Knowledge Management in the Design of Safety-Critical Systems

Guy Boy (boy@onecert.fr) and Yvonne Barnard (barnard@onecert.fr)
European Institute of Cognitive Sciences and Engineering
EURISCO International, 4 Avenue Edouard Belin, 31400 Toulouse, France

Abstract
Knowledge management (KM) in the design of safety-critical systems addresses the question of how designers can share, capitalize and re-use knowledge in an effective and reliable way. KM is situated in groups, organizations and communities, playing different roles in the design process. Design of safety-critical systems has specific properties, such as dealing with complexity, traceability, maturity of knowledge, interaction between experts, awareness of the status of information, and trust in knowledge. Documentation of such properties is crucial in design processes. However, writing is not an easy task for engineers and support is needed. Several KM solutions are available to support design work, such as active notification of changes, personal and team workspaces, active design documents and knowledge portal solutions.

Introduction
A system qualifies as safety-critical when it involves risks for human beings in its environment. These people may interact with the system or not. In safety-critical environments, people need to be aware of these risks and know what to do in the case of problems. For this reason, knowledge management (KM) is a crucial issue. It takes the form of experience feedback, traceability and various types of communication channels and processes. People need to have the right information at the right time and under the right format. But situation awareness is not only a matter of availability of information or data; it is also a matter of knowledge and know-how to process information. Today, safety-critical systems are tremendously reliable from a technological viewpoint. However, system reliability and safety remains very local; the overall reliability falls when people interact with these systems. 80% of aviation incidents are caused by human errors (Johnson, 2003). Human reliability is then a very important issue. Human errors have been extensively investigated during the last fifteen years, and the distinction between patent and latent errors has emerged. The former type deals with end user errors, i.e., the errors that pilots commit. The latter type is committed during the design, development and repair of the system. Patent errors are deeply rooted in organizations, communities and teams. The development of a system is the fruit of many types of interactions among people of many kinds. Making an aircraft for example is the fruit of the collaboration of several types of actors, e.g., professionals, industries, services and regulatory authorities. These actors result in an organization that is safety-critical by itself. Therefore it involves careful knowledge management in order to produce safe products. This article presents a human factors view of knowledge management in the design of safety-critical systems. It is based on several knowledge management projects performed in cooperation with groups of engineers in large aerospace and telecommunication companies. In particular, most recent findings come from the European Research and Development project WISE (IST-2000-29280). In WISE (Web-Enabled Information Services for Engineering) we study work-practices of engineers in large manufacturing companies and we design practical methods to easily share and access essential knowledge and information for their tasks (Barnard & Rothe, 2003, Barnard & Pöyry, 2004). These methods are supported by the development of an engineering knowledge portal application. The industrial partners involved in this project are Nokia and Airbus. Other partners are Cyberstream Interface SI, PACE, EURISCO International, Norwegian Computing Centre Helsinki University of Technology, and Technical University of Berlin.

Background
Knowledge management has become an important research topic, as well as a crucial issue in industry today (e.g. Davenport & Prusak, 1998; Barnes, 2002; Holsapple, 2003). People have always tried to organize themselves in order to capitalize, reuse and transfer knowledge and skills among each other within groups. Poltrock and Grudin (2001) propose the triple distinction team-organization-community for groups. KM tools and organizational setups usually emerge from the requirements of one of these kinds of groups. Note that we don’t dissociate a KM tool from the group that is likely to use it.
A team is a small group of persons that work close with each other, but not necessarily in the same location. A leader often coordinates its work. Team participants typically fulfill different roles. They strongly need to communicate. The following groups are examples of teams: software development teams; proposal writing teams; conference program committees; small operational groups such as customer support or research project teams. Support technologies include: buddy lists; instant messaging; chat; Groove (a peer-to-peer team collaboration environment); Quickplace (provides an instant virtual team room where information is managed); BSCW (both a product and a free service for managing information for self-organizing groups; Bentley et al, 1997); video conferencing; data conferencing; and eRoom (team workspaces with shared workspaces, calendars, and discussions through a web browser).

The structure of an organization is typically hierarchical. Modern organizations are usually geographically distributed. They strongly need to be coordinated. The following groups are examples of organizations: companies; governments or government agencies; and non-profit organizations. Support technologies include: E-mail; calendars; workflow; Lotus Notes (an integrated collaboration environment); intranet applications and web; document management systems; and broadcast video.

Communities share a common interest but no structure. They are usually geographically distributed and provide services to people (e.g., the European KM Forum, Amazon.com). The following groups are examples of communities: citizens of a city or neighborhood; special-purpose chat groups; virtual world citizens; auction participants; stamp collectors; and retired people. Support technologies include: web sites; chat rooms; and virtual worlds.

In the field of safety-critical systems, teams, organizations and communities inter-relate in order to insure quality on both products and processes. They are highly constrained. Usually teams are made to carry out projects and programs; they may be multi-national for example. Organizations are made to manage people within a consistent space, e.g., a national company that is more appropriate to handle social laws and customs of the country where it is chartered. Communities are made to help people who share the same kind of work practice to refer among each other, e.g., a community of electrical engineers. We summarize these distinctions in Figure 1.

A project team exists only during the time of the related project. A company may have several projects or programs that themselves may involve people from others companies. A company may become obsolete when the type of products it produces is no longer appropriate with the current market. Professional communities survive the obsolescence of both projects and companies. They actually may also become obsolete when either technology and/or the social world change.

This article presents specific issues brought by the design of safety-critical systems and human factors related to documentation generated and used in design processes. It also focuses on related current design issues. The specificity of safety-critical design-knowledge will be briefly developed. Several KM management solutions will be discussed. The article concludes with a discussion on the difficulties and challenges of KM in engineering.
Designing safety-critical systems

Safety-critical systems have specific properties that directly affect the way knowledge management is carried out. Examples of safety-critical systems are aircraft, power plants, medical equipment, and telecommunication systems. They are basically complex, as complete as possible and described by mature knowledge. Safety is not only a matter of end-user emotion, attention and cognition; it is also a matter of organization and people involved in the whole life cycle of related products. They involve experts that need to cooperate. For that matter, traceability of decisions is crucial. Nonaka, Toyama & Byosiere (2001) describe how knowledge is created in organizations.

Safety-critical systems communities as families

People working on safety-critical systems, in domains such as aerospace, nuclear power plants and medicine, form strongly connected communities of practice that could be seen as families. They have their own meetings, workshops and conferences, even journals, where they can exchange experience, foster research and improve knowledge on safety-critical systems in general. These communities work across organizations and teams. They tend to become references and initiate standards in the related field. They are recognized bodies for knowledge validation, providing principles for assessing knowledge maturity. How people share knowledge in engineering organizations was investigated in the WISE project, see Pöyry, Meriluoto, & Luoma (in press).

Dealing with complexity

Even if the designers of safety-critical systems should always have in mind to design for simplicity, what they have to do is inherently complex. Such systems are usually complex. Their design and development processes are also complex. In the design process, designers rely on knowledge that is available in the form of handbooks, lessons learned, and best practices. Designers have to take into account the experiences with older systems, on which the new system is usually building, making sure that incidents and accidents that have happened are no longer possible in the new design. Designs are verified and validated in extensive, well-defined processes. In the end of the design process, certification by different authorities and certification bodies can also play a large role. In order to get a system certified, one has to be able to justify the choices that were made, to prove, as far as possible, that all knowledge about problems with similar systems has been taken care of, and that the system will function safely in all kind of difficult and even disastrous scenarios.

Targeting completeness in an open world

Safety-critical systems require complete definition of their (cognitive) functions that they involve in terms of role, context of validity and use, and appropriate resources that they need to accomplish their functions (Boy, 1998). A cognitive function analysis is usually mandatory when we need to demonstrate that the system being designed satisfies a set of safety requirements. Completeness does not apply only to the mandatory kinds of functions but also to the situations that end-users may encounter when they are using the systems. Completeness is difficult and often impossible to reach. This is why groups that design safety-critical systems use simulators in order to multiply the number of situations and cover a broader spectrum. They incrementally accumulate and articulate related knowledge by categorizing relevant situations.

Maturity of knowledge and maturity in design

We claim that knowledge is constructed; let’s say designed. The design of knowledge is incremental. Safety-critical systems are designed over time. They are tested, modified several times and certified. Their use is carefully observed and documented. The resulting observation product, usually called experience feedback, is provided to designers who use it to modify their current understanding of the artifacts they have designed. Knowledge about these artifacts becomes progressively mature through this incremental process. There are short-loop design-knowledge validation processes that lead to official documents guiding the design process. There are also long-loop design-knowledge validation processes that involve experience feedback on already mature artifacts. In particular, engineers involved in safety analysis have an everyday need in using internal official documents. For example for a system safety analyst, requirements, courses, applicable documents, lessons learned (In-Service experience), FAQ, list of experts, previous similar deliverables, review results, validation and verification checklists/action-lists, and system review action lists are crucial information that needs to be easily accessed.

Dealing with drafts

Not all documents are finalized and approved at any time. Designers have to deal with draft documents, addressing questions such as how one can recognize that a document is on-going, how versioning is taken into account, how
revisions are managed. The validation of a document is related to the appropriate list of signatures. When a document is validated it becomes “official”. Each design rationale description should be appropriately contextualized including its status (i.e., mainly the revision and approval dates) and background information (where it is coming from and who did it). In order to follow appropriate guidelines to edit and publish such a document, training may be necessary and guidelines should be easily available. From a broader standpoint, our investigations led to the distinction between private and public spaces of a document, i.e., each technical document has a private space where it is invisible outside of a specific community, and a public space where it is visible by a wider community.

**Awareness and communication between experts**

Situation awareness is a key issue in safety-critical systems. It is much studied at use time. However, it requires more attention at design and development times. People may make errors because they are not aware of the current situation or state-of-the-art. Is Team-1 aware of current actions and productions of Team-2 at the right time? Is Team-1 aware of what Team-3 did a few years ago on the same topic or a similar one? How can we increase awareness? In some cases, it would be nice to have the appropriate information pushed to the front so potential users are aware of its existence. In addition, efficient search mechanisms should provide the necessary means to pull appropriate information when needed. In both cases, context-sensitive algorithms, that may take the form of software agents, are necessary.

When designers know about a type of incident or accident that involved a piece of equipment that they are designing, they (at least try to) design artifacts in order to provide users with the necessary means to handle related situations in the best possible way. They are expert in their field, i.e., design. People who are likely to provide this “incident/accident” knowledge are human factors specialists, end-users themselves or experiences laid down in appropriate databases and knowledge bases. In any case, experts need to communicate either in a live way, e.g., using computer-supported cooperative work environments, or a remote way, e.g., using knowledge bases. Space-time constraints usually impose choices in the way such communication would happen.

**Traceability in space and time**

One of us carried out an exhaustive study on traceability within a large aircraft manufacturing company (Boy, 2001). Traceability is not only information retrieval, it also deals with awareness that potential knowledge exists somewhere, and finally when this knowledge is found, it must be understood correctly (see also Heijst, Spek, & Kruizinga, 1997). Whether they are project teams, corporate organizations or professional communities, groups have difficulty to provide clear explanations of why things were done in a certain way. This is due to the geographical spread-out of people composing these groups, speed of technology evolution, high turn-over of personnel, and lack of documentation of the design process. We will see below that writing is a key issue that cannot be removed from the design activity. People need to know salient reasons that pushed other people to design artifacts they are currently responsible for. This remains true during the whole life cycle of an artifact.

**Trust in knowledge**

Whenever someone gets knowledge from someone else, a crucial issue is to figure out if it is reliable. Do I trust this knowledge without experiencing it? What are the processes that I would need to implement to believe that this knowledge is reliable? The use of Web technology opens our horizons to a wider spectrum of knowledge providers, but we are less sure that acquired knowledge might be trusted. The level of maturity needs to be clearly understood. Consequently, knowledge should come with contextual information that reinforces our understanding of its maturity and context if use.

In the study of Bonini, Jackson and McDonald (2001) three dimensions of trust were found of importance: belief, dependence and experience. If you have to trust the information coming from others, you have to be confident in the other and the information provided, you are dependent because you need the information and you rely on the experience you have with this person and the information. In design processes, the designer is regularly in a dependent position, because preliminary versions are shared between group members and designs of other, related systems are often also in a not yet stable version (participatory design).

Especially in the design of safety critical systems one has to make sure that the knowledge that is shared is correct and can be trusted. For this reason extensive validation and document version management is in place in industries.
One should avoid the risk to base one’s design on information that has not been verified and designers should be aware what the latest version of a document is in order to use it.

**Design is writing and writing is design**

Knowledge management for safety-critical systems mostly deals with documentation since everything should be traceable and formally validated. Consequently, the way things are written is crucial. However, writing is not always perceived as a key issue in engineering and design. Engineers are not scientists who base their careers on the number and quality of the papers they produce. A technical document may be generated the day before delivery just because it was planned to do so. Engineering culture is based on creativity and efficiency, based on very specific languages, often in the form of drawings and schematics that cannot be understood by an outsider. Engineers do not perceive the writing-for-all philosophy as relevant.

**Two separate worlds: engineering and literature**

The distinct worlds of engineering and literature barely met during the last century. The human-computer interaction (HCI) community has nicely introduced design “into the picture” since user interfaces require a subtle combination amount of technique and graphical art. There, science and arts met. In knowledge management, a deeper step is required. Designers need to step into literature. They need to write technical documents describing requirements, specifications, job orders, evaluation rationale, training and performance support, experience feedback and a large variety of official documents. It has been observed that people who are already in senior positions in an organization know the benefit of good documentation, and tend to write more that younger employees who do not have as much experience. Document content should satisfy the objectives, i.e., answering the question: why and for whom are we writing this document?

In addition, in international environments such as contemporary European multi-national companies, writing in English may be a difficult task for non-native English-speaking personnel. The result is that produced English-written documents may be difficult to understand.

**The time-for-writing issue**

Project deadlines are always very short and do not allow for enough time for decent writing. In an engineering organization, the real job is design, not writing. People are usually awarded on design performance issues, not on documentation issues. Writing time should be clearly planned in a project schedule and given the same priority as other activities, so that when there is an extension in the duration of the project, writing is not the last item on the agenda when there is little time left to perform it, as it is often the case.

**What is obvious for someone (expert) is not necessarily for someone else**

There is no consensus whether writing has improved over the years in the aeronautics domain for example. However, some people think that most aerospace technical documents generated during the 1960s are remarkably precise. They were not ambiguous. Work was very well done. People had time and resources to write properly. Other people think that current engineers do not capitalized the necessary technical background to produce appropriate and sufficiently detailed technical documents. It is very important that a selected group of readers reviews all documents. If someone does not understand a technical document then it should be modified and improved towards a better comprehension. We should apply to documents the same kind of usability testing and user-centered design procedures as for systems. Human factors principles are very similar. After Norman (1992), sometimes we say, “writing is design, and design is writing.”

**Redefining prose rules using multimedia**

This statement claims that the quality of technical documentation contributes to the quality of design. We usually write for potential readers. In the same way, we design for potential users. Researchers know that several persons must review papers before being delivered outside. We also know that several persons must test artifacts before being delivered outside. The reader of a multimedia document has become a user of a software application. From this viewpoint, reading a physical note, report or book has evolved towards interacting with a computer. Writing has also evolved towards design of interactive software. Writing words, phrases, paragraphs and chapters has become designing objects and software agents. Static paper documents have become (inter)active documents.
The active part of a book (system) is the reader (user). In addition, the organization of the book (system), the way phrases (objects) are written (designed), style and lexicon used suggest reader (user) activity. Sometimes, the reader (user) hardly understands what the author (designer) wanted to express. Instead of mobilizing the cognition of the reader (user) on interaction problems, the most important part of the cognitive activity of the reader (user) should be centered on the understanding and interpretation of (active) document content.

Toward simplicity
Design documents are not only outputs of design processes but also inputs, i.e., formulating design rationale contributes to improving the design itself. There are two issues of simplicity: documenting to improve the simplicity of use of a system being developed; and reducing the difficulty of generating technical documents, i.e., making it simpler. Simplest systems are best used. In most cases, when systems are too complicated, they are not used at all. This is true both for the system being developed and for its documentation.

Writing from bottom-up (annotations) vs. top-down (requirements)
People tend to write little notes either by using post-its, personal notebooks, page marks, and so on. They annotate what they do and use these notes in order to improve the capacities of their own short-term and long-term memories. If this kind of practice is very useful to people themselves, on a short term, interoperability becomes a problem when such knowledge needs to be exchanged with others or reused by the same person after a longer period of time. Annotations can be considered as pragmatic knowledge that needs to be structured if it is to be used by others. People cannot structure such knowledge in the first place because it is intrinsically situated, i.e., it is captured in context to keep its full sense. This is why a mechanism that would support annotation generation and structuring can be a powerful tool.

Appropriate tools and organizational setups
In industries that develop safety-critical systems, a variety of knowledge management tools are available. Also in R&D projects (including projects in the European Frameworks), many KM tools have been developed. It is clear that tools cannot be designed and used without appropriate organizational setups. People adapt to technology and groups, whether they are teams, organizations or communities. However, adaptation can be limited by the constraints imposed by tools and socio-cultural habits of the people involved.

Active notifications of changes in design
Designers of safety-critical systems are expected to be proactive people who manage information using available tools in their organizational setups. However, information technology is capable of augmenting their initial skills. Software agents may provide assistance in a variety of tasks that require routine, and usually boring, actions. Safety-critical technology always incrementally changes due to accidents and incidents, customer requirements and needs that continuously evolve and refinement of the technology itself. There is always a discrepancy between these effective changes of technology and its related operational documentation. People need to be notified about changes in order to operate such technology in a safe way. When such notification is timely, it is usually passive and left to the expertise or intuition of the user, it may not be noticed. This is why a system that would provide proactive notification of changes would be tremendously useful. In the WISE environment, people can subscribe to documentation, indicating about which changes (updates, deletion, status changes etc.) they want to be notified, by email or in the active work environment.

Supporting the writing process
Above we have emphasized the importance of writing for the design process. Tools are available that can support engineers in documenting their work, and capture annotations during the design work, not just after the design is finished. An example of such a tool is the Computer Integrated Documentation (CID) system developed at NASA (Boy, 1991). Another example can be found in the IMAT (Integrating Manuals and Training) system developed for designing learning material (de Hoog et al., 2002). Also in the WISE workspace the engineer is enabled to make annotations to all different kinds of knowledge objects and to choose whether to share them with team-members or others.

Crisp and clearly understood design rationale is a good indicator of maturity in design. Formalisms have been developed to describe design rationale such as gIBIS (graphical Issue-Based Information System) (Conklin & Begeman, 1989) or QOC (Questions Options Criteria) (MacLean et al., 1991). They support the elicitation of design
rationale and enable the documentation of design decisions, development plans and systems that are effectively developed.

**Organization of personal and team work-spaces**

In current communication and cooperation software, very efficient search engines are available; bottlenecks are elsewhere. They are in the way people categorize incoming information with respect to what is already available on their desktop. This categorization is a strong condition for further retrieval and traceability. People organize their workspace in order to perform their tasks efficiently and manage time and content accordingly. They use post-its, bookmarks, documents piles, proximity for urgent or frequent access, and so on. In any case, people don’t stop to fine-tune their initial categorization to better fit their everyday needs. In the WISE project we have developed an environment in which users have a personal workspace in which they can organize the knowledge they need for their task, as well as a workspace for groups in which knowledge can be pre-structured and shared. The environment consists of a portal that gives access to the companies documentations, databases and tools, including search facilities on all knowledge objects thus available, of whatever format or location.

**Active design documents**

The concept of active design document (ADD) (Boy, 1997) was developed to support designers of safety-critical systems in knowledge management. Active documentation may take various forms and involve different kinds of content. An ADD is defined by four categories that organize designer’s workspace: interface objects, interaction descriptions, contextual links and design rationale.

*Interface objects* (IOs) provide appropriate, useful and natural illusions of designed artifacts. IOs have their own behavior reflecting the behavior of related artifacts. They enable users to test usefulness and usability of related artifacts. They provide concrete feeling and grasp of the use of an artifact, its learning requirements, its purpose hands-on, etc. Their progressive integration leads to a series of prototypes and, in the end, the final product.

*Interaction descriptions* (IDs) provide the specification of user-artifact dialogue. IDs may be expressed in either natural language, or a domain-specific technical language ranging from textual descriptions in simplified English (operational procedures for example) to a knowledge representation like the interaction blocks (Boy, 1998). A main advantage of using interaction blocks is to enable formal testing of interaction complexity, and expressing contexts and abnormal conditions of use explicitly.

A test user either follows IDs and produces an activity by using appropriate related IOs, or interacts directly with IOs and verifies the validity of related IDs. In both cases, he or she tests the links between IOs and IDs in context (i.e., in the context of the task being performed). The corresponding category is called *contextual links* (CLs). This is where usefulness and usability evaluations (sometimes annotations) are stored in the form of either free text or specific preformatted forms.

*Design rationale* (DR) provides the reasons why the IOs and IDs of an artifact have been designed the way they are, and design alternatives that were not chosen. DR is commonly implemented by using semi-formal languages such as gIBIS or QOC already mentioned.

ADDs are tools that support not only communication and mediation, but also prototyping and evaluation. They enable their users to store design knowledge according to a concrete and systematic formalism. Creation and maintenance of such ADDs enable an entire organization to maintain awareness of their design processes and products.

**Interoperable documents and the Portal concept**

In today information systems, documents are not always interoperable, i.e., when an electronic document is received from somewhere else the recipient should be able to read it, understand it and process it as he/she wants. This requirement induces two kinds of issues: standards and integrated environments. When people exchange documents across teams, organizations and communities, they expect the others to be able to process what they provide. This is commonly a matter of standards. In a closed world where an organization can cope with an integrated environment
in the form of intranet for example, people don’t have to worry about standards. Nevertheless, standards progressively emerge from the extensive use of specific types of documents.

Designers require KM environments that are user-centered (easy to use, and avoiding overload) and integrated within their current tasks. They should have easy access to KM services at each design step. For example, in a safety assessment process, there should be information provided for performing safety analysis and related documents. In other words, the designer workspace should be (re)designed in such a way that he or she has easy access to experience feedback (e.g., not only a list of what is necessary to do and forbidden (checklists), but providing deep knowledge to foster preventive design actions and avoid later corrective actions) at any time. Having this knowledge available at the designer’s desktop at all times can be realized by a KM portal. A portal means that it provides access to knowledge, wherever it is located, but does not contain this knowledge itself. In the KM portal developed in WISE designers have access to knowledge available in for example databases with experience feedback, lessons-learned and best practices, to all kinds of relevant documents, and to people who can bring interesting knowledge and experience. Access to all these sources is provided in the same manner and with a single search facility.

**Future Trends**

Important questions remain to be answered: does KM technology change the job of engineers? Does it free up engineers from boring tasks? Or does it create new ones? Answers to these questions are complex. However, this paper contributes by providing categories of KM solutions such as the organization of personal and team workspaces, active design documents, and knowledge portals. Usefulness and usability of such solutions need to be tested carefully in a real-world environment with a critical mass of people involved. This is very difficult to do since experts and specialists (e.g., designers of safety critical systems) are always occupied, busy and constrained into an already existing KM system, often very far from the solutions proposed. Transformations should be incremental, accepted by the people involved. Implementing a new KM system is also redefining a new philosophy of work, a new culture. This is hard to do and hard to implement! This is the main reason why the design of new KM technology must be human-centered, i.e., team-centered, organization-centered and community-centered. Each of these types of group has its own motivations, requirements and constraints.

**Conclusion**

The way knowledge is exchanged during the design and the further life cycle of a safety-critical system induces several technological factors (complexity, completeness, maturity, traceability) and human factors (expertise, writing, simplicity, drafts, information credibility, uncertainty and awareness).

Several actual developments influence the design processes of safety-critical systems: more people from different organizations (within the company or (sub)contractors) get involved, more procedures are in place (such as certification procedures, involving human factors in particular), product development needs to be faster than before. These evolutions have a direct impact on the increase of both the number and content of documents. Information technology provides new means to generate, maintain and use such documents. A main issue is to improve the use of such means.

**References**


The concept of active design document (ADD) (Boy, 1997) supports designers of safety-critical systems in knowledge management. Active documentation may take various forms and involve different kinds of content. An ADD is defined by four categories that organize designer’s workspace: interface objects, interaction descriptions, contextual links and design rationale.

### Complexity/simplicity
Safety-critical systems are usually complex because they include hardware and software that tend to improve local and global safety. Complexity often comes from the inter-connection of processes and people. The designers of safety-critical systems should always have in mind to design for simplicity.

### Information awareness
Information awareness is provided by clues or mechanisms that suggest to a user that an appropriate piece of information (or knowledge) exists and is reachable.

### Interoperable document
An electronic document is interoperable if it can be processed by anyone on any computer. Operating system compatibility, versioning and publishing philosophies are typical issues of lack of document interoperability.

### Maturity of knowledge
Systems and knowledge associated with these systems are designed and refined almost at the same time in a spiral process: one supports the design of the other and conversely. Maturity occurs when both system and associated knowledge are mastered by both designers and users. Maturity is often related to trust.

### Safety-critical system
A system is safety-critical when it involves human risks in its environment.

### Traceability
Traceability is the process of suggesting appropriate information-awareness clues or mechanisms, retrieving appropriate information, and providing understandable information or knowledge.