Preface

The Concept of Sociotechnical Systems: History and Definition

The term "Sociotechnical System" comes from the field of organizational development. The goal of this field is to improve the performance and effectiveness of human organizations. The term was introduced by Emery and Trist (1960), organizational development researchers at the Tavistock Institute of Human Relations in London (*www.tavinstitute.org*). By coining the term "sociotechnical system" they were challenging the conventional position at the time, which was based on technological determinism. Technology determinism postulated that:

- Technology is autonomous.
- The individual and social conditions of human work must follow the technical structures (Ropohl, 1999).
- Training is a process that adapts humans to machines, so humans can be quickly replaced (if need be).

The organization of labor known as Taylorism can be seen as a natural consequence of technological determinism, and Henry Ford's synchronized mass production methods are its most prominent example. This is the way of thinking that Charlie Chaplin's film *Modern Times* and many similar dystopias narrated in 20th century literature criticize. By contrast, Emery and Trist thought that there is, or there should be, an interdependent and reciprocal relationship between humans and technology. Hence, from the point of view of work design, both the social and the technological aspects of work need to be in harmony to increase effectiveness and "humanize" the workplace. This would be achieved mainly by user participation in the design of the systems and devices that users are to operate at the workplace.

From this discussion, it can be seen that the term "sociotechnical" comes from the analysis of labor relations in the industrial era. This new view of the interdependency of the technical and the social also emerged in high-tech industries. For instance, after an in-depth analysis of the development process of a defense-related aircraft, Law and

Callon (1988) found that engineers “are not just people who sit in drawing offices and design machines;” they are also “social activists who design societies or social institutions to fit those machines.”

The introduction of computers at the workplace soon led to new views and extensions of this work. Research into labor relations and work design became more and more concerned with the use of computing systems (Scacchi, 2004). An outstanding contribution came from the so-called “Scandinavian School.” This school advocates that, at design time, apart from user participation, there is also a need to address the politics of labor and democratize the workplace (Scacchi, 2004). This position had a heavy bearing on software and systems engineering, so much so that Friedman and Kahn (1994) later affirmed, in a purely computing context such as the “Communications of the ACM”, that “computer technologies are a medium for intensely social activity” and “system design –though technical as an enterprise — involves social activity and shapes the social infrastructure of the larger society.”

It is also important to note that, at the same time, the field of computer ethics was developing in response to the risks inherent to computing systems, and the ACM “Code of Ethics” was published in 1992. The term “sociotechnical” is widely embraced by people interested in computer ethics, and it is from this field that we have borrowed, slightly modified, what we believe to be the most complete definition: *A sociotechnical system is a complex inter-relationship of people and technology, including hardware, software, data, physical surroundings, people, procedures, laws and regulations (*www.computingcases.com*, 2004).*

Soon the software engineering community realized that systems are dynamic, as the organizational and social context in which they are embedded is also dynamic (Truex, 1999). Projects became more and more *socially self-conscious,* or, in other words, more aware that their objectives are to alter *both the technical and the social* (Blum, 1996). Accordingly the term “sociotechnical” started to be adopted in software engineering and systems engineering. Two main points can be made as to the use of the term: (i) the term normally applies to the product, not the process, because the process is tacitly recognized as sociotechnical by the software engineering community; (ii) the term is normally used in an attempt to emphasize the *socially self-conscious* feature, as defined above, and underline opposition to technological determinism.

Sociotechnical Systems and Requirements Engineering

In no other field is the need to attach just as much importance to the system as to the people so clear as in software and systems engineering. This is because, due to its inherent flexibility (at least theoretically), software can be configured by the designer/developers to fit any particular situation and to achieve almost any purpose. In practice, however, this flexibility comes at a price, because the number of different software + hardware + people configurations is so high that it is extremely difficult to find out exactly which is the best one at a particular point in time to satisfy the stakeholders’ goals. This has been less of a problem in "traditional" engineering, like mechanical or civil engineering, at least until now. But nowadays, in the so-called Information Society, traditional engineering is not free from this problem. Software is now an essential part of products and services offered by industries traditionally unrelated to software, like the automotive industry, photography, telephony, medicine, and so forth (for instance, as Paech, Denger, Kerkow, and von Knethen say in this book, a modern car contains more executable code than the first Apollo that flew to the moon). At the same time, software is an essential instrument for the designers of these new products and services.

But a software system is of no help unless it is built according to its requirements. Requirements engineering (RE) provides the methods, tools, and techniques to build the roadmaps that designers and developers of complex software/people systems should follow, as it is the discipline concerned with the real-world goals for, functions of, and constraints on those systems (Zave, 1997). It is the discipline that most helps to achieve success in system development and, in particular, in sociotechnical system development.

The RE discipline plays an important role in raising the socially self-conscious factors related to complex system development. Additionally, success in RE essentially depends on it being founded on a sociotechnical position. The goal of this book, written by practitioners and researchers, is to promote the sociotechnical aspects that permeate RE. The book adds to existing literature revealing that the RE field (both in research and in practice) is immensely mature as regards accepting and dealing with the multidisciplinary issues required to properly build sociotechnical systems, even though there is still a lot of ground to be covered.

In this book, we present 20 contributions from different authors, divided into three sections: (I) Basics, (II) Challenges, and (III) Approaches.

Section I: Basics

Section I presents eight chapters that introduce important topics in the RE area, always from a sociotechnical perspective. These chapters are, however, not confined to a mere description of the topics. Instead the authors criticize some of the existing approaches and move into new territory.

In Chapter I Parviainen, Tihinen, Lormans, and van Solingen introduce RE for sociotechnical systems. The authors describe the terminology and the process in detail. Nasr, in Chapter II, introduces the topic of RE for embedded systems, in which software is just a part of a complex system. An important topic closely related to the sociotechnical side of RE is that of elicitation. In Chapter III Coulin and Zowghi review the topic and propose some future directions. The problems related to, and the alternatives to, conceptual modelling in RE are the topic of Chapter IV by Andrade Garda, Ares Casal, García Vázquez, and Rodríguez Yáñez. In Chapter V de Antonio and Imbert clarify the use of the term "agent" in RE and in agent-oriented software, and conclude that the different uses of "agent" are not unrelated as they may appear. Sawyer, in Chapter VI, reviews the topic of software process improvement from a sociotechnical perspective and considers lessons learned from industrial pilot studies. Chapter VII, by Greer, discusses the topic of requirements prioritization for incremental and iterative projects and proposes a method for optimally assigning requirements to product releases. The topic of requirements management tools is considered by Carvallo, Franch, and Quer in Chapter VIII, in which a method is presented for building quality models for requirements management tools.

Section II: Challenges

The six chapters included in Section II introduce some important and difficult topics that requirements engineers and system developers have to deal with to build genuine sociotechnical systems.

In Chapter IX Periorellis explains the problems and opportunities related to the composition of existing systems in order to build new systems, even transcending organizational boundaries. The complexity of modern technical products that incorporate software is the subject of Chapter X, with a focus on the automotive industry. In this chapter Paech, Denger, Kerkow, and von Knethen present the QUASAR process that faces some of the challenges posed by those systems. Grützner and Paech, in Chapter XI, introduce the challenges and possible solutions for courseware development, clearly a kind of system with a strong sociotechnical component. The Open Source software development offers a new playground for RE, based on a collaborative, feedback-based process. Chapter XII, by Dietze, presents some insight into this process. The multidisciplinary task of creating value webs, and a methodology for their development, is the topic of Chapter XIII by Gordijn. Technology is opening many possibilities for workgroups. In Chapter XIV Garrido, Gea, and Rodríguez review the topic of RE for cooperative systems and propose a methodology based on behavior and task models.

Section III: Approaches

Finally, Section III proposes some methods and techniques that can help practitioners to solve the complex problems involved in sociotechnical system development.

In Chapter XV Jones and Maiden present RESCUE, a method for requirements specification that has been applied to complex Sociotechnical Systems like air traffic control. Hospital information systems have a clear sociotechnical nature. Chapter XVI, by Sørby, Melby, and Seland, proposes observational studies and drama improvisation as a means to identify and analyze requirements for those systems. An approach to elicit the sometimes subjective and elusive non-functional requirements is described in Chapter XVII, by Kerkow, Dörr, Paech, Olsson, and Koenig. McCall and Mistrik, in Chapter XVIII, propose to use a lightweight natural language processing approach for discovering requirements from transcripts of participatory design sessions. In Chapter XIX Hall and Rapanotti bring one of the most innovative topics in RE, namely, problem frames, closer to the topic of sociotechnical systems. Finally, in Chapter XX, Cronholm and

Goldkuhl present a method based on the perception that the main purpose of information systems is to support communications between different actors.

References

Blum, B. (1996). *Beyond programming: To a new era of design.* Oxford University Press.

Computing Cases (2004). Site devoted to computer ethics. Retrieved from *www.computingcases.org.*

Emery, F. E., & Trist, E. L. (1960). Sociotechnical systems. In C. W. Churchman & M. Verhulst (Eds.), *Management sciences: Models and techniques* (Vol. 2, pp. 83-97). Pergamon Press.

Friedman, B., & Kahn, P. H. (1994). Educating computer scientists: linking the social and the technical. *Communications of the ACM*, *37*(1), 64-70.

Law, J., & Callon, M. (1988). Engineering and sociology in a military aircraft project: A network of analysis of technological change. *Social Problems, 35*, 284-297.

Ropohl, G. (1999). Philosophy of sociotechnical systems. *Society for Philosophy and Technology* Vol. 4, No 3. Retrieved *http://scholar.lib.vt.edu/ejournals/SPT/v4n3/.*

Scacchi, W. (2004). Sociotechnical design. In W. S. Bainbridge (Ed.), *The Encyclopedia of Human-Computer Interaction.* Berkshire Publishing Group.

Truex, D. P., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organizations. *Communications of the ACM*, *42*(8), 117-123.

Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Computing Surveys, XXIX*(4), 315-321.