27

Information Requirements Determination

27.1 Introduction

Information requirements determination (IRD) is the process of eliciting, representing, and verifying the functional and non-functional needs for an information system (Browne and Ramesh 2002; Davis 1982; Hickey and Davis 2004; Larsen and Naumann 1992; Vitalari 1992). More broadly, IRD is a form of needs analysis, an activity required in any designed artifact, ranging from consumer products to software to industrial processes (Simon 1996).

The primary IRD process occurs early in the development of an information system, generally following the project planning process. IRD has traditionally been conceptualized within the traditional systems development life cycle (SDLC) (Kendall and Kendall 2010) as an early stage in the process. However, IRD is also utilized in its basic form in new approaches to systems development, including most “agile” development methods (Baskerville and Pries-Heje 2004; Lee and Xia 2010).

The importance of IRD to systems development is difficult to overstate. Because the IRD process occurs early in development and determines the needs for the system, all remaining activities in development, from modeling to design to coding to implementation, depend on specifying requirements that are as accurate and complete as possible. Elegantly designed systems that do not meet user requirements will not be used. Queries to databases that do not contain information users need will not be made. Thus, IRD is arguably the central activity in systems development.

Computer information systems serve a wide range of purposes in society. Regardless of whether the system is fully automated, such as a control system (e.g., a system operating temperature controls for a building) or a system that acts as a replacement for human decision makers (such as a loan application system), or is a system designed for interactive decision support (e.g., web-based shopping assistants), the requirements for the system must be determined by the systems analyst and developer. For this reason, requirements determination is also key to the computing profession generally.
27.2 Underlying Principles

The full information determination requirements process is often conceptualized as the development of a strategy for the elicitation of the requirements, their representation, and their verification (Larsen and Naumann 1992; Vitalari 1992). Development of a strategy was described well by Davis (1982), but little recent research has addressed the subject. Elicitation, or gathering or capturing of requirements, can be accomplished in at least four ways: asking, deriving from an existing system, synthesis from characteristics of the utilizing system, and discovering from experimentation from an evolving system (prototyping) (Davis 1982). Although all these sources are used at various times, this chapter will focus on the asking strategy, as it is the primary strategy utilized.

The asking strategy for elicitation is performed primarily through discussions with people who use or will use the system. Two other methods also important in elicitation include (1) examination of (a) internal documents such as organizational forms, reports, procedures manuals, policy documents, strategic plans, training materials, documents concerning existing systems (whether computerized or manual), and other company records and (b) external documents such as books, scholarly articles, white papers, and web-based materials and (2) observations of people performing tasks relevant to the system being developed. The focus of this chapter will be primarily on elicitation of requirements from users, but examining documents and observation will also be discussed briefly.

Representation of requirements is accomplished by analysts using informal models (such as flow charts, conceptual maps, knowledge maps, and cognitive maps) and semi-formal models (such as entity-relationship diagrams, data-flow diagrams, and Unified Modeling Language diagrams) (Browne and Ramesh 2002; Montazemi and Conrath 1986; Robertson and Robertson 2006). Verification is conducted by discussing the analyst’s understanding of the requirements with the user, utilizing models, diagrams, written notes, and/or any other representational or verbal means available. Although the full process is critical to the systems development process, as noted, this chapter will focus on the elicitation of requirements from users.

Information system development can proceed in several different ways. The traditional SDLC model (usually termed the “waterfall” model or the “planned” model) is still used in a large majority of systems development efforts (Light 2009; Norton 2008) as it brings numerous significant advantages to the development process. However, the lack of quick user feedback on design efforts in the waterfall model has led to the creation of several alternative methods for systems development over the past 15 years or so. These methods are usually grouped together under the term “agile.” Agile methods rely on short cycle times, small work modules, and rapid user feedback to improve system development outcomes (Baskerville and Pries-Heje 2004; Lee and Xia 2010). A set of basic requirements is needed even in agile methodologies, and so the discussion of IRD in this chapter is relevant to all methodologies. Nonetheless, there are differences in IRD between the two general approaches, and these differences are discussed in the Impact on Practice section.

27.2.1 Individual Level Principles

Asking users to specify information requirements relies on several fundamental cognitive and behavioral principles. Knowledge elicitation utilizes an interactive process in which the analyst stimulates the user to describe his needs in terms of inputs, processes, and outputs in the business process. Users are often encouraged not just to state “facts” but also to tell stories that reveal process activities that may otherwise remain unsaid (e.g., Alvarez 2002; Davidson 2002; Urquhart 2001). Stimulating users to evoke their knowledge requires that the analyst ask the right questions (Browne and Rogich 2001; Marakas and Elam 1998; Moody et al. 1998; Wetherbe 1991) to cause the user to access the correct “path” in memory. The more substantively knowledgeable (in the application domain) and procedurally knowledgeable (in systems development) the analyst, the more likely he will be to stimulate the user in appropriate ways. Failure to stimulate the correct path for the user can result in serious
Information Requirements Determination

(although unintentional) “misinformation” or lack of information elicited from the user and inaccurate requirements. Thus, the analyst’s role in eliciting knowledge from users is usually both critical and challenging.

Analysts (and users) are limited in their information processing capabilities (Reisberg 2009; Simon 1996), and this constrains their ability to gather accurate and reasonably complete requirements (note that it can be persuasively argued that requirements are never “complete”). Perception is vastly limited; humans perceive a tiny fraction of information available in the environment at any point in time. Similarly, attention is also significantly limited. Most of what is perceived is not attended to and therefore does not enter working memory. Working memory is the mechanism through which everything we refer to as “thinking” takes place (e.g., reasoning, judgment, decision making, reflection, and introspection). Information in working memory that is elaborated upon and that has sufficient time to be stored (“consolidation”) enters long-term memory. Working memory is severely constrained both by speed of processing (generally speaking, we are serial processors and can think about only one thing at a time) and capacity of processing (we are generally limited to four to nine pieces of information in working memory at one time, although the more complex the piece of information, the lower the number of pieces that can be considered). Long-term memory is also constrained. Because working memory processes so little information so slowly, most information to which we attend either does not enter long-term memory or enters only as fragments. We remember the “gist,” or essence, of events but not the details (Brainerd and Reyna 1990). When asked to recall an event, a person must fill in the details of the event that were not stored. So, for example, when a user is asked by an analyst to recall an event that occurred, he will have to fill in the details (i.e., invent them) because he has no memory of those details. This can, of course, mislead the analyst. The other major problem with long-term memory is that people forget information. Nonetheless, it is generally believed that nothing is ever truly forgotten, and thus the goal for analysts is finding the right stimulus material to help the user locate the correct “path” to the memory (Reisberg 2009). This effort can be more or less successful.

Researchers have investigated numerous issues related to user and analyst cognition in IRD. As one example, recent research in requirements determination has investigated specific issues in long-term memory that affect the accuracy of requirements. Appan and Browne (2010) investigated a phenomenon referred to as “retrieval-induced forgetting.” This psychological phenomenon shows that the act of recalling certain information (e.g., in response to a question from an analyst) causes related but currently unrecalled information to be suppressed in memory and therefore to be unavailable for subsequent recall from the same or similar stimulus. Thus, the first relevant information an analyst receives from a user concerning a particular issue is also likely to be the last; if the analyst returns to the issue in subsequent IRD sessions, he is likely to receive the same information from the user (Appan and Browne 2010; see also MacLeod 2002). This problem reduces the amount of information an analyst will learn about user needs and suggests the need to use multiple users at multiple points in time for IRD.

A second unrelated study concerning long-term memory, also by Appan and Browne (2012), investigated the impact of the misinformation effect on IRD. The “misinformation effect” occurs when false or misleading information is suggested to someone (e.g., a user) following the occurrence of an event and the person recalls the misinformation rather than the information he originally observed during the event. Appan and Browne (2012) demonstrated empirically the existence of the misinformation effect during IRD and extended previous findings to show that the effect exists even with factual information learned over a long period of time. The latter finding is particularly troubling for IRD, since the results show that if an analyst suggests some misinformation contrary to users’ long-held beliefs or factual knowledge that they have developed through their own experiences, those users are likely to report in response to subsequent questions the information provided by the analyst (which may be arbitrary or due to ulterior motives) rather than their own knowledge or beliefs. The potential impact of the misinformation effect on eliciting accurate user requirements and on ultimate system success is therefore of significant concern.
Beyond specific problems with long-term memory, the overall limitations on information processing mean that systems analysts, and humans in general, are not cognitively equipped to gather all possible information in any but the simplest of problems. Requirements determination is not a simple problem. Therefore, analysts employ heuristics to help them narrow the problem space. A heuristic is a short-cut, or rule-of-thumb, for achieving satisfactory (and perhaps better) results in solving a problem (Simon 1996). People evolve heuristics for solving problems based on experience; the more experience a person has with a type of problem, the better his heuristics typically are. Heuristics allow humans to circumvent their cognitive limitations, particularly those associated with working memory, but they come at a cost. Because they are not optimization techniques (people do not gather all available information and do not integrate it optimally), heuristics result in errors (Tversky and Kahneman 1974). Some of the errors created are random and some are systematic. The systematic errors are usually referred to as cognitive biases. A bias is a systematic deviation from an accurate answer; accuracy in this case can mean normatively accurate information or a target standard resulting from consensus.

In the elicitation of requirements from users, numerous biases have been theorized and several have been investigated empirically. Basic issues in cognitive biases in requirements determination have been discussed by Browne and Ramesh (2002), Ralph (2011), and Stacy and MacMillan (1996). Prompting strategies for overcoming cognitive biases in requirements elicitation have been suggested by Browne and Rogich (2001), and a list of cognitive biases and strategies for overcoming them is provided in Pitts and Browne (2007).

The influence of cognitive biases on human behavior, including requirements elicitation, is extensive, and a full discussion is beyond the scope of this chapter. It is known that people are affected by the ease with which information can be recalled, by insensitivity to base rates (i.e., the actual rate of occurrence of events in populations), by vividness of stimuli, by giving inappropriate weight to small samples, by misconceptions of chance events (e.g., believing that certain patterns in data cannot occur without a causal explanation), by seeking primarily confirmatory evidence for conclusions, by anchoring on initial values (or existing procedures), and by overwhelming overconfidence, among many other biases (see, e.g., Bazerman and Moore 2013; Kahneman 2011). These biases all impact the information elicited in requirements determination as they affect both the users and the analysts. As one example, important research concerning the anchoring bias in requirements modeling has been performed by Parsons and Saunders (2004) and Allen and Parsons (2010) and in requirements elicitation by Jayanth et al. (2011). The interested reader is referred generally to Bazerman and Moore (2013) and Kahneman (2011) and, for examples specific to IRD and systems development, to Browne and Ramesh (2002) and Stacy and MacMillan (1996). In general, the study of cognitive biases is an under-researched area in the information systems field.

A second type of bias that affects IRD is motivational biases. Motivational biases result from perceived incentives in the decision-making environment or from internal desires or preferences (see Bazerman and Moore 2013 for a general discussion). Examples of motivational biases include self-serving attributions (i.e., taking credit for successes and blaming failures on external sources beyond the person’s control), reporting false facts or beliefs to curry the favor of influential others (e.g., bosses), the illusion of control (i.e., the belief that one can control random events), real and anticipated regret, and deliberately escalating commitment to a failing project. A large body of literature in the information systems discipline concerning escalation of commitment has been developed by Keil (1995) and Mähring and Keil (2008). Most other motivational biases have not received attention, however. Motivational biases generally (although not universally) arise in IRD because requirements determination involves numerous parties in a social process, and both analysts and users are motivated to portray themselves in a positive light, to hide errors, and to avoid blame.

### 27.2.2 Group Level Principles

When analysts elicit requirements from users, the parties are engaged in a social process. In addition to the motivational biases just discussed, social influence concerns such as conformity, persuasion, and
Information Requirements Determination

27.2.3 Examination of Internal and External Documentation

As noted, a second method usually used to accompany the asking strategy is for the analyst to examine internal and external documents (e.g., Davis 1982; Kotonya and Summerville 1998; Maiden and Rugg 1996). Analysts must decide which documents contain valuable and relevant information for the firm and for the users and should be incorporated into the proposed information system and which documents are redundant, not used, or not valuable and should be omitted (Robertson and Robertson 2006; Watson and Frolick 1993). For example, Watson and Frolick (1993) discussed requirements determination for executive information systems and noted the importance of examining memos, e-mails, and various internal company documents as well as external documents such as articles, books, marketing research information, and government and industry publications. Company forms and reports often determine the desired inputs and outputs, respectively, of systems and are therefore a natural starting point for analysts. However, with all existing documents, analysts must be concerned about improper anchoring on current data and processes (Davis 1982). Simply because internal documentation exists or external data seem to support requirements does not mean they should be included in the system (Robertson and Robertson 2006). Because people prefer concrete examples to abstract concepts, there is a strong temptation to utilize an existing document, design, interface, etc. as a starting point. This initial anchoring point, however, typically precludes other classes of potential alternatives, including more innovative ideas. Oftentimes, business process reengineering using outside experts (with no...
pre-conceived notions or company “baggage”) may be required to be certain that existing documentation reflects requirements that are useful and appropriate for the system currently under design.

### 27.2.4 Observation of Users and Organizational Task Performance

Observation of users as they perform organizational tasks is also an important method for eliciting requirements (Alvarez 2002; Byrd et al. 1992; Kendall and Kendall 2010; Leifer et al. 1994; Robertson and Robertson 2006). Users often have difficulty verbalizing how they perform their everyday tasks. As experience increases, people typically lose conscious access to the steps they take in performing a task (Leifer et al. 1994; Simon 1979). Observation can help analysts either as a first step in understanding work processes or in filling in gaps in user descriptions of how they perform their jobs (Kendall and Kendall 2010). Observation can be performed either silently or interactively. Silent observation requires the analyst to observe the user performing his task without interruption. The advantage of this method is that the analyst can observe only behavior, not cognition, and therefore may not understand how or why an action is being performed. Interactive observation permits the analyst to interrupt the user’s task performance when an action or the reasons for it are not clear (Robertson and Robertson 2006). This allows the analyst to understand how and why the action is being performed, assuming the user can articulate the reasons. The main disadvantage is that the user’s behavior and thought processes are interrupted, which may lead to task performance that is not typical. This could cause the analyst to model the requirements inappropriately. Based on the strengths and weaknesses of the two techniques, using silent observation first and following that technique with interactive observation may be the best approach.

A disadvantage of observation generally is the Hawthorne effect, in which users act differently because they know they are being observed (Parsons 1992). There is no perfect cure for the Hawthorne effect, but analysts should assure users that they are not being judged and that the results of their individual task performance will not be reported (only aggregated results), if true. If the organization is unwilling to promise anonymity for individual performance, it is recommended that the analyst utilize a different technique.

Another method for observing task performance is through protocol analysis (Byrd et al. 1992; Ericsson and Simon 1993). Protocol analysis requires users to think aloud while they perform a task, and their thoughts (sometimes accompanied by movements) are recorded on tape, video, or digital media. The resulting vocalized thoughts can be analyzed to trace task performance and to capture certain aspects of behavior and cognition that both silent and interactive observation may miss. When the correct procedures are utilized, results from protocol analysis can be valid and highly valuable sources of information (see Ericsson and Simon 1993).

### 27.3 Impact on Practice

Requirements determination is arguably the principal job of systems analysts. Regardless of whether the proposed system is intended to automate or provide decision support for an existing manual system, upgrade an existing computerized information system, or tailor a purchased (“off-the-shelf”) software program for a company’s needs, user and organizational requirements must be determined. Because information systems are now the lifeblood of organizations (and, increasingly, individuals), requirements determination is a critical and ubiquitous activity in companies today. Thus, all of the issues discussed in the previous section are important and relevant for IRD practice.

Faulty requirements are generally thought to be the most common cause of information system failure in organizations (Bostrom 1989; Byrd et al. 1992; Hofmann and Lehner 2001; Wetherbe 1991). Underutilization, in which systems are used but not to their full functionality, is also a problem that results from inaccurate or incomplete requirements. The cost of information systems failures to businesses is...
estimated in the hundreds of billions or even trillions of dollars worldwide each year.* Thus, the impact of improving information systems on organizational performance is difficult to overestimate.

Requirements determination research over the years has directly influenced systems analysis practice. For instance, analysts are now trained in multiple elicitation techniques and are often warned about the influence of various problems in elicitation (Kendall and Kendall 2010). For example, analysts ideally should provide little input during early stages of the interview process, stimulating the user to talk and probing using only neutral questions. This allows the user to express his views without any priming or undue influence from the analyst. At later stages of the process, the analyst can probe to elicit specific knowledge not yet revealed (Appan and Browne 2012).

Research demonstrating the advantages of involving many stakeholders and multiple points of view has led to the use of JAD sessions, noted earlier. JAD sessions allow for informed discussion of various potential needs both between users and analysts and between users themselves. The assignment of devil’s advocates in JAD sessions to improve and polish ideas is also a result of research on group decision making (see Browne and Ramesh 2002). Although JAD sessions have weaknesses, such as wasted time during group brainstorming (Brown and Paulus 2002; see also Sandberg 2006), they bring significant strengths to the IRD process, including user involvement and increased “buy-in” by stakeholders (Davidson 1999).

One practical implication of research in IRD that has yet to appear on a large-scale basis is training analysts to recognize the heuristics they use and the biases and other cognitive challenges to which they are subject. Articles in the practitioner literature have appeared (e.g., Stacy and MacMillan 1996; West 2008), but they have not been widespread and there is no evidence of systematic training. Such training has the potential to make a profound contribution to the effectiveness and efficiency of requirements elicitation, perhaps more than any other set of findings from the research literature. Motivational issues are also critical, but they are generally more easily recognized and remedied. Cognitive biases, recall issues, attentional deficits, and other cognitive problems operate primarily subconsciously and are largely hidden from view. This makes their effects particularly pernicious and difficult to discern.

As noted in the Underlying Principles section mentioned earlier, there are differences in requirements determination in the SDLC approach and agile approaches to systems development. In the SDLC, requirements are gathered primarily in one of the earliest stages of the process. After they are modeled and verified, users often “sign off” on the requirements, and system designers then proceed with development of the system. There are feedback loops in the SDLC, and analysts and designers often must return to users for additional requirements and/or clarifications. However, the initial stage subsumes the majority of the requirements determination effort. Research has demonstrated that one of the weaknesses of this approach is that users often do not know what they need during the early stages of the development of a system (especially a new system) and that their “sign off” is of dubious validity (Wetherbe 1991). Additionally, analysts may not understand the proposed system well enough to ask appropriate questions at this point, resulting in an inadequate determination of requirements (Davis 1982). A second weakness is that requirements often change during development (e.g., Maruping et al. 2009), and the “frozen” requirements document with which designers work when using the SDLC can therefore make the resulting system inappropriate or out of date by the time it is completed. These findings led to the agile methods discussed earlier, including such methods as scrum and extreme programming, which rely on short bursts of IRD followed by mock-ups or prototypes to which users can quickly react. It is likely that such agile methods will continue to be used and perhaps increase in popularity as the techniques are improved and the advantages, particularly quick user feedback, are fully utilized. It is unlikely, however, that an initial requirements gathering will disappear from agile methods that survive. Absent an initial substantial investment in requirements elicitation, the first prototypes will reflect what

the analyst thinks he knows about the system or what he thinks should be included. There is no more fertile ground for the types of cognitive and motivational biases and other problems discussed in this chapter than this type of approach to systems development.

27.4 Research Issues

Despite the large body of research that has developed concerning requirements determination over the past 30 years, there is no shortage of work remaining. Because the activity involves individual and group behavior, and all the cognitive and motivational issues associated with such behavior, IRD progresses and improves largely as our understanding of human behavior progresses in the social sciences. This section describes a limited subset of research issues that are currently of interest in the field or will be important in the near future.

One important set of issues is the enduring topic of cognition in IRD. As we gain a better understanding of judgment and reasoning, and how decisions are made, we can improve the efficiency of IRD and elicit and model requirements more accurately. This process includes a better understanding of how analysts allocate their attention, how they reason and make judgments and choices, and what information they commit to long-term memory. Additionally, the recall of information from long-term memory, and the problems that have been shown to occur, remain issues that deserve further research. As noted in the Underlying Principles section in the preceding text, recent research in the information systems field has addressed some of these issues. In addition, neuroscience techniques such as functional magnetic resonance imaging and electroencephalography are allowing information systems researchers to develop an understanding of the brain at the neural level (e.g., Dimoka et al. 2011) that will aid immeasurably in our understanding of cognition and behavior in IS development.

A related concern is the issue of eliciting requirements that help analysts directly with decision making. Decisions about requirements require reasoning, judgment, and choices by systems analysts, and the direct elicitation of user preferences about requirements can aid immeasurably in these decisions. Some research in IRD has addressed direct elicitation of user preferences (e.g., Liaskos et al. 2011) and decision making more generally (e.g., Alenljung and Persson 2008). However, much more research from decision theory could be incorporated to improve these outcomes. Preference elicitation, probability elicitation, and many other decision analytic techniques can be incorporated in IRD to provide a stronger foundation for the choice and prioritization of requirements by analysts (see, e.g., Goodwin and Wright 2004; Jain et al. 1991; Keeney and Raiffa 1993; von Winterfeldt and Edwards 1986).

Another important issue for systems analysts in IRD is when to stop eliciting and modeling requirements. In unstructured problems with nearly limitless information, such as IRD, analysts can conceivably elicit information and model requirements indefinitely. However, time, cost, cognitive, and motivational considerations prevent them from doing so. Research has revealed that analysts (unsurprisingly) do not gather all available information in requirements elicitation settings and has demonstrated cognitive stopping rules that analysts employ to cease requirements determination efforts (Browne and Pitts 2004; Pitts and Browne 2004). Additionally, research has investigated stopping rules in information search behavior more generally (Browne et al. 2007). Stopping rules are particularly important in IRD because of the costs of underacquisition and overacquisition of information. Overacquisition results in wasted time and money and may lead to unnecessary features in software. Underacquisition results in underspecified requirements, leading to systems that likely do not meet user needs. Much additional work is necessary to understand both cognitive and motivational reasons that analysts stop during all activities in IRD, including elicitation, representation, and verification of requirements.

In addition to basic cognitive issues, an improved understanding of analysts’ use of heuristics and the biases that result can have a profound impact on IRD. As noted earlier, human decision making is often biased in predictable ways. Both analysts and users are biased in their provision and interpretation of information and in the decisions they make based on that information. Therefore, an improved understanding of the conscious and subconscious heuristics that analysts use to elicit, model, and verify
information, and of the biases that are present in both analysts and users, has the potential to improve IRD and the remainder of the systems development process significantly.

Another important issue is that software development is increasingly performed by people who are not colocated geographically. This means that requirements determination must be performed by people in different places, perhaps of different cultures, who likely have never met. The telecommunications challenges of such interactions have largely been solved, but the lack of face-to-face communication may have significant impacts on the process and outcomes of IRD. Research in communication theory and organizational behavior has demonstrated the differences between face-to-face communication, computer-mediated communication, and other forms of communication. Although some excellent research concerning IRD has been performed both in this area and the related area of off-shoring (e.g., Aranda et al. 2010; Dibbern et al. 2008; Evaristo et al. 2005), much more work is necessary to understand the full implications of distributed requirements determination.

Another important future research issue concerns requirements determination for purchased software. Much software is now purchased rather than produced “in-house,” and requirements determination can be a complicated process with such software. Ideally, requirements are determined before software is purchased to ensure that the software meets user needs. However, software can be purchased for a variety of reasons that are unrelated to requirements, including political reasons, purchasing from a vendor with which the company already has a relationship, better pricing, etc. If requirements are determined beforehand, systems analysts and designers will still need to integrate the system into the company’s environment. If requirements are not determined a priori, or if only a cursory attempt is made, the challenges upon the arrival of the software can be daunting. Analysts will need to determine user requirements and then assess how (and if) the software can be tailored to those requirements. Some research has addressed these issues (e.g., Parthasarathy and Ramachandran 2008). However, as software is increasingly purchased off-the-shelf or in large-scale company-wide packages (e.g., enterprise resource planning systems), analysts’ challenges in determining requirements escalate dramatically. In such cases, arguments have been made that it is easier to change user processes rather than the software (e.g., Davenport 1998; but for a contrary view, see Hong and Kim 2002). In many cases, dramatic IT implementation failures have resulted from systems that could not be properly adapted to company requirements after purchase (see, e.g., Chen et al. 2009). More research is needed to help analysts manage the challenges associated with IRD in these contexts.

Another important research question concerns how users can be trained and motivated to provide better and more complete requirements. Users are the domain content experts in most systems development efforts, and systems analysts (and the organization) rely on them to provide requirements that will lead to successful systems. However, users suffer from the same biases as analysts. There is also often inertia among users who do not want to change how they perform work, which is nearly inevitable with the introduction of a new information system (Gallivan and Keil 2003; Markus and Benjamin 1997). Thus, users may not provide necessary information or may deliberately introduce misinformation (Appan and Browne 2012). Additional research investigating requirements elicitation issues from the user side would be highly valuable.

Specific content areas are also important. Although most research has focused on methods and tools for IRD that can be generalized across content domains, some tailoring is often necessary depending on the domain. For example, Schneider et al. (2012) note that many systems are now security-related, and requirements determination expertise for such systems is often lacking. They state that “Identifying security-relevant requirements is labor-intensive and error-prone” (p. 35) and suggest organizational learning tools to aid analysts in determining requirements for security-related software. Similarly, Vivas et al. (2011) point out the difficulties in designing security-related systems that ensure privacy, trust, and identity management. Fabian et al. (2010) provide a conceptual framework for understanding requirements elicitation for security-related applications. Considering the continuing threat to system security, further research into this content area, with methods and tools for appropriately determining system requirements, is crucial. Other content areas may similarly benefit from IRD research directed specifically toward them.
Convincing practitioners to use the methods and tools suggested by research is another topic worthy of further investigation. Sikora et al. (2012) point out that many methods generated by academics and supported by empirical research do not make their way into common usage among systems analysts. The authors attribute this adoption problem in part to a lack of understanding among academics concerning the tools practitioners need (ironically, perhaps, a requirements determination problem), but a lack of reading of the research literature and general suspicion of academics by practitioners also likely play a role. Better dissemination of research results and better connections with practicing systems analysts by academics have the potential to improve this situation, but more research into the full nature and causes of the problem would be useful.

Requirements determination is a critical and enduring topic in systems development. Many other issues are important for research, and the discussion in this section should be considered a starting point for topics worthy of investigation.

27.5 Summary

IRD is central to systems development because it occurs early in the process and directly influences all activities that follow it. Much research has been performed in the area, but because it is critical to all systems development it is an enduring research topic in computer information systems. Both successes and difficulties in eliciting requirements have been discussed in this chapter, as well as methods for mitigating the problems that occur. IS practice is directly impacted by research in the area because systems analysts are always gathering requirements and are under increasing competitive pressure to produce systems quickly with appropriate, and often innovative, features. Excellent requirements determination is like creative problem solving, with innovations flowing directly from requirements users may not have even imagined before talking with a systems analyst. Needless to say, technology is ubiquitous throughout world societies today. Technology directly improves productivity and standards of living. Advancements and innovations in technology can only be achieved through an improved understanding of users’ needs and desires and through the proper management of these requirements during the entire systems development process.

Further Informations


References


Information Requirements Determination


MacLeod, M.D. Retrieval-induced forgetting in eyewitness memory: Forgetting as a consequence of remembering. *Applied Cognitive Psychology*, 16, 2002, 135–149.


