4. The Design Process

Three Mile Island

The accident at TMI-2 began at 4 a.m. on March 28, 1979. A minor malfunction, in the non-nuclear part of the system would evolve a series of automated responses in the reactor's coolant system, and during all of this, the relief valve on the pressuriser would become stuck open.

Symptoms were misread by the operators over a 2 1/4-hour period before the relief valve was closed.

Following misunderstanding of the state of the reactor, an automatic emergency cooling system was turned off.

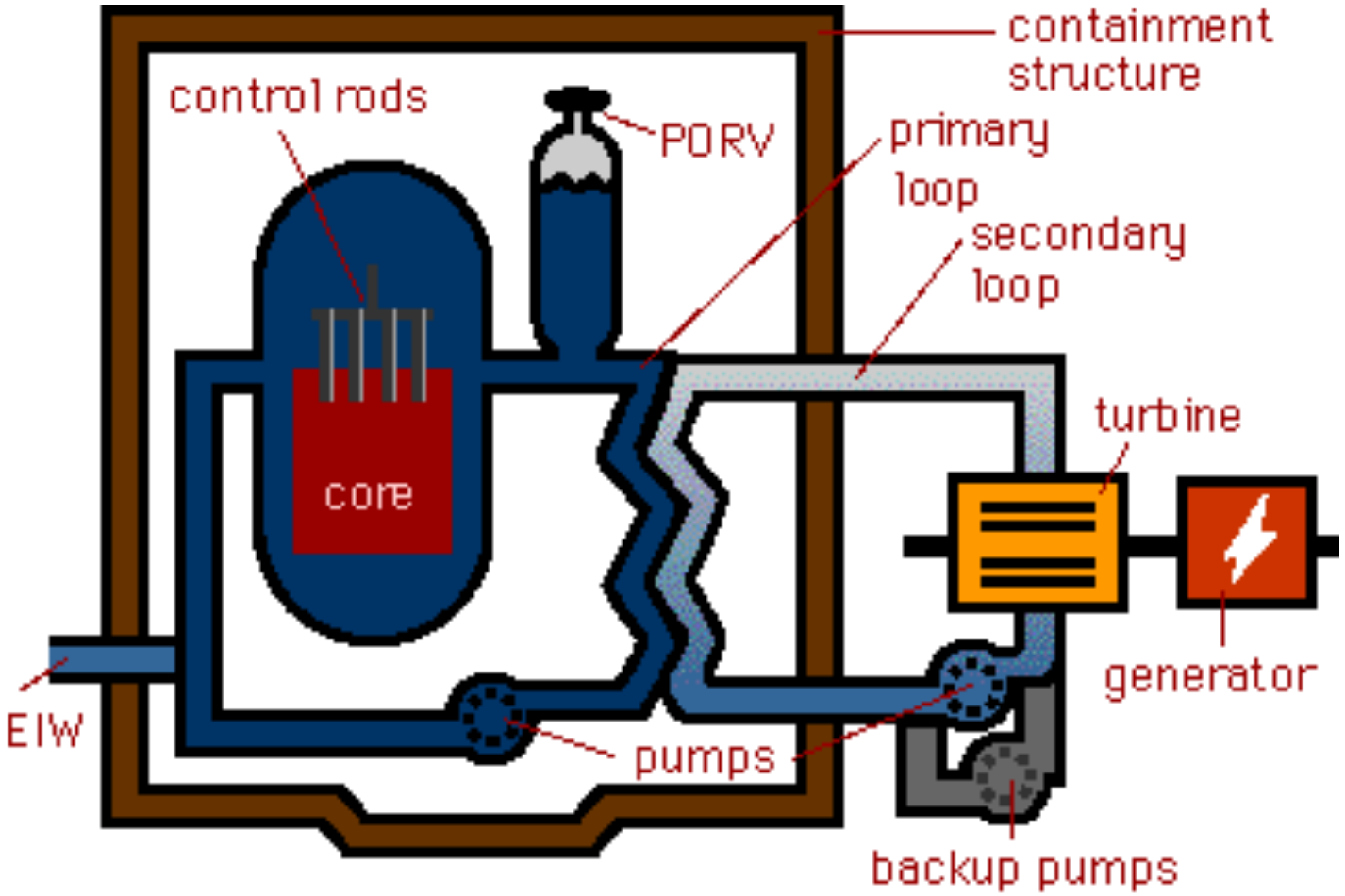
The reactor core became partially uncovered and severely damaged. It would be another 12 hours before the plant crew and the engineers would concur in effective corrective action.

http://www.threemileisland.org for details.

TMI Sequence

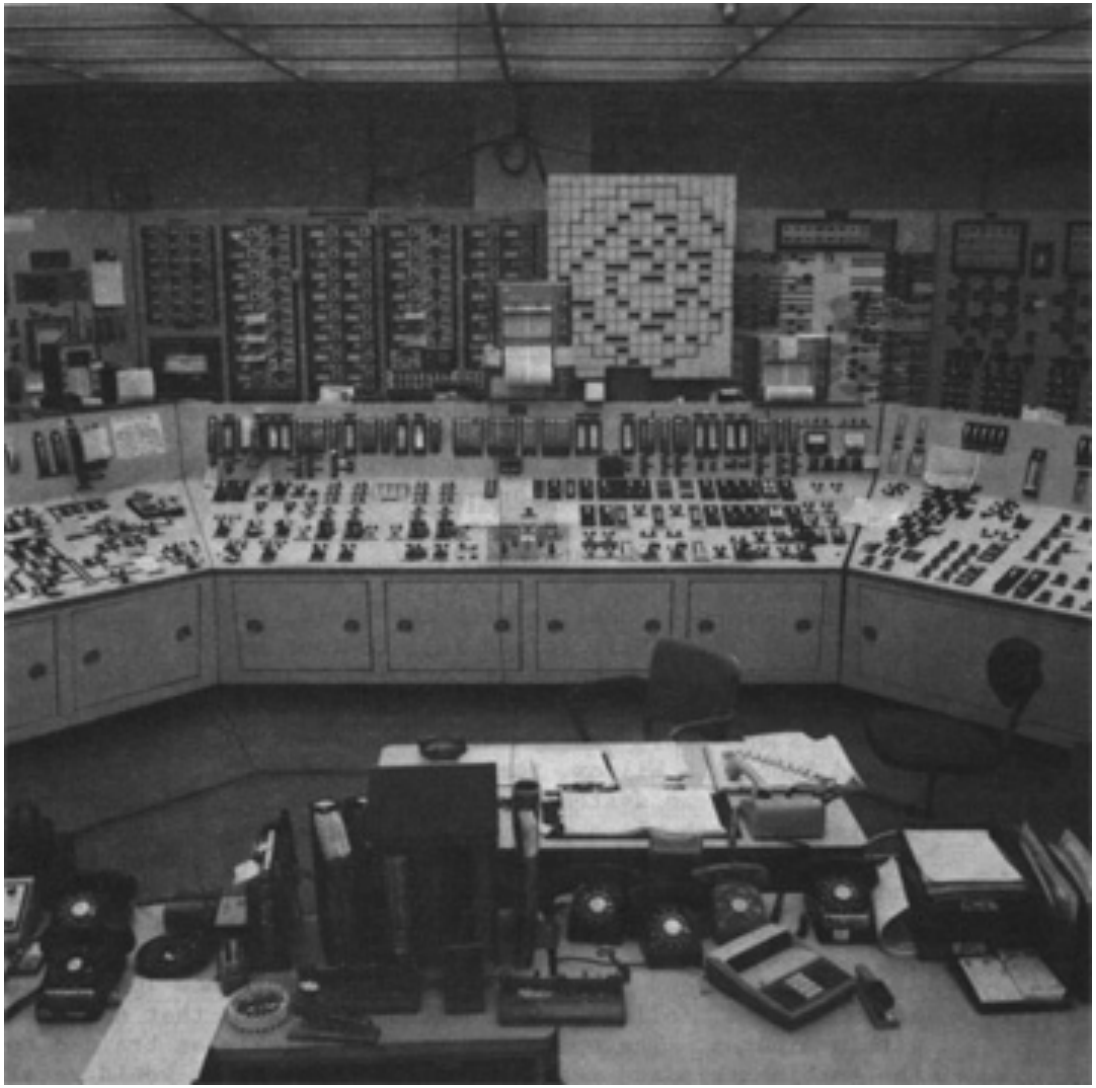
- Feed pump broke down. Pressuriser relief valve opened. When pressure reduced, valve failed to close (accident started).
- Emergency feed pump had been tested 42 hours before; involved closing valve, performing test, opening valve. Valve had not been re-opened.
- Pressure dropped, water in reactor turned to steam. Pressuriser filled with water, steam surrounding fuel rods.
- Emergency cooling system kicked in, but operators saw pressuriser full of water and turned it off.
- Steam bubbles in coolant pumps causing vibrations (and potential failure), so operators turned them off. Fuel pellets began to melt.
- 6:18 am, one of the operators finally realised PORV was stuck open. Immediate emergency halted.
- 2pm small hydrogen explosion. Actions over several days to remove hydrogen in reactor vessel.
- Water from relief valve overflowed tanks – LOCA.

TMI Sequence



From pbs.org

Three Mile Island Control Room



Three Mile Island Control Panel

Human Error?

- The control room alarm provided audible and visual indication for more than 1500 alarm conditions.
- An important light indicator was covered by a caution tag attached to another valve controller.
- An automatic relief valve on top of the pressuriser failed to close, even though a poorly designed indicator led operators to believe that it had. The light, which one operator described as perhaps the brightest light on the entire panel, indicated only what the valve had been **commanded** to do, not what it was **actually doing**. It remained dark, because the valve was commanded to close.
- The accident investigation identified the root cause of the disaster as "human error"

Design error

- The control panels did not indicate to the operators the true state of affairs in the reactor plant.
- No instrument showed coolant level in the reactor.
- They did not provide the necessary information in a form that the operators could understand and use to rectify the situation.
- Operators thought relief valve closed and water level too high – overrode emergency cooling system.
- The operators' training covered regular tasks, but not unusual situations.
- Maintenance tags covered one of the closed emergency feedwater valve indicator lights during the first 8 minutes of the accident.

Design process

- Now we focus on the process of design.
- What is design? Achieving goals within constraints. Trade-offs.
- Software engineering is the emerging discipline for understanding the design process, or life cycle.
- Designing for usability occurs at all stages of the life cycle, not as a single isolated activity

Interactions and Interventions

Design *interactions* not just interfaces

- not just the immediate interaction
- technology can change interaction style

Designing *interventions* not just artefacts

not just the system, but also …

- documentation, manuals, tutorials
- what we say and do as well as what we make

Interaction design

- Four basic activities:
- Identifying needs and establishing requirements for the user experience
- Developing alternative designs that meet those requirements.
- Building interactive versions of the designs so that they can be communicated and assessed.
- Evaluating what is being built throughout the process and the user experience it offers.

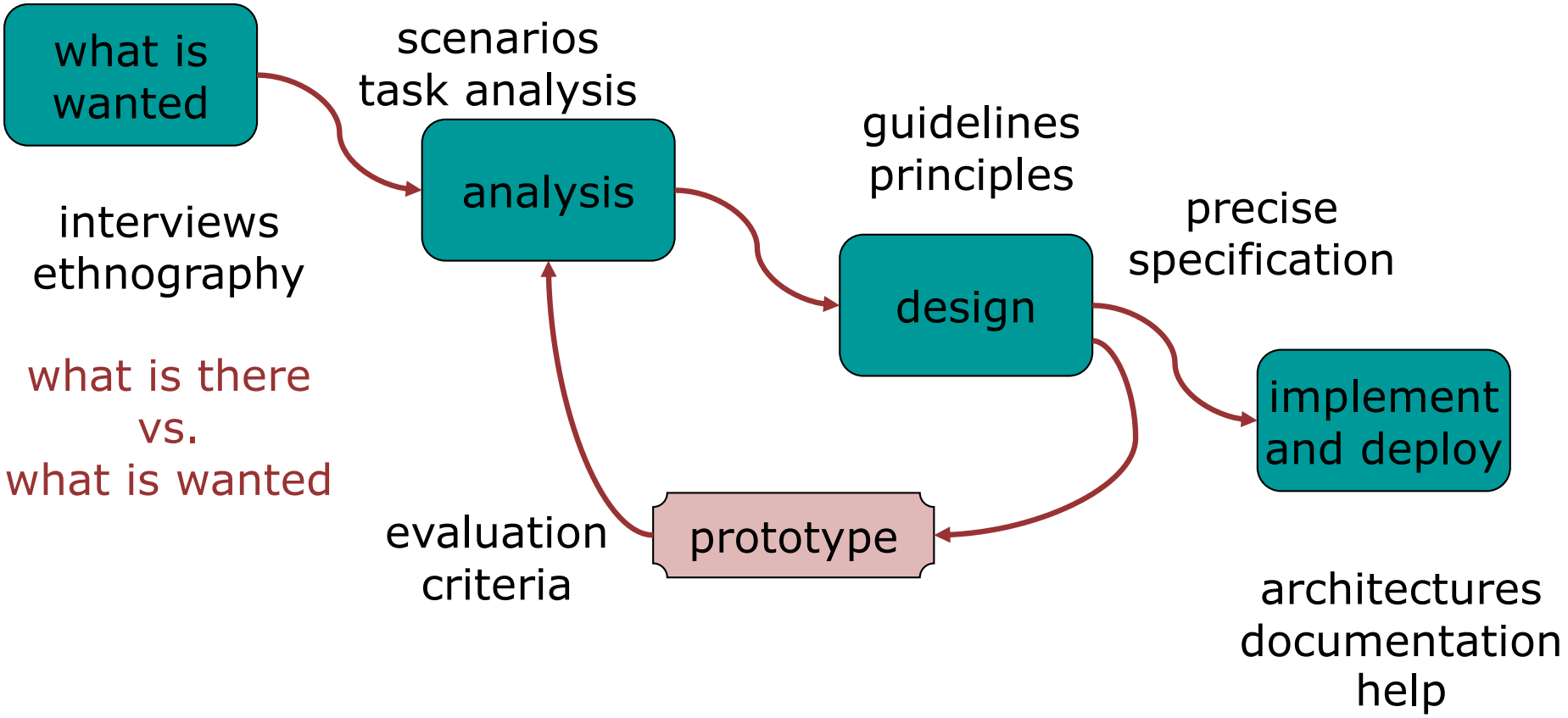
Interaction Design

- Requirements gathering:
  - Determine characteristics of the user population, types of user, frequency of use, discretion about use, experience of the task, level of training, experience of computer systems.
  - Determine characteristics of the task, complexity of task, breakdown of task, context/environment of task.
  - Determine constraints and objectives: choice of hardware and software, desired throughput, acceptable error rate.

Interaction Design

- Design of the interface
  - Allocate elements of task to user or system; determine communication requirements between users and system.
  - Design elements of the interface to support the communication between users and system in the light of characteristics of the users, characteristics of the task, and constraints on design.
- Interface evaluation:
  - Develop prototypes of interface designs
  - Test prototypes with users to determine if objectives are met.

Process of design



Design rules - standards

- Set by national or international bodies to ensure compliance by a large community of designers standards require sound underlying theory and slowly changing technology
- Hardware standards more common than software high authority and low level of detail
- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks

Design rules - guidelines

- Guidelines
  - More suggestive and general
  - Many textbooks and reports full of guidelines
- Abstract guidelines (principles) applicable during early life cycle activities
- Detailed guidelines (style guides) applicable during later life cycle activities
- Understanding justification for guidelines aids in resolving conflicts

ISO Usability Standard 9241

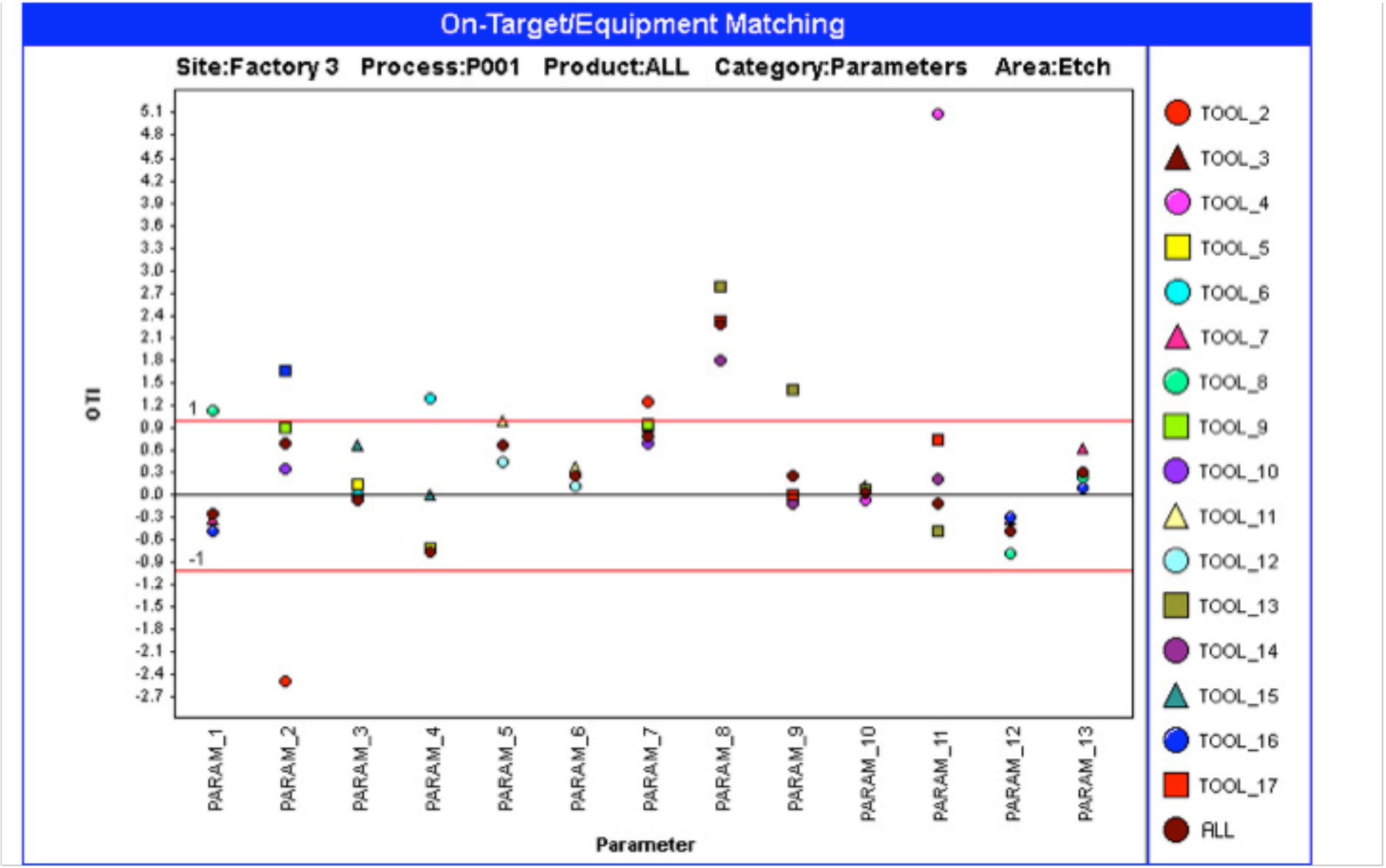
Adopts traditional usability categories:

- effectiveness
  - can you achieve what you want to?
- efficiency
  - can you do it without wasting effort?
- satisfaction
  - do you enjoy the process?

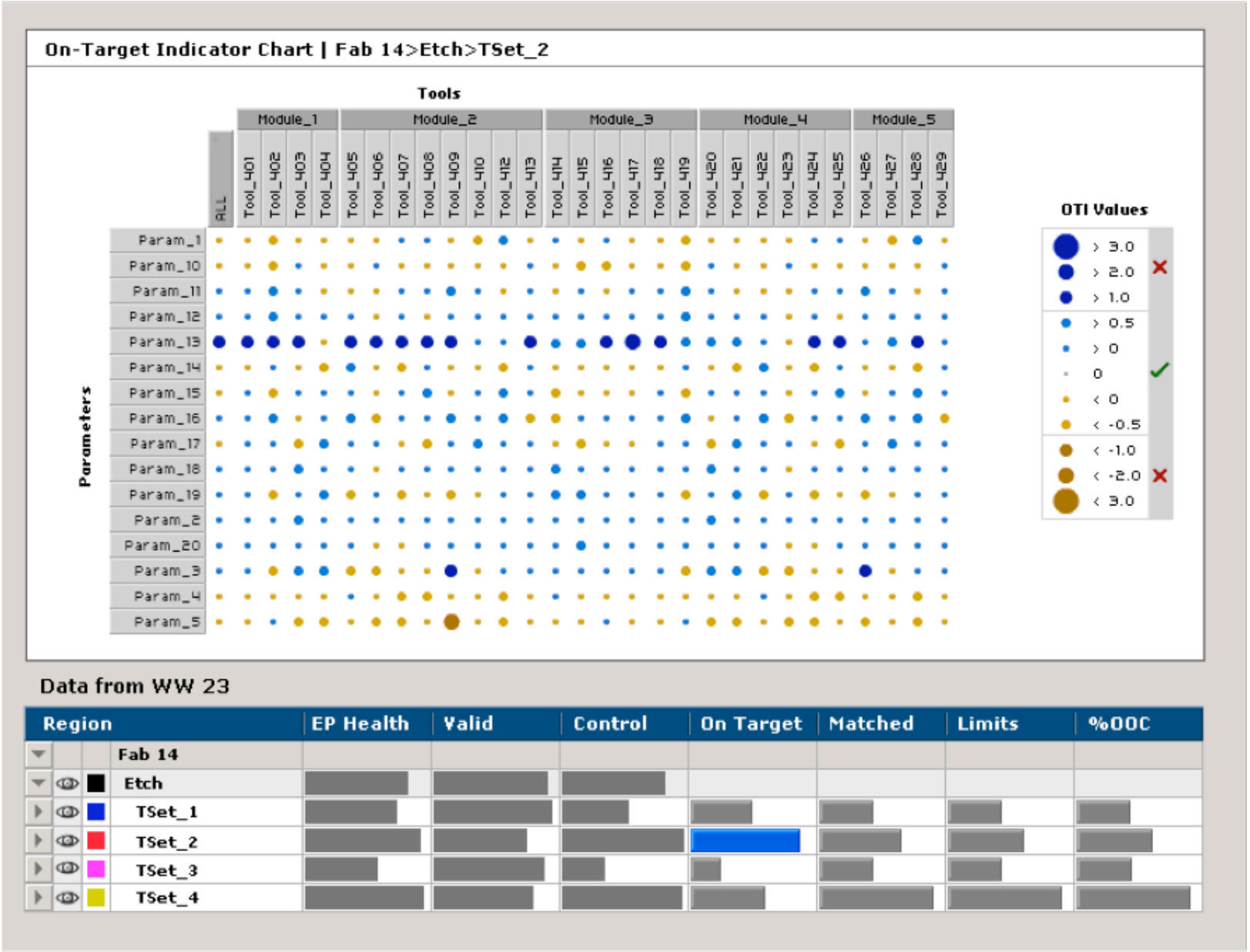
Some metrics from ISO 9241

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn function	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

Original design



New design

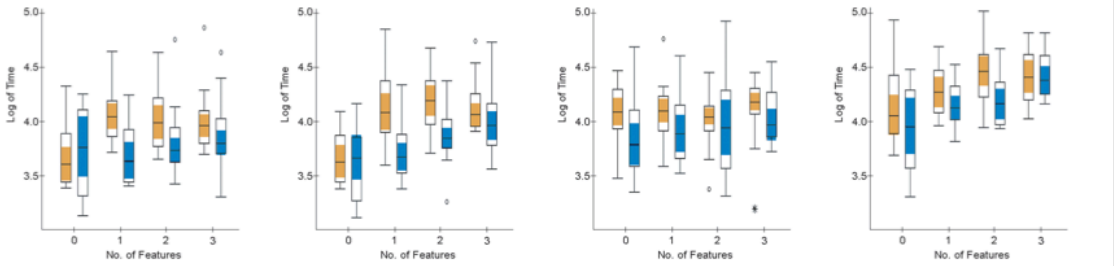

On-Target Indicator Chart | Fab 14>Etch>TSet_2
Tools
Module_1
Module_2
Module_3
Module_4
Module_5
ALL
Tool_401
Tool_402
Tool_403
Tool_404
Tool_405
Tool_406
Tool_407
Tool_408
Tool_409
Tool_410
Tool_412
Tool_413
Tool_414
Tool_415
Tool_416
Tool_417
Tool_418
Tool_419
Tool_420
Tool_421
Tool_422
Tool_423
Tool_424
Tool_425
Tool_426
Tool_427
Tool_428
Tool_429
Parameters
Param_1
Param_10
Param_11
Param_12
Param_13
Param_14
Param_15
Param_16
Param_17
Param_18
Param_19
Param_2
Param_20
Param_3
Param_4
Param_5
OTI Values
> 3.0
> 2.0
> 1.0
> 0.5
> 0
0
< 0
< -0.5
< -1.0
< -2.0
< 3.0
Data from WW 23
Region
EP Health
Valid
Control
On Target
Matched
Limits
%OOC
Fab 14
Etch
TSet_1
TSet_2
TSet_3
TSet_4

Results

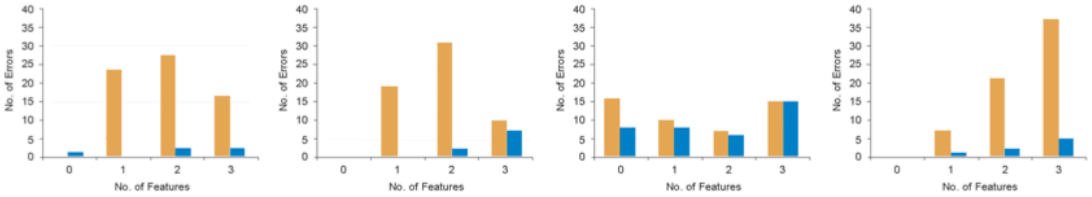
= Chart A (Original)

= Chart B (Redesign)

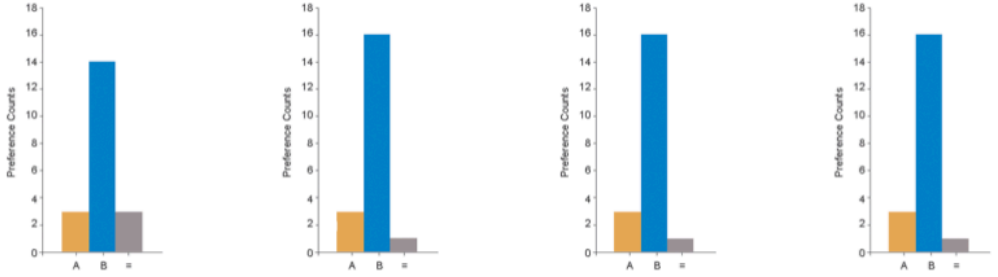
Task 1 | Task 2 | Task 3 | Task 4



Efficiency plot charts showing the log of time against the number of features



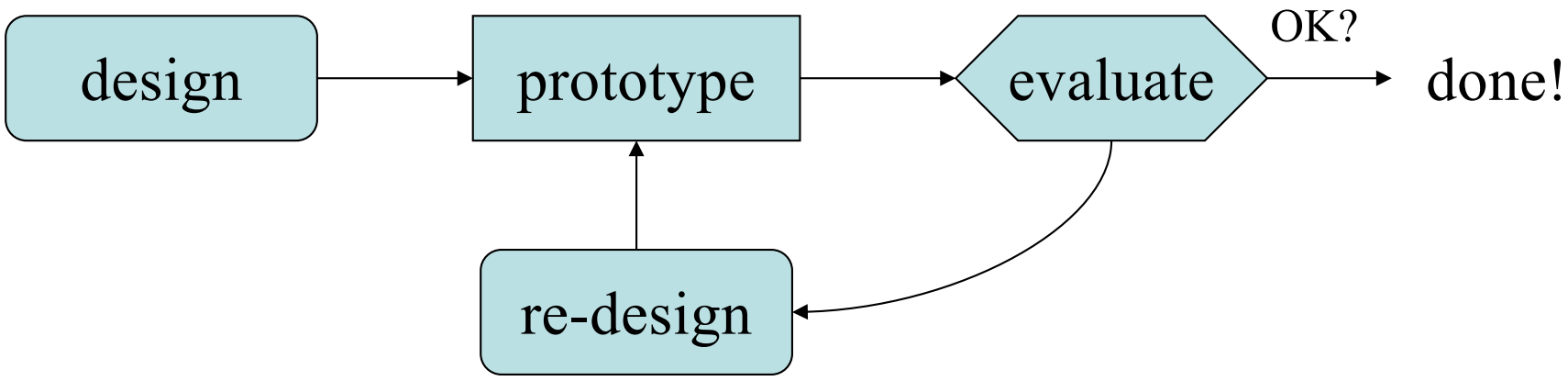
Accuracy bar charts showing the total number of errors against the number of features



Satisfaction barcharts showing users preferred chart for each task

Prototyping

- Never get it right first time. When you stop depends on goal of prototype



Iterative design and prototyping

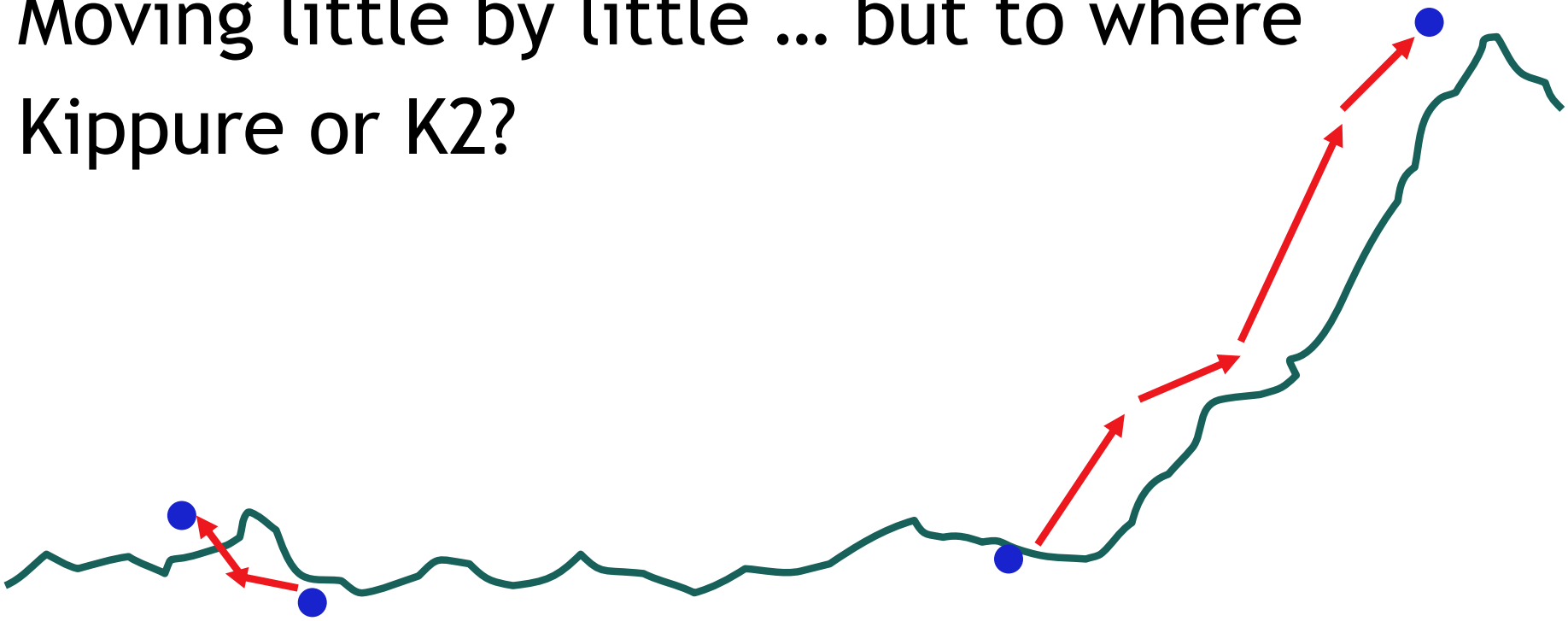
- Iterative design overcomes inherent problems of incomplete requirements
- Prototypes
  - simulate or animate some features of intended system
- Different types of prototypes
  - throw-away
  - incremental
- Management issues
  - time, planning
  - non-functional features (safety, reliability, response time)
  - contracts

Techniques for prototyping

- Storyboards
  - need not be computer-based
  - can be animated
- Limited functionality simulations
  - some part of system functionality provided by designers
  - tools are common for these, also VB, Flash, HTML…
  - Wizard of Oz technique
- Warning about iterative design
  - design inertia – early bad decisions stay bad
  - diagnosing real usability problems in prototypes….
    …. and not just the symptoms

Pitfalls of prototyping

- Moving little by little … but to where
- Kippure or K2?



1. Need a good start point
2. Need to understand what is wrong

Scenario based approaches

- Scenarios are step-by-step descriptions of a user's actions that can be used as a tool in requirements gathering, interface design and evaluation.
- Scenarios can be textual narrative describing the users actions or they can be in the form of storyboards (a series of pictures depicting those actions).
- Scenarios can capture both the actions the user carries out in the current system, and how they would see themselves performing with a new system.
- Detailed scenarios can also form the basis for evaluations of designs, and the constructed system.

Participatory Design

- User is an active member of the design team.
- Characteristics
  - Context and work oriented rather than system oriented
  - Collaborative
  - Iterative
- Methods
  - Brain-storming
  - Storyboarding
  - Workshops
  - Pencil and paper exercises

Ethnographic approaches

- The term ethnography refers to a range of techniques used in sociology and anthropology. The focus is on being part of the situation under study, observing work practices in the field.
- It is obviously a qualitative approach.
- It involves the ethnographer participating, overtly or covertly in people's daily lives for an extended period of time, watching what happens, listening to what is said, asking questions and collecting whatever data is available to throw light on the issues of interest.
- Thus contextual factors which are omitted from laboratory evaluations are captured.
- Hawthorne effect – people behave differently if watched.

What can ethnography provide us?

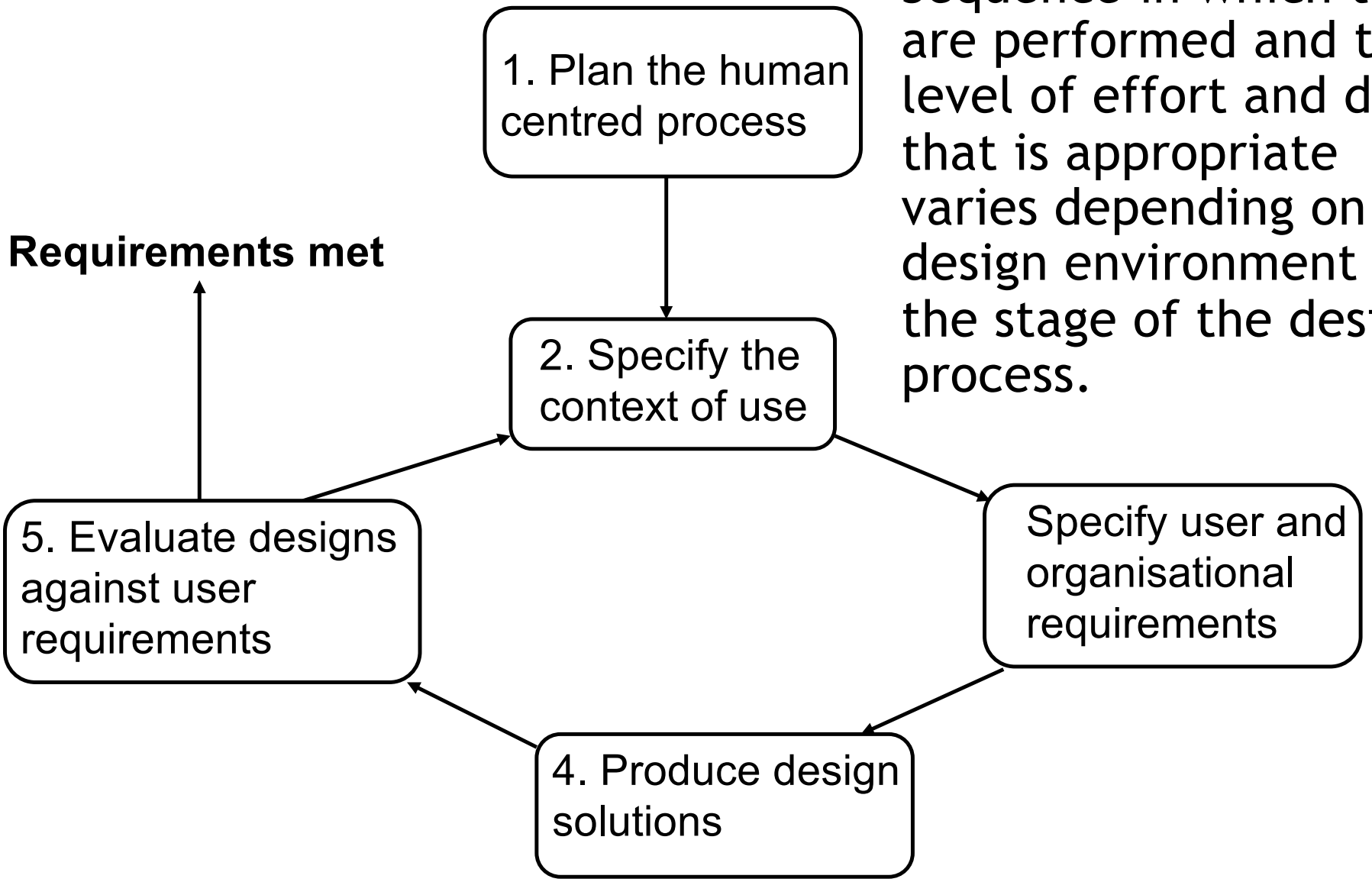
- Additional domain knowledge.
- An overall view of complex settings which would otherwise be difficult to obtain.
- Perspectives from, and practices of, a variety of stakeholders.
- Some assessment of the scope and limitations of systems and products that might be envisaged.
- A balanced view of the relationship between standardised processes, human skills, and how to deal with contingencies.
- A fuller view of the real-world nature of the problems that need to be solved.
- Detailed knowledge of the routine ways in which technologies actually get used, and what for.
- A critique of 'snake oil' salesmen — i.e., those who offer simplistic technical or organisational solutions.

[Randall & Rouncefield 2013]

ISO 9241 – Human Centred Design

- Human Centred Design Processes for Interactive Systems.
- Provides guidance on achieving quality in use by incorporating user centred design activities throughout the life cycle of interactive systems.
- Describes user centred design as a multi-disciplinary activity. There are four activities which start at the earliest stages of a project.
  - understand and specify the context of use
  - specify the user and organisational requirements
  - produce design solutions
  - evaluate designs against requirements.

ISO 9241



Sequence in which these are performed and the level of effort and detail that is appropriate varies depending on the design environment and the stage of the design process.

Exercise

- What sort of design process would you use for:
  - A system to capture emergency medical records
  - An augmented reality location based game
  - An e-commerce site for selling consumer electronics

Summary

- The software engineering life cycle
  - distinct activities and the consequences for interactive system design
- Using design rules
  - standards and guidelines to direct design activity
- Usability engineering
  - making usability measurements explicit as requirements
- Iterative design and prototyping
  - limited functionality simulations and animations