these four dimensions are particularly relevant in the case of ERP-based information systems, which tend to be complex, may imply radical changes in business processes, and prefer borrowing (configuration) rather than adaptation (customization). Overall, all these three factors accentuate implementation challenges, only divisibility (i.e., modularity) of ERP systems mitigating them. Yet, the four factors and their interactions have not been systematically discussed in the ERP literature.

† The IT substance is just a matter operationalization of perceived ease of use and perceived usefulness.
Furthermore, modeling application types and specific applications in terms of variables such as designable qualities in Table 3.2 allows us to conduct more general and time-invariant research, since application types typically include several specific applications that vary in terms of designable qualities just as specific IT applications do during their evolution. For example, we have currently numerous social networking websites. When researching them, it would be advisable to include a number of sites in a study rather than a single one in order to increase “objective” variance of each designable quality. Assuming that there are standard measurement instruments for designable qualities, independent studies investigating different websites or a single website at different times are also more easily comparable.

But above all, designable qualities constitute parameters for design. Therefore, when conducting nomothetic research, this chapter suggests that the incorporation of designable qualities is primary and the inclusion of non-designable and non-actionable contextual factors only secondary. Although the latter may significantly increase the variance explained and therefore may be intellectually interesting to include, interpreting IS as a practical discipline underlines that it is more important to understand how designable qualities explain the success of IT applications and in this way to help practitioners to design better systems.

References


Disciplinary Foundations and Global Impact


© 2014 by Taylor & Francis Group, LLC


