
Human & Social Factors in Critical Systems Design

Objectives

- Understand the importance of human factors in systems design
- Understand that social and organisational issues as well as individual human psychology is important
- Appreciate the impact that human factors can have on the success or failure of critical systems.

Human & social factors

- System development is a human intensive process
 - Especially the requirements phase
- Human science research indicates various modes of human behaviour that are prone to errors and failure
 - Individual work
 - cognitive psychology
 - Social group and team work
 - social psychology, sociology
 - Organisational processes and context
 - sociology, organisational behaviour

Human Error

- People are fallible
- Human Error a problematic term
 - Tends to be used when ‘scapegoating’
 - Can cover a multitude of sins
- Human activity in the systems development process can be thought of as *individuals* working in isolation or as part of *social groups or teams*, within a particular *organisational context*
- A focus on these three areas can lead to a better understanding of process *vulnerabilities* to error, and the *defences* which can be put in place against them

Sources of error

- **In the specification of the system**
 - As a result of individual, social or organisational factors, errors are made in the specification of a system
 - Misunderstood responsibility is probably the most common error here
- **In the development of the system**
 - System developers make errors and hence introduce faults into the system
 - Excessive complexity ‘strains our brains’
- **In the use of the system**
 - System operators make mistakes for various reasons
 - The system as understood by the operator is different from the system as understood by the system developers

Types of individual tasks

- Based on work in cognitive psychology
- Rasmussen's s-r-k framework
 - Skill-based: routine, automatic tasks in familiar settings
 - Rule-based: applying pre-packaged solutions to problem solving tasks
 - Knowledge-based: tackling unfamiliar problems in unfamiliar settings
- Skill-based tasks
 - Initiating a program, logging in
- Rule-based tasks
 - Installing a program, making some computation using a spreadsheet
- Knowledge-based tasks
 - Designing a spreadsheet, debugging a program

Individual errors

- Reason's Generic Error Modelling System (GEMS)
 - Based on Rasmussen's framework, informed by diary studies
 - Skill-based slips and lapses
 - Rule- and knowledge-based mistakes
- **Skill-based errors**
 - Login using the wrong password
 - Plug a cable into the wrong socket
- **Rule-based errors**
 - Miss out a stage in the configuration process
 - Enter numbers at the wrong precision
- **Knowledge-based errors**
 - Fail to understand how C pointers work
 - Fail to understand the distinction between references and values in a spreadsheet

Violations

- Differ from errors in that deliberate action is involved
- Often in name only
 - The ‘violation’ keeps the process going
- Can be thought of in terms of s-r-k framework
 - Skill-based: routine and optimising violations
 - Rule-based: Situational violations and ‘misventions’
 - Knowledge-based: Exceptional violations

Individual error recovery

- Errors are unavoidable
- Most research has concentrated on application to the design of interfaces to critical systems
 - But equally applicable to the design process itself
- Slips & lapses easiest to identify and recover from
- Recovering from rule-based errors means that the system has to understand the process and rules associated with some specific intention
 - Automating the process e.g. via wizards can help
- Recovering from knowledge-based errors is very difficult as the system has to know the intentions of the user

Social factors

- Social factors are those factors which relate to people working together in groups. They include:
 - Group interaction
 - Group leadership
 - Group decision making
- Unlike human factors, social factors may be culture specific
 - Models of group working that work in one country/organisation may not work in another
- Group working can reduce errors because of several people are looking at the problem
 - However, good group communications is essential if this is to work properly

Group interaction

- Errors can arise because of poor communication within a group
 - Alice tells Bob that she has changed the configuration file but neither Bob nor Alice tell Charlie because Charlie has been rude to them
 - Charlie changes the configuration file in a way that is incompatible and the system fails
- **Status and experience affect group interaction**
 - Alice is senior to Bob and decides that a particular component should be used. Bob knows that there are problems with this but also knows that Alice is sensitive to criticism. He needs a reference from her for promotion so he says nothing. The component and the system fails.

Group leadership

- Good leadership is critical for good group performance
 - Autocratic leaders tend to inhibit communications and so undetected errors are likely to arise
 - Poor leadership exposes the group to competing demands and conflicts and this, again, tends to lead to errors being made
 - Political issues become important if leadership is poor
 - Failures may be deliberately allowed to happen in order to undermine the group leader
 - Group dissension increases and hence communication tends to break down

Group decision making

- Poor decision making in either a design or operational context can lead to errors
- Group decision making is influenced by
 - *Status and seniority*. People with a higher status have more say in the decision making process
 - *Group organisation*. If the group organisation does not support open discussion and consideration of alternatives then decision errors are more likely
 - *Group cohesiveness*. Very cohesive and tightly-knit groups suffer from ‘groupthink’. They deliberatively avoid choices that threaten the cohesiveness of the group

Group performance failures

- Working in groups can counter some of the errors which individuals are prone to make
 - But it can introduce problems of its own
- Group performance differs from the collective effort of the individuals concerned based on the type of task performed
 - Can vary from better than the best, to equal to the worst individual in the group
- The quality of decisions made is dependent upon the make-up of the group (personalities, seniority, skills, etc.), the nature of the task, and external factors beyond the group's control

Vulnerabilities and defences

- As with other types of failure, failure-based analysis can consider the issues of human and social factors
- Vulnerabilities
 - Identify the areas where human error can lead to system failure
- Defences
 - Put defences in place that avoid, tolerate or recover from human error
- Consider both the development process and the system being developed

User interface design

- Systems should be designed with the assumption that the operators of these systems will make mistakes
- Human cognitive capabilities must be considered in the user interface design
 - People are not good at repetitive work without making errors
 - People are not good at monitoring but are good at reacting
 - People's behaviour is not consistent - they may respond to the same stimulus in different ways at different times
 - People rarely follow processes exactly as specified but change them to suit local circumstances and expertise

Operating under stress

- As people are placed under stress, then their ability to work without making mistakes is reduced
- A failing critical system is a stressful situation for the system operator so once failures start, they have a tendency to generate more failures
- User interfaces for critical should therefore be designed for failure situations not normal situations

Key points

- System design is a human intensive activity
- Understanding the possible human causes of failures systems development can lead to improved processes and fewer failures
- Human factors have been reviewed in terms of *individuals* working in *social* groups and teams, within an *organisational* context.
- System user interfaces must be designed to take account of human error

The London Ambulance fiasco

- The London Ambulance Service (LAS) Computer Aided Despatch (CAD) system
- Failed dramatically on October 26th 1992 shortly after it was introduced:
 - The system could not cope with the load placed on it by normal use
 - The response to emergency calls was several hours
 - Ambulance communications failed and ambulances were lost from the system
- Catalogue of errors made in the procurement, design, implementation, and introduction of the system

London Ambulance Service

- Managed by South West Thames Regional Health Authority
- Largest ambulance service in the world (LAS inquiry report)
 - Covers geographical area of over 600 square miles
 - Resident population of 6.8 million people (greater during daytime, especially central London)
 - Carries over 5,000 patients every day
 - 2,000-2,500 calls received daily, of which 1,300-1,600 are 999 calls

Computer-aided despatch systems

- Provide one or more of the following:
 - Call taking
 - Resource identification
 - Resource mobilisation
 - Ambulance resource management
- Consist of:
 - CAD software & hardware
 - Gazetteer and mapping software
 - Communications interface (RIFS)
 - Radio system
 - Mobile data terminals (MDTs)
 - Automatic vehicle location system (AVLS)

The manual system to be replaced

- **Call taking**
 - Recorded on form; location identified in map book; forms sent to central collection point on conveyor belt
- **Resource identification**
 - Form collected; passed onto resource allocator depending on region; duplicates identified. Resource allocator decides on which resource to be mobilised; recorded on form and passed to dispatcher
- **Resource mobilisation**
 - Dispatcher telephones relevant ambulance station, or passes mobilisation instructions to radio operator if ambulance already on road
- **Whole process meant to take <3 minutes**

Concept/design of the CAD system

- Existing systems dismissed as inadequate and impossible to modify to meet LAS's needs
 - Intended functionality “greater than available from any existing system”
- **Desired system:**
 - to consist of Computer Aided Dispatch; Computer map display; Automatic Vehicle Location System (AVLS)
 - Must integrate with existing MDTs and RIFS (Radio Interface System)
- **Success dependent upon:**
 - Near 100% accuracy and reliability of technology
 - Absolute cooperation from all parties including CAC staff and ambulance crews

Problems: Procurement (i)

- Contract had to be put out to open tender
 - Regulations emphasis is on best price
 - 35 companies expressed interest in providing all or part of the system
 - Most raised concerns over the proposed timetable of less than 1 year until full implementation
- Previous Arthur Andersen report largely ignored
 - Recommended budget of £1.5M and 19 month timetable for packaged solution. Both estimates to be significantly increased if packaged solution not available
 - Report never shown to new Director of Support Services
- Only 1 out of 17 proposals met all of the project team's requirements, including budget of £1.5M

Problems: Procurement (ii)

- **Successful consortium**
 - Apricot, Systems Options (SO), Datatrak; bid at £937k was £700k cheaper than the nearest bid
 - SO's quote for the CAD development was only £35k
 - Their previous development experience for the emergency services was only for administrative systems
 - Ambiguity over lead contractor
- **2 key members of evaluation team:**
 - Systems manager: Career ambulance man, not an IT professional, already told that he was to make way for a properly qualified systems manager
 - Analyst: Contractor with 5 years experience working with LAS

Problems: Project management

- **Lead contractor responsible**
 - Meant to be SO, but they quickly became snowed under, so LAS became more responsible by default
 - No relevant experience at LAS or SO
- Concerns raised at project meeting not followed-up
- **SO regularly late in delivering software**
 - Often also of suspect quality, with software changes put through ‘on the fly’
- **Formal, independent QA did not exist at any stage throughout the CAD system development**
- **Meanwhile, various technical components of the system are failing regularly, and deadlines missed**

Problems: Human resources & training (i)

- Generally positive attitude to the introduction of new technology
- Ambiguity over consultation of ambulance crews for development of original requirements
- Circumstantial evidence of resistance by crews to Datatrak equipment, and deliberate misleading of the system
- Large gap between when crews and CAC staff were trained and implementation of the system
- Inability of the CAC and ambulance staff to appreciate each others' role
 - Exacerbated by separate training sessions

Problems: Human resources & training (ii)

- Poor industrial relations
- Management ‘fear of failure’
- CAD system seen as solution to management’s desire to reduce ‘outdated’ working practices
- System allocated nearest resource, regardless of originating station
- System removed flexibility in resource allocation
- Lack of voice contact exacerbated “them and us”
- Technical problems reduced confidence in the system for ambulance crews and CAC staff

System problems

- **Need for near perfect information**
 - Without accurate knowledge of vehicle locations and status, the system could not allocate optimum resources
- **Poor interface between crews, MDTs & the system**
 - There were numerous possible reasons for incorrect information being passed back to the system
- **Unreliability, slowness and operator interface**
 - Numerous technical problems with the system, including:
 - Failure to identify all duplicated calls
 - Lack of prioritisation of exception messages
 - Exception messages and awaiting attention queues scroll off top of screen

Configuration changes

- Implementation of the system on 26 October involved a number of significant changes to CAC operation, in particular:
 - Re-configuring the control room
 - Installing more CAD terminals and RIFS screens
 - No paper backup system
 - Physically separating resource allocators from radio operators and exception rectifiers
 - Going 'pan London' rather than operating in 3 divisions
 - Using only the system proposed resource allocations
 - Allowing some call takers to allocate resources
 - Separate allocators for different call sources

So, what happened?

- Changes to CAC operation made it extremely difficult for staff to intervene and correct the system.
- As a consequence, the system rapidly knew the correct location and status of fewer and fewer vehicles, leading to:
 - Poor, duplicated and delayed allocations
 - A build up of exception messages and the awaiting attention list
 - A slow up of the system as the messages and lists built up
 - An increased number of call backs and hence delays in telephone answering

Why did it fail?

- Technically, the system did not fail on October 26th
 - Response times did become unacceptable, but overall the system did what it had been designed to do!
 - Failed 3 weeks later due to a program error
- It depends who you ask!
 - Management
 - Union
 - System manager
 - Government

Lessons learned

- Inquiry report makes detailed recommendations for future development of the LAS CAD system, including:
 - Focus on repairing reputation of CAD within the service
 - Increasing sense of ‘ownership’ for all stakeholders
 - They still believe that a technological solution is required
 - Development process must allow fully for consultation, quality assurance, testing, training
 - Management and staff must have total, demonstrable, confidence in the reliability of the system
 - Any new system should be introduced in a stepwise approach