10 Sociotechnical systems

Objectives (understanding)

1. what **is meant by a sociotechnical system**
2. **difference between a technical**, **computer-based system and a sociotechnical system**
3. concept of **emergent system properties, such as reliability, performance, safety, and security;**
4. the **procurement, development, and operational activities**
5. why software **dependability and security should not be considered in isolation** and how they are affected by systems issues, such as operator errors.

Sociotechnical systems are so complex that it is practically impossible to understand them as a whole. Rather, you have to view them as layers, as shown in Figure 10.1. These layers make up the sociotechnical systems stack:

10.1 Complex systems

Abstract systems**, such as the system of government.**

A system is a purposeful collection of interrelated components, of different kinds, which work together to achieve some objective.

Systems that include software fall into two categories:

1. ***Technical computer-based systems*** These are systems that include hardware andsoftware components **but not procedures and processes**. Examples of technical systems include **televisions, mobile phones, and other equipment with embedded software**. Most **software for PCs, computer games, etc., also falls into this category**
2. ***Sociotechnical systems*** These include **one or more technical systems** but, crucially, also **include people who understand the purpose of the system within the system itself**. Sociotechnical systems have defined operational **processes and people** (the operators) are inherent **parts of the system**. They are governed by organizational policies and rules and may be affected by external constraints such as national laws and regulatory policies

Organizational factors from the system's environment that may affect the requirements, design, and operation of a sociotechnical system include:
1***. Process changes*** The system **may require changes** to the work processes in the environment. If so, **training will certainly be required**. **If changes are significant**, or if they involve people losing their jobs, there **is a danger that the users will resist the introduction of the system**.

2. ***Job changes*** New systems may **de-skill the users** in an environment or **cause them to change the way they work**. If so, users may actively resist the introduction of the system into the organization. Designs that involve managers having to change their way of working to fit a new computer system are often resented. The managers may feel that their status in the organization is being reduced by the system.
3. ***Organizational changes*** The **system may change the political power structure in an organization**. For example, if an organization is dependent on a complex system, those who control access to that system have a great deal of political power.

Sociotechnical systems have three characteristics that are particularly important when **considering security and dependability**:
1. **They have emergent properties** that are properties of the system as a whole, rather than associated with individual parts of the system. Emergent properties **depend on both the system components and the relationships between them**. Given this complexity, the emergent properties can only be evaluated once the system has been assembled. **Security and dependability are emergent system properties**.
2**. They are often nondeterministic**. This means that when presented with **a specific input, they may not always produce the same output.** The system's behavior **depends on the human operators and people do not always react in the same way**. Furthermore, use of the system may create new relationships between the system components and hence change its emergent behavior. System faults and failures may therefore be transient, and people may disagree about whether or not a failure has actually occurred.
3. **The extent to which the system supports** organizational objectives does **not just depend on the system itself. It also depends on the stability of these objectives, the relationships, and conflicts between organizational objectives and how people** in the organization **interpret these objectives**. New management may reinterpret the organizational objectives that a system was designed to support so that a 'successful' system may then be seen as a 'failure'.

10.1.1 Emergent system properties

There are two types of emergent properties:
1**. Functional emergent properties** when the purpose of a system only emerges after its components are integrated. For example, a bicycle has the functional property of being a transportation device once it has been assembled from its components.
2**. Non-functional emergent properties**, which relate to the behavior of the system in its operational environment. Reliability, performance, safety, and security are examples of emergent properties. These are critical for computer-based systems, as failure to achieve a minimum defined level in these properties usually makes

Emergent dependability properties, such as reliability, depend on both the properties of individual components and their interactions
In a sociotechnical system, you need to consider reliability from three perspectives:
1. Hardware reliability What is the probability of hardware components failing and how long does it take to repair a failed component?
2. Software reliability How likely is it that a software component will produce an incorrect output? Software failure is distinct from hardware failure in that software does not wear out. Failures are often transient. The system carries on working after an incorrect result has been produced.
3. Operator reliability How likely is it that the operator of a system will make an error and provide an incorrect input? How likely is it that the software will fail to detect this error and propagate the mistake?

10.1.2 Non-determinism

Sociotechnical systems **are non-deterministic** partly because they include people and partly because changes to the hardware, software, and data in these systems **are so frequent**. The interactions between these changes are complex and so the behavior

10.1.3 Success criteria

it is difficult to define the success criteria for a system. How do you decide if a new system contributes, as planned, to the business goals of the company that paid for the system?

The judgment of success is not usually made against the original reasons for procuring and developing the system. Rather, it is based on whether or not the system is effective at the time it is deployed. As the business environment can change very quickly, the business goals may have changed significantly during the development of the system.

The situation is even more complex when there are multiple conflicting goals that are interpreted differently by different stakeholders. For instance, the system on which the MHC-PMS is based was designed to support two distinct business goals:

1. Improve the quality of care for sufferers from mental illness.
2. Increase income by providing detailed reports of care provided and the costs of that care.

Systems engineering

Systems engineering **encompasses all of the activities involved in procuring, specifying, designing, implementing, validating, deploying, operating, and maintaining sociotechnical systems**. Systems engineers are not just concerned with software but also with hardware and the system's interactions with users and its environment.

They must think about the services that the system provides, the constraints under which the system must be built and operated, and the ways in which the system is used to fulfill its purpose or purposes.

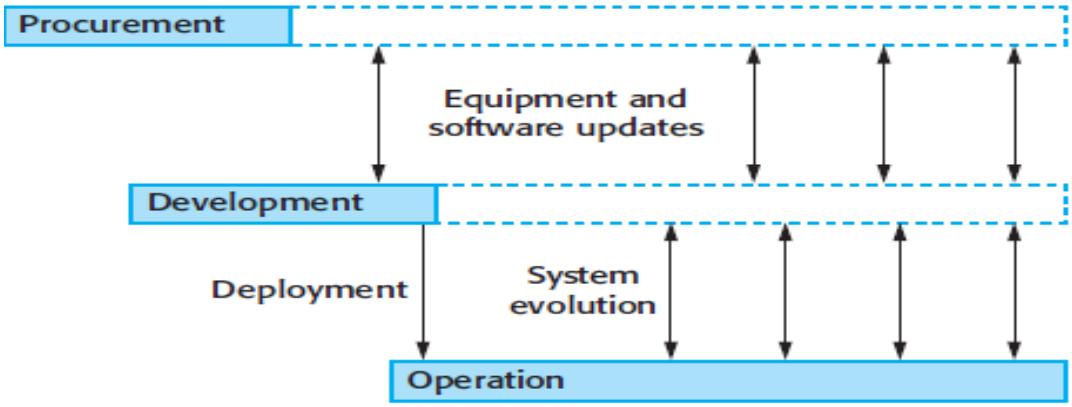
There are three overlapping stages in the lifetime of large and complex sociotechnical systems:

1. *Procurement or acquisition* During this stage, the purpose of a system is decided; high-level system requirements are established; decisions are made on how functionality will be distributed across hardware, software, and people; and the components that will make up the system are purchased.

2. Development During this stage, the system is developed. Development processes include all of the activities involved in system development such as **requirements definition, system design, hardware and software engineering, system integration, and testing**. Operational processes are defined and the trening courses for system users are designed.

3. Operation At this stage, **the system is deployed, users are trained, and the system is brought into use.** The planned operational processes usually then have to change to reflect the real working environment where the system is used. Over time, **the system evolves as new requirements are identified**. Eventually, the system declines in value and it is decommissioned and replaced.

These stages are not independent.



An important difference between systems and software engineering **is the involvement of a range of professional disciplines throughout the lifetime of the system.**

For example,

- **the technical disciplines** that may be involved in the procurement and development of a new system for air traffic management
- **Architects and civil engineers** are involved because new air traffic management systems usually have to be installed in a new building.
- **Electrical and mechanical engineers** are involved to specify and maintain the power and air conditioning.
- **Electronic engineers** are concerned with computers, radars, and other equipment.
- **Ergonomists design the controller** workstations and software engineers and user interface designers are responsible for the software in the system.

The involvement of a range of professional disciplines is essential because there are so many different aspects of complex sociotechnical systems. However, differences between disciplines can introduce vulnerabilities into systems and so compromise the security and dependability of the system being developed:

1. Different disciplines use the same words to mean different things. Misunderstandings are common in discussions between engineers from different backgrounds. If these are not discovered and resolved during system development, they can lead to errors in delivered systems. For example, an electronic engineer who may know a little bit about C# programming may not understand that a method in Java is comparable to a function in C.
2. Each discipline makes assumptions about what can or can't be done by other disciplines. These are often based on an inadequate understanding of what is actually possible. **For example, a user interface designer may propose a graphical UI for an embedded system that requires a great deal of processing and so overloads the processor in the system**.
3. Disciplines try to protect their professional boundaries and may argue for certain design decisions because these decisions will call for their professional expertise. Therefore, a software engineer may argue for a software-based door locking system in a building, although a mechanical, key-based system may be more reliable.

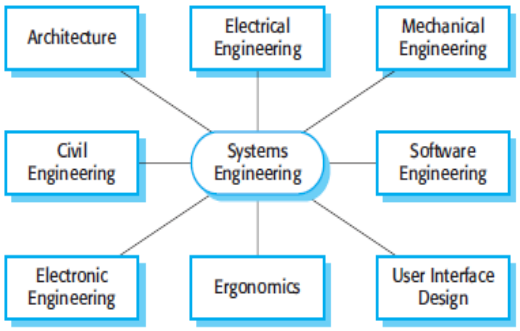


Figure 10.5 Professional disciplines involved in systems engineering

System procurement

The initial phase of systems engineering is system procurement (sometimes called system acquisition). At this stage, decisions are made on the scope of a system that is to be purchased, system budgets and timescales, and the high-level system requirements. Using this information, further decisions are then made on whether to procure a system, the type of system required, and the supplier or suppliers of the system.

The drivers for these decisions are:

1. *The state of other organizational systems* If the organization has a mixture of systems that cannot easily communicate or that are expensive to maintain, **then procuring a replacement system may lead to significant business benefits**.
2. *The need to comply with external regulations* Increasingly, businesses are regulated and have to demonstrate compliance with externally defined regulations (e.g., Sarbanes-Oxley accounting regulations in the United States). This may
require the replacement of noncompliant systems or the provision of new systems specifically to monitor compliance.
3. External competition If a business needs to compete more effectively or maintain a competitive position, investment in new systems that improve the efficiency of business processes may be advisable. For military systems, the need to improve capability in the face of new threats is an important reason for procuring new systems.
4. Business reorganization **Businesses and other organizations frequently restructure with the intention of improving efficiency and/or customer service. Reorganizations lead to changes** in business processes that require new systems support.
5. Available budget The **budget available is an obvious factor in determining the scope of new systems that can be procured**.

System development

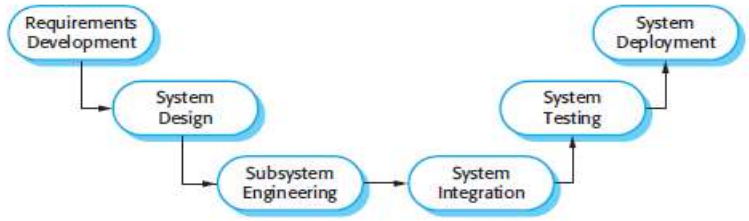


Figure 10.7 Systems development

The goals of the system development process are to develop or acquire all of the components of a system and then to integrate these components to create the final system. The requirements are the bridge between the procurement and the development processes. During procurement, business and high-level functional and nonfunctional system requirements are defined. You can think of this as the start of development, hence the overlapping processes shown in Figure 10.4. Once contracts
for the system components have been agreed, more detailed requirements engineering
then takes place.

Plan-driven processes are used in systems engineering because different parts of the system are being developed at the same time. For systems that include hardware and other equipment, changes during development can be very expensive or, sometimes, practically impossible. It is essential therefore, that the system requirements are fully understood before hardware development or building work begins.

Reworking the system design to solve hardware problems is rarely possible. For this reason, more and more system functionality is being assigned to the system software. This allows some changes to be made during system development, in response to new system requirements that inevitably arise.

One of the most confusing aspects of systems engineering is that companies use different terminology for each stage of the process. **The process structure also varies**.Sometimes, **requirements engineering is part** of the development process and **sometimes it is a separate activity.**

However, there are essentially six fundamental activities in systems development:
1. *Requirements development*
2. System design
3. Subsystem engineering

4. System integration
5. System testing
6. System deployment

The requirements and the highlevel design are developed concurrently. Constraints posed by existing systems may limit design choices and these choices may be specified in the requirements. You may have to do some initial design to structure and organize the requirements engineering process. As the design process continues, you may discover problems with existing requirements and new requirements may emerge. Consequently, you can think of these linked ***processes as a spiral, as shown in*** Figure 10.8.

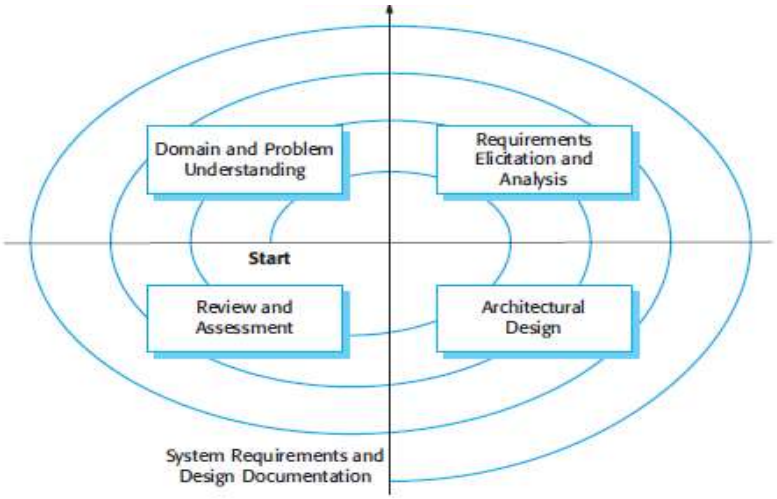


Figure 10.8
Requirements and design spiral

1. It is usually impossible to schedule the development of the subsystems so that they are all finished at the same time.
2. Incremental integration reduces the cost of error location. If many subsystems are **simultaneously integrated, an error that arises during testing may be in any of these subsystems**. When a single subsystem is integrated with an already working system, e**rrors that occur are probably in the newly integrated subsystem** or in the interactions between the existing subsystems and the new subsystem

System operation

Operational processes are the processes that are involved in using the system for its defined purpose. For example, operators of an air traffic control system follow specific processes when aircraft enter and leave airspace, when they have to change height or speed, when an emergency occurs, and so on. For new systems, these operational processes have to be defined and documented during the system development process. Operators may have to be trained and other work processes adapted to make effective use of the new system. Undetected problems may arise at this stage because the system specification may contain errors or omissions. Although the system may perform to specification, its functions may not meet the real operational needs.

Consequently, the operators may not use the system as its designers intended. The key benefit of having system operators is that people have a unique capability of being able to respond effectively to unexpected situations, even when they have never had direct experience of these situations.

Therefore, when things go wrong, the operators can often recover the situation although this may sometimes mean that the defined process is violated. Operators also use their local knowledge to adapt and improve processes. Normally, the actual operational processes are different from those anticipated by the system designers.

. A problem that may only emerge after the system goes into operation is the operation of the new system alongside existing systems. There may be physical problems of incompatibility or it may be difficult to transfer data from one system to another.

More subtle problems might arise because **different systems have different user interfaces**. Introducing a new system may **increase the operator** error rate, as the operators use user interface commands for the wrong system.

10.5.1 Human erorr

I suggested earlier in the chapter that non-determinism was an important issue in sociotechnical systems and that one reason for this is that the people in the system do not always behave in the same way. Sometimes they make mistakes in using the system and this has the potential to cause system failure. For example, an operator may forget to record that some action has been taken so that another operator (erroneously) repeats that action. If the action is to debit or credit a bank account, say, then a system failure occurs as the amount in the account is then incorrect. As Reason discusses (2000) human errors will always occur and there are two ways to view the problem of human error:

1. *The person approach.* Errors are considered to be the responsibility of the individual and 'unsafe acts' (such as an operator failing to engage a safety barrier) are a consequence of individual carelessness or reckless behavior. People who adopt this approach believe that human errors can be reduced by threats of disciplinary action, more stringent procedures, retraining, etc. Their view is that the error is the fault of the individual responsible for making the mistake.

2. *The systems approach.* The basic assumption is that people are fallible and will make mistakes. The errors that people make are often a consequence of system design decisions that lead to erroneous ways of working, or of organizational factors, which affect the system operators. Good systems should recognize the possibility of human error and include barriers and safeguards that detect human errors and allow the system to recover before failure occurs. When a failure does occur, the issue is not finding an individual to blame but to understand how and why the system defences did not trap the error.

I believe that the systems approach is the right one and that systems engineers **should assume that human errors will occur during system operation**. Therefore, to improve the security and dependability of a system, **designers have to think about the defenses and barriers to human error that should be included in a system.** They should also think about whether these barriers should be built into the technical components of the system. If not, they could be part of the processes and procedures for using the system or could be operator guidelines that are reliant on human checking and judgment.

Examples of defenses that may be included in a system are:

1. An air traffic control system may include an automated conflict alert system. When a controller instructs an aircraft to change its speed or altitude, the system extrapolates its trajectory to see if it could intersect with any other aircraft. If so, it sounds an alarm.
2. The same system may have a clearly defined procedure to record the control instructions that have been issued. These procedures help the controller check if they have issued the instruction correctly and make the information available to others for checking.
3. **Air traffic control usually involves a team of controllers who constantly monitor each other's work**. Therefore, when a mistake is made, it is likely that it will be detected and corrected before an incident occurs.

Inevitably, all barriers have weaknesses of some kind. Reason calls these 'latent conditions' as they usually only contribute to system failure when some other problem occurs. For example, in the above defenses, a weakness of a conflict alert system is that it may lead to many false alarms. Controllers may therefore ignore warnings from the system. A weakness of a procedural system may be that unusual but essential information can't be easily recorded**. Human checking may fail when all of the people involved are under stress and make the same mistake**. Latent conditions lead to system failure when the defenses built into the system do not trap an active failure by a system operator. **The human error is a trigger for the failure but should not be considered to be the sole cause of the failure**. Reason explains this using his well-known 'Swiss cheese' model of system failure

(Figure 10.9).

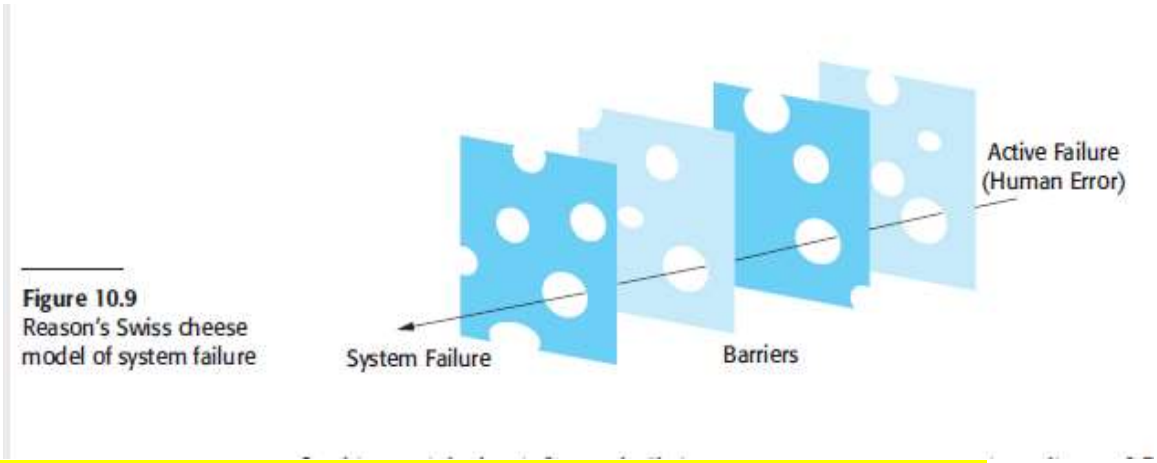


Figure 10.9
Reason's Swiss cheese model of system failure

To reduce the probability that system failure will result from human error, designers should:
1. Design a system so that different types of barriers are included. This means that the 'holes' will probably be in different places and so there is less chance of the holes lining up and failing to trap an error.
2. Minimize the number of latent conditions in a system. Effectively, this means reducing the number and size of system 'holes'. Of course, the design of the system as a whole should also attempt to avoid the active failures that can trigger a system failure. This may involve designing the operational processes and the system to ensure that operators are not overworked, distracted, or presented with excessive amounts of information.

10.5.2 System evolution

Large, complex systems have a very long lifetime. During their life, they are changed to correct errors in the original system requirements and to implement new requirements that have emerged. The system's computers are likely to be replaced with new, faster machines. The organization that uses the system may reorganize itself and hence use the system in a different way. The external environment of the system may change, forcing changes to the system. Hence evolution, where the system changes to accommodate environmental change, is a process that runs alongside normal system operational processes. System evolution involves reentering the development process to make changes and extensions to the system's hardware, software, and operational processes.

Legacy systems
Legacy systems are sociotechnical computer-based systems that have been developed in the past, often using older or obsolete technology. These systems include not only hardware and software but also legacy processes and procedures—old ways of doing things that are difficult to change because they rely on legacy software. Changes to one part of the system inevitably involve changes to other components. Legacy systems are often business-critical systems. They are maintained because it is too risky to replace them.
http://www.SoftwareEngineering-9.com/LegacySys/

System evolution, like software evolution is inherently **costly for several reasons**:
1. **Proposed changes have to be analyzed very carefully from a business and a technical perspective**. Changes have to contribute to the goals of the system and should not simply be technically motivated.
2. **Because subsystems are never completely independent**, changes to one subsystem may adversely affect the performance or behavior of other subsystems. Consequent changes to these subsystems may therefore be needed.
3. The reasons for original design decisions are often unrecorded. Those responsible for the system evolution have to work out why particular design decisions were made.
4. **As systems age, their structure typically becomes corrupted by change so the costs of making further changes increases**.

Systems that have evolved over time are often reliant on obsolete hardware and software technology. If they have a critical role in an organization, they are known as '**legacy**

systems'. These are usually systems that the organization would like to replace but don't do so as the risks or costs of replacement cannot be justified.

From a dependability and security perspective, changes to a system are often a source of problems and vulnerabilities. If the people implementing the change are different from those who developed the system, they may be unaware that a design decision was made for dependability and security reasons. Therefore, they may change the system and lose some safeguards that were deliberately implemented when the system was built. Furthermore, as testing is so expensive, complete retesting may be impossible after every system change. Adverse side effects of changes that introduce or expose faults in other system components may not then be discovered.

E XERCISES

10.1. Give two examples of government functions that are supported by complex sociotechnical systems and explain why, in the foreseeable future, these functions cannot be completely automated.
10.2. Explain why the environment in which a computer-based system is installed may have unanticipated effects on the system that lead to system failure. Illustrate your answer with a different example from that used in this chapter.
10.3. Why is it impossible to infer the emergent properties of a complex system from the properties of the system components?
10.4. Why is it sometimes difficult to decide whether or not there has been a failure in a sociotechnical system? Illustrate your answer by using examples from the MHC-PMS that has been discussed in earlier chapters.
10.5. What is a 'wicked problem'? Explain why the development of a national medical records system should be considered a 'wicked problem'.
10.6. A multimedia virtual museum system offering virtual experiences of ancient Greece is to be developed for a consortium of European museums. The system should provide users with the facility to view 3-D models of ancient Greece through a standard web browser and should also support an immersive virtual reality experience. What political and organizational difficulties might arise when the system is installed in the museums that make up the consortium?
10.7. Why is system integration a particularly critical part of the systems development process? Suggest three sociotechnical issues that may cause difficulties in the system integration process.
10.8. Explain why legacy systems may be critical to the operation of a business.
10.9. What are the arguments for and against considering system engineering as a profession in its own right, like electrical engineering or software engineering?
10.10. You are an engineer involved in the development of a financial system. During installation, you discover that this system will make a significant number of people redundant. The people in the environment deny you access to essential information to complete the system installation. To what extent should you, as a systems engineer, become involved in this situation? Is it your professional responsibility to complete the installation as contracted? Should you simply abandon the work until the procuring organization has sorted out the problem?

EXERCIȚII

10.1. Dați două exemple de funcții guvernamentale care sunt susținute de sisteme sociotehnice complexe și explicați de ce, în viitorul previzibil, aceste funcții nu pot fi complet automatizate.
10.2. Explicați de ce mediul în care este instalat un sistem bazat pe computer poate avea efecte neprevăzute asupra sistemului care duc la eșecul sistemului. Ilustrați-vă răspunsul cu un exemplu diferit de cel folosit în acest capitol.
10.3. De ce este imposibil să se deducă proprietățile emergente ale unui sistem complex din proprietățile componentelor sistemului?
10.4. De ce este uneori dificil să decidem dacă a existat sau nu un eșec într-un sistem sociotehnic? Ilustrați-vă răspunsul utilizând exemple din MHC-PMS care au fost discutate în capitolele anterioare.

10.5. Ce este o „problemă rea”? Explicați de ce dezvoltarea unui sistem național de evidență medicală ar trebui considerată o „problemă rea”.
10.6. Un sistem de muzee virtuale multimedia care oferă experiențe virtuale ale Greciei antice urmează să fie dezvoltat pentru un consorțiu de muzee europene. Sistemul ar trebui să ofere utilizatorilor posibilitatea de a vizualiza modele 3D ale Greciei antice printr-un browser web standard și ar trebui să susțină, de asemenea, o experiență imersivă de realitate virtuală. Ce dificultăți politice și organizaționale ar putea apărea atunci când sistemul este instalat în muzeele care alcătuiesc consorțiul?
10.7. De ce este integrarea sistemului o parte deosebit de critică a procesului de dezvoltare a sistemelor? Sugerează trei probleme sociotehnice care pot cauza dificultăți în procesul de integrare a sistemului.
10.8. Explicați de ce sistemele vechi pot fi esențiale pentru funcționarea unei afaceri.
10.9. Care sunt argumentele pro și contra considerării ingineriei de sistem ca o profesie de sine stătătoare, cum ar fi ingineria electrică sau ingineria software?
10.10. Sunteți un inginer implicat în dezvoltarea unui sistem financiar. În timpul instalării, descoperiți că acest sistem va face ca un număr semnificativ de persoane să fie redundante. Persoanele din mediu vă refuză accesul la informații esențiale pentru a finaliza instalarea sistemului. În ce măsură, ca inginer de sistem, ar trebui să vă implicați în această situație? Este responsabilitatea dumneavoastră profesională să finalizați instalarea conform contractului? Ar trebui pur și simplu să abandonezi munca până când organizația de achiziții a soluționat problema?