GÉNIE DES SYSTÈMES INTERACTIFS

David Navarre / Philippe Palanque

navarre@irit.fr

Interactive Critical Systems Team - IRIT

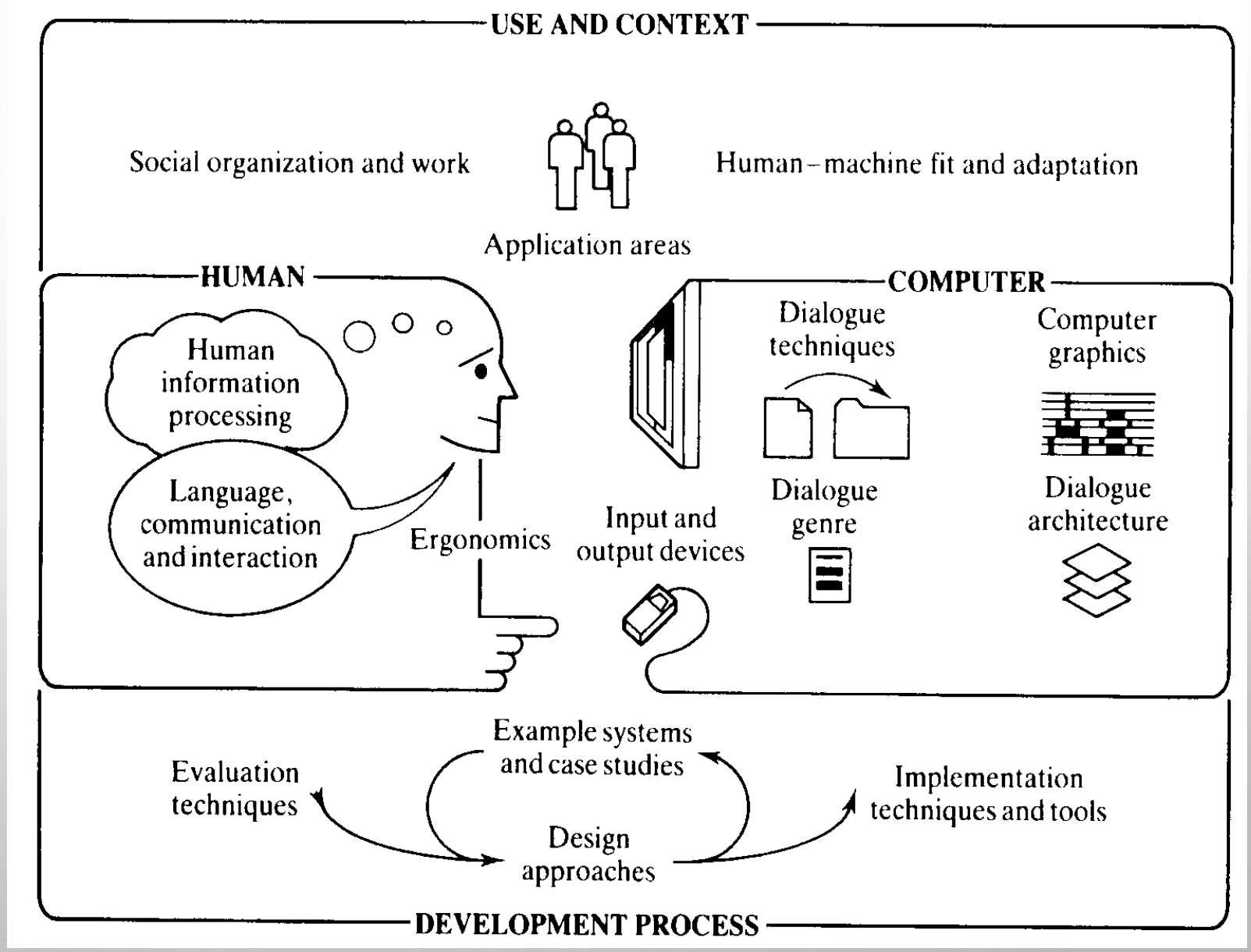
Université Toulouse 1 - Capitole

PLAN DU COURS

- Introduction et définitions
- Processus de développement des SI
- Les techniques de spécification formelle pour les SI
- Les standards existants (DO 178C, ESSAR, ISO, …) – cours Safety !!
- Les techniques d'ingénierie des SI
- Les méthodes de conception de SI
- Perspectives de recherche (futur du GSI)

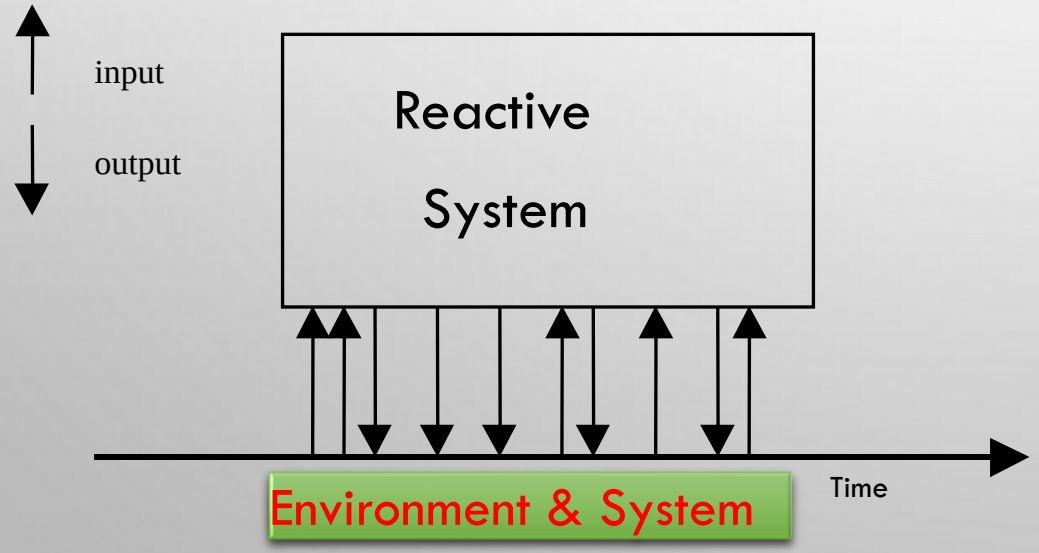
INTRODUCTION À L'IHM

Gardez votre esprit critique !



DEFINITION DES SI

- Un principe: Des systèmes réactifs
- Un challenge: variabilité (contextes d'utilisation, des utilisateurs, …)
- Une philosophie: Outil comme approche (utilisateur dans la boucle IN THE LOOP)



APPLICATIONS OF TEMPORAL LOGIC TO THE SPECIFICATION AND VERIFICATION OF REACTIVE SYSTEMS: A SURVEY OF CURRENT TRENDS

A. Pnueli
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Acknowledgement:

Being a survey, this paper summarizes the work of many authors, in particular joint work with Z. Manna, H. Barringer, R. Kuiper, L. Zuck, and O. Lichtenstein, as well as independent work by L. Lamport, Z. Manna, P. Wolper, S. Owicki, B. Hailpern, V. Nguyen, D. Gries, F. Schneider, E. Clarke, and P. Sistla. Explicit references to these works are given whenever we discuss points specific to particular articles. On the other hand, when discussing general points which are common to several authors, we usually avoided explicit references for the sake of continuity of exposition, and hope that this general acknowledgement suffices in order to acknowledge their valuable general contribution to the subject.

Range of Applicability of TL — Reactive Systems

There are two basically different views of computerized systems (programs). The first view regards programs as *functions* from an initial state to a final state; in the non-deterministic case as *relations* between initial and final states. This view is particularly appropriate for programs that accept all of their inputs at the beginning of their operation and yield their outputs at termination. We call such programs *transformational*, referring to their interpretation as *state transformers*. Typical examples of transformational programs are batch, off-line data-processing, and other computational programs. For transformational programs, adequate and fully abstract description and specification tools are provided by denotational semantics based on state-functions and Hoare logic, or Dijkstra's predicate transformers.

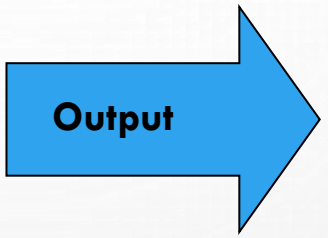
On the other hand, there are systems that cannot be covered by the transformational view. Some systems, such as operating systems, process control programs, seat reservation systems, etc., ideally never terminate. Moreover, the purpose for which they are run is not to obtain a final result, but rather to maintain some interaction with their environment. We refer to

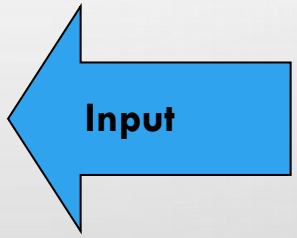
There are two basically different views of computerized systems (programs). The first view regards programs as *functions* from an initial state to a final state; in the non-deterministic case as *relations* between initial and final states. This view is particularly appropriate for programs that accept all of their inputs at the beginning of their operation and yield their outputs at termination. We call such programs *transformational*, referring to their interpretation as *state transformers*. Typical examples of transformational programs are batch, off-line data-processing, and other computational programs. For transformational programs, adequate and fully abstract description and specification tools are provided by denotational semantics based on state-functions and Hoare logic, or Dijkstra's predicate transformers.

On the other hand, there are systems that cannot be covered by the transformational view. Some systems, such as operating systems, process control programs, seat reservation systems, etc., ideally never terminate. Moreover, the purpose for which they are run is not to obtain a final result, but rather to maintain some interaction with their environment. We refer to such systems as *reactive systems.* Clearly, reactive systems cannot be adequately described by referring only to their initial and final states. An adequate description must refer to their on-

Time

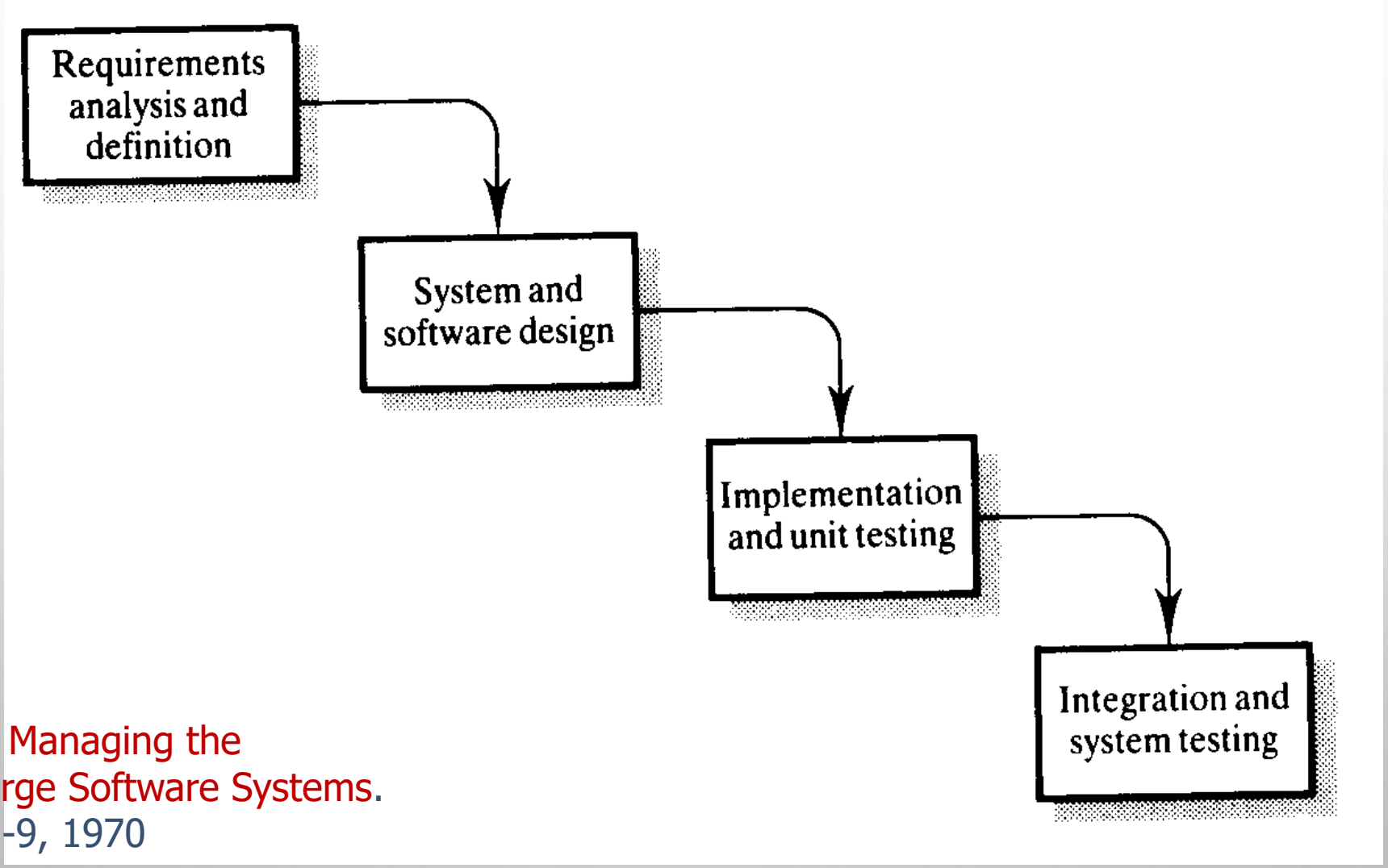
DÉFINITION DES SI





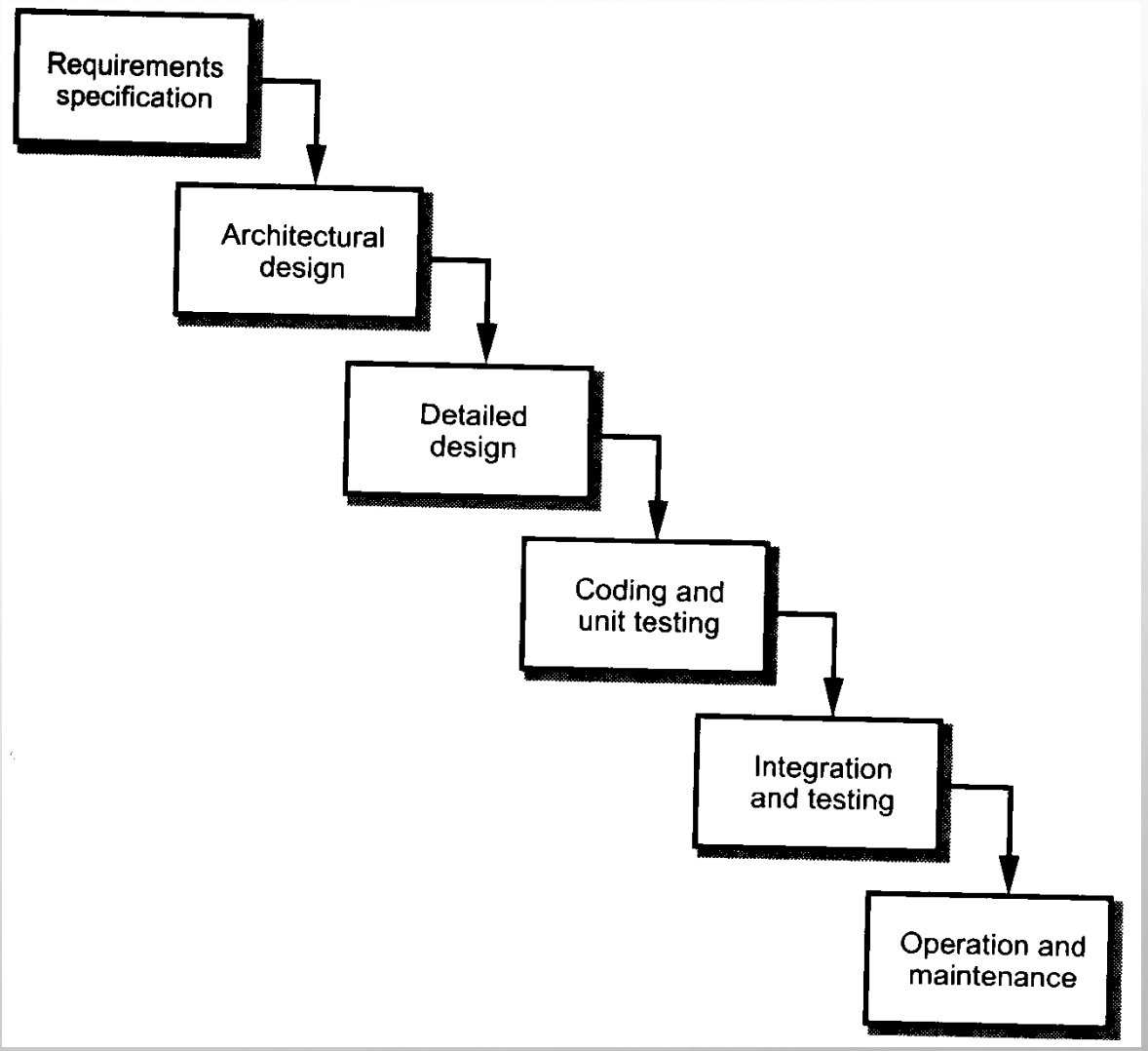
Pnueli A. (1986) Applications of temporal logic to the specification and verification of reactive systems: A survey of current trends. In: de Bakker J.W., de Roever W.P., Rozenberg G. (eds) Current Trends in Concurrency. Lecture Notes in Computer Science, vol 224. Springer, Berlin, Heidelberg

CYCLE CASCADE COURT



Winston W. Royce. Managing the Development of Large Software Systems. IEEE Wescon, pp 1-9, 1970

CYCLE DE DÉVELOPPEMENT EN CASCADE



Winston W. Royce. Managing the Development of Large Software Systems. IEEE Wescon, pp 1-9, 1970

VRAIMENT?

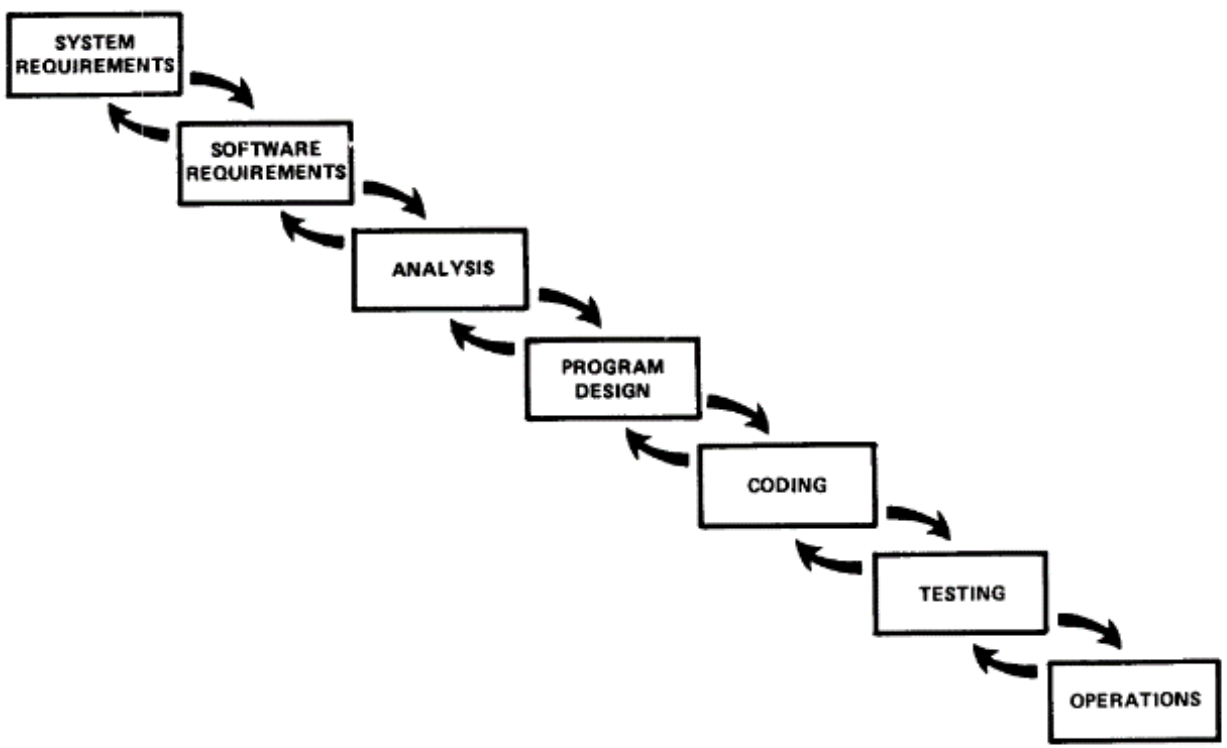


Figure 3. Hopefully, the iterative interaction between the various phases is confined to successive steps.

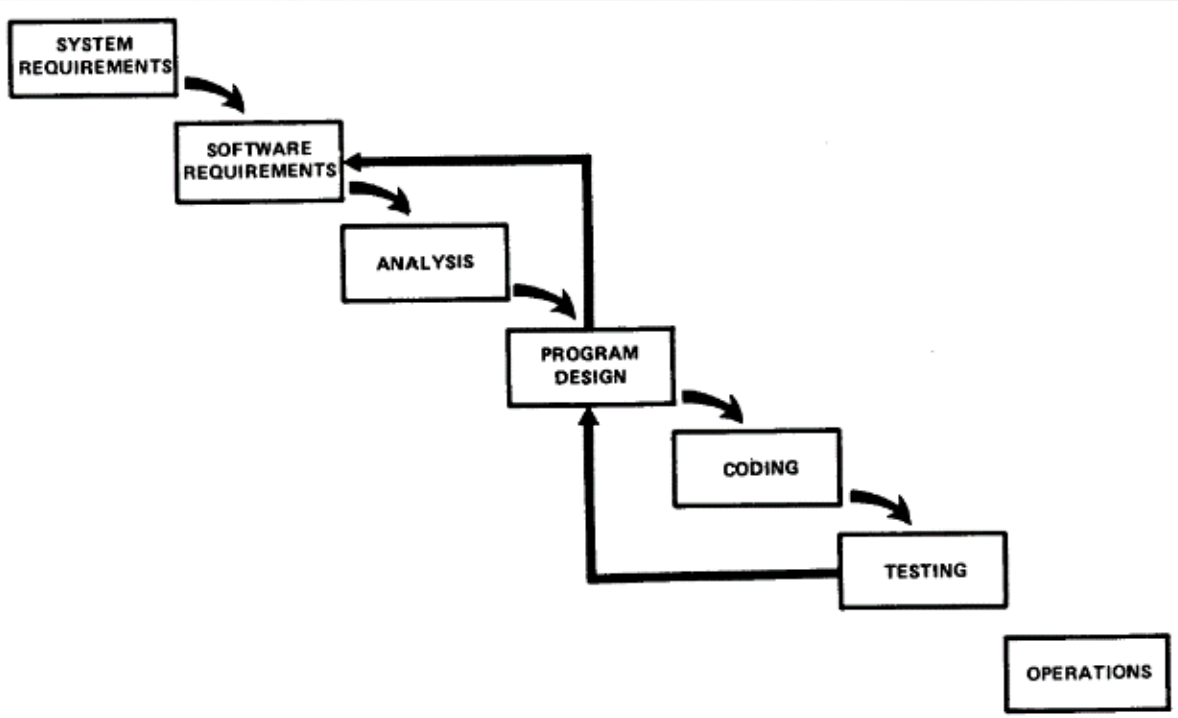
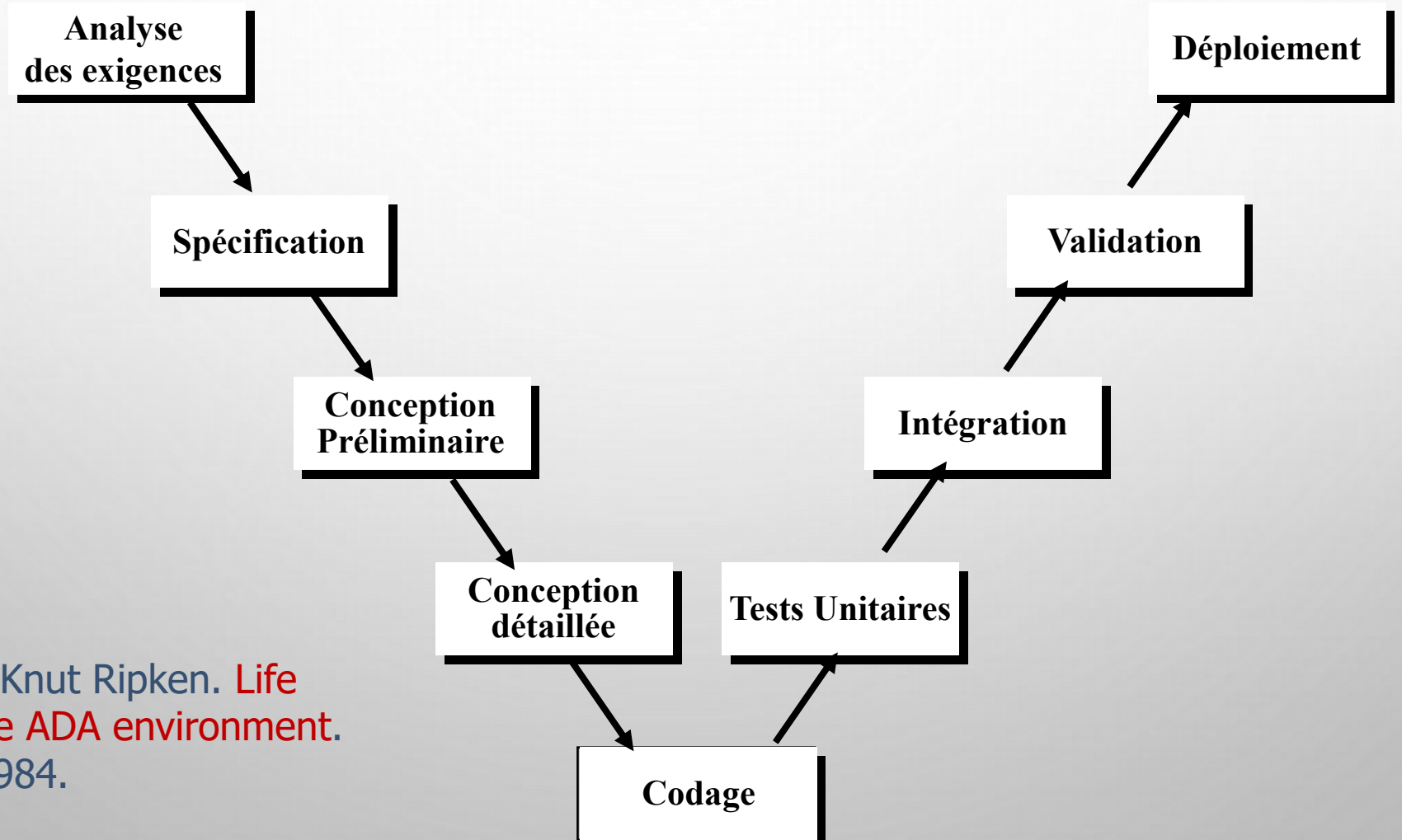


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

CYCLE DE DÉVELOPPEMENT EN V



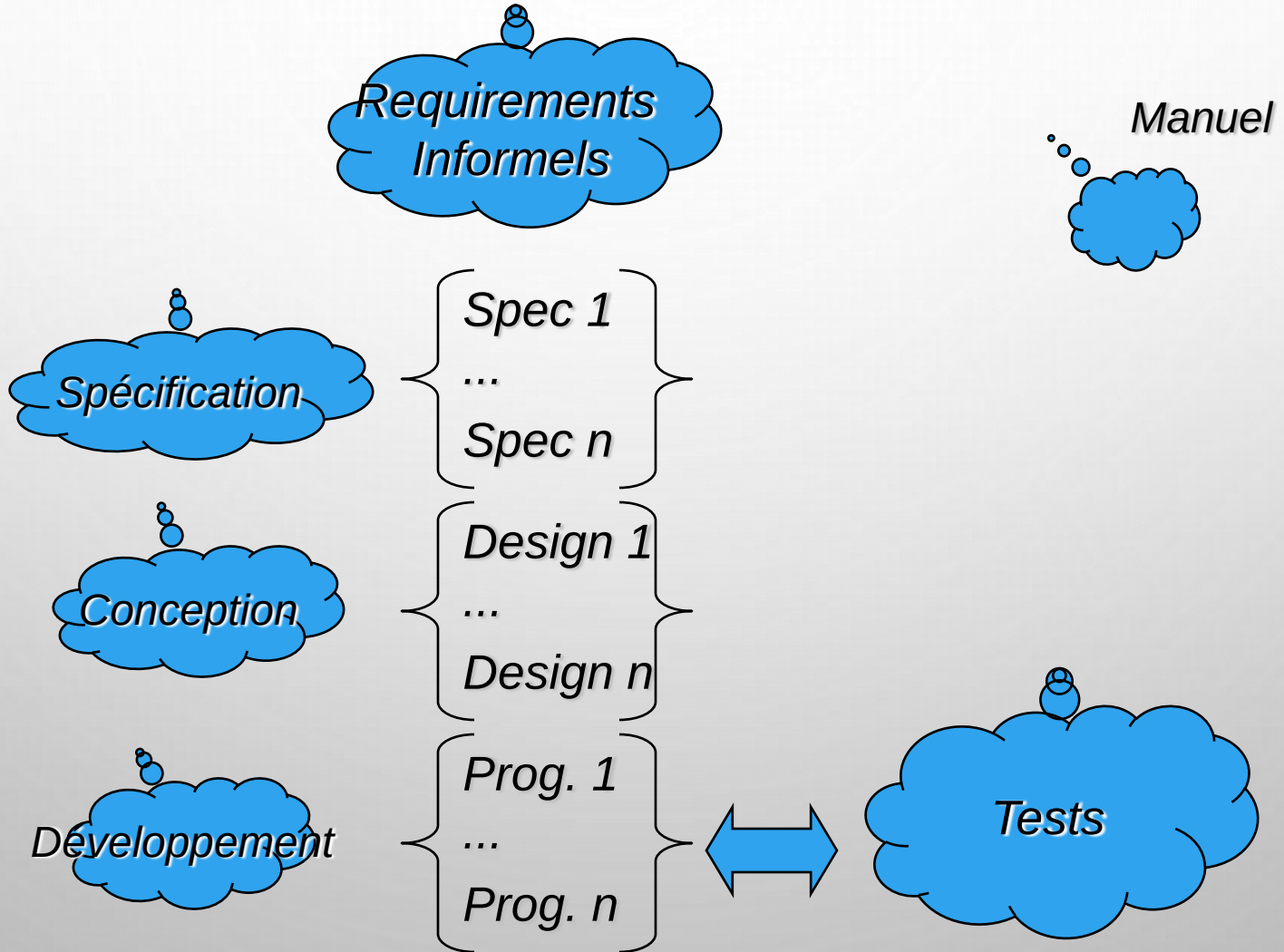
John McDermid et Knut Ripken. Life cycle support in the ADA environment. University Press, 1984.

SPÉCIFICATION DES SYSTÈMES INTERACTIFS

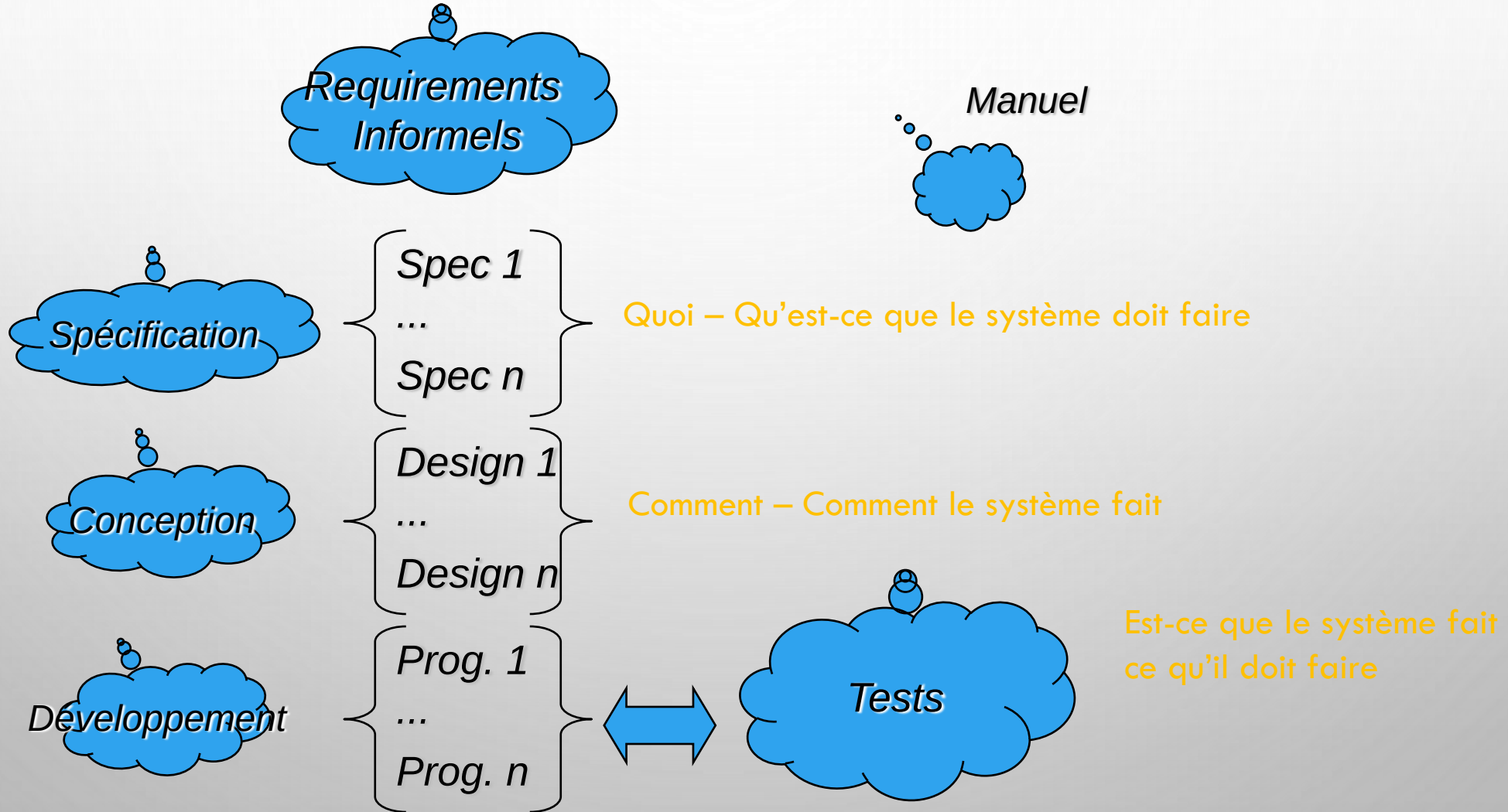
SPÉCIFICATION

- Norbert E. Fuchs, Specifications are (preferably) executable, Software Engineering Journal, Volume 7, Issue 5, September 1992, p. 323 – 334
- Ian Hayes and C. B. Jones. 1989. Specifications are not (necessarily) executable. *Softw. Eng. J.* 4, 6 (November 1989), 330-338.

PRINCIPE DE CONTINUITÉ ENTRE REQUIREMENTS ET PROGRAMME



PRINCIPE DE CONTINUITÉ ENTRE REQUIREMENTS ET PROGRAMME



MODÈLES ET MODÉLISATION DE SYSTÈMES INTERACTIFS

QU'EST-CE QU'UN "MODÈLE"

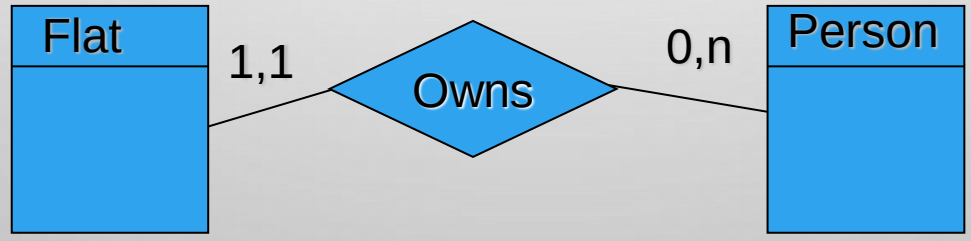
- Concepts et liens entre concepts
- Exemple : le Modèle Entité/Association
  - Concepts : Classe d'Entité, Classe d'Association, Attribut, Domaine, Identifiant....
  - Liens entre concepts : "Une CA est un sous-ensemble du produit cartésien de 2 CE"

QUE DOIT-ON ATTENDRE D'UN "MODÈLE"

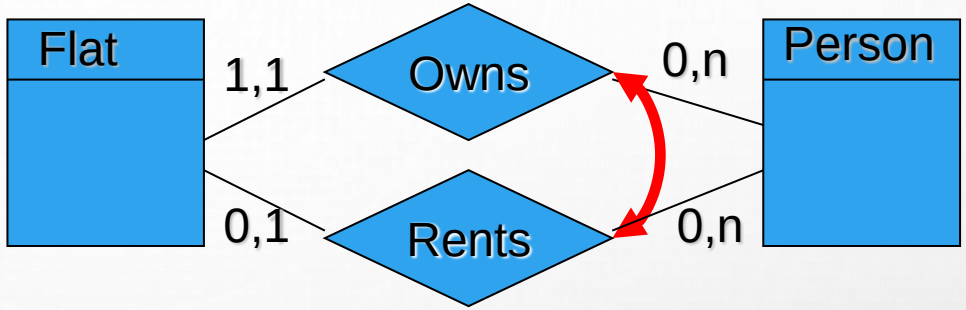
- Complétude (On peut exprimer tout ce que l'on souhaite)
- Consistance (On ne peut pas exprimer d'énoncés contradictoires)
- Généralité (Indépendant d'un domaine d'application particulier... Toujours ?)

QU'EST-CE QU'UN "FORMALISME"?

- Conventions de représentation des concepts d'un Modèle
  - Lexique (graphique ou textuel)
  - Syntaxe concrète (séparateurs, terminaux, ...)
  - Sémantique (un sous-ensemble de celle du Modèle)



QUE DOIT-ON ATTENDRE D'UN "FORMALISME" ?



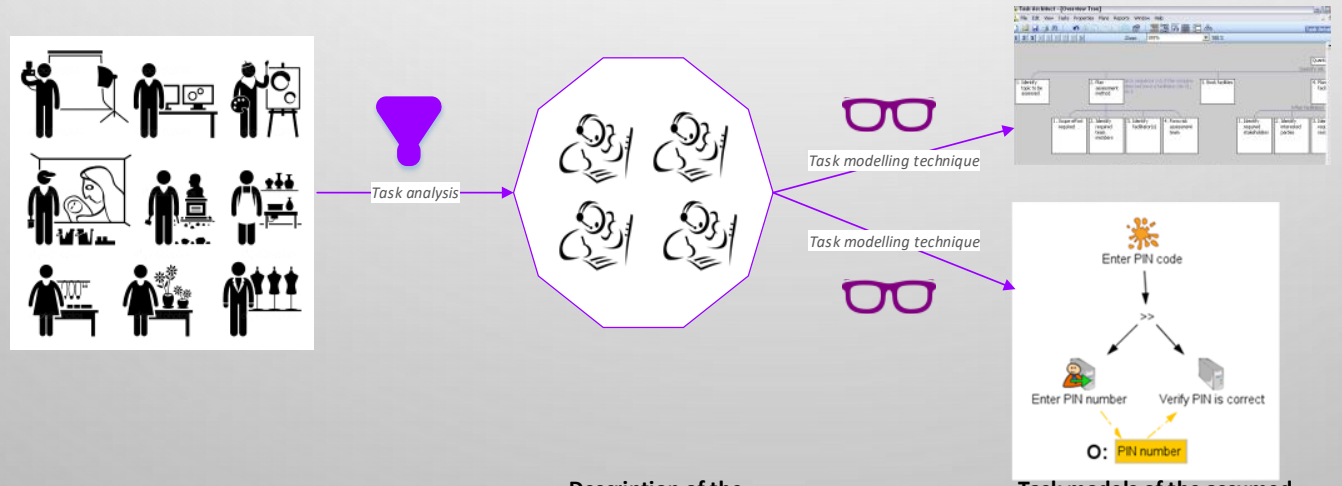
- Expressivité
- Concision
- Complétude par rapport au Modèle (contre-ex : le formalisme Entité Association)
- Proximité accrue de la représentation avec un domaine d'application
- "the only difficult problem the author solved was understanding his own notation"L.Lamport

BUG : AVEC UN "FORMALISME" ON FAIT DES "MODÈLES"

- modèle : représentation idéale d'une situation du monde réel, limitée aux concepts couverts par un Modèle.
- Objectif : analyser le modèle pour en tirer des enseignements sur la situation du monde réel
- Méta-modèle : …

LEÇONS À EN TIRER

- Le Modèle et son formalisme associé ont un impact majeur sur ce que l'on va décrire
- Ce que l'on décrit a un impact sur ce que l'on peut analyser
- Plus c'est simple, moins on dit de choses
- Plus c'est simple plus on laisse des choses dans le flou



POURQUOI MODÉLISER UN SYSTÈMES INTERACTIFS ?

- Une description abstraite du système
  - indépendante de l'implémentation
  - qui n'intègre pas trop tôt les détails
- Décrire les sorties à produire en fonction des entrées
- Permettre la discussion entre les différents intervenants (à un instant donnée et tout au long du processus de développement)
  - garder le résultat des discussions
  - garder la trace des discussions qui amènent aux décisions ???

QU'EST-CE QU'UN MODÈLE DU SYSTÈME POUR LES SYSTÈMES INTERACTIFS ?

- Décrit à la fois données et actions du système
- Décrit le comportement du système
  - Quelles sont les actions offertes
  - Quand une action est disponible (en fonction de l'état du système)
  - Quel est l'effet d'une action sur l'état du système
- Décrit l'aspect externe
  - comment le système est rendu perceptible à l'utilisateur
  - comment l'utilisateur peut interagir avec le système

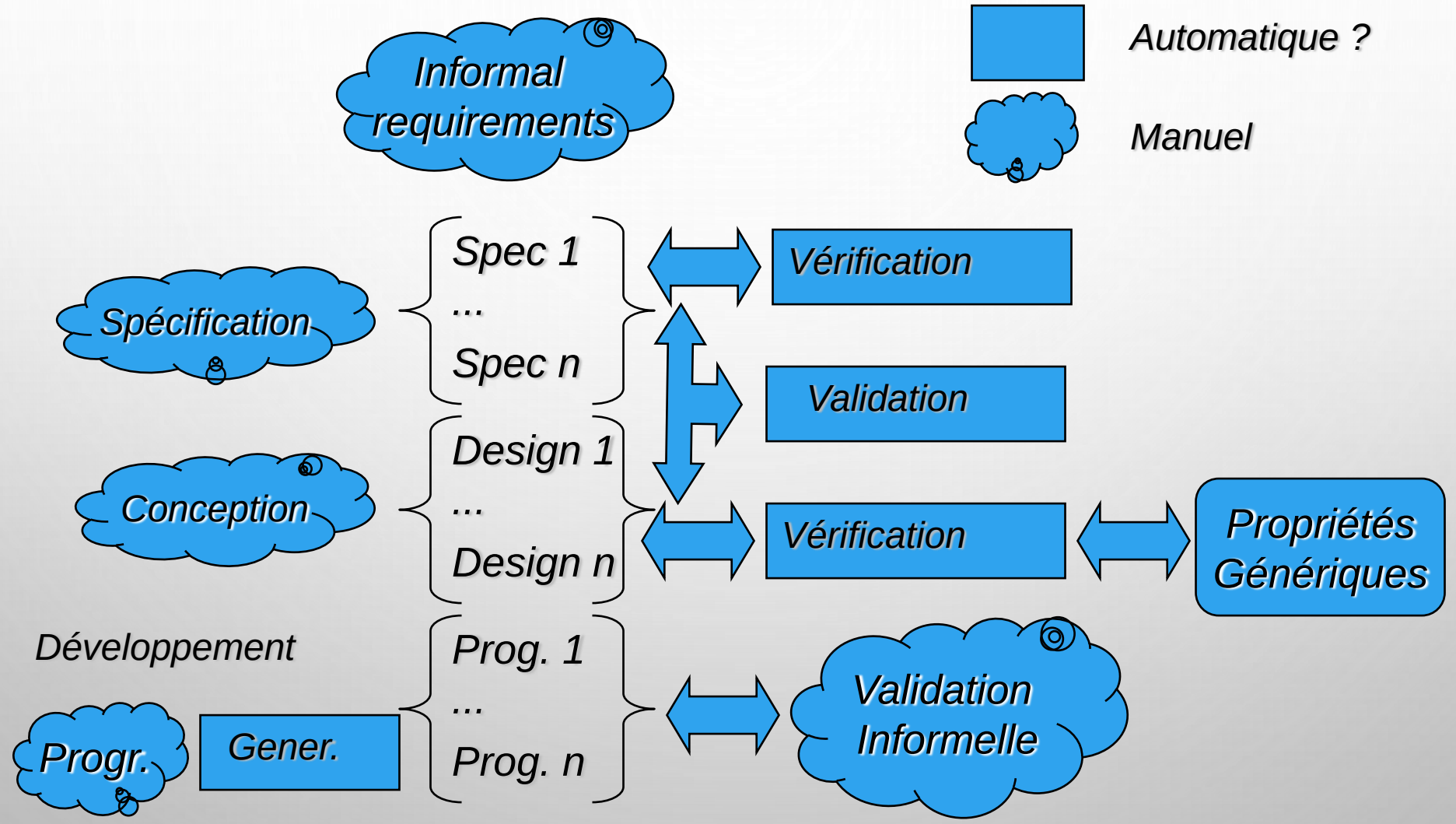
EXEMPLE - EXERCICE

- 1- Décrire avec le formalisme entité association une base de donnée avec des clients et des représentants
  - ID, nom, prénom, âge, …
- 2- Décrire avec le même formalisme l'accès à ces tables
  - Ouvrir la table, rechercher un élément, modifier l'élément, …
- Discuter …
- Refaire la question 2 avec un automate à états

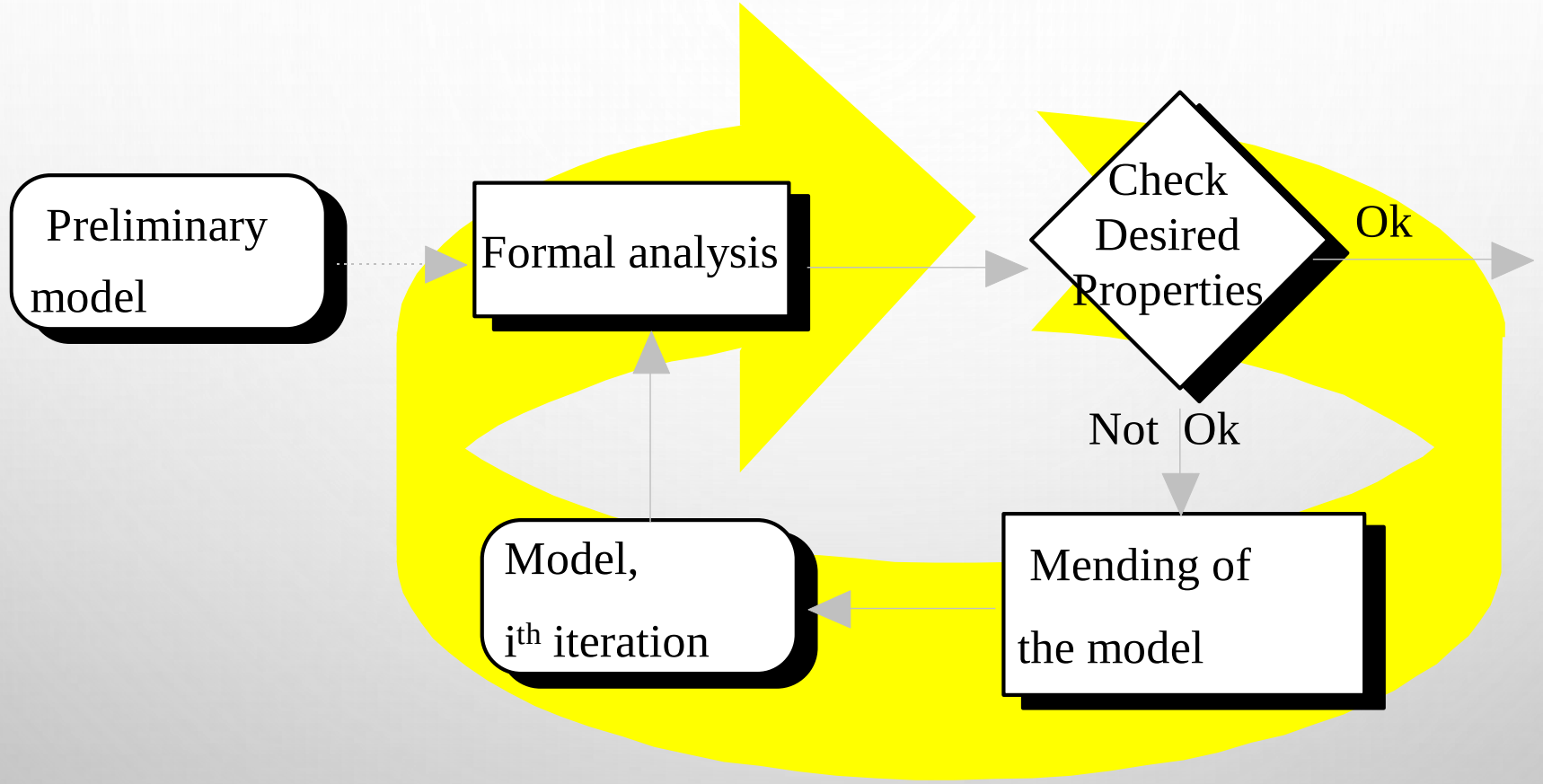
POURQUOI CONCEVOIR LES SI **FORMELLEMENT** ?

- Pour gérer la complexité
- Pour limiter l'intervention humaine en observation (contrôler les modèles)
- Pour limiter l'intervention humaine en traduction (écriture du code)
- Pour raisonner
  - en validation
  - en vérification
- Pour atteindre trois buts fondamentaux
  - fiabilité : propriétés spécifiques et génériques
  - efficacité : performances du système, de l'utilisateur (charge de travail) et du couple utilisateur/système (au niveau des tâches)
  - utilisabilité

PROCESSUS DE CONCEPTION AVEC DES APPROCHES FORMELLES

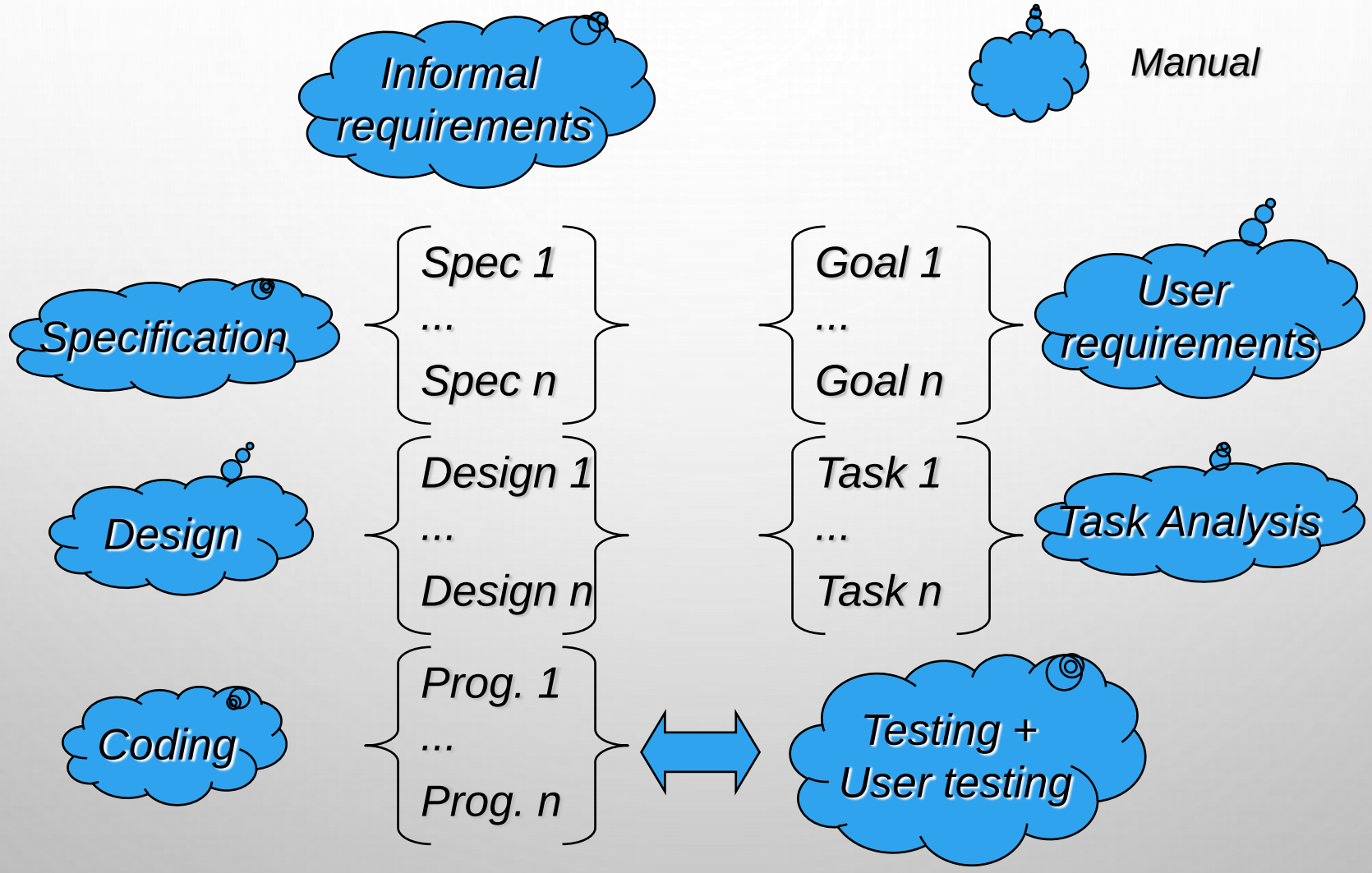


PROCESSUS DE CONSTRUCTION DES MODÈLES

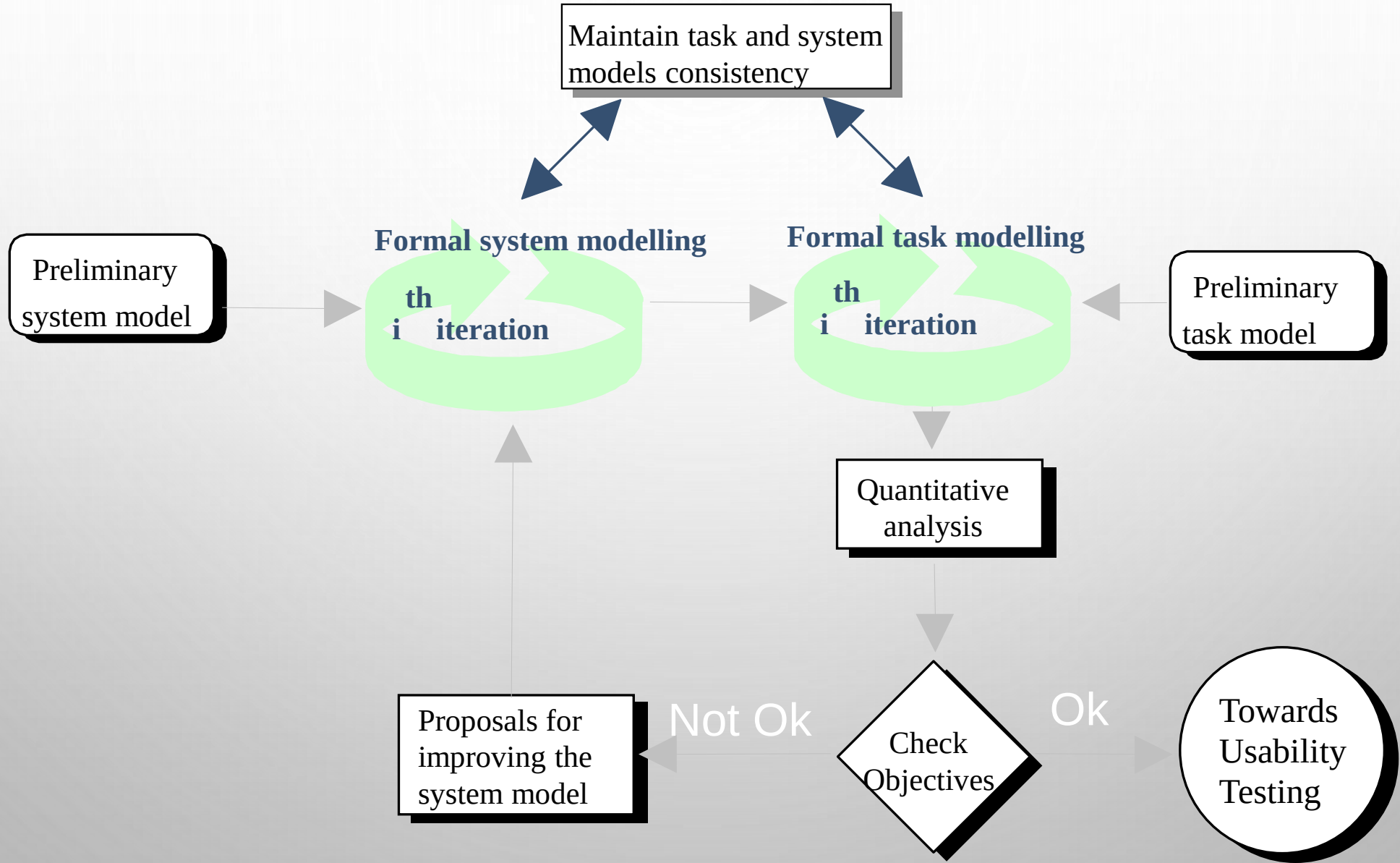


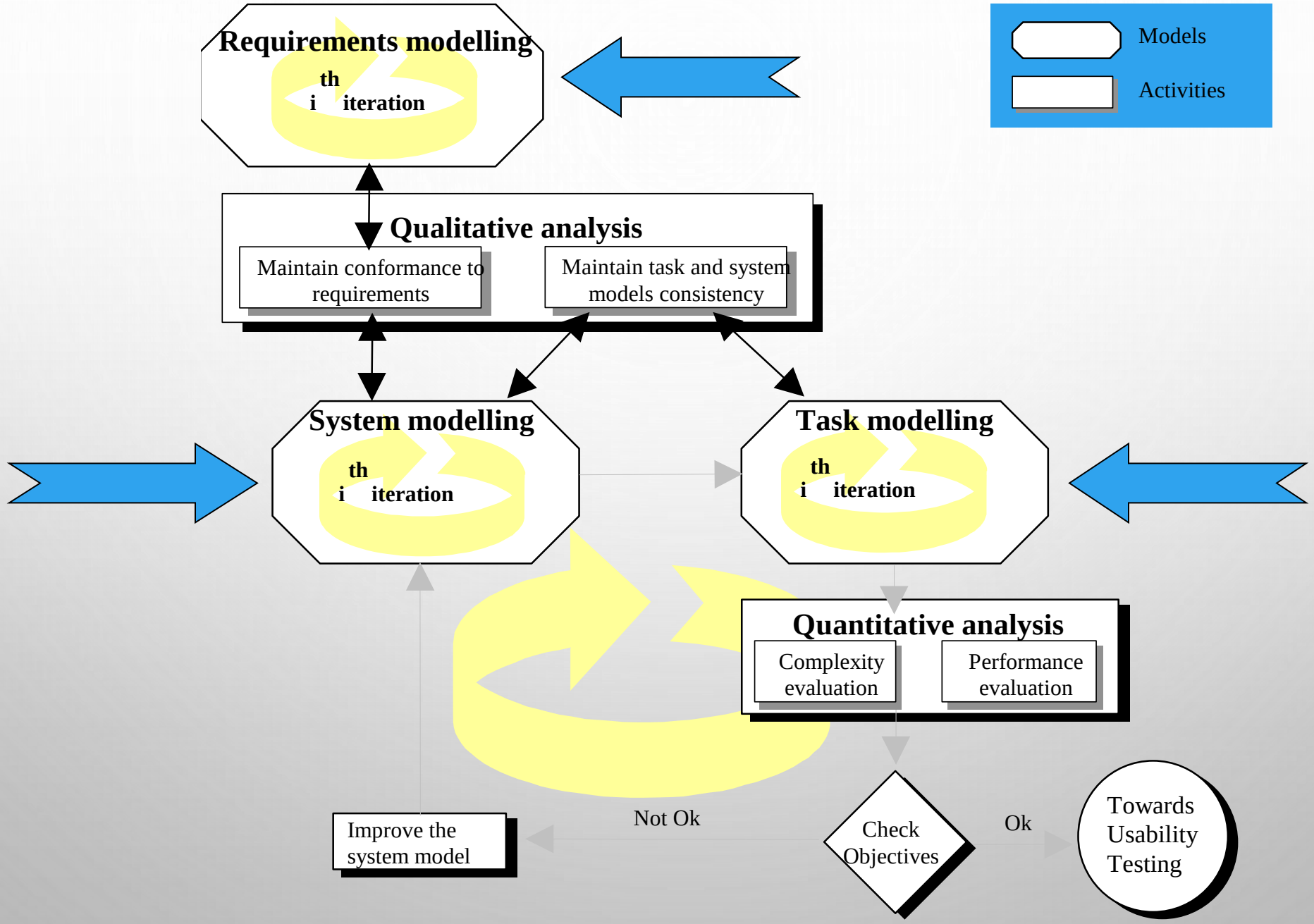
Palanque P. & Bastide R. Synergistic modelling of tasks, system and users using formal specification techniques Interacting With Computers, Academic Press, 9, 12, pp. 129-153, 1997.

SPÉCIFICITÉ DES SI (L'UTILISATEUR ET LES TÂCHES)?



RELATIONS ENTRE TÂCHES ET SYSTÈME




Requirements modelling
th
i iteration
Models
Activities
Qualitative analysis
Maintain conformance to requirements
Maintain task and system models consistency
System modelling
th
i iteration
Task modelling
th
i iteration
Quantitative analysis
Complexity evaluation
Performance evaluation
Improve the system model
Not Ok
Check Objectives
Ok
Towards Usability Testing

REQUIREMENTS - EXIGENCES

EXEMPLES DE MODÈLES (REQUIREMENTS)

- Un formalisme déclaratif est nécessaire
- Requirements
  - Toute clearance envoyée par un contrôleur à un avion p est reçue par cet avion

    $\forall$ p Planes, $\forall$ req DLRequest,

    AG[send(req,p)]AF<receive(req,p)>true
  - Une clearance data-link envoyée par un contrôleur à un avion p sera uniquement reçue par cet avion

    $\forall$ p,p' Planes !(p=p'), $\forall$ req DLRequest,

    AG[send(req,p)]AG[not(receive(req,p')]true

Philippe A. Palanque, Rémi Bastide, Fabio Paternò: Formal Specification as a Tool for Objective Assessment of Safety-Critical Interactive Systems. INTERACT 1997: 323-330

LA LOGIQUE TEMPORELLE CTL * (COMPUTATIONAL TREE LOGIC STAR)

- Un état a un ou plusieurs successeurs
- L'ensemble des états représente un arbre infini
- Opérateurs
    - A (tous les futurs possibles); E (un futur possible) + {F inévitablement, G toujours, X suivant, U jusqu'à}
- Connecteurs logiques
    - $\wedge$ (et); $\vee$ (ou); $\neg$ (non); $\Rightarrow$ (implication)

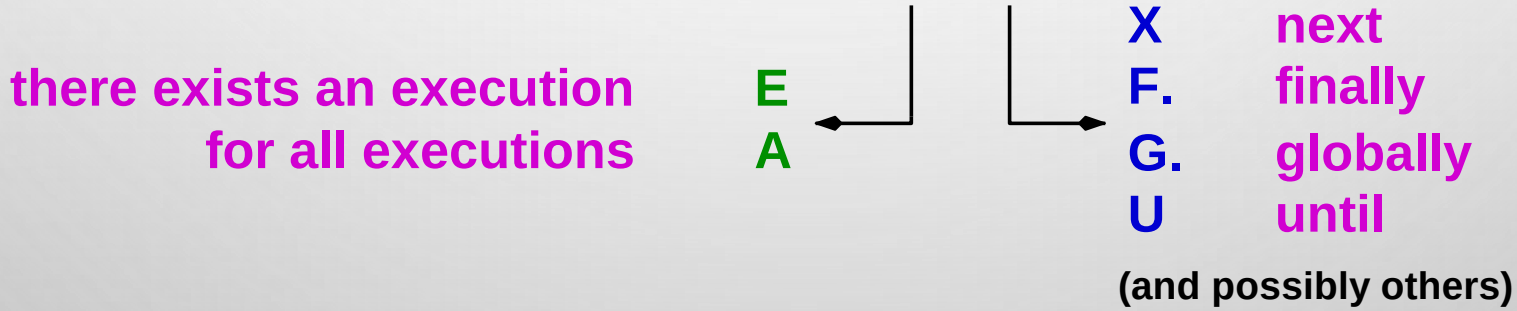
CTL: SURVOL DES OPÉRATEURS

CTL = Computation-Tree Logic

Combines temporal operators with quantification over runs

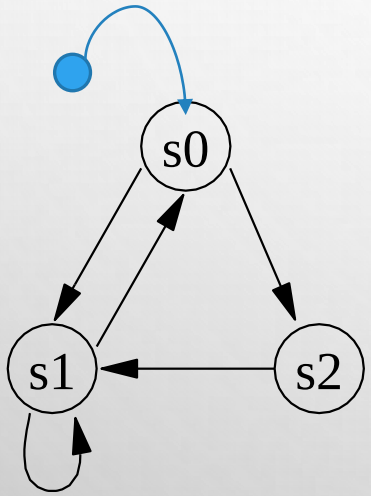
Operators have the following form:

Quantification over runs: Q T: temporal operators

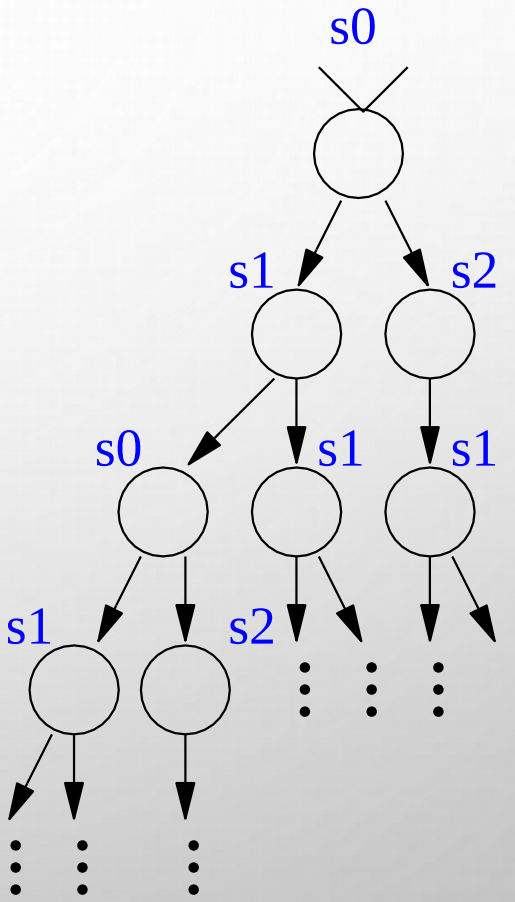


CTL: EXAMPLE

Un système de transition et son arbre calculé (états atteignables en bleu à partir de l'état initial)



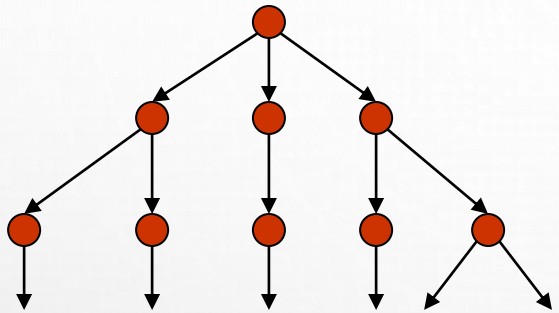
Graphe d'état en intention



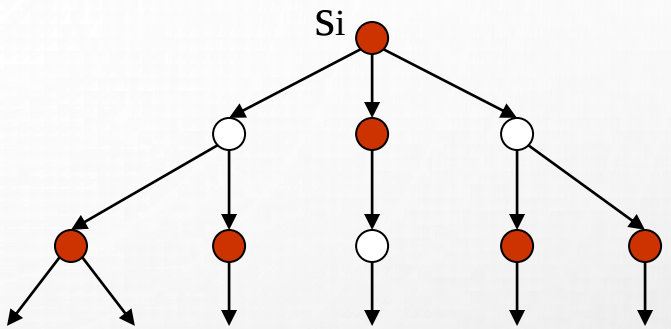
Arbre d'état en extension

LA LOGIQUE TEMPORELLE CTL *

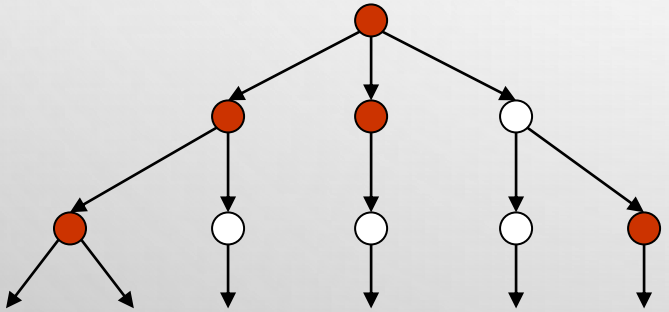
Explicitation de propriétés spécifiques sur les arbres d'états



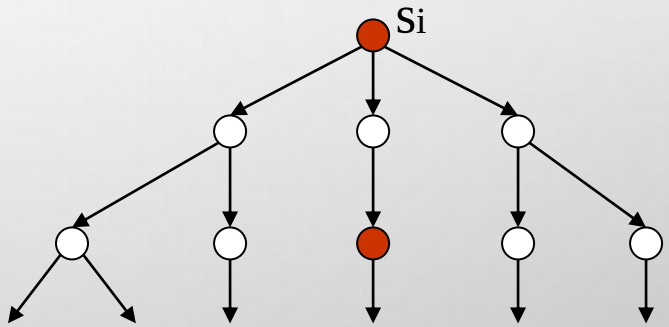
Si |= **AG***p*



Si |= **AF***p*



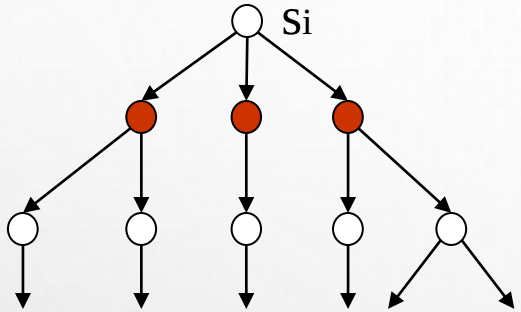
Si |= **EG***p*



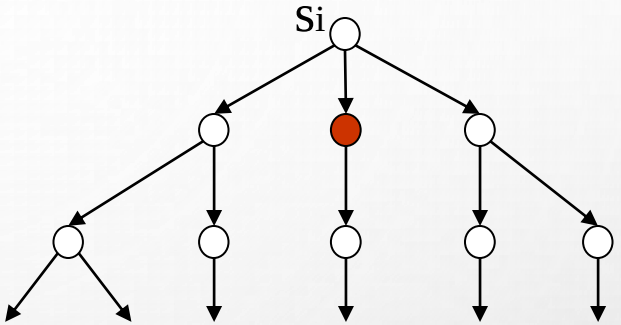
Si |= **EF***p*

La formule ***p*** est vraie si dans l'état rouge et fausse dans l'état blanc

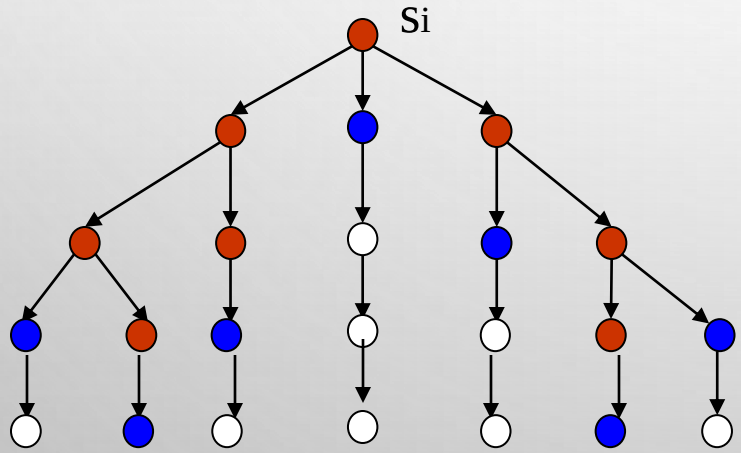
LA LOGIQUE TEMPORELLE CTL *



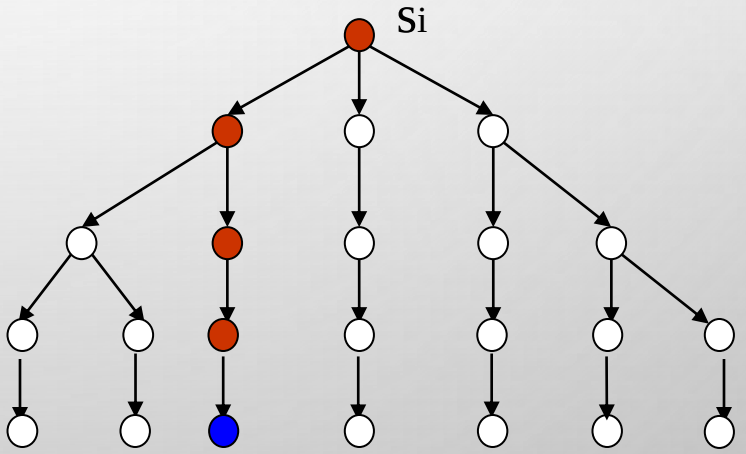
Si |= **AX***p*



Si |= **EX***p*

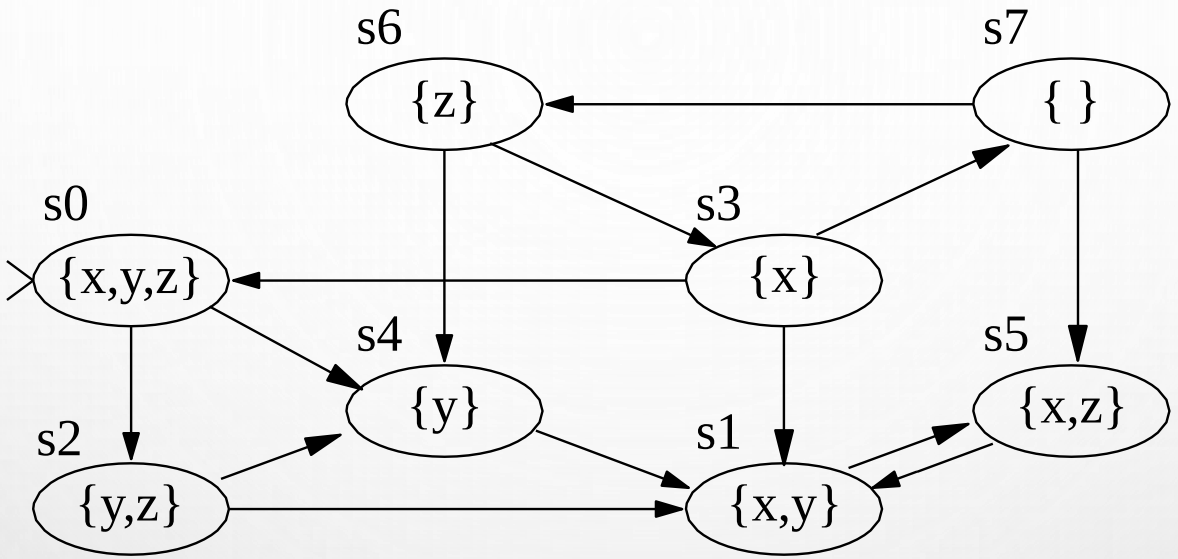


Si |= **A(***p* **U** *q***)**



Si |= **E(***p* **U** *q***)**

SOLVING NESTED FORMULAS: IS S0 ∈ [[AF AG X]]?



To compute the semantics of formulas with nested operators, we first compute the states satisfying the innermost formulas; then we use those results to solve progressively more complex formulas.

In this example, we compute [[*x*]], [[**AG** *x*]], and [[**AF AG** *x*]], in that order.

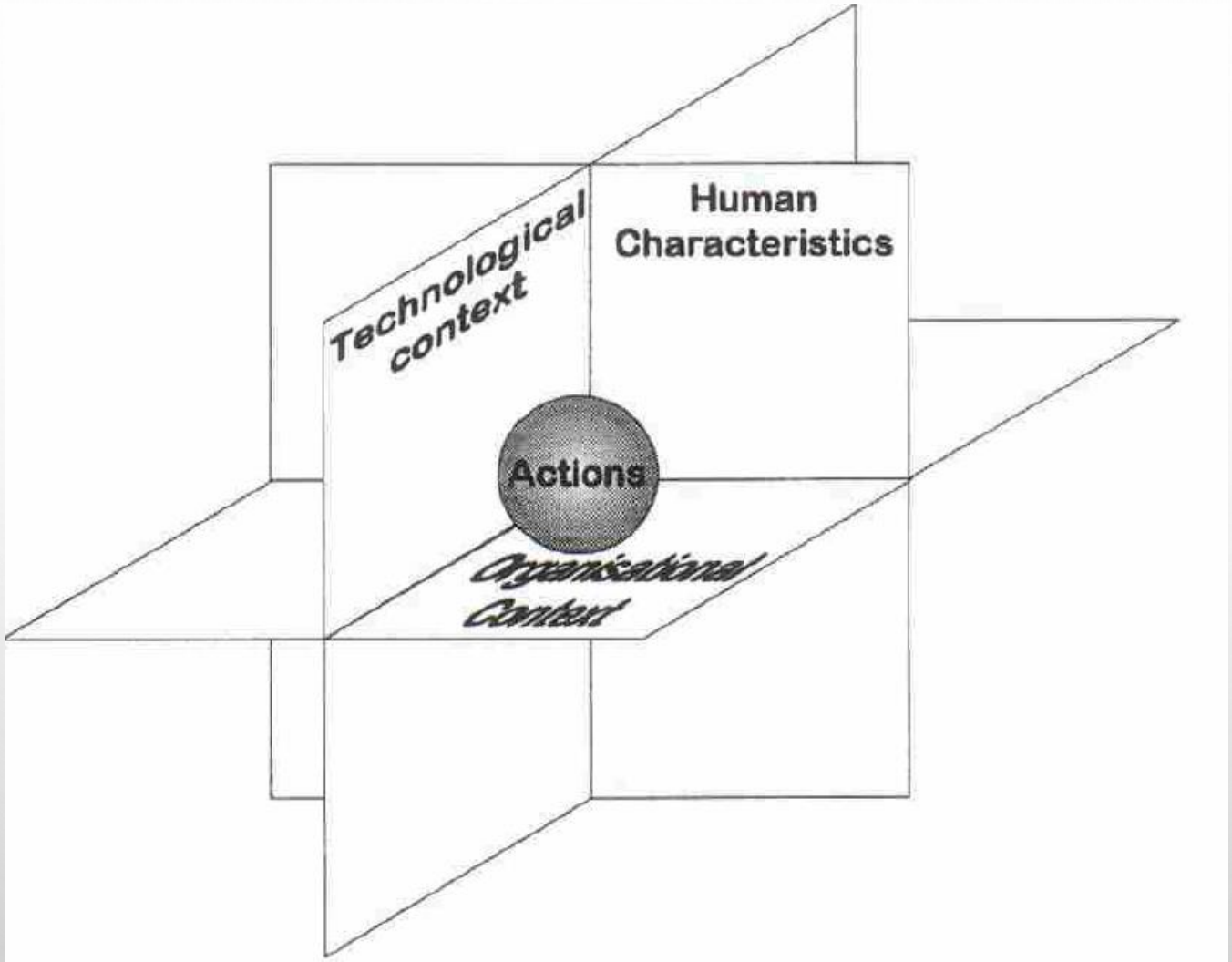
CLASSES DE PROPRIÉTÉS (SAFETY / LIVENESS)

Pnueli, A. (1986). Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends (pp. 510-584). Springer Berlin Heidelberg

- Safety (sûreté)
  - A safety property is an invariant, "always X";
  - "nothing bad [that would be NOT-X] ever happens"
  - A system can be fulfilling a [or a set of] safety property
- Liveness (vivance)
  - A liveness property is a task-progression assertion "always eventually Y".
  - "something good (Y) always eventually happens".
  - A system can be fulfilling a [or a set of] liveness property

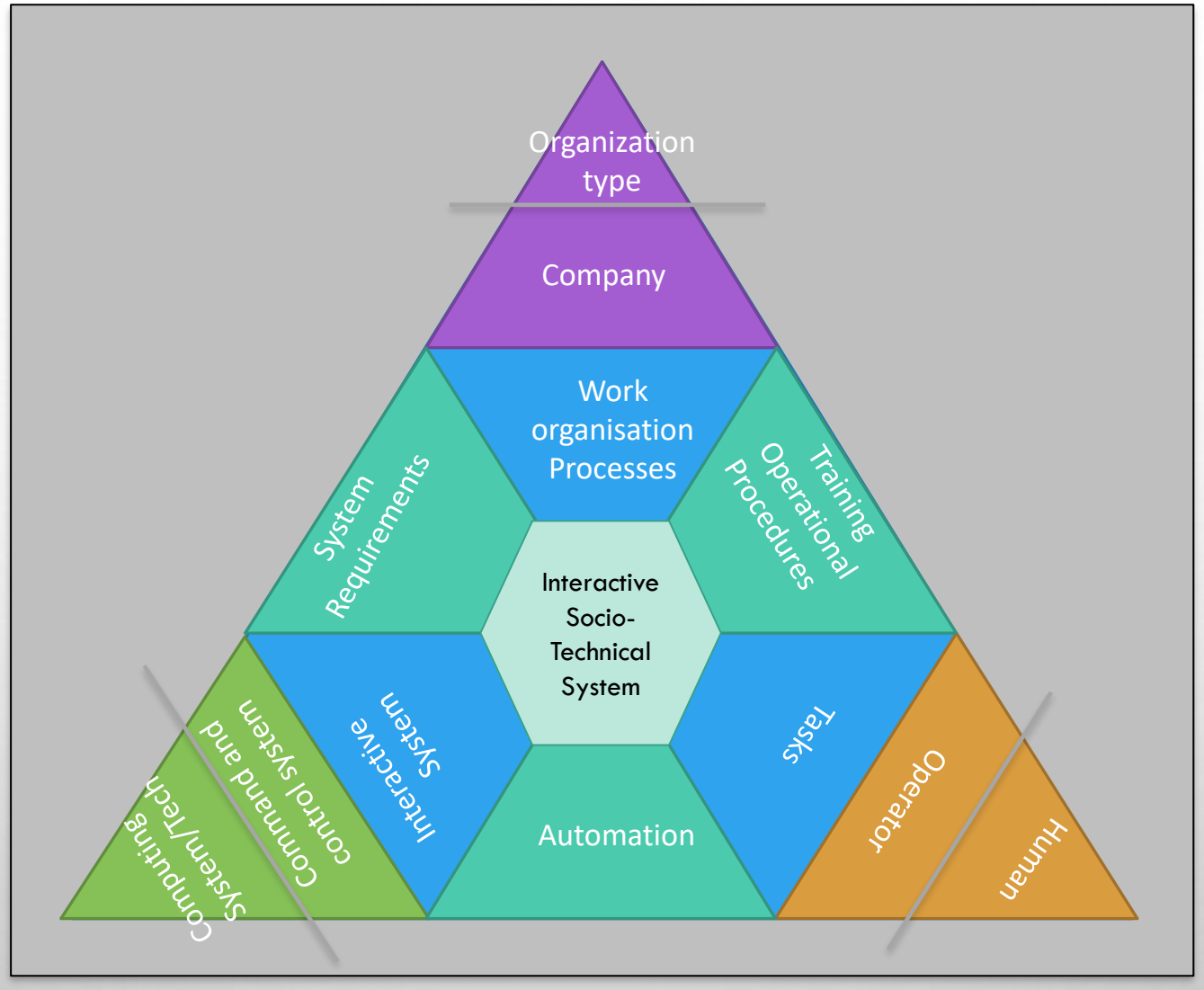
UNE VUE PLUS GLOBALE (ORGANISATION)

Provenance des requirements



Distinction between ergonomics and cognitive ergonomics

Erik Hollnagel. 1997. Cognitive ergonomics: it's all in the mind. Ergonomics 40, 10 (1997), 1170–1182



Martina Ragosta, Célia Martinie, Philippe Palanque, David Navarre, and Mark Alexander Sujan. 2015. Concept Maps for Integrating Modeling Techniques for the Analysis and Re-Design of Partly-Autonomous Interactive Systems. In *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems* (ATACCS '15), ACM, New York, NY, USA, 41-52.

DO178-C FORMAL METHODS SUPPLEMENT 333

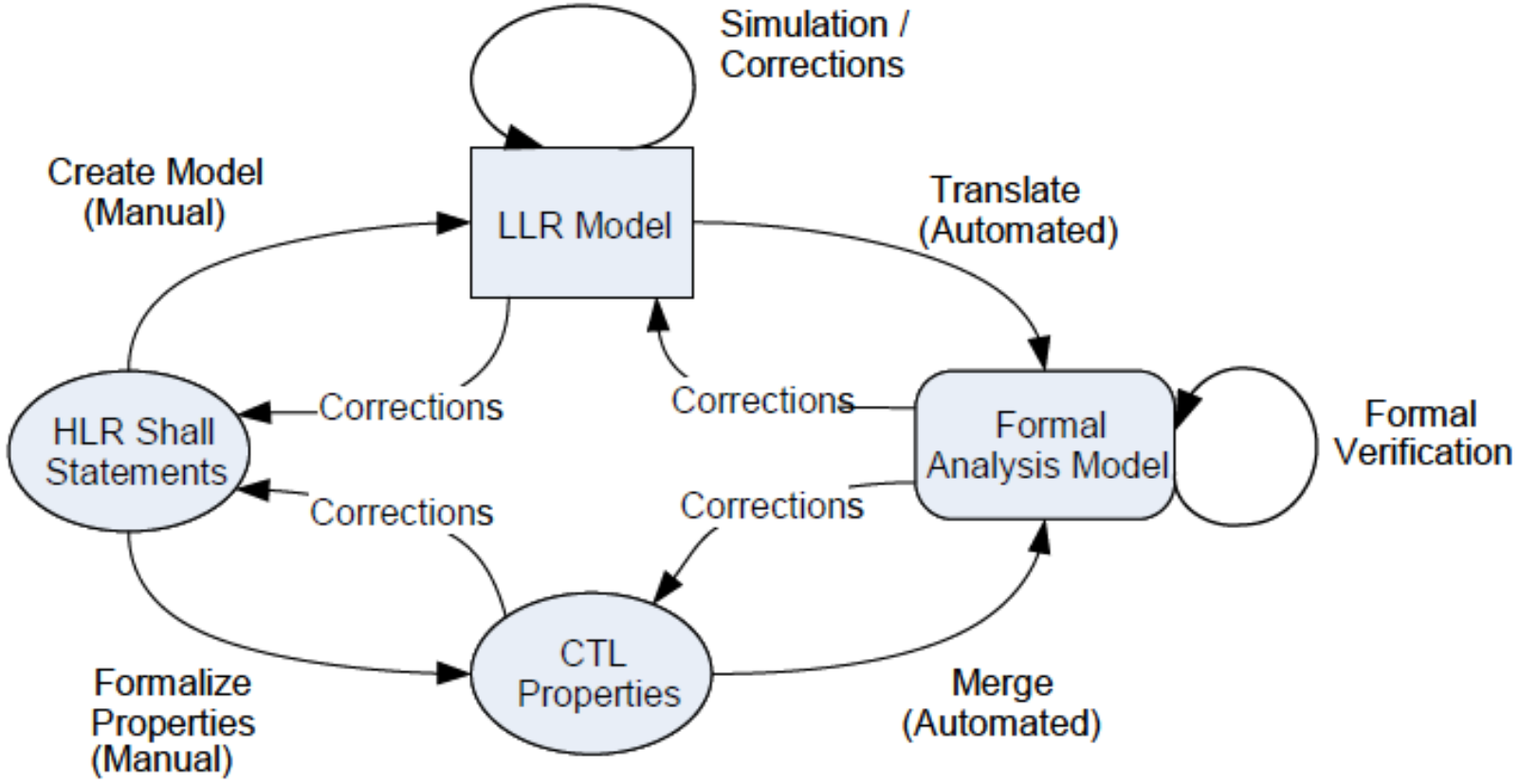


FIGURE FM.B.1-6
ANALYSIS PROCESS

DO178-C FORMAL METHODS SUPPLEMENT 333

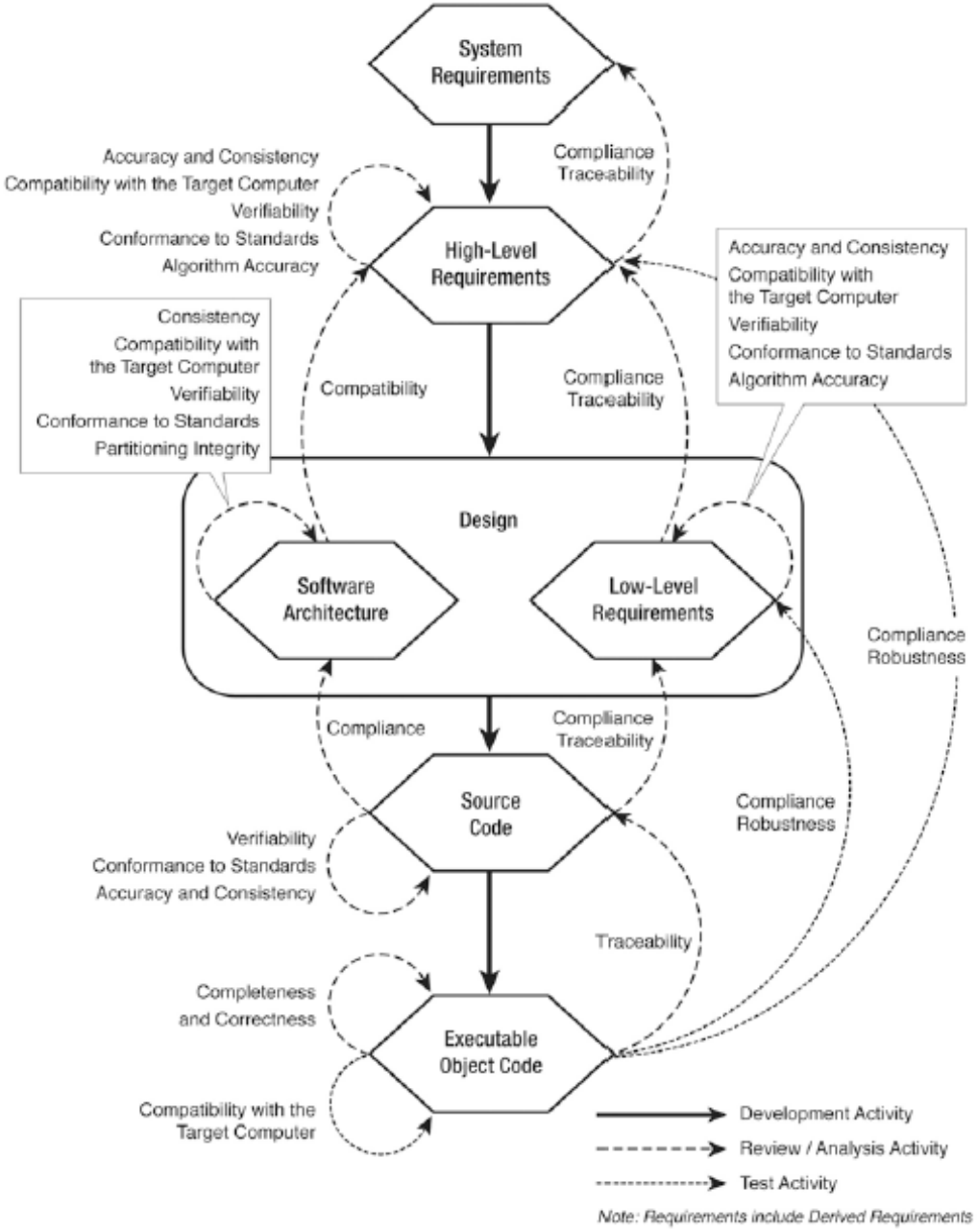


Figure FM.6-1 Level A Software Verification Processes (DO-178C TABLES A-3 TO A-6)

DS

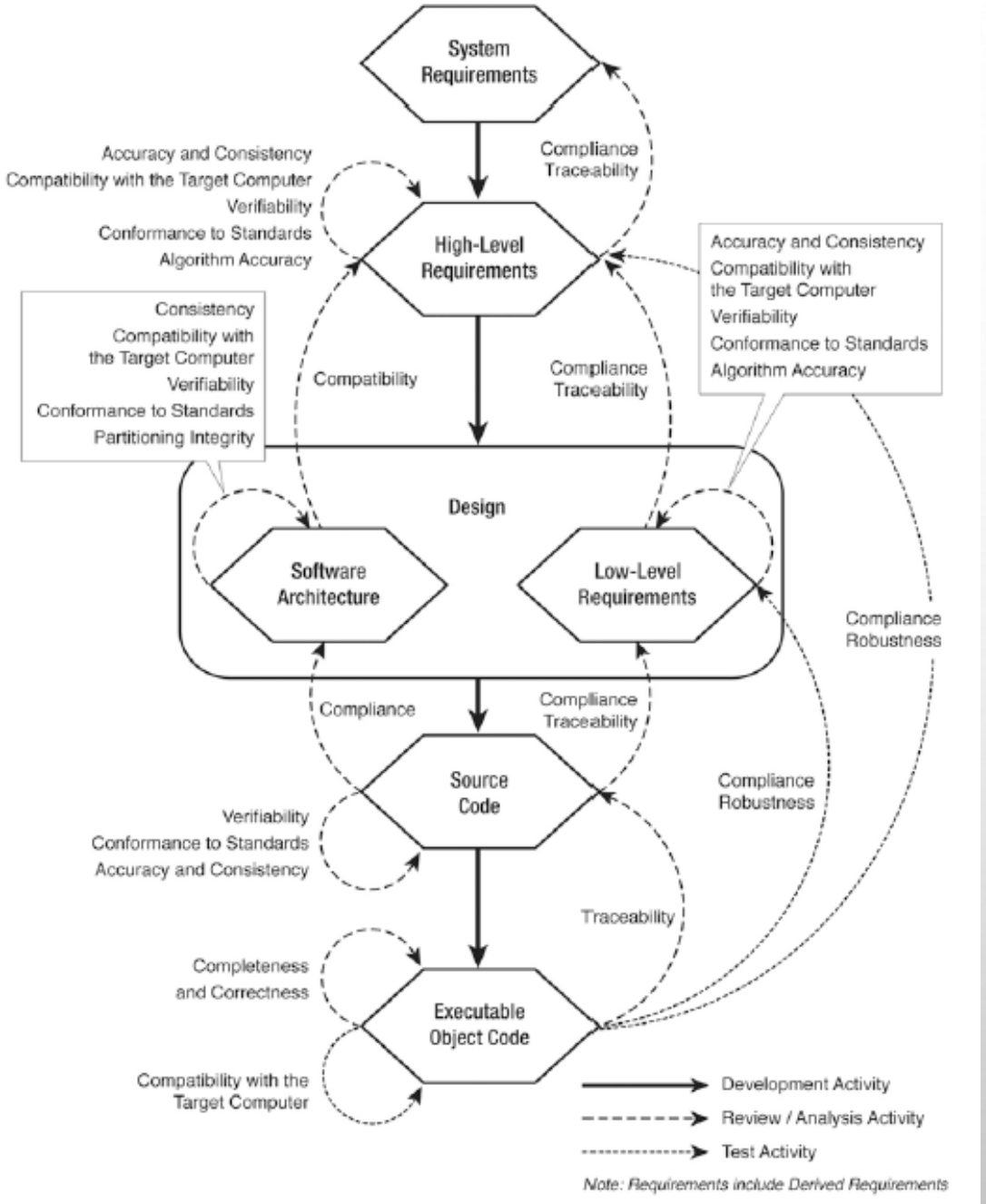


Figure FM.6-1 Level A Software Verification Processes (DO-178C TABLES A-3 TO A-6)

V&V: VERIFICATION AND VALIDATION

- Verification

– The process of determining that a model implementation accurately represents the developer's conceptual description and specifications and that it exhibits desired properties

"Did I build the system right?"

- Validation

– The process of determining (a) the manner and degree to which a model is an accurate representation of the 'real-world' from the perspective of the intended uses of the model

"Did I build the right system?"

ELÉMENTS À DÉCRIRE LORS DE LA MODÉLISATION DE SYSTÈMES INTERACTIFS

FONDEMENTS DES APPROCHES FORMELLES POUR LES SI

- décrire à la fois les états et les événements
- décrire à la fois la structure de donnée et la structure de contrôle
- offrir des mécanismes de structuration
- décrire la concurrence
- décrire les aspects temporels
- faire tout cela de façon formelle

- 3 problèmes: quoi décrire, avec quoi décrire et comment décrire

ASPECTS ÉTATS ET ÉVÉNEMENTS

- Systèmes réactifs
- Dirigé par événement
- Code difficile à gérer sans représentation exploitable des états
- Approches
  - approches venant de la conception des SR
  - langages réactifs ou synchrones (esterel, Lotos)
  - utilisation méthodologique des RdP

ASPECT STRUCTURE DE DONNÉE / STRUCTURE DE CONTRÔLE

- La crise du logiciel à montré les limites de la séparation données/traitements (maintenabilité, évolutivité, réutilisabilité, ...)
- Approches
  - mélanger deux approches (CSP-Z, Object-Z, Full LOTOS, …)
  - intégrer deux approches (ICOs)

MÉCANISMES DE STRUCTURATION

- Gestion de composants complexes
- Compréhensibilité des modèles
- Réutilisabilité
- Evolutivité
- Approches
  - composition (agrégation, association, …)
  - communication (client-server, actors)
  - macros and plugs

CONCURRENCE

- Concurrence avec les dispositifs d'entrée et de sortie
- Collecticiels
- Dialogues multi-fils
- Approches
  - formalismes textuels (CSP, CCS, LOTOS, Logiques Temporelles)
  - formalismes graphique (réseaux de Petri, Statecharts)


Captain
First Officer
FMS
AI1001
ACTIVE
POSITIO
SEC
DATA
CONFIG
ACTIVE / FPLN
FROM
UTC
TRK
DIST
FPA
KIAD
SWANN
V26
GOLDA
V268
BROSS
J42
OOD
(T/C)
J42
DAVYS
J42
BRAND
J42
RVB
INSERT WPT
DELETE *
OFFSET
HOLD
AIRWAYS
OVERFLY *
ENABLE ALTN *
NEW DEST
CONSTRAINTS
CMS
WIND
BRG
089°
58
193°
8
094°
7
057°
37
060°
2
060°
22
061°
10
061°
14
DEST
LFPG
07:42
FPLN-INFO
DIR TO
LINE 1 - 35 CHAR
LINE 2 - 35 CHAR
MSG LIST

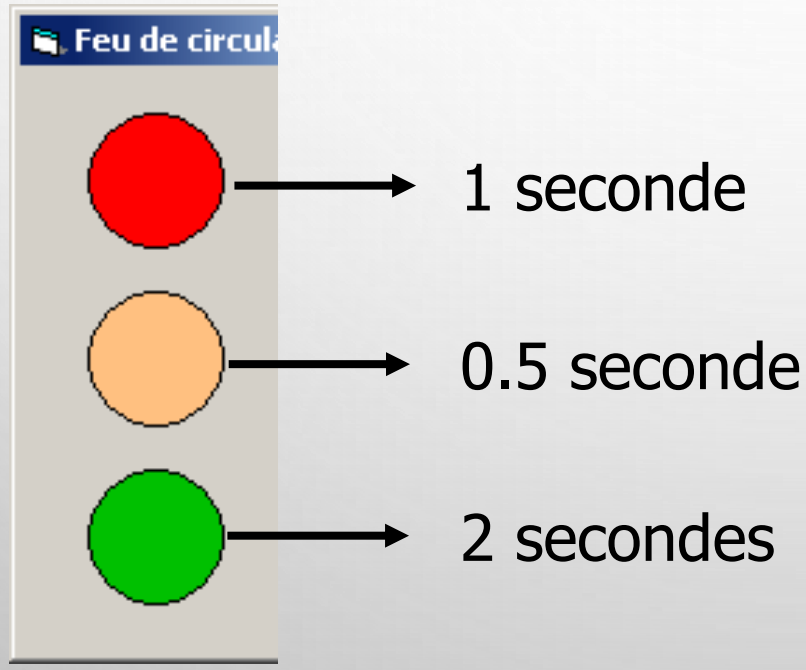
ASPECTS TEMPORELS

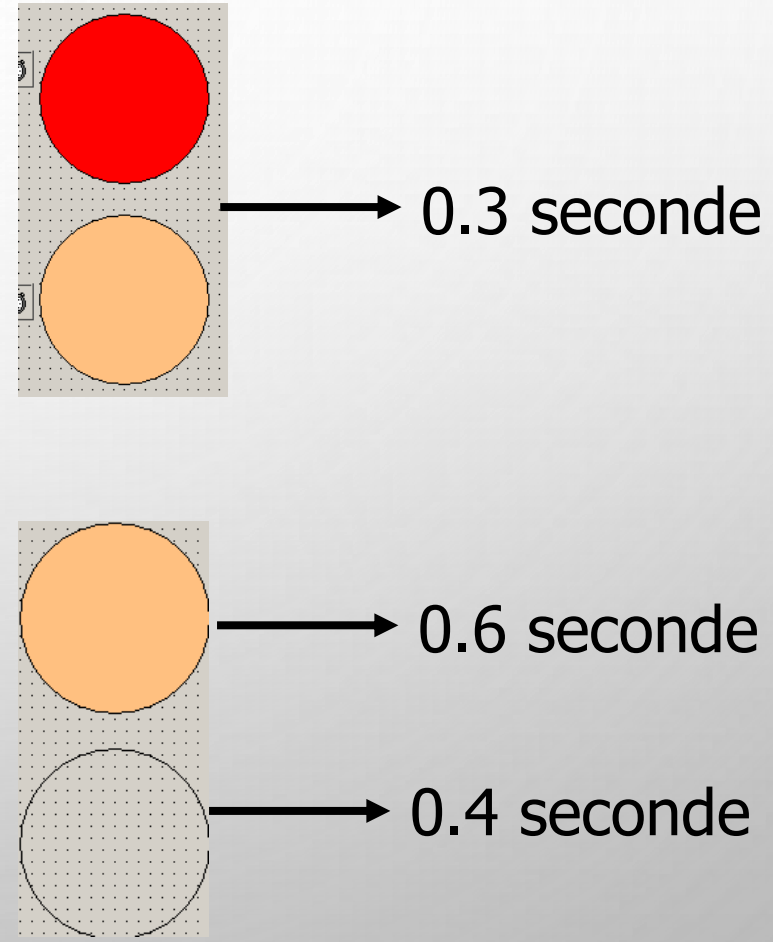
- Systèmes multimodaux
- Animation (évolution basée sur le temps)
- Alarmes
- Événements calendaires

- Approches
  - procédurales (RdP, Lotos, …)
  - déclaratives (Logiques temporelles, …)

EXEMPLE: LE FEU DE CIRCULATION BRITANIQUE

- Aspects temporels





CARACTÈRE FORMEL

- Plus facile de prouver que de tester – on verra ça en programmation
- Systèmes critiques – on doit démontrer le « bon fonctionnement »
- Complétude, concision et non-ambiguïté
- Exécutabilité
- Approches
  - validation mathématique
  - génération de jeux de tests
  - génération automatique de code

ETAT DE L'ART EN APPROCHES FORMELLES POUR LES SYSTÈMES INTERACTIFS

- Compréhension des SI
  - Qu'est-ce qu'un système réactif ?
  - Quelles sont les propriétés fondamentales des SI ?
  - Comment trouver et décrire les propriétés spécifiques à telle ou telle application interactive ?
- Ingénierie des SI
  - Quels sont les composants de base des SI
  - Comment décrire ces composants et leurs inter-relations ?
  - Comment ceci se situe par rapport au processus de développement ?

COMPRÉHENSION DES SI

- Architectures génériques pour les SI
  - PIE and red-PIE models (Dix 91)
  - Seeheim (Green 85) and Arch/Slinky (Bass 92)
- Propriétés génériques pour les SI
  - (Sufrin & He 90), (Dix 91)
  - Propriétés externes et internes (Gram & Cockton 96)
- A mi-chemin entre compréhension et ingénierie
  - CNUCE interactors (Paternò & Faconti 92)
  - York interactors (Duke & Harrison 93)

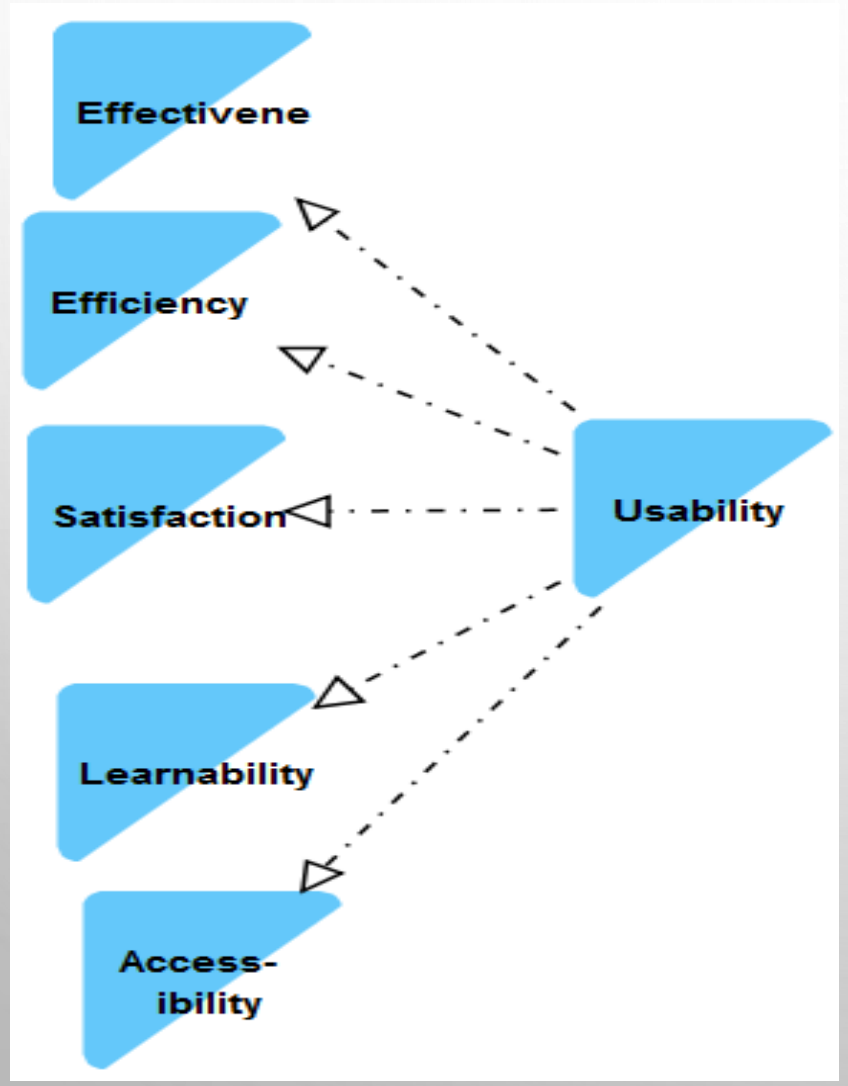
INGÉNIERIE DES SI

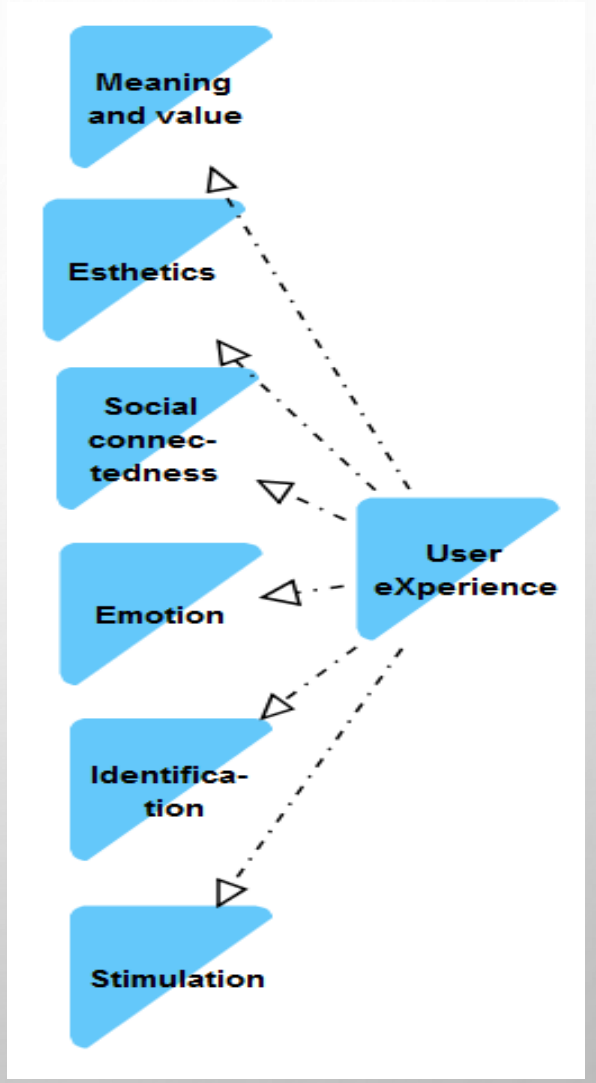
- Dialogue modelling WIMP
  - (Bastide & Palanque 90)
  - (Beck et al. 95)
- Model-Based UIMS (WIMP) (CADUI'96)
  - UIDE (Foley et al. 93)
  - Tadeus (Elwert & Schlungbaum 95)
  - PetShop (Bastide & Palanque 95)
- Design Patterns: PAC (Coutaz 87) & MVC (Goldberg 83)

PROPRIÉTÉS GÉNÉRIQUES

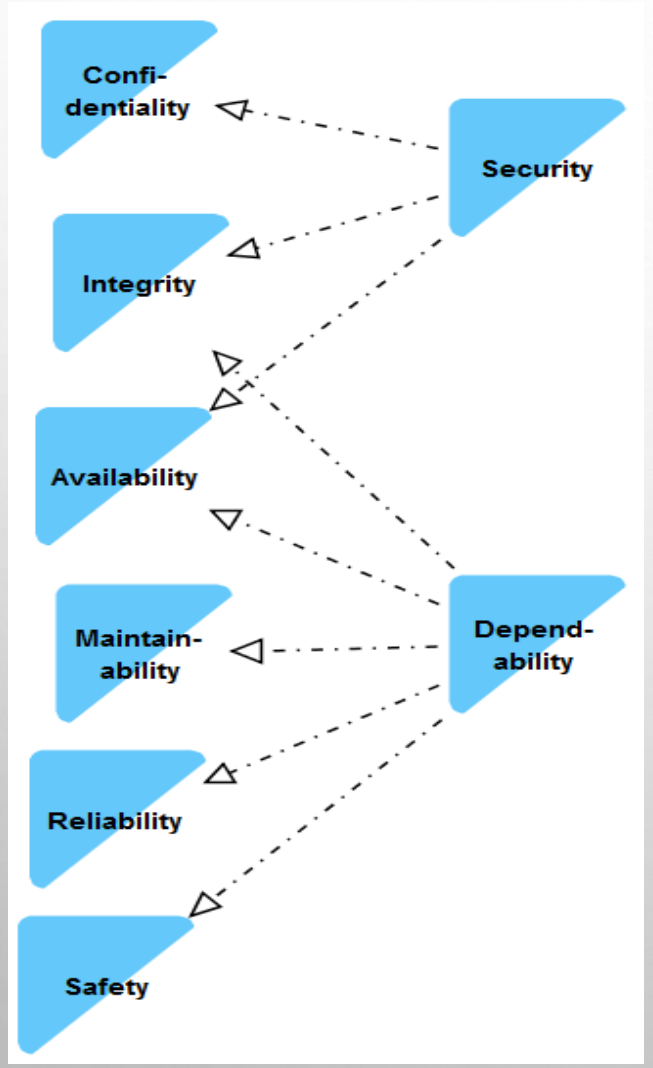
- Propriétés génériques des systèmes
  - vivacité
  - sécurité
  - usure
- Propriétés génériques des systèmes interactifs
  - prédictibilité
  - observabilité
  - atteignabilité

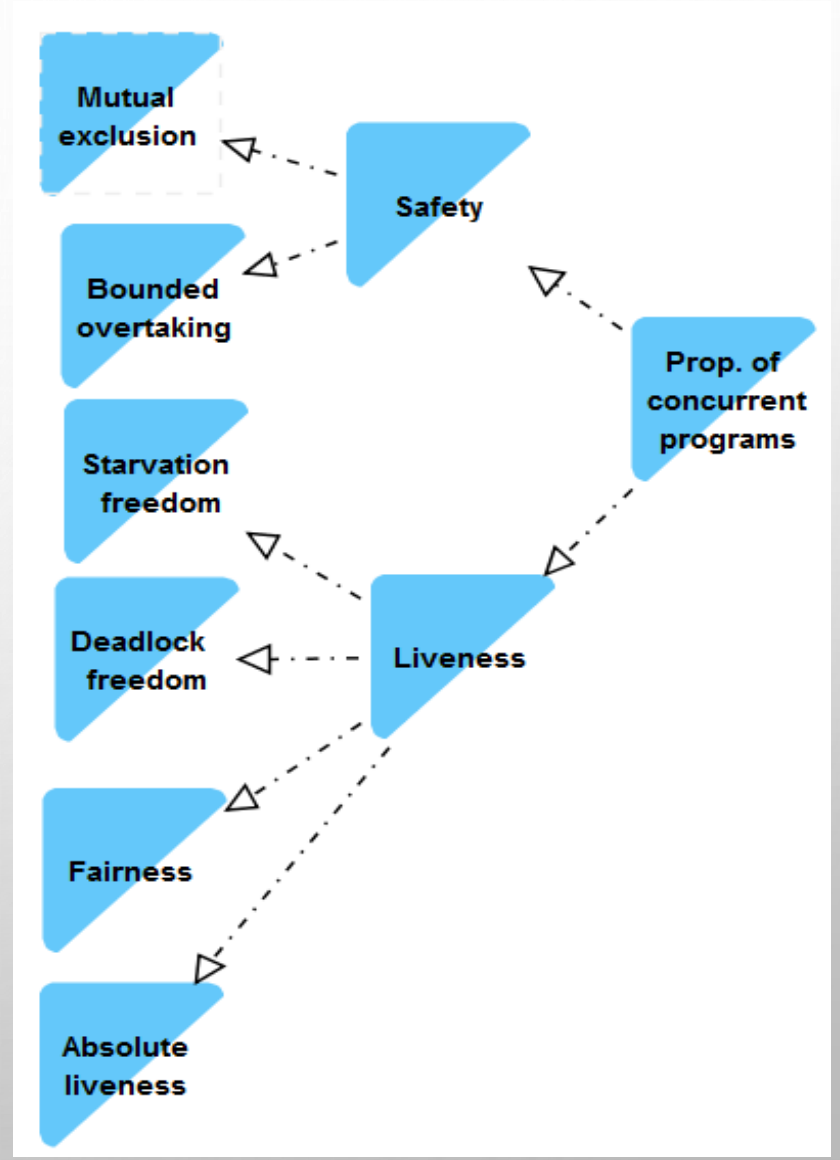
DESCRIPTION OF EXISTING CLASSIFICATION OF PROPERTIES (HCI)





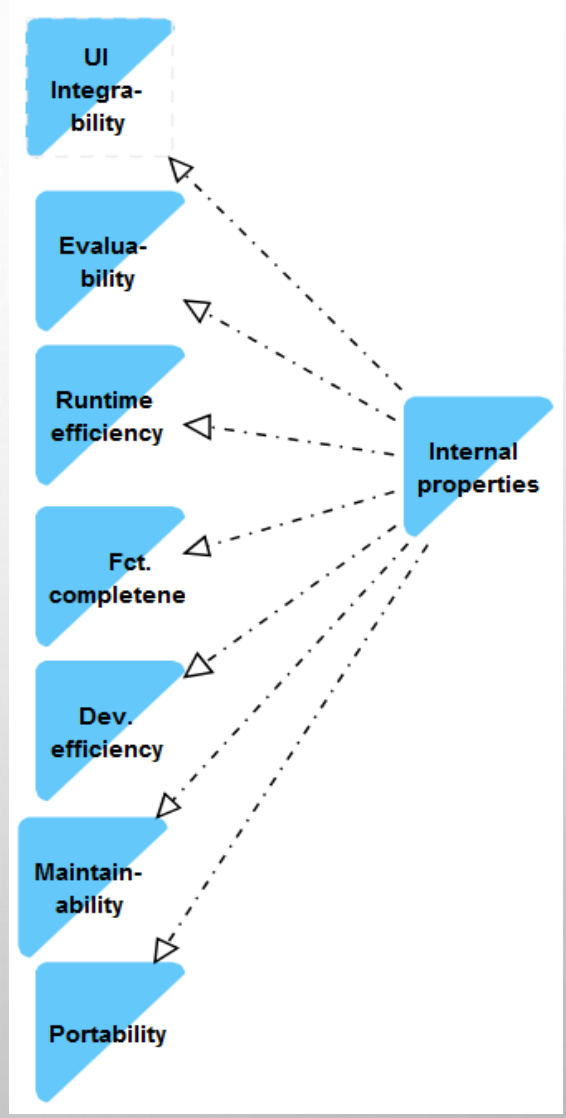
DESCRIPTION OF EXISTING CLASSIFICATION OF PROPERTIES (DEPENDABILITY)





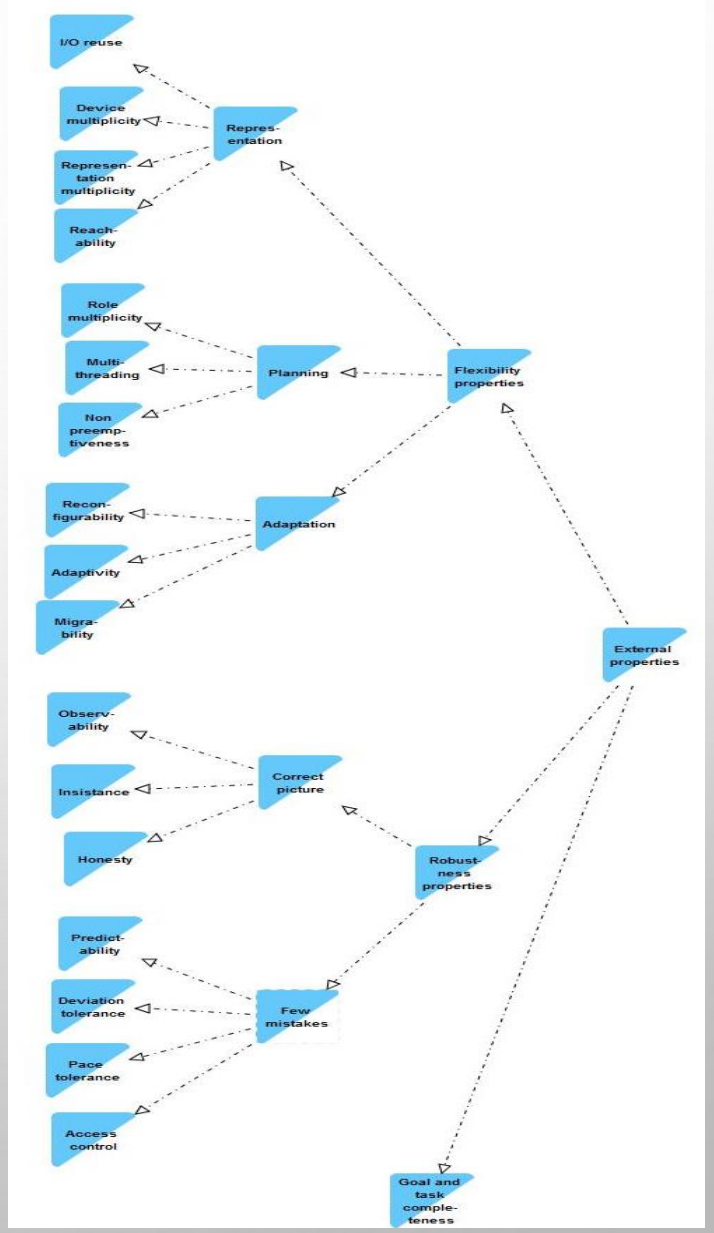
DESCRIPTION OF EXISTING CLASSIFICATION OF GENERIC PROPERTIES (INTERACTIVE SYSTEMS)

- Internal properties
- External properties



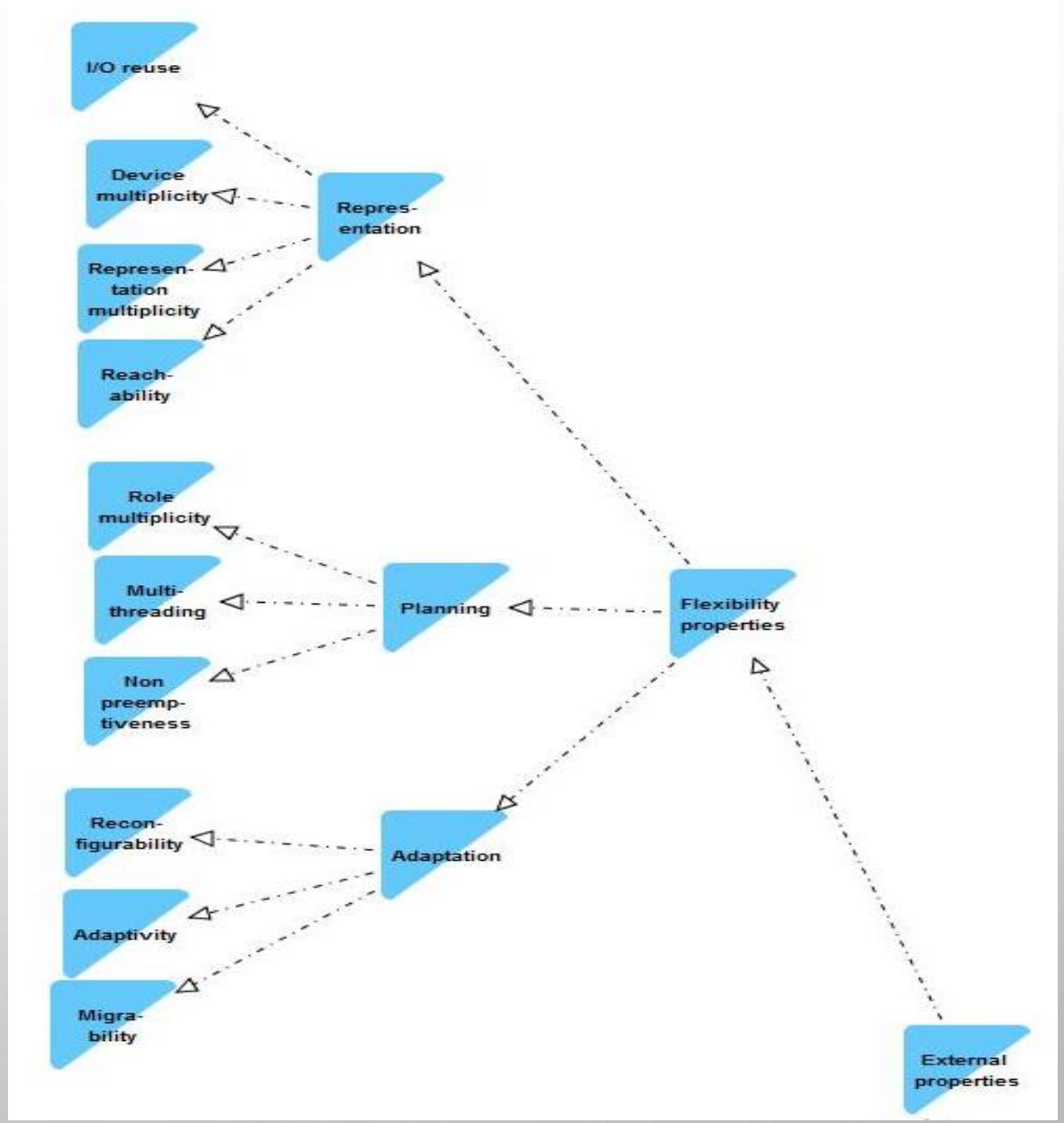
DESCRIPTION OF EXISTING CLASSIFICATION OF GENERIC PROPERTIES (INTERACTIVE SYSTEMS)

- Internal properties
- External properties
  - Flexibility properties
  - Robustness properties



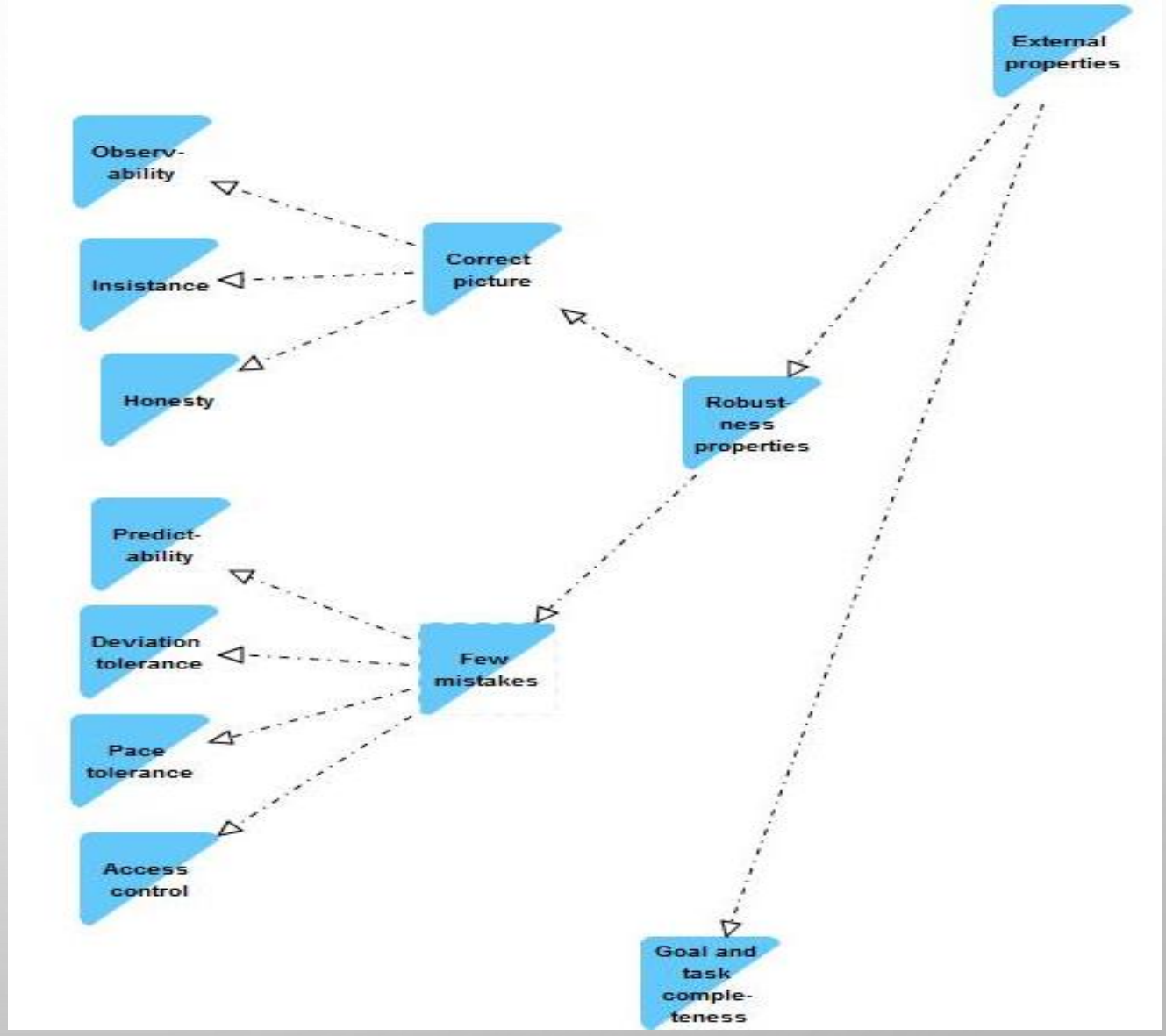
DESCRIPTION OF EXISTING CLASSIFICATION OF FLEXIBILITY PROPERTY

- Internal properties
- External properties
  - Flexibility properties
  - Robustness properties



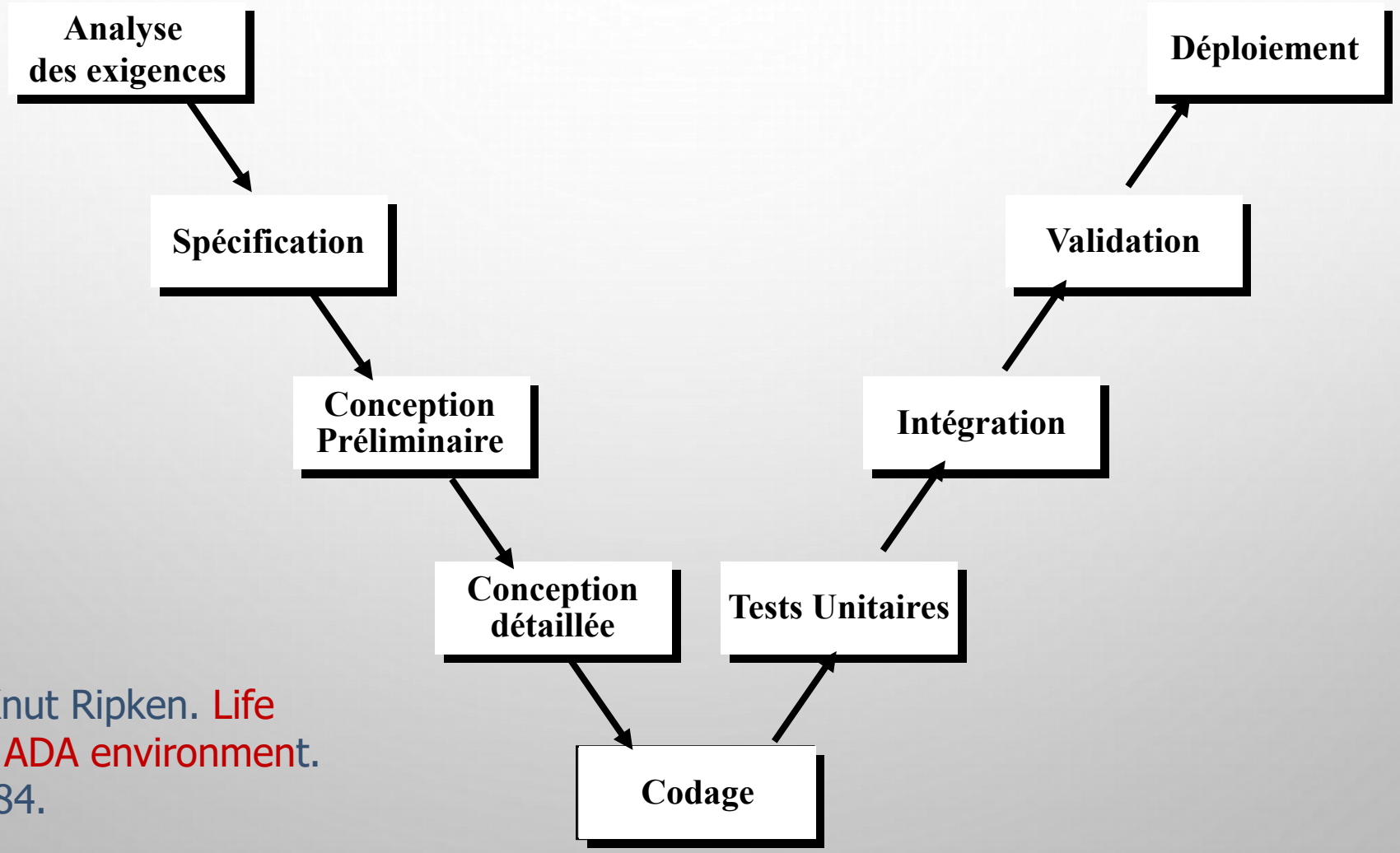
DESCRIPTION OF EXISTING CLASSIFICATION OF ROBUSTNESS PROPERTY

- Internal properties
- External properties
  - Flexibility properties
  - Robustness properties



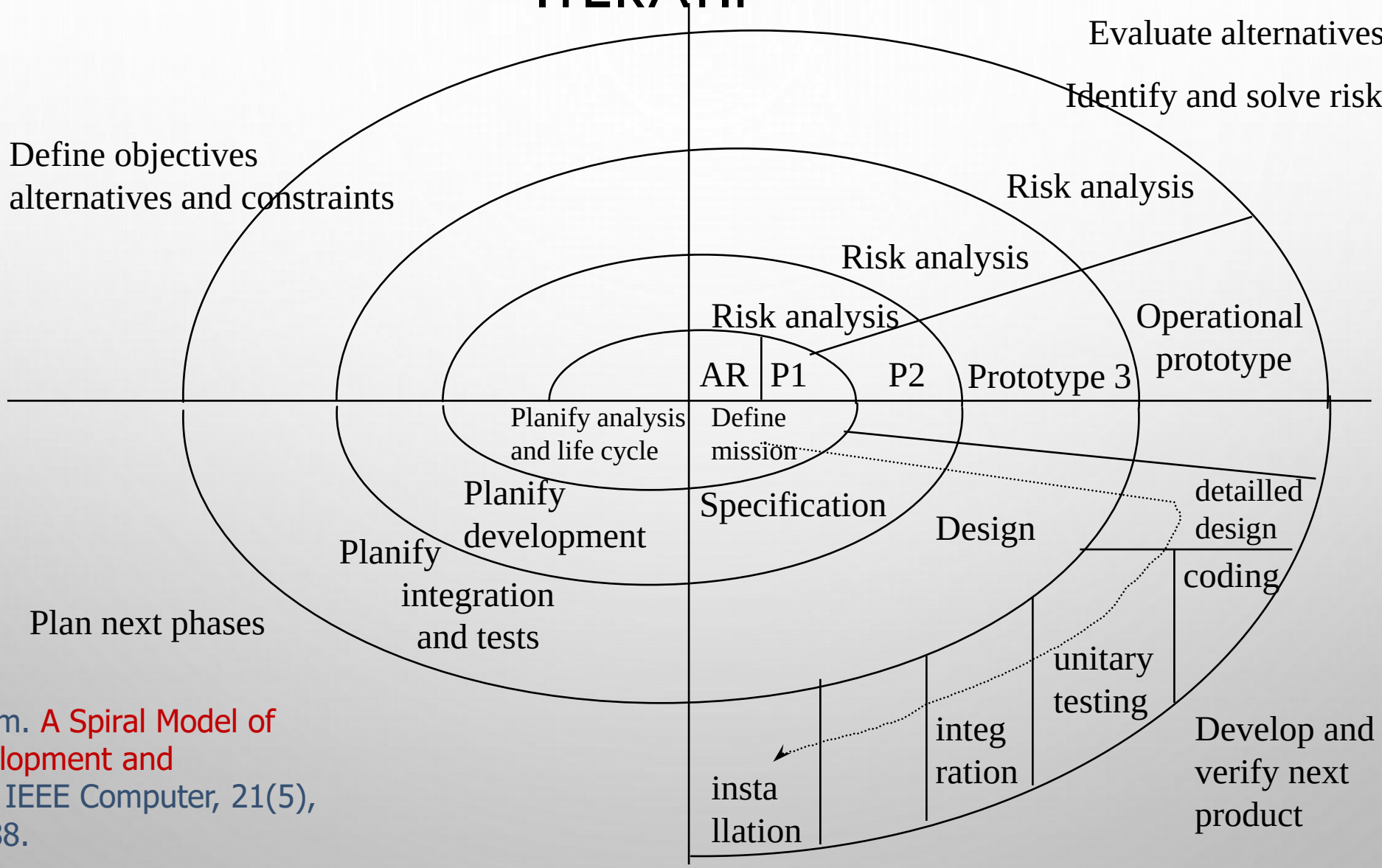
QUELS PROCESSUS AVEC DE TELS OUTILS?

CYCLE DE DÉVELOPPEMENT EN V

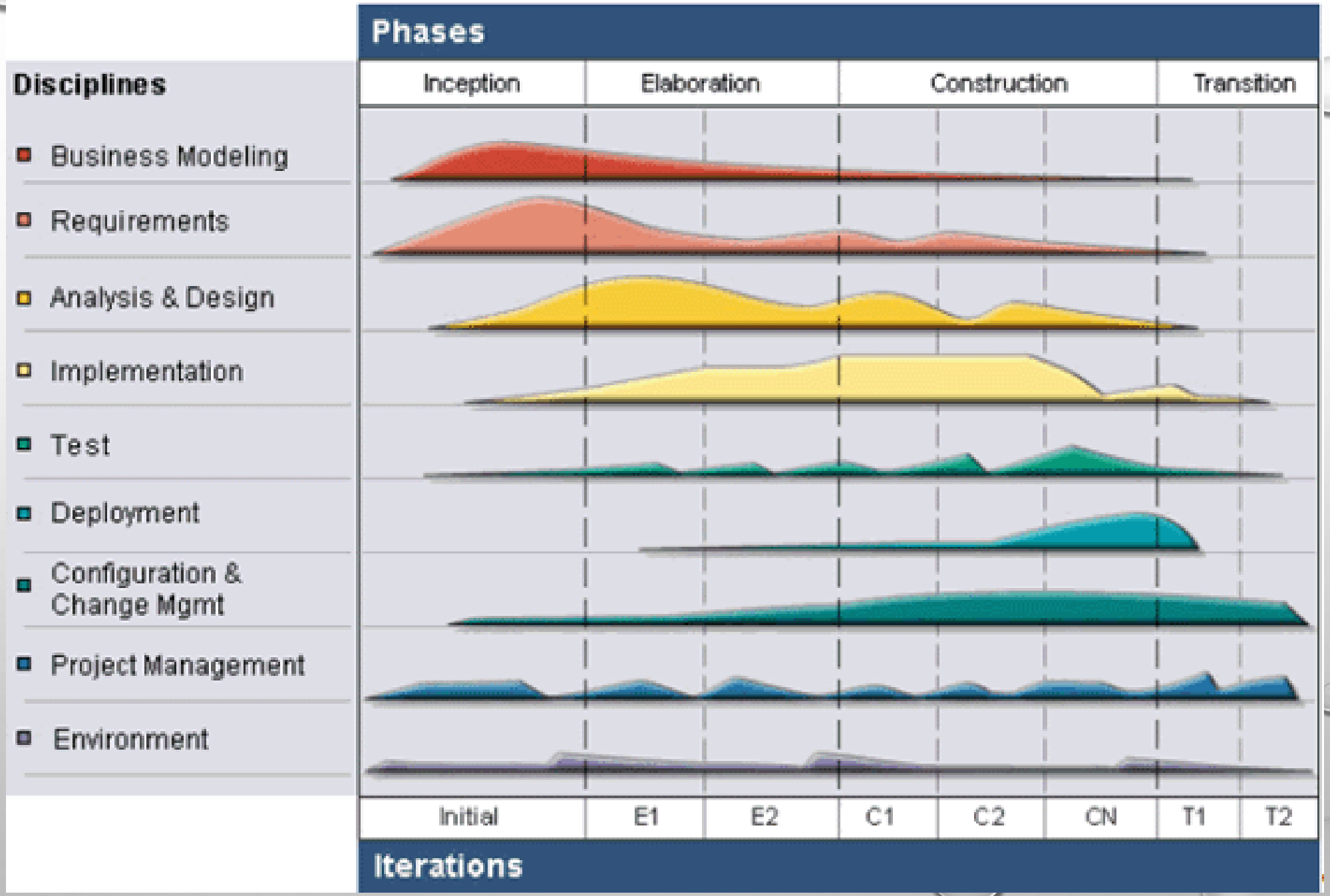


John McDermid et Knut Ripken. Life cycle support in the ADA environment. University Press, 1984.

CYCLE DE DÉVELOPPEMENT ITÉRATIF

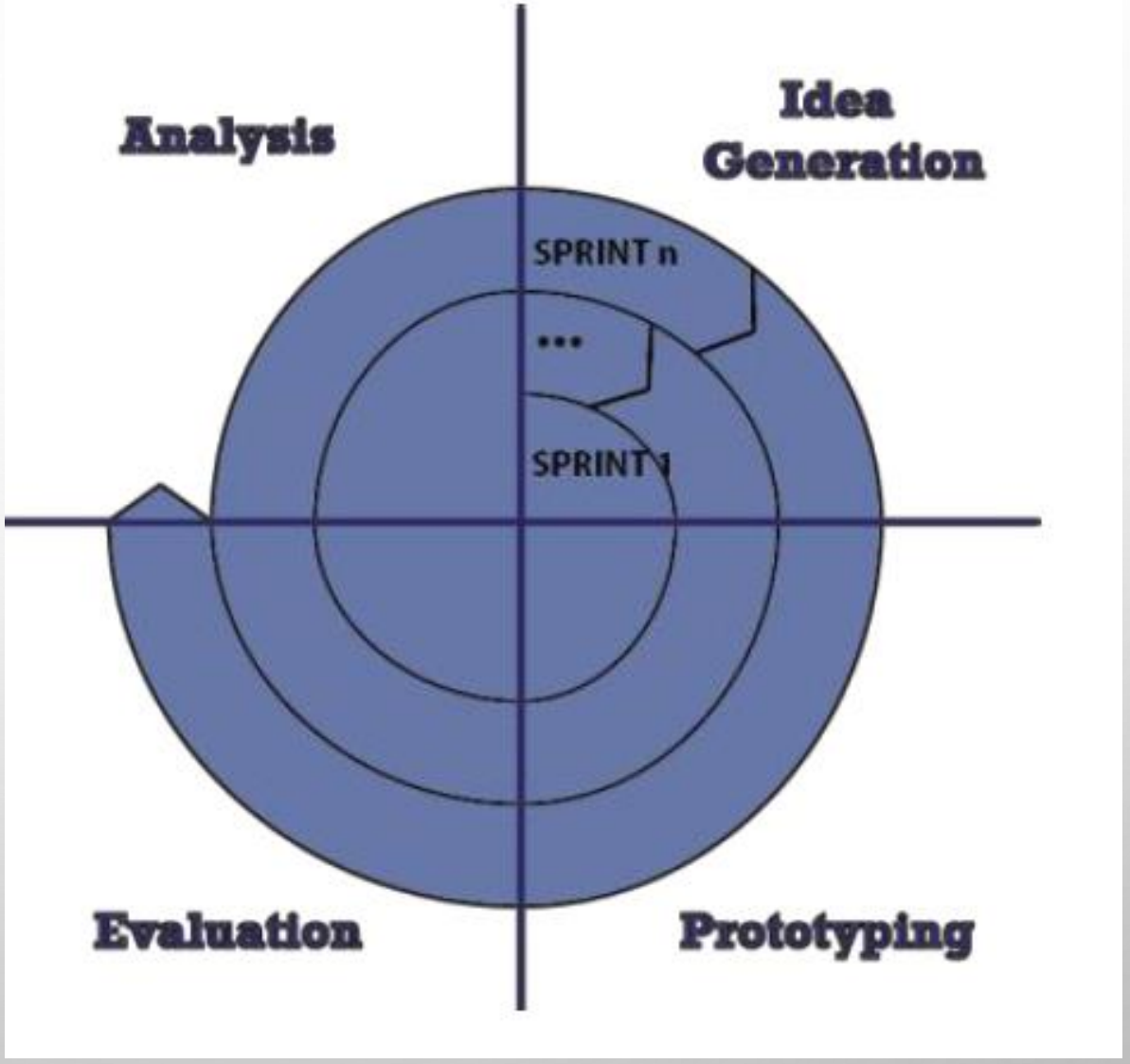


Barry W. Boehm. A Spiral Model of Software Development and Enhancement. IEEE Computer, 21(5), pp. 61-72, 1988.

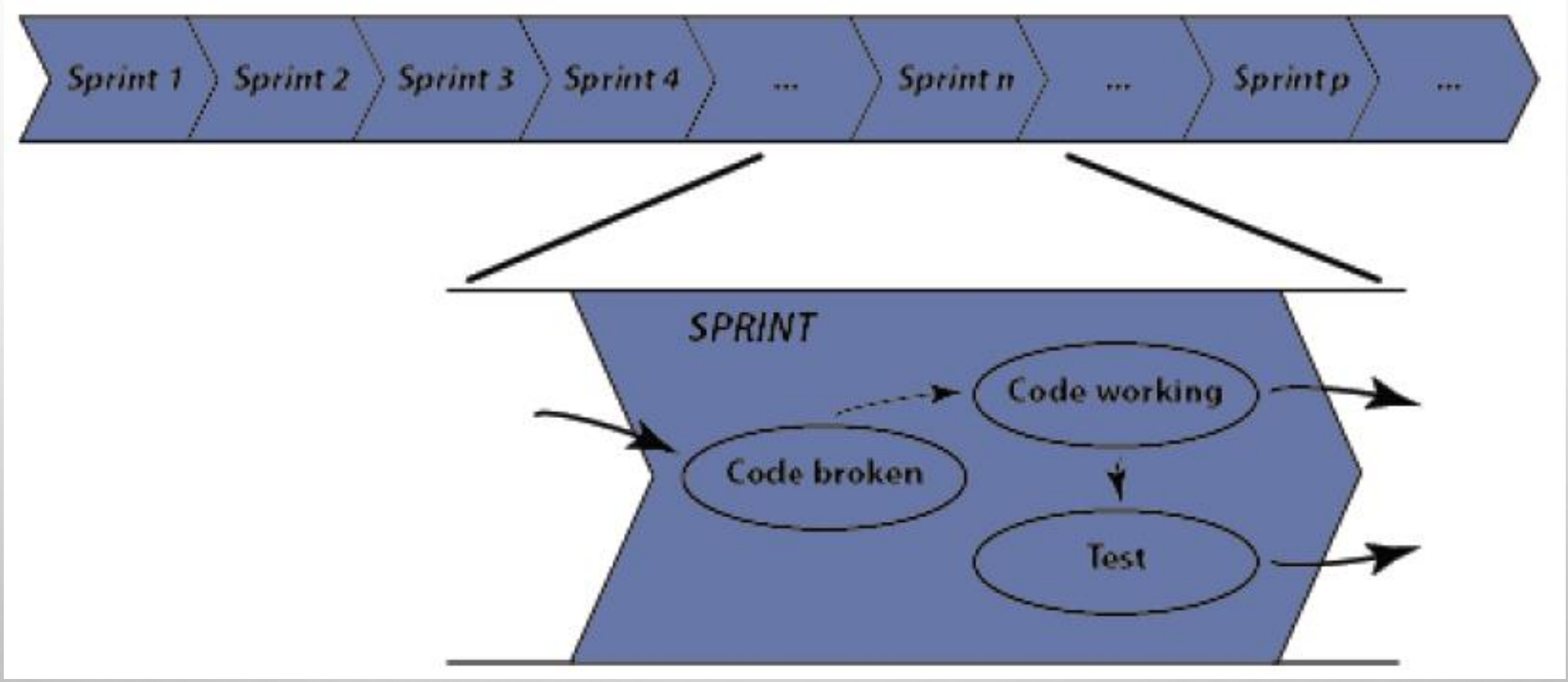


UML RUP Process

AGILE ET ITÉRATIF



AGILE PROCESSES – XTREME PROGRAMMING



K. Schwaber and M. Beedle, Agile Software Development with SCRUM. Upper Saddle River, NJ: Prentice-Hall, 2002.

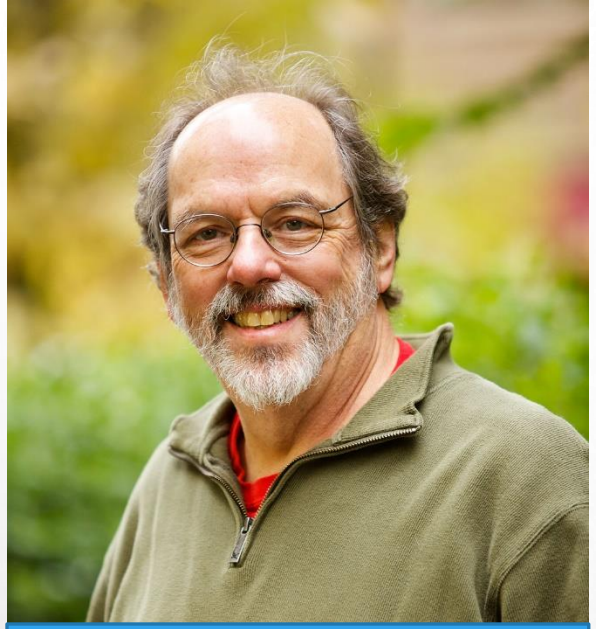
TECH DEBT – DIGRESSION IMPORTANTE

WHAT IS TECHNICAL DEBT?

Technical Debt results in decisions made to complete a feature quickly with or without regard for technically accepted principles or practices by engineering and product management (cause) and in turn makes future development or maintenance costlier (effect)

Deliver now but put off other stuff for later

Good Enough VS The Best



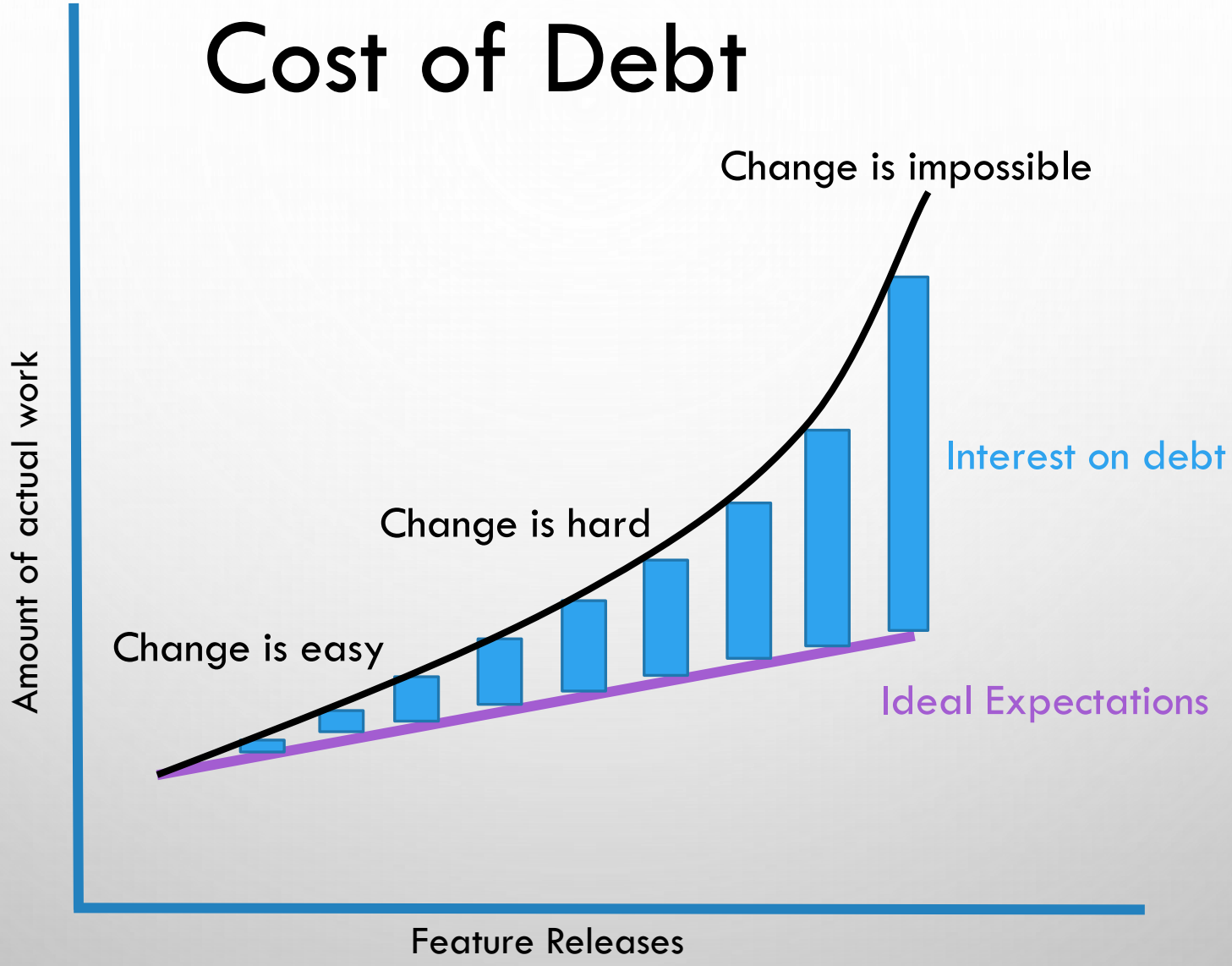
Cunningham's Law

"Shipping first time code is like going into debt.

A little debt speeds development so long as it is paid back promptly with a rewrite...

The danger occurs when the debt is not repaid."

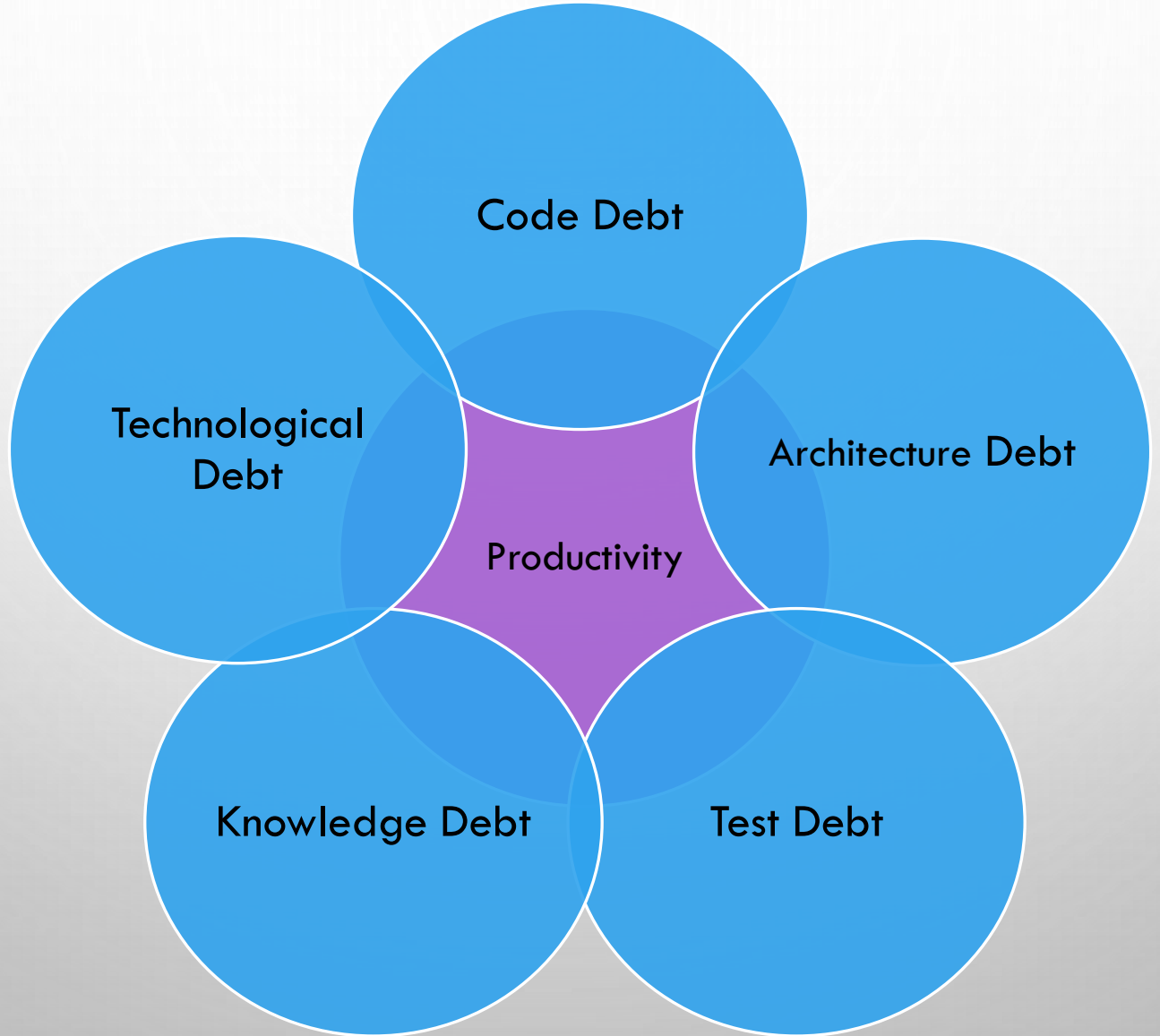
Cost of Debt


Change is impossible
Interest on debt
Change is hard
Change is easy
Ideal Expectations
Amount of actual work
Feature Releases

HOW TO MANAGE TECH DEBT

- Reflect
- Repress
- Repair
- Repay

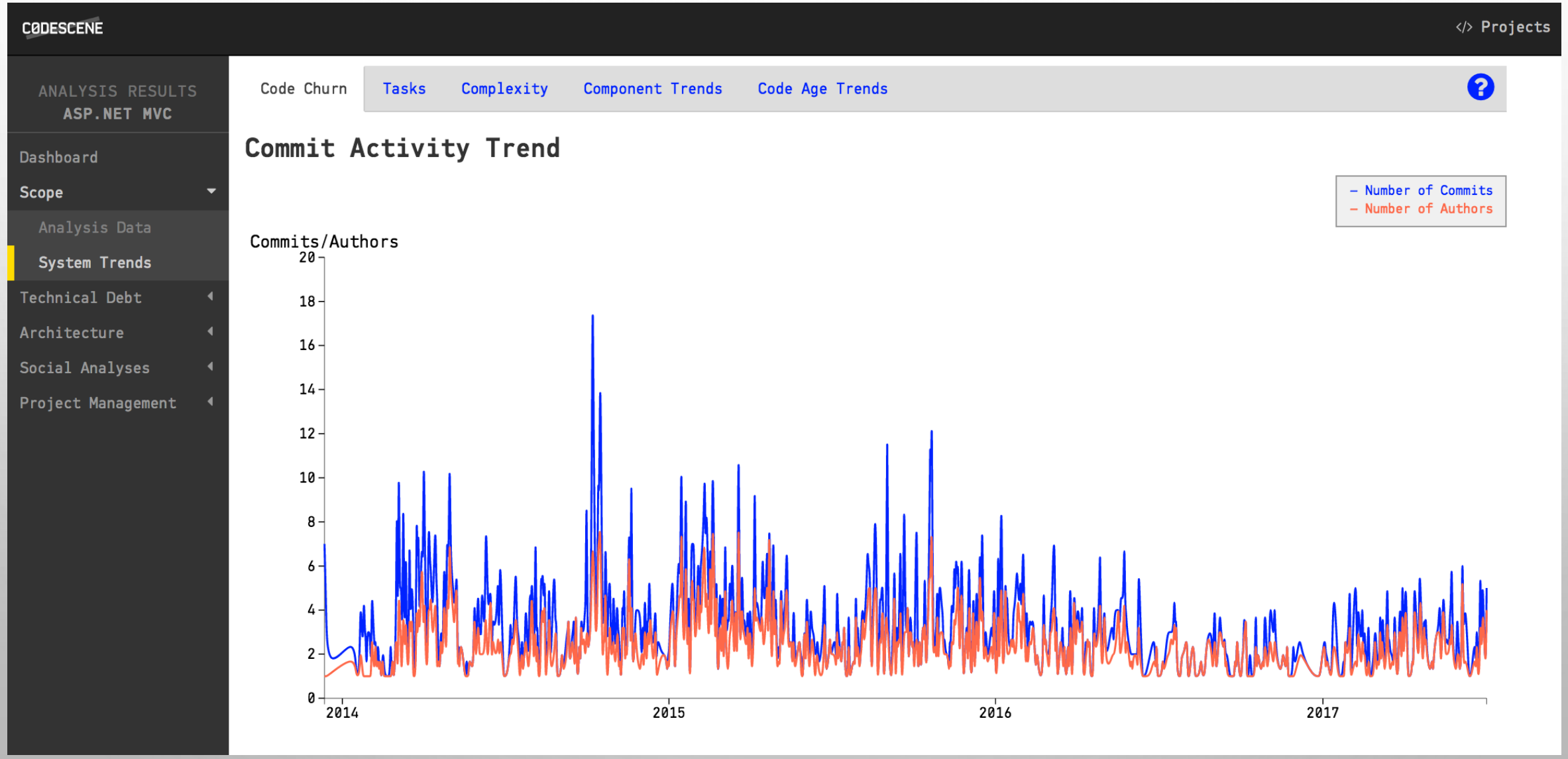
Reflect: 5 Categories on Technical Debt



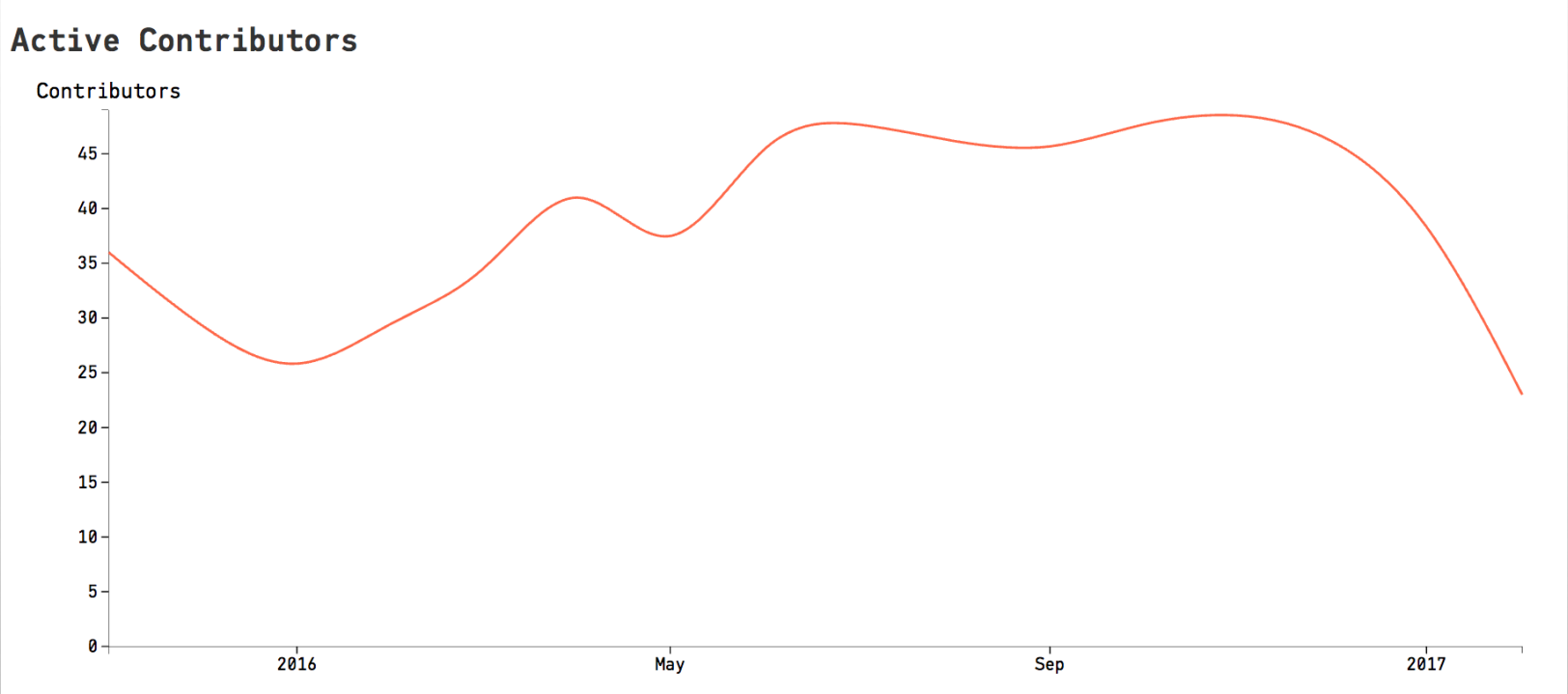
REFLECT: DEVELOPMENT

- TIME
  - Estimate vs Reality – Based on efforts did we meet the business objectives?
  - Velocity – Too perfect, margin of effort.
- CODE
  - LOC
  - Cyclomatic Complexity
- SOURCE CODE : CODE CHURN
- TEST
  - Defect Data
  - Coverage

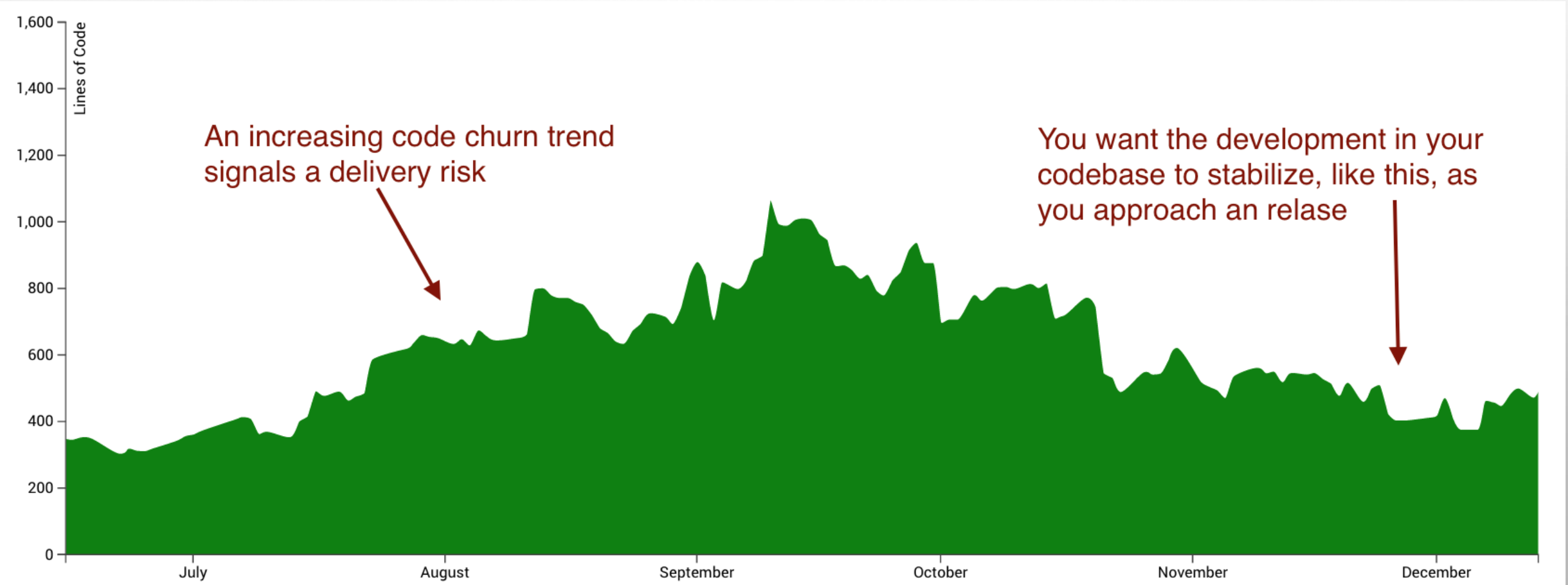
CODE CHURN


CODESCENE
Projects
ANALYSIS RESULTS
ASP.NET MVC
Dashboard
Scope
Analysis Data
System Trends
Technical Debt
Architecture
Social Analyses
Project Management
Code Churn
Tasks
Complexity
Component Trends
Code Age Trends
Commit Activity Trend
Number of Commits
Number of Authors
Commits/Authors
20
18
16
14
12
10
8
6
4
2
0
2014
2015
2016
2017

CODE CHURN


Active Contributors
Contributors
45
40
35
30
25
20
15
10
5
0
2016
May
Sep
2017

CODE CHURN


1,600
1,400
1,200
1,000
800
600
400
200
0
Lines of Code
An increasing code churn trend signals a delivery risk
You want the development in your codebase to stabilize, like this, as you approach an relase
July
August
September
October
November
December

REPRESS: MANAGE BEHAVIORS

- PRESSURE ON SCHEDULES MAKING COMMITMENTS ON BEHALF OF THE TEAM W/O CONSULTATION
  - Produced by:
    - Scope Creep – Features added without change in anything else.
    - Third-Party Estimates – leads give initial estimates and team held to that standard
    - Change in team makeup – Unmanaged Interrupts, full utilizations
    - Artifact of estimation process – not knowing the business need and not negotiable
    - Late integration – lead to surprises at unknown costs
  - Prevented by
    - Engaged business regularly (and users!!)
    - Knowledge shared freely – fostering sharing and collaboration
    - CI/CD (Continuous Integration/Continuous Delivery) for fast development feedback
- DUPLICATION
- URGE TO DO IT "RIGHT" THE FIRST TIME
- AVOID DARK SCRUM

REPRESS: MANAGE BEHAVIORS

- PRESSURE ON SCHEDULES MAKING COMMITMENTS ON BEHALF OF THE TEAM W/O CONSULTATION
- DUPLICATION - MAKE SYSTEM HARD TO MANAGE
  - Produced by
    - Lack of experience – new problem
    - Copy and Paste Programming – if it worked somewhere else, then
    - Conforming to poor design – take the style of other coders
    - Pressure to deliver – what is at stake is developers performance
  - Prevented by
    - Pair Programming
    - Don't Copy Paste Rule
    - Evolve Software Design
- URGE TO DO IT "RIGHT" THE FIRST TIME
- AVOID DARK SCRUM

REPRESS: MANAGE BEHAVIORS

- PRESSURE ON SCHEDULES MAKING COMMITMENTS ON BEHALF OF THE TEAM W/O CONSULTATION
- DUPLICATION - MAKE SYSTEM HARD TO MANAGE
- URGE TO DO IT "RIGHT" THE FIRST TIME – TOO MUCH TIME SPEND ON PLANNING AND DESIGN, MAKING FUTURE PROOF, AND SOLUTIONS CAN BE REUSED AND NO REWORK

  Prevented by Automation and Refactoring
- AVOID DARK SCRUM
  - Prevented by
    - Back to the basics – Reflect on agile principles (drop them when needed – extend them when needed)
    - Observe how others are doing it
    - Create or join a technical user group
    - Contribute to shared learning

REPAIR

- Cover it with tests then modify
- Extensible then extend
- Modularize then rewrite

- In a nutshell
  - Prerequisite – Create or Review your automation strategy. Seek consultation.
  - Adopt tools that help with fast feedback and increase flow
    - Source Control Management
    - Build Server
    - Test Harnesses and Unit Testing
    - Design Patterns and Code Stewardship
    - Migrate to Microservices or more modern architectures/infrastructures

REPAY

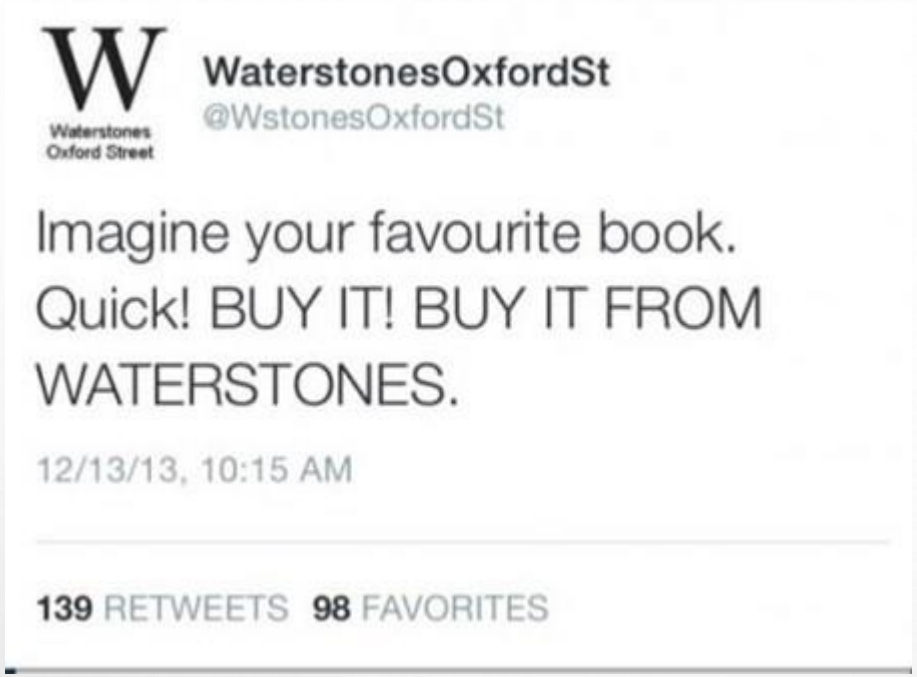
- Who will be responsible?
  - Everyone, but Engineers should make everyone care and describe the **hidden pressures** on the business
  - Prioritize work as enablers or value
  - Include in product roadmap and direction
  - Engage business and customers

REPAY (2)

1. Manage Conflict or Misunderstandings(Developer vs Manager)
   - Developer
     - "You don't care about maintainability of code"
     - "You don't give enough time"
   - Manager
     - "You just want to use the coolest tech"
     - "You don't care about the customer"
2. Balance Costs
   - Developer
     - **Time Lost** not paying interest right away or Cost of Delay
     - **Time Required to Repay** or Effort
   - Business
     - First to market ahead of competition
     - Perceived Reputation
     - Change direction or stick with current plan
3. Make It Visible

TECH DEBT DANS LA VRAIE VIE

- Design
- Development
- Maintenance
- Work organization


W
Waterstones
Oxford Street
WaterstonesOxfordSt
@WstonesOxfordSt
Imagine your favourite book.
Quick! BUY IT! BUY IT FROM
WATERSTONES.
12/13/13, 10:15 AM
139 RETWEETS 98 FAVORITES


W
Waterstones
Oxford Street
WaterstonesOxfordSt
@WstonesOxfordSt
Imagine your favourite book. Quick! BUY IT! BUY IT FROM WATERSTONES.
12/13/13, 10:15 AM
139 RETWEETS 98 FAVORITES
W
Waterstones
Oxford Street
WaterstonesOxfordSt
@WstonesOxfordSt
It turns out most of you already own a copy of your favourite book. We haven't thought this marketing campaign through.
12/13/13, 10:16 AM
198 RETWEETS 207 FAVORITES

WaterstonesOxfordSt
@WstonesOxfordSt

Imagine your favourite book. Quick! BUY IT! BUY IT FROM WATERSTONES.

12/13/13, 10:15 AM

139 RETWEETS **98** FAVORITES

WaterstonesOxfordSt
@WstonesOxfordSt

It turns out most of you already own a copy of your favourite book. We haven't thought this marketing campaign through.

12/13/13, 10:16 AM

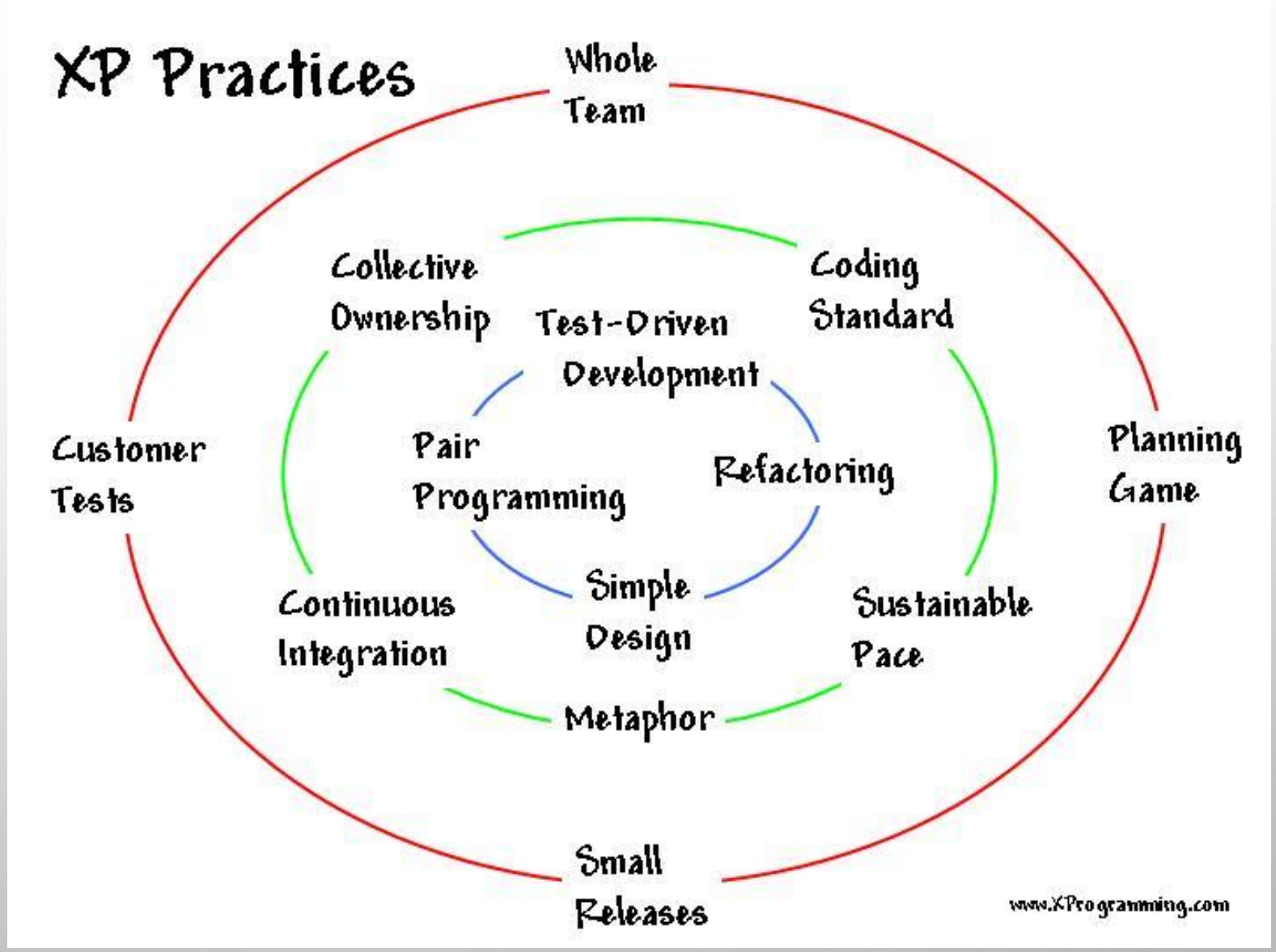
198 RETWEETS **207** FAVORITES

PRECURSORS FOR HUMAN ERROR (AND FOR ACCIDENTS OR

PRECUR ND FOR ACCIDENTS OR

PRECURSOR DENTS OR

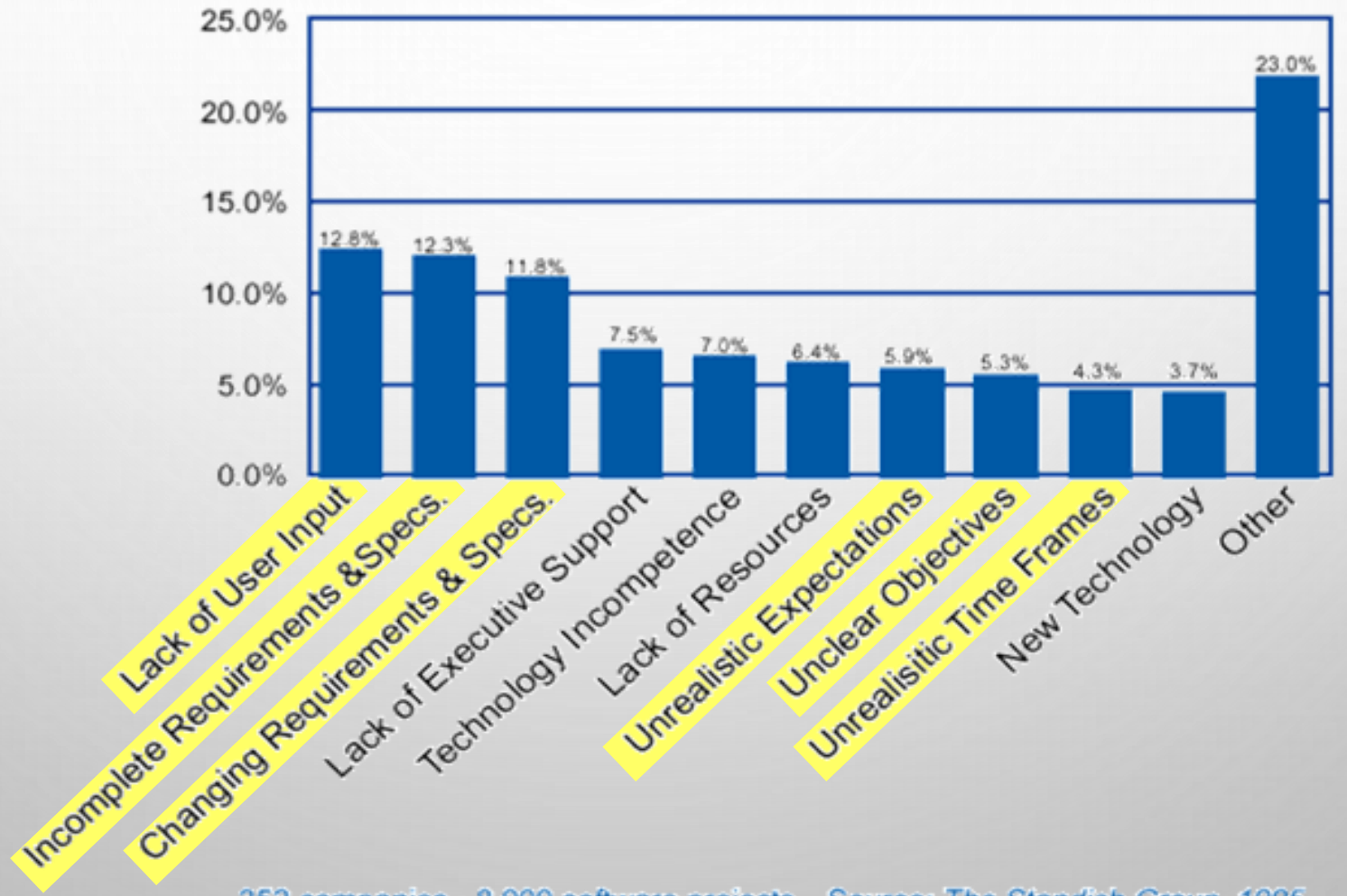
EXTREME PROGRAMMING



Kent Beck. Extreme Programing Explained: First Edition September 29, 1999. ISBN: 0201616416, 224 pages.

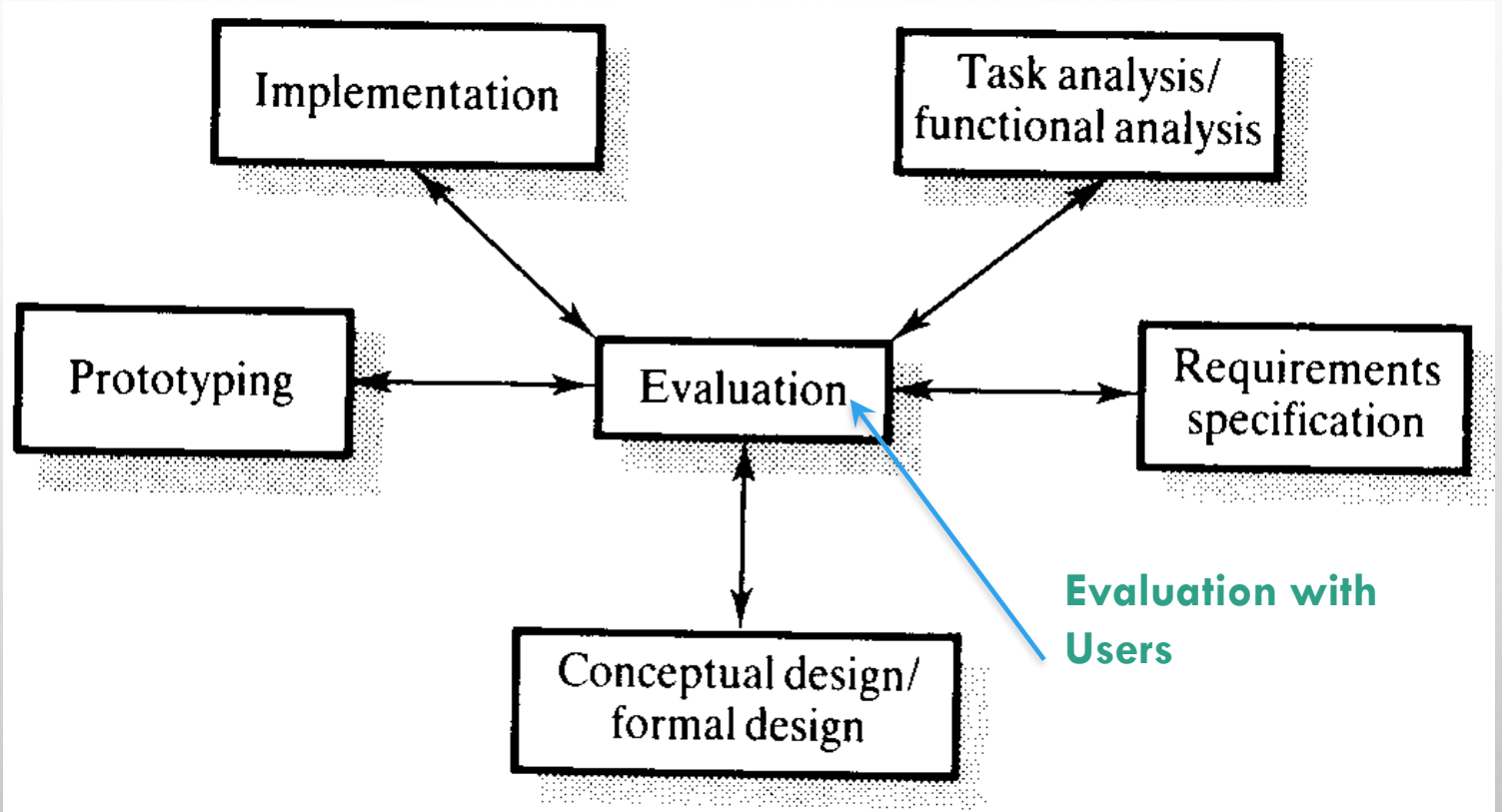
WHY SOFTWARE PROJECTS FAIL

(source Boehm 2006 – invited Talk IEEE ICSE 2006)
Average overrun: 89.9% on cost, 121% on schedule, with 61% of content



BESOINS ET CAFARDS

CYCLE DE VIE EN ÉTOILE



The star life cycle (adapted from Hix and Hartson, 1993).

MÉTHODOLOGIE DE CONCEPTION

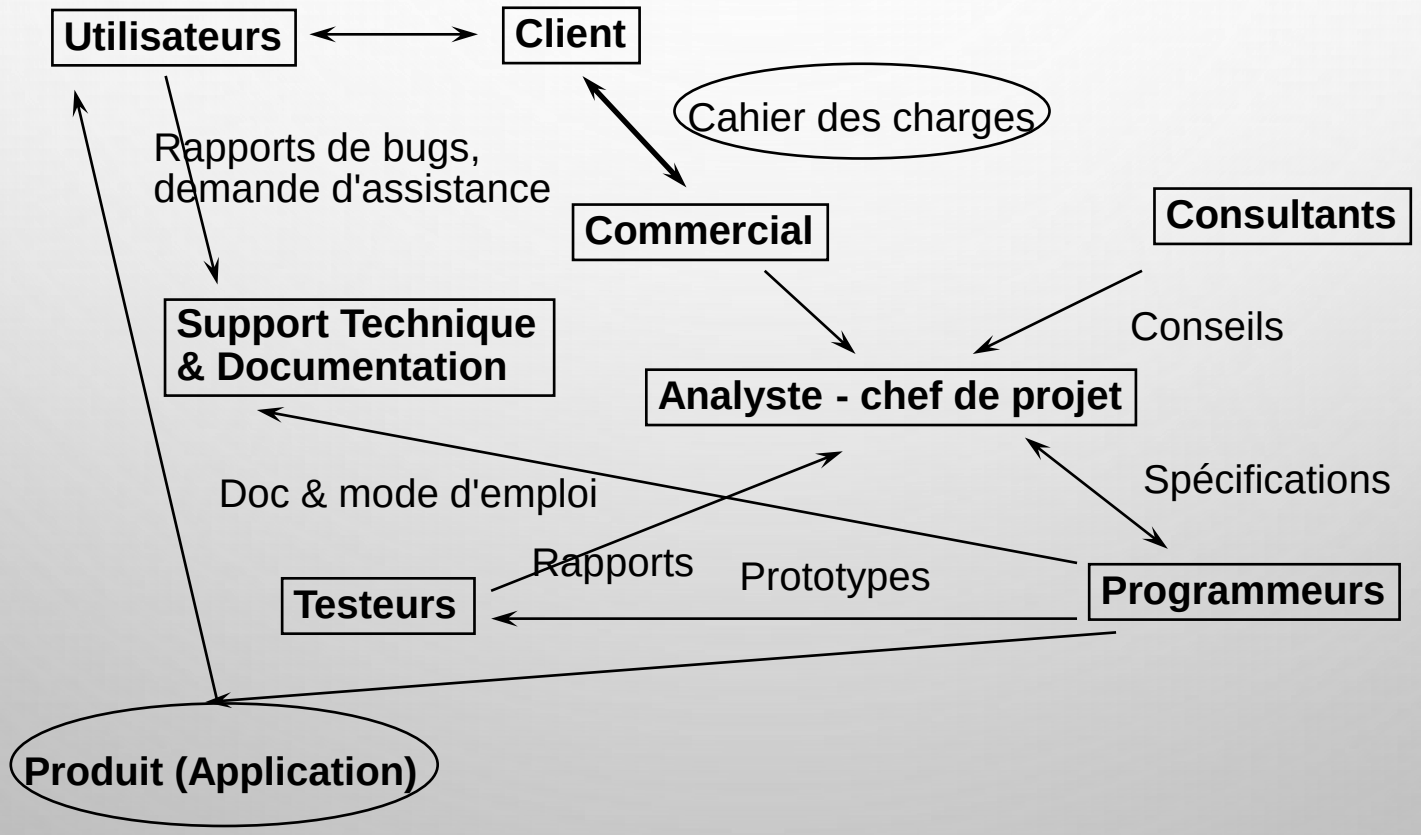
Systèmes interactifs nécessitent
- des connaissances et compétences pluridisciplinaire
- une science non exacte impliquant fortement l'humain
- une intégration matériel et logiciel
    -> ça va être difficile et intéressant

Les systèmes interactifs sont différents des autres systèmes informatiques

Il faut donc des méthodes et des outils différents de ce qui est utilisé pour les autres

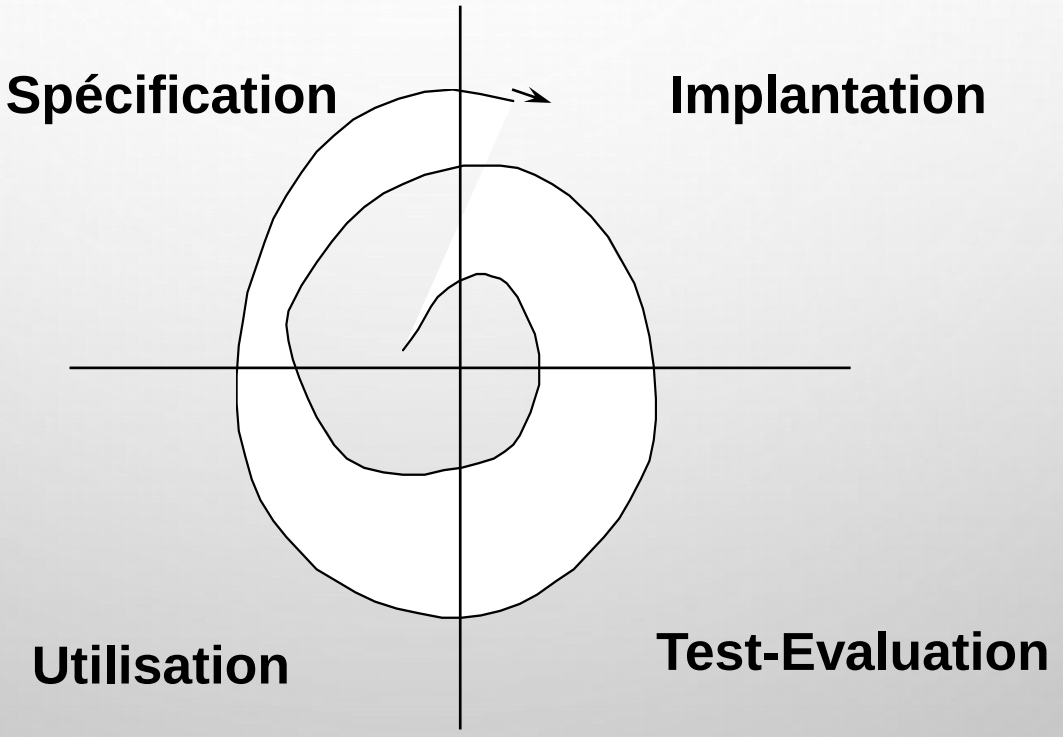
CONCEPTION DANS LA PRATIQUE – UN GROS DÉCALAGE ENTRE L'INDUSTRIE ET LA RECHERCHE/ENSEIGNEMENT

Circuits de l'information beaucoup plus complexes



LES APPROCHES ITÉRATIVES

Évolution "En spirale" vers le produit final



PROTOTYPAGE – MAQUETTAGE (VOIR COURS PROTOTYPAGE)

Prototype : diffère du produit final par le processus de conception

Maquette : diffère du produit par l'échelle (taille, nombre de fonctionnalités, …)

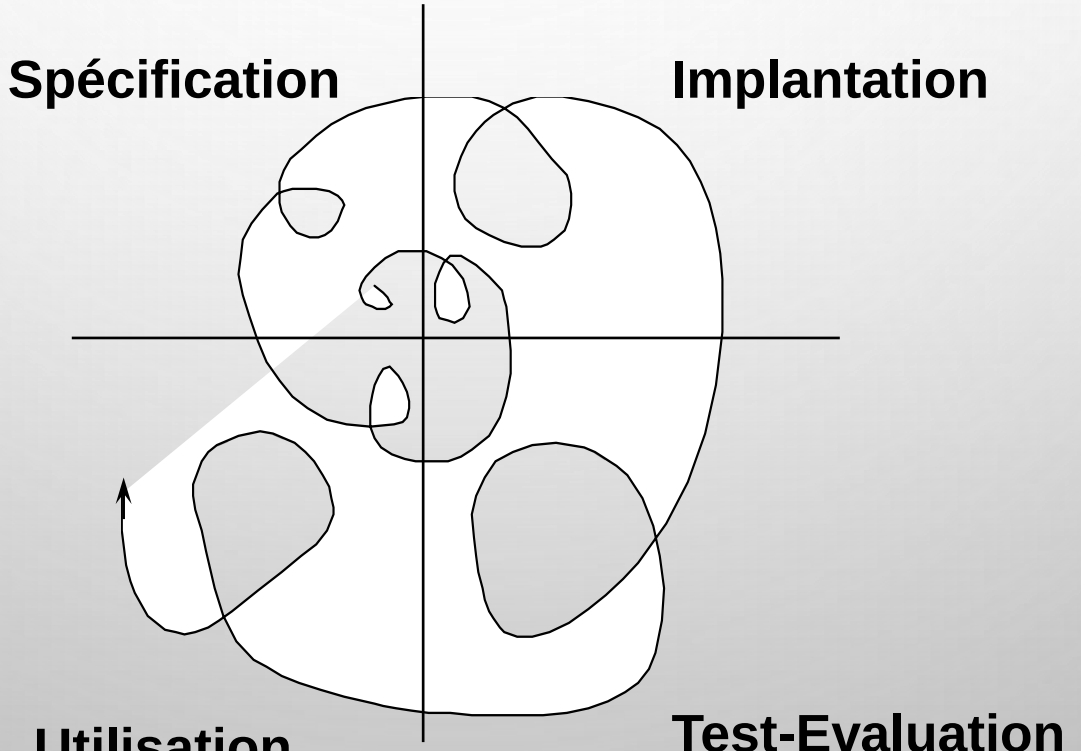
En IHM le sens de ces mots a été altéré

Prototype : produit qui fonctionne (des parties de chacune des couches du modèle de Seeheim ont été développées)

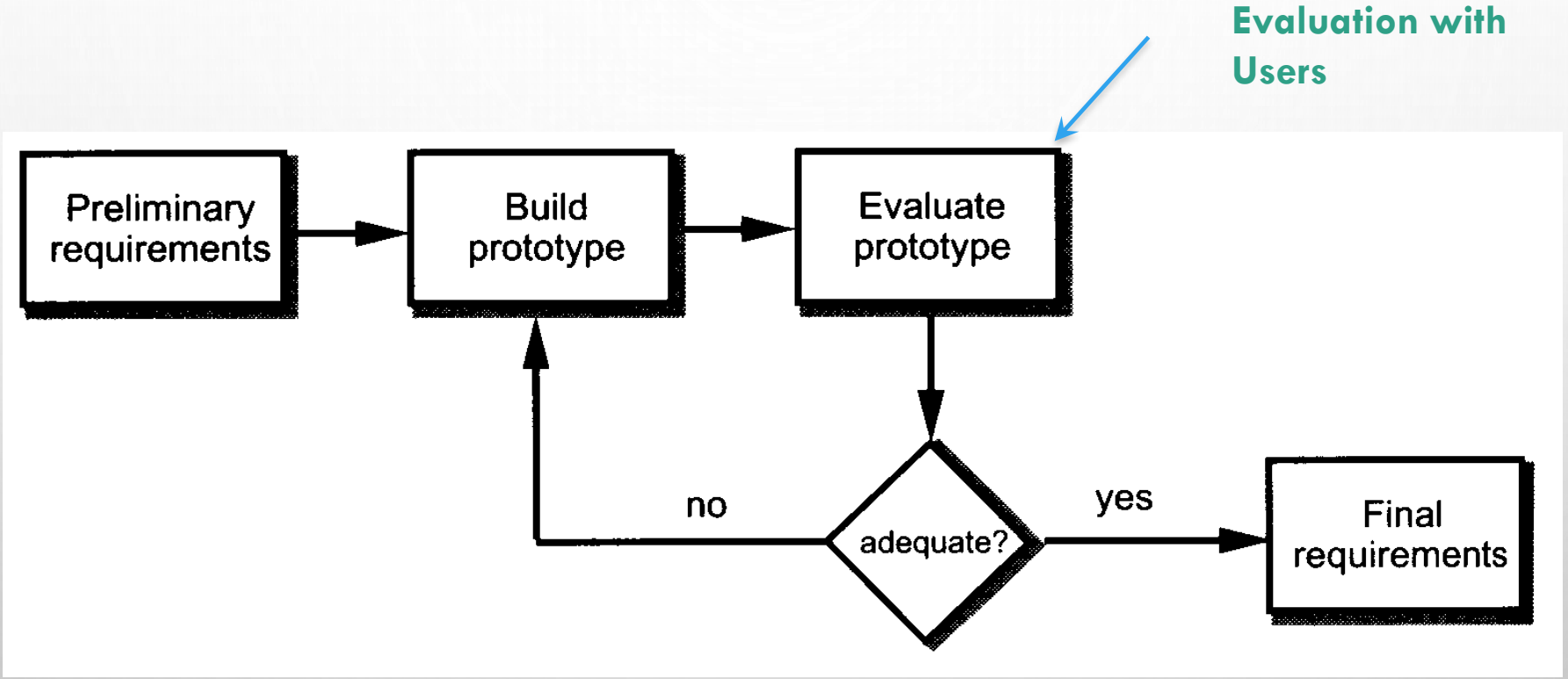
Maquette : l'ensemble de la partie présentation a été réalisée mais les fonctionnalités ne sont pas mises en œuvre (on voit la statique de l'interface mais pas la dynamique)

APPROCHES "SUPER-ITÉRATIVES"

Réaliser des itérations à chaque étape du processus:

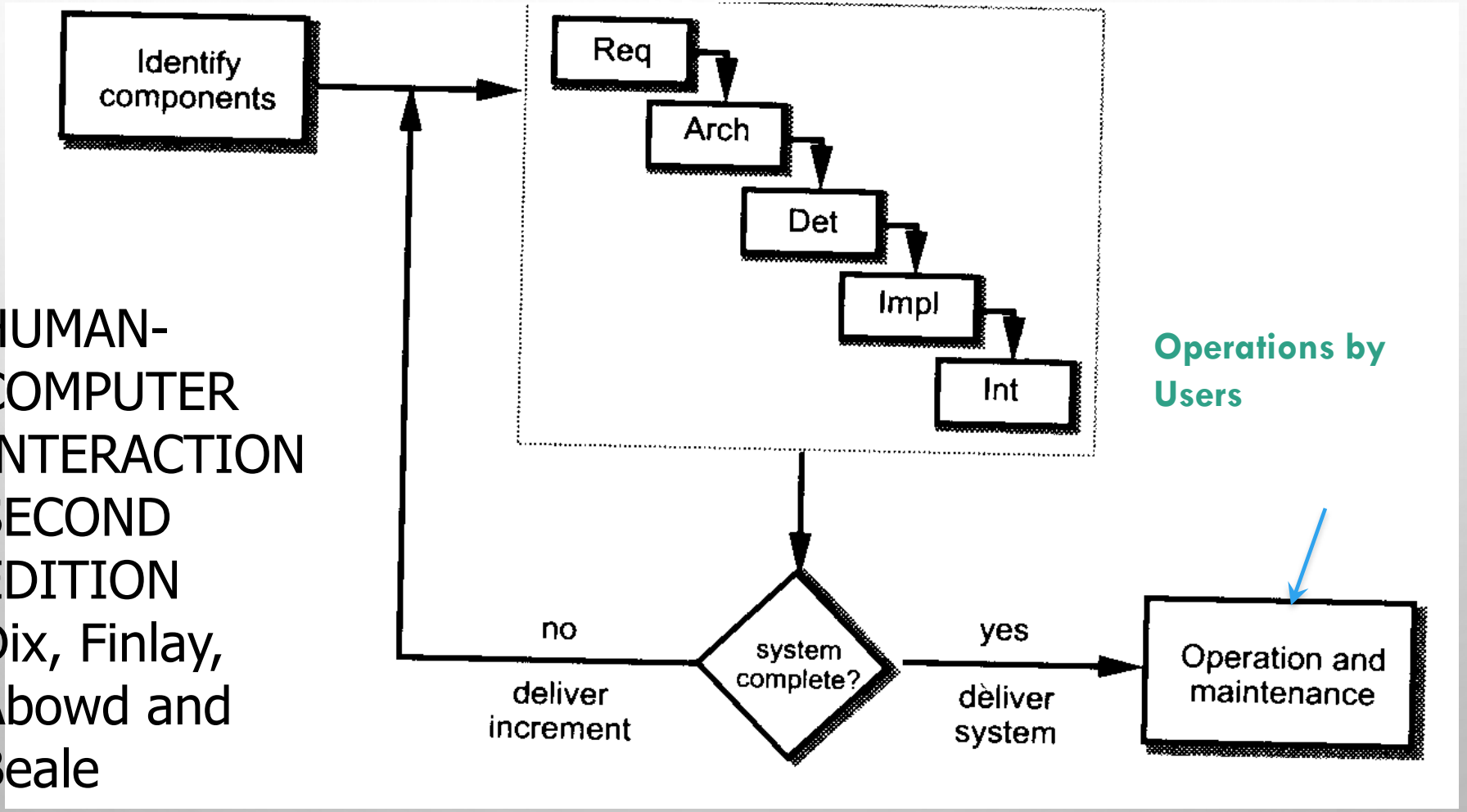


PROTOTYPAGE JETABLE



HUMAN-COMPUTER INTERACTION SECOND EDITION
Dix, Finlay, Abowd and Beale

PROTOTYPAGE INCRÉMENTAL



HUMAN-COMPUTER INTERACTION SECOND EDITION
Dix, Finlay, Abowd and Beale

PROTOTYPAGE BASSE FIDÉLITÉ

Partir de schémas bruts et très simples qui exposent
uniquement les problèmes importants puis raffinements progressifs
Dessins et maquettes "manuelles" suggérant le mode de fonctionnement
sans rentrer dans les détails de l'interface, distrayants.

+ efficace et + abstrait qu'un prototype

Interaction constante avec les futurs utilisateurs

Faire intervenir à ce moment là seulement les consultants externes:
graphistes, ergonomes, spécialistes du domaine…

Envisager des **Cours de dessin, d'expression graphique.**

PROTOTYPAGE HAUTE FIDÉLITÉ

-> réduire le nombre de boucles pour aller au plus vite vers l'application finale

Utilisation des outils interactifs de développement.

Inconvénients :
- lenteur des boucles d'itération
- difficultés à dégager clairement à chaque étape les problèmes essentiels:
-> la réalisation du prototype fait intervenir presque en même temps les 4 phases
- Les outils de prototypages imposent leur limites, parfois assez importantes

Early design

Brainstorm different representations

Choose one or two representations

Rough out interface style

Low fidelity paper prototypes

Task centered walkthrough and redesign
Behaviour (dialogue & interaction) modelling

Fine tune interface, screen design

Heuristic evaluation and redesign

Usability testing and redesign

Medium fidelity prototypes

(Re) development

Limited field testing
Alpha/Beta tests

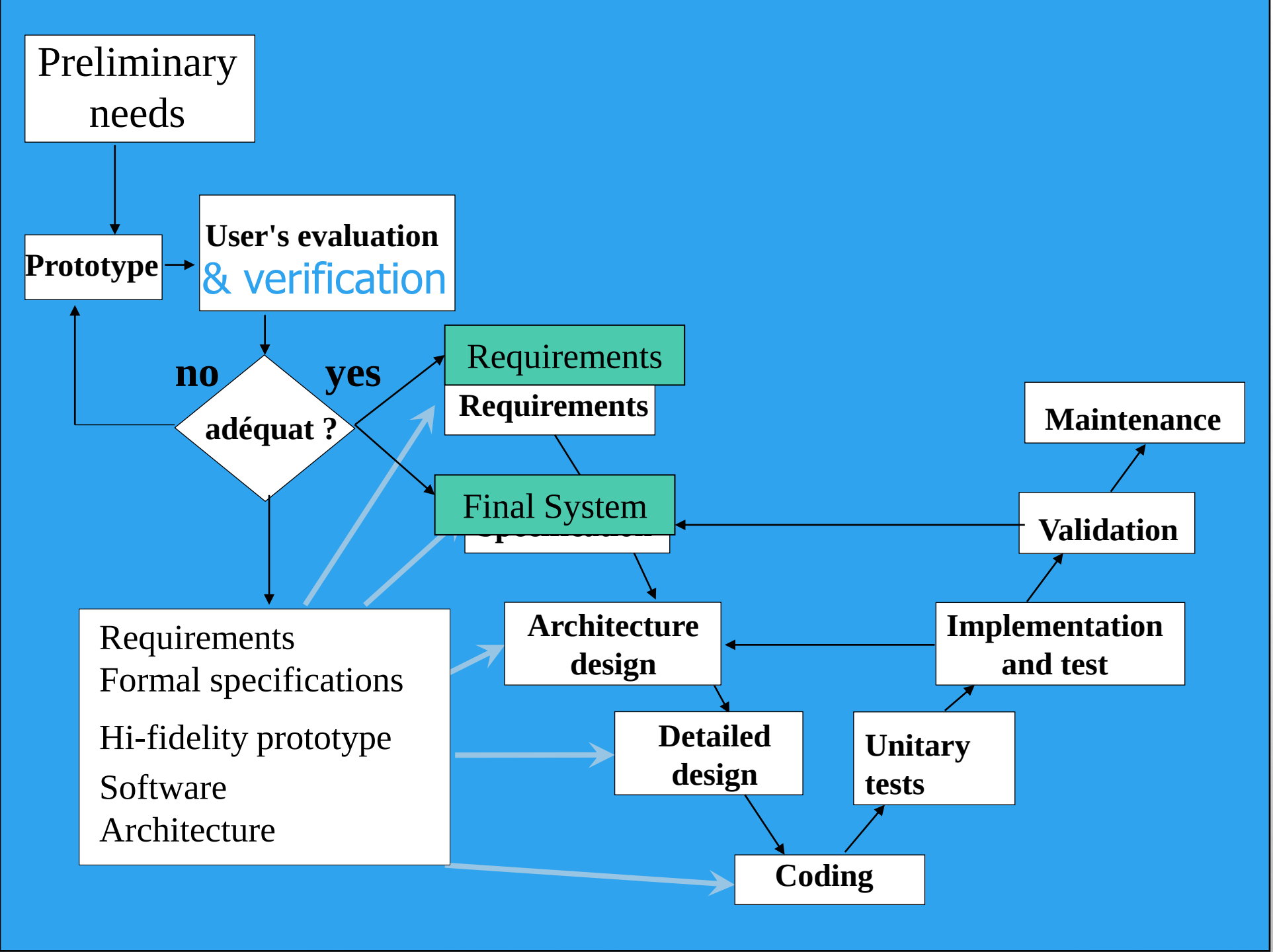
Working systems

High fidelity prototypes

Late design

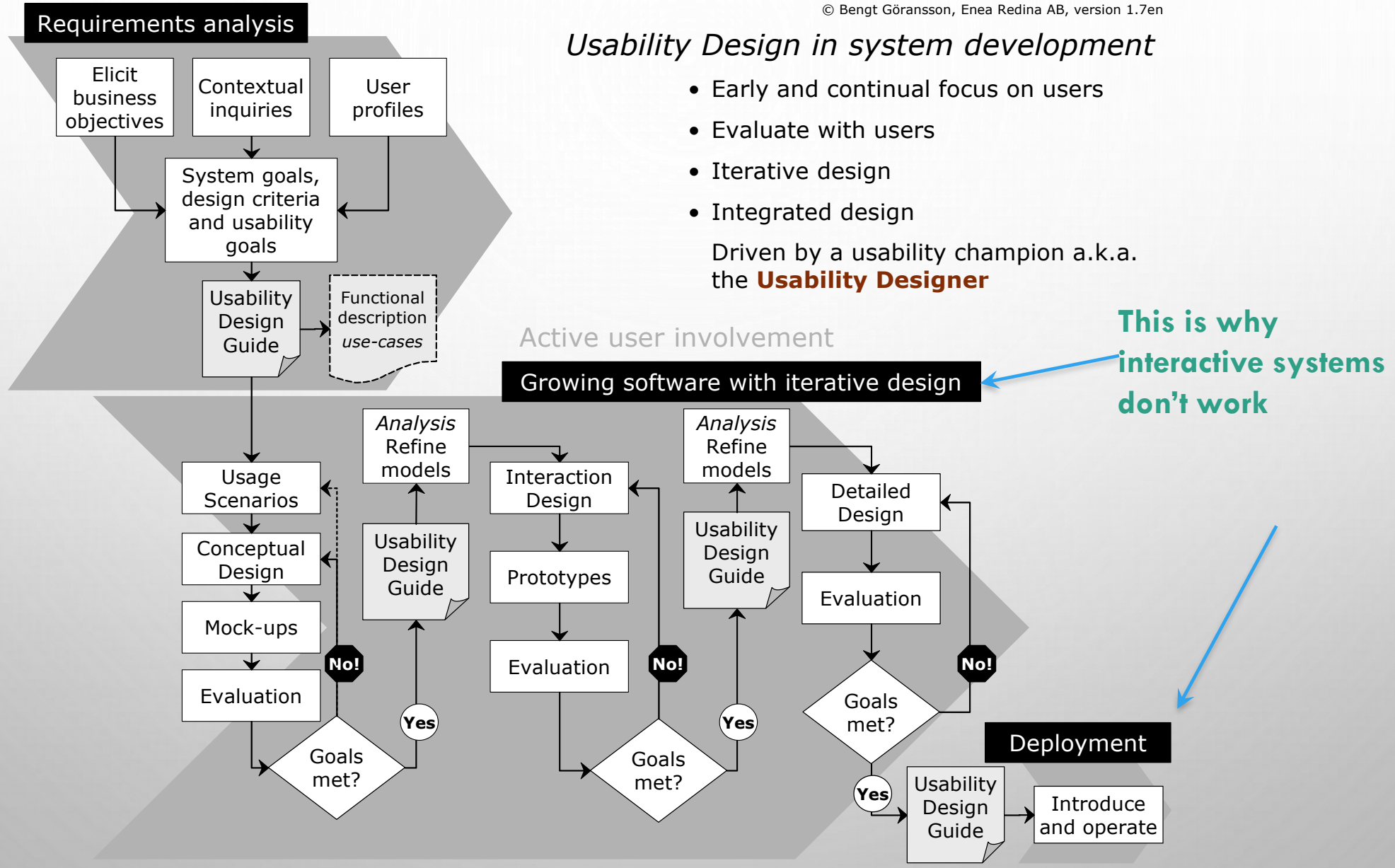
PEUT-ON INTÉGRER PROTOTYPAGE ET APPROCHES FORMELLES ?

- Approches itératives
- Approches en V ou en cascade



R. Bastide, D. Navarre, P. Palanque: A tool-supported design framework for safety critical interactive systems. Interacting with Computers 15(3): 309-328 (2003)

USABILITY DESIGN PROCESS



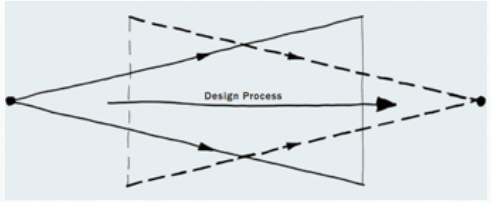
Usability Design in system development

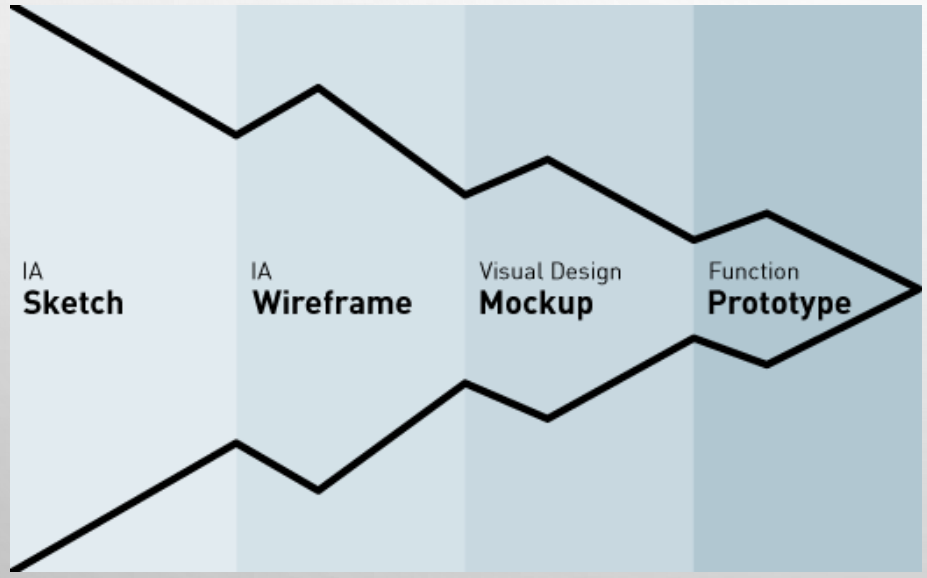
- Early and continual focus on users
- Evaluate with users
- Iterative design
- Integrated design

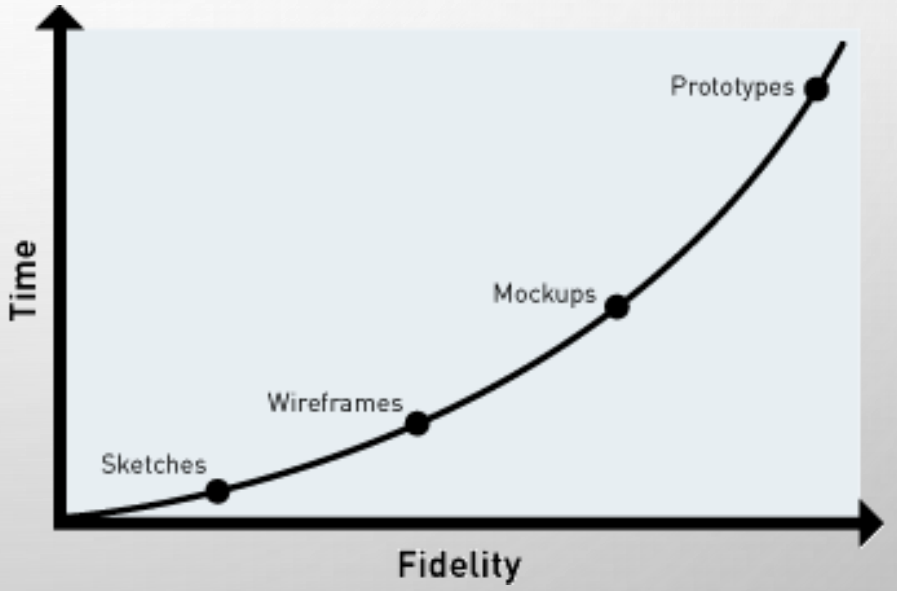
Driven by a usability champion a.k.a. the **Usability Designer**

This is why interactive systems don't work

USABILITY AND UX DESIGN PROCESS







B. Buxton. 2007. Sketching User Experiences : Getting the design right and the right design. M. Kaufmann

USABILITY AND UX DESIGN PROCESS

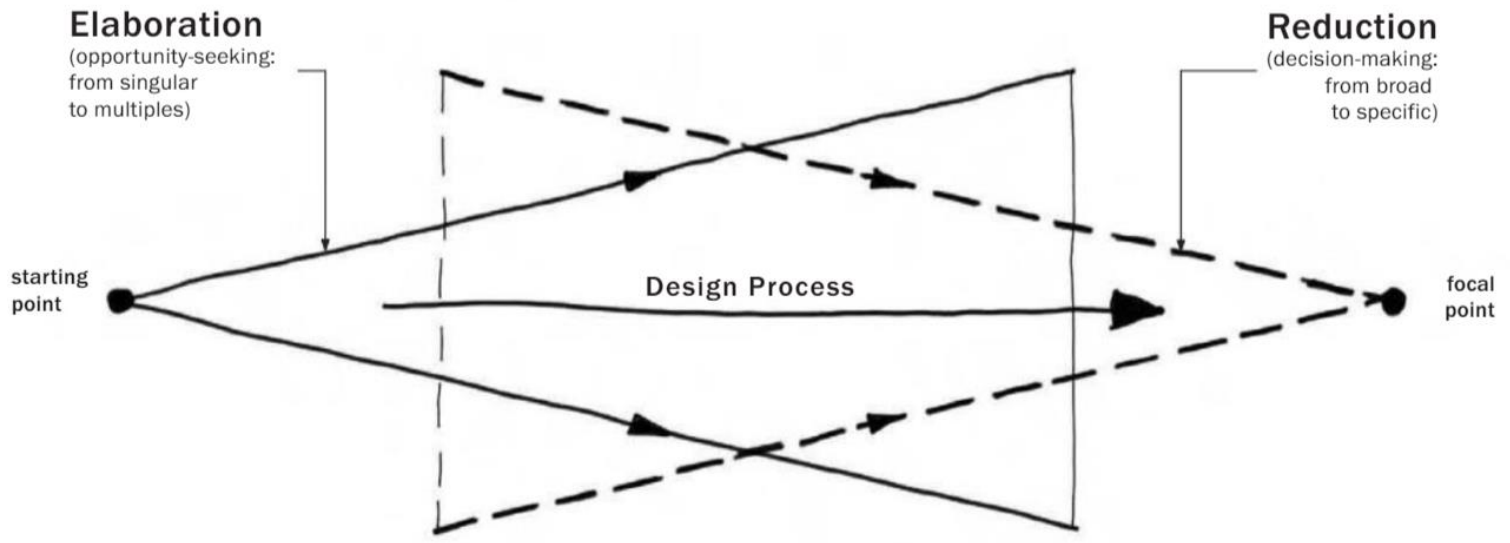


Figure 53: Overlapping Funnels

The reduction that results from decision making is balanced by the constant generation of new ideas and creativity that open up new opportunities to improve the design.

Source: Laseau 1980; p. 91

USABILITY AND UX DESIGN PROCESS

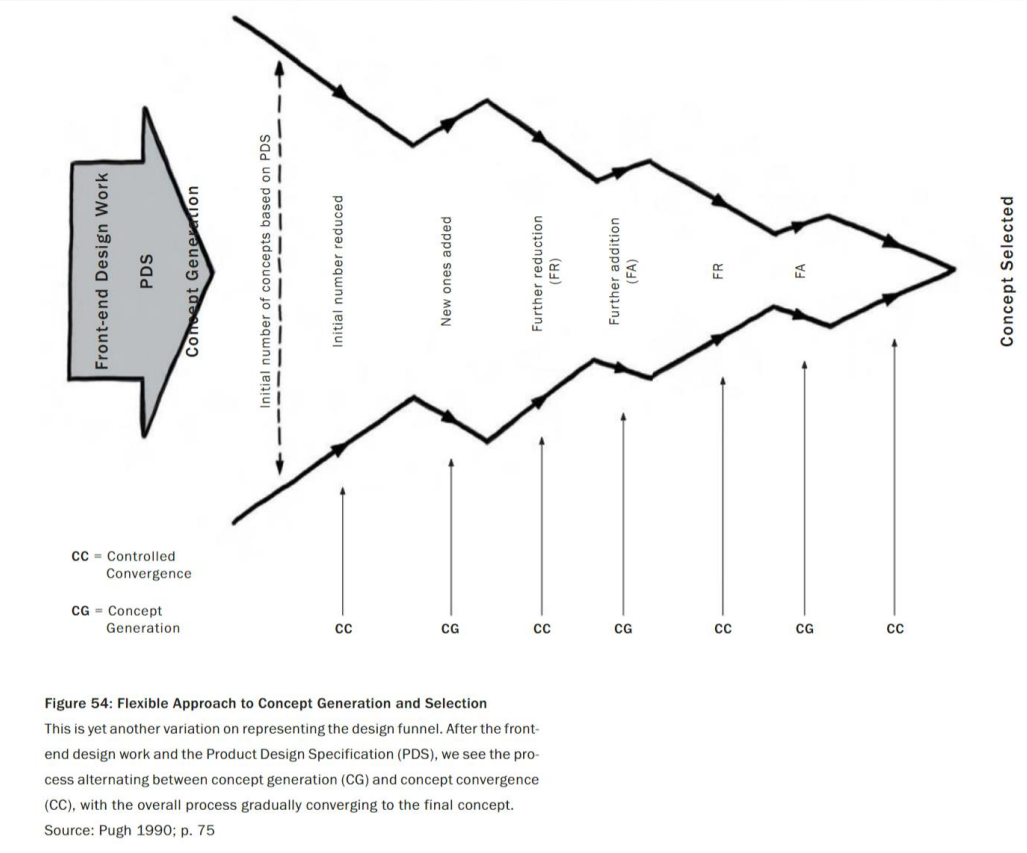
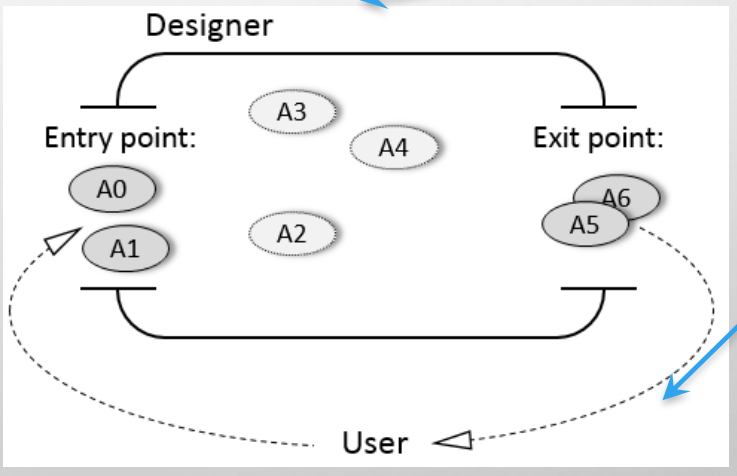


Figure 54: Flexible Approach to Concept Generation and Selection
This is yet another variation on representing the design funnel. After the front-end design work and the Product Design Specification (PDS), we see the process alternating between concept generation (CG) and concept convergence (CC), with the overall process gradually converging to the final concept.
Source: Pugh 1990; p. 75

Laseau's overlapping funnels

This is why interactive systems don't work

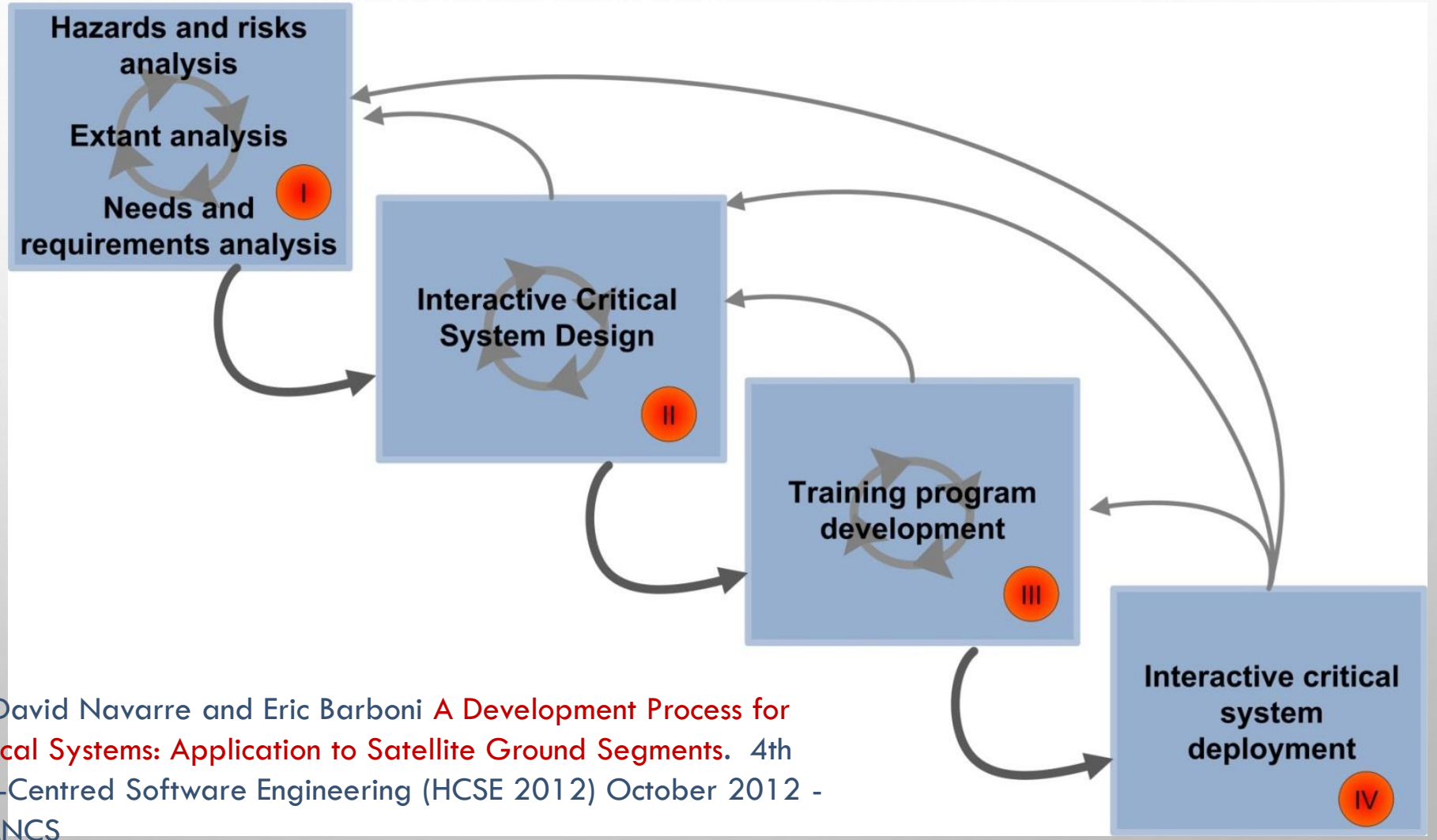


B. Buxton. 2007. Sketching User Experiences : Getting the design right and the right design. M. Kaufmann

TRAINING – FORMER LES UTILISATEURS

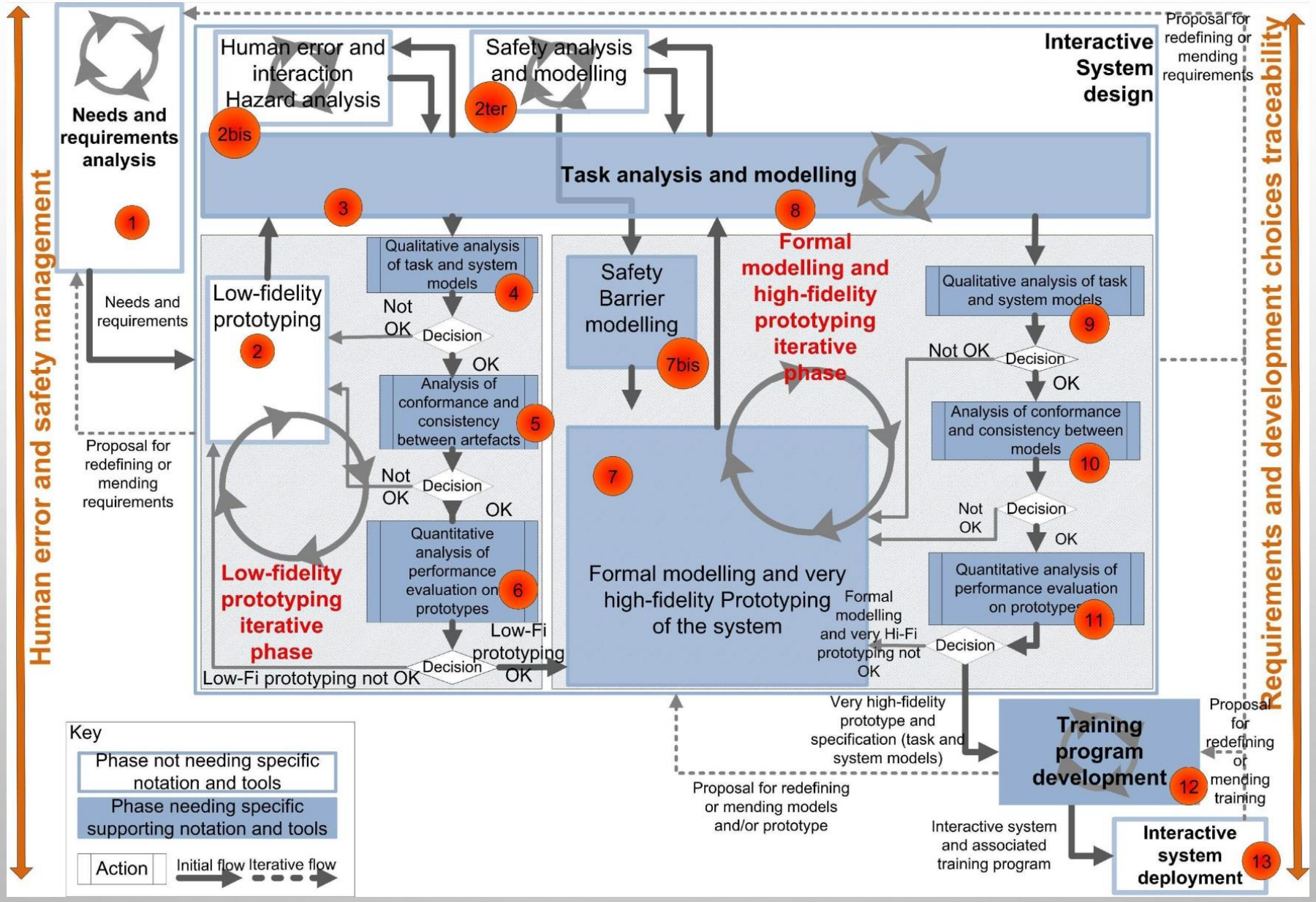
OVERVIEW OF THE DESIGN-AND-DEVELOPMENT PROCESS

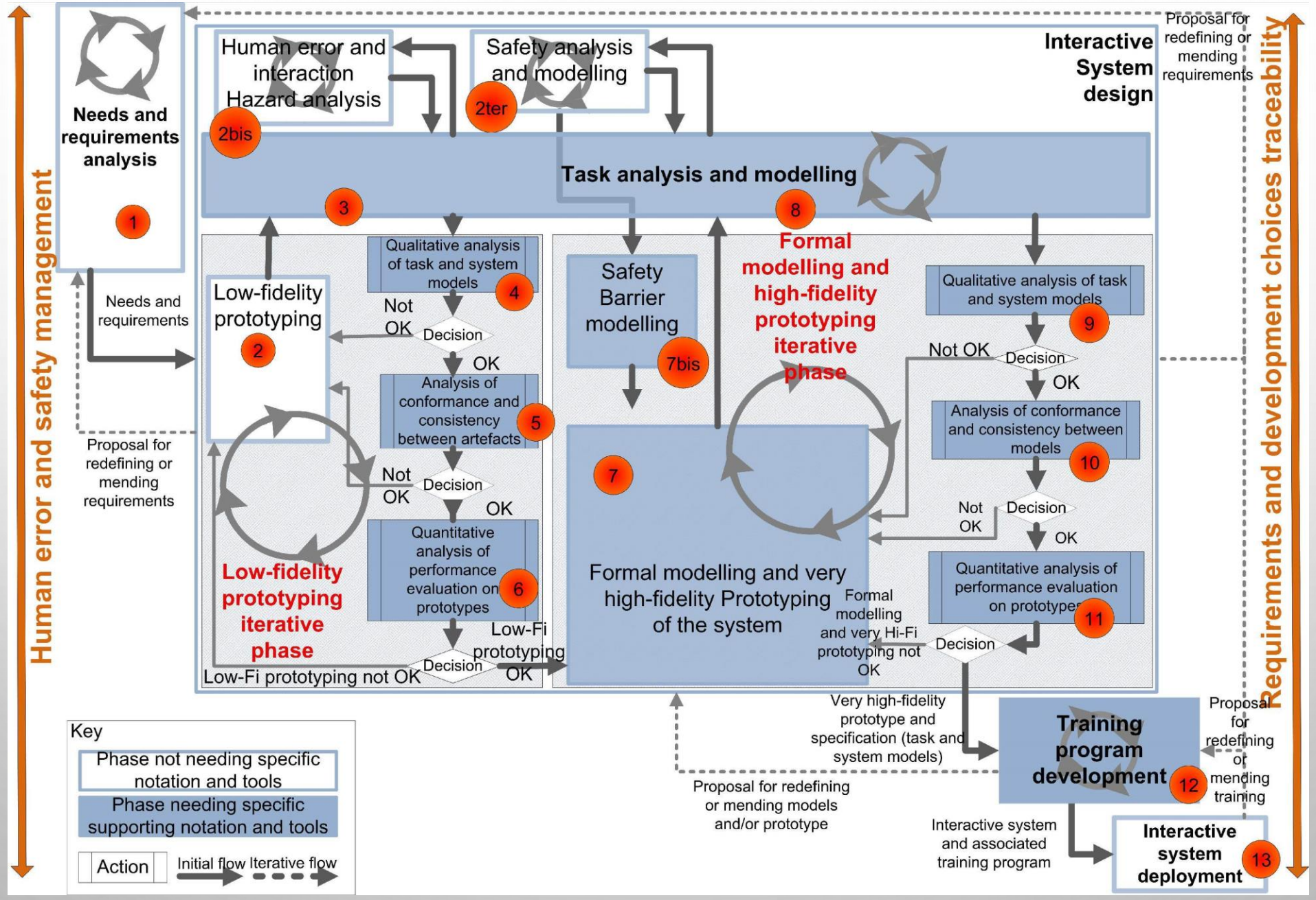
- Training & Human Error explicit



Célia Martinie, Philippe Palanque, David Navarre and Eric Barboni A Development Process for Usable Large Scale Interactive Critical Systems: Application to Satellite Ground Segments. 4th International Conference on Human-Centred Software Engineering (HCSE 2012) October 2012 - Toulouse, France, Springer Verlag, LNCS

DESIGN-AND-DEVELOPMENT PHASE DETAILED




Human error and safety management
Requirements and development choices traceability
Needs and requirements analysis
1
Human error and interaction Hazard analysis
2bis
Safety analysis and modelling
2ter
Interactive System design
Proposal for redefining or mending requirements
Task analysis and modelling
3
8
Needs and requirements
Proposal for redefining or mending requirements
Low-fidelity prototyping
2
Qualitative analysis of task and system models
4
Not OK
Decision
OK
Analysis of conformance and consistency between artefacts
5
Not OK
Decision
OK
Quantitative analysis of performance evaluation on prototypes
6
Low-fidelity prototyping iterative phase
Low-Fi prototyping not OK
Decision
Low-Fi prototyping OK
Safety Barrier modelling
7bis
Formal modelling and high-fidelity prototyping iterative phase
7
Formal modelling and very high-fidelity Prototyping of the system
Qualitative analysis of task and system models
9
Not OK
Decision
OK
Analysis of conformance and consistency between models
10
Not OK
Decision
OK
Quantitative analysis of performance evaluation on prototypes
11
Formal modelling and very Hi-Fi prototyping not OK
Decision
Very high-fidelity prototype and specification (task and system models)
Proposal for redefining or mending models and/or prototype
Training program development
12
Proposal for redefining or mending training
Interactive system and associated training program
Interactive system deployment
13
Key
Phase not needing specific notation and tools
Phase needing specific supporting notation and tools
Action
Initial flow
Iterative flow

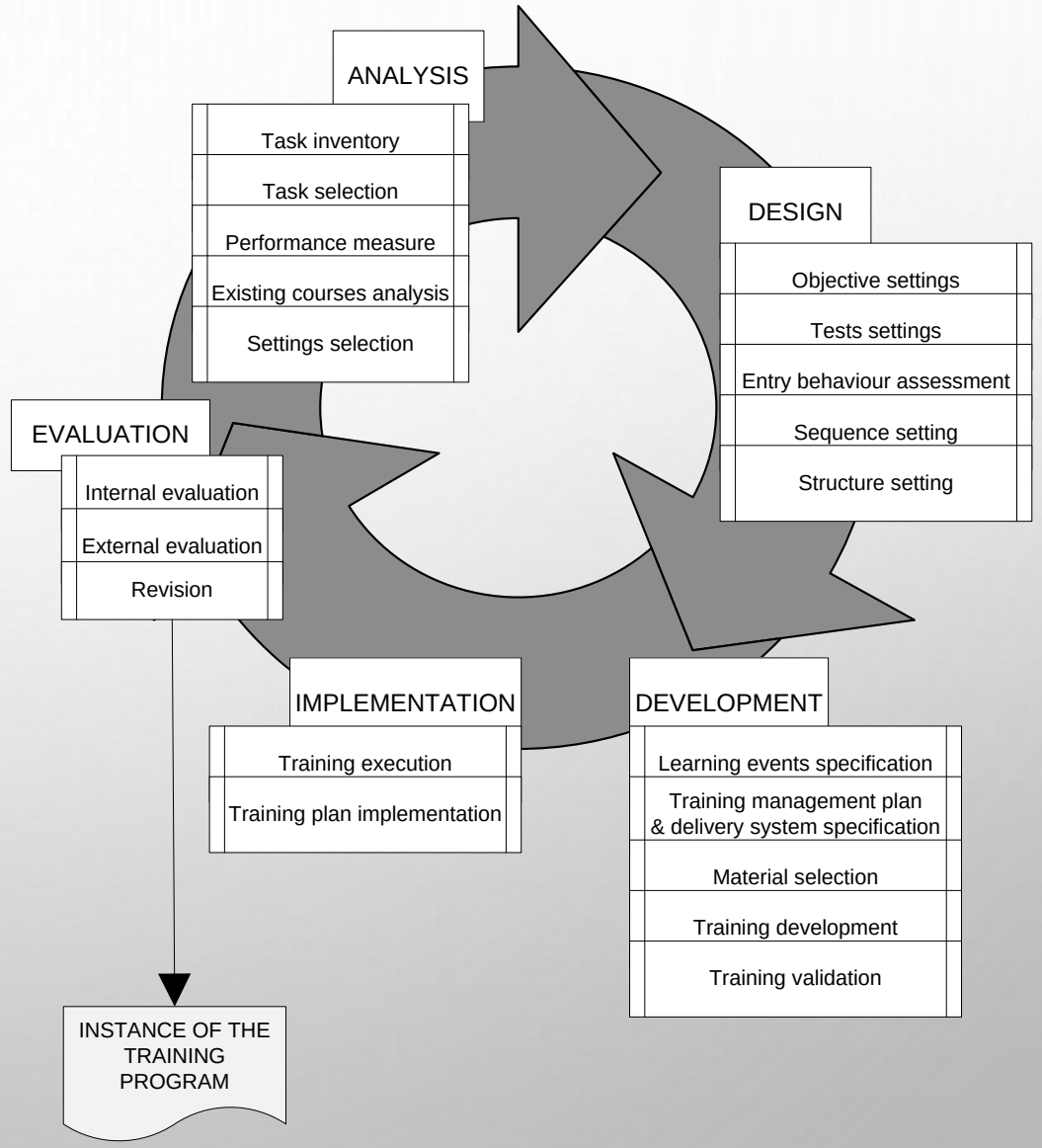
ADDIE

Clark, D. R., ISD Concept Map, retrieved January 2010 from http://nwlink.com/~donclark/hrd/ahold/isdmap.jpg

Branson et al., IPISD, Florida State University, 1975

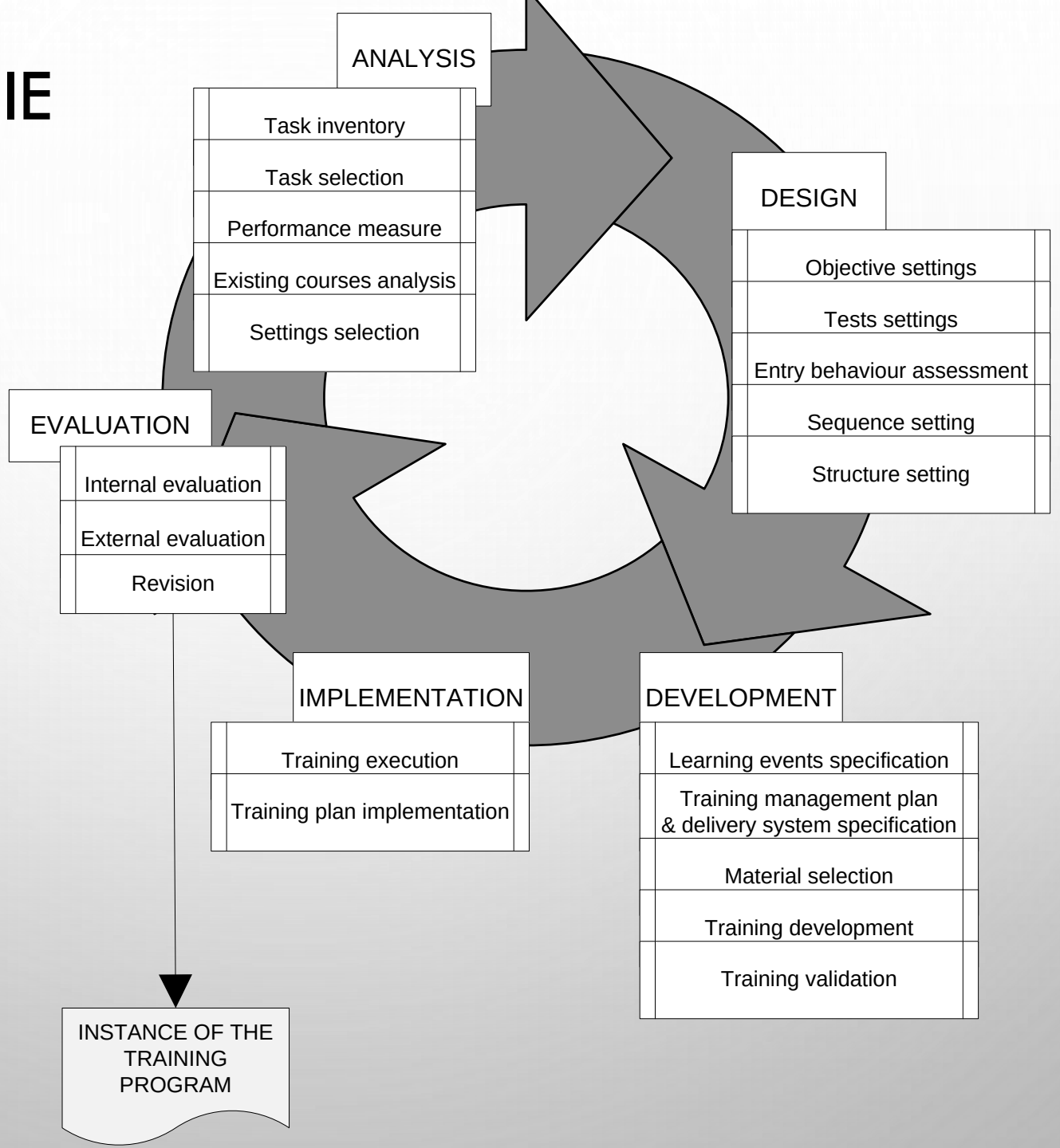
U.S. Department of Defense Training Document (1975). Pamphlet 350-30. August, 1975

- Analysis
- Design
- Development
- Implementation
- Evaluation



ADDIE

- Analysis
- Design
- Development
- Implementation
- Evaluation



TASKS AND TRAINING RESEARCH

- Research on task analysis is a pillar of training design

HTA Hierarchical Task Analysis : Anett J. & Duncan K.D. (1967, 1971)

Shepherd A.(1985)

Patrick J. (1992)

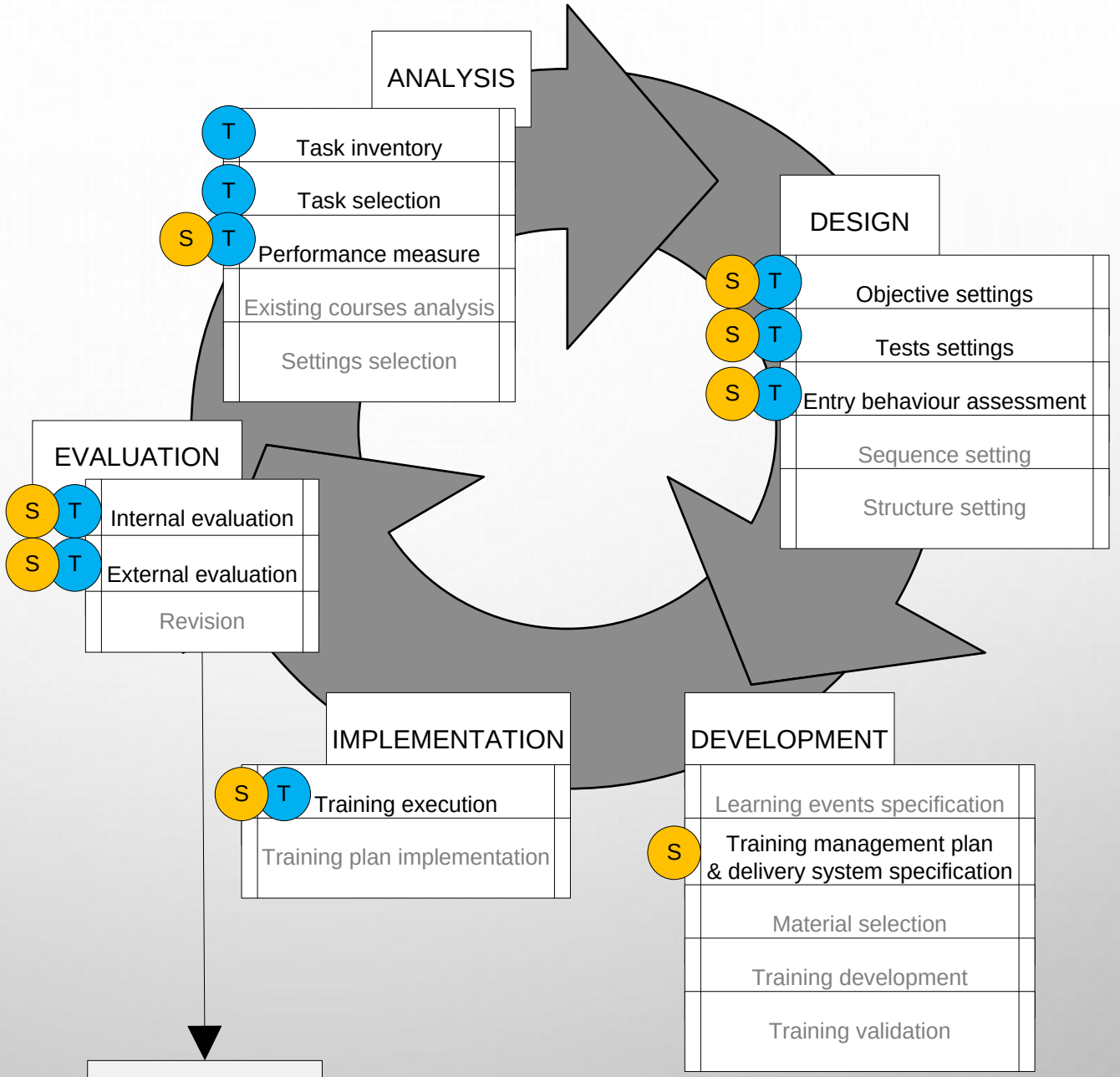
- Training research explicitly requires tasks identification together with:
  - Situation
  - Objects
  - Actions (including tasks)
  - Tools

The Conditions of learning and the theory of Instruction

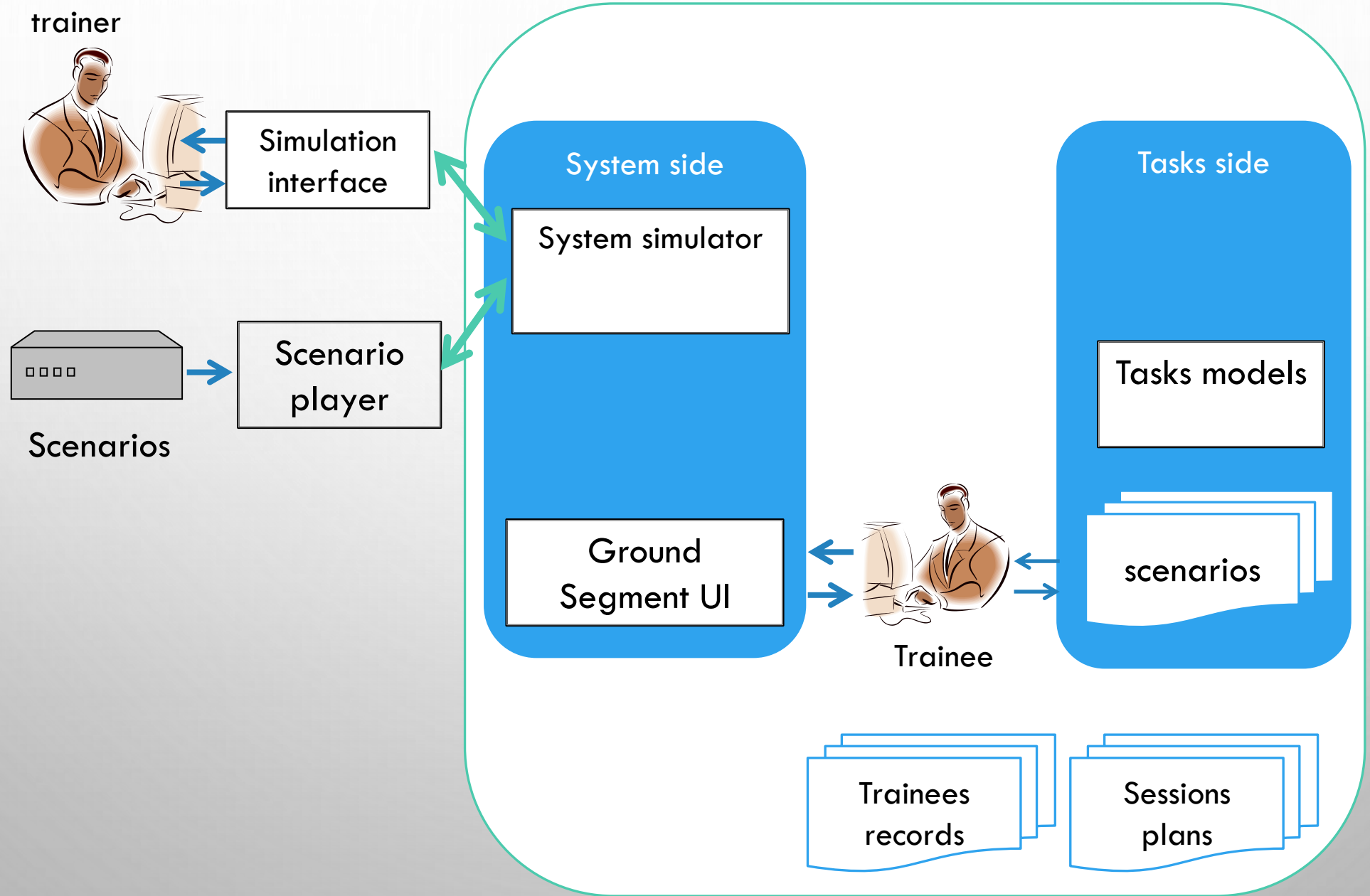
Gagné R. (1985)

Evaluation and setting of training objectives

INTEGRATION WITHIN ADDIE



PRACTICAL TRAINING SESSION

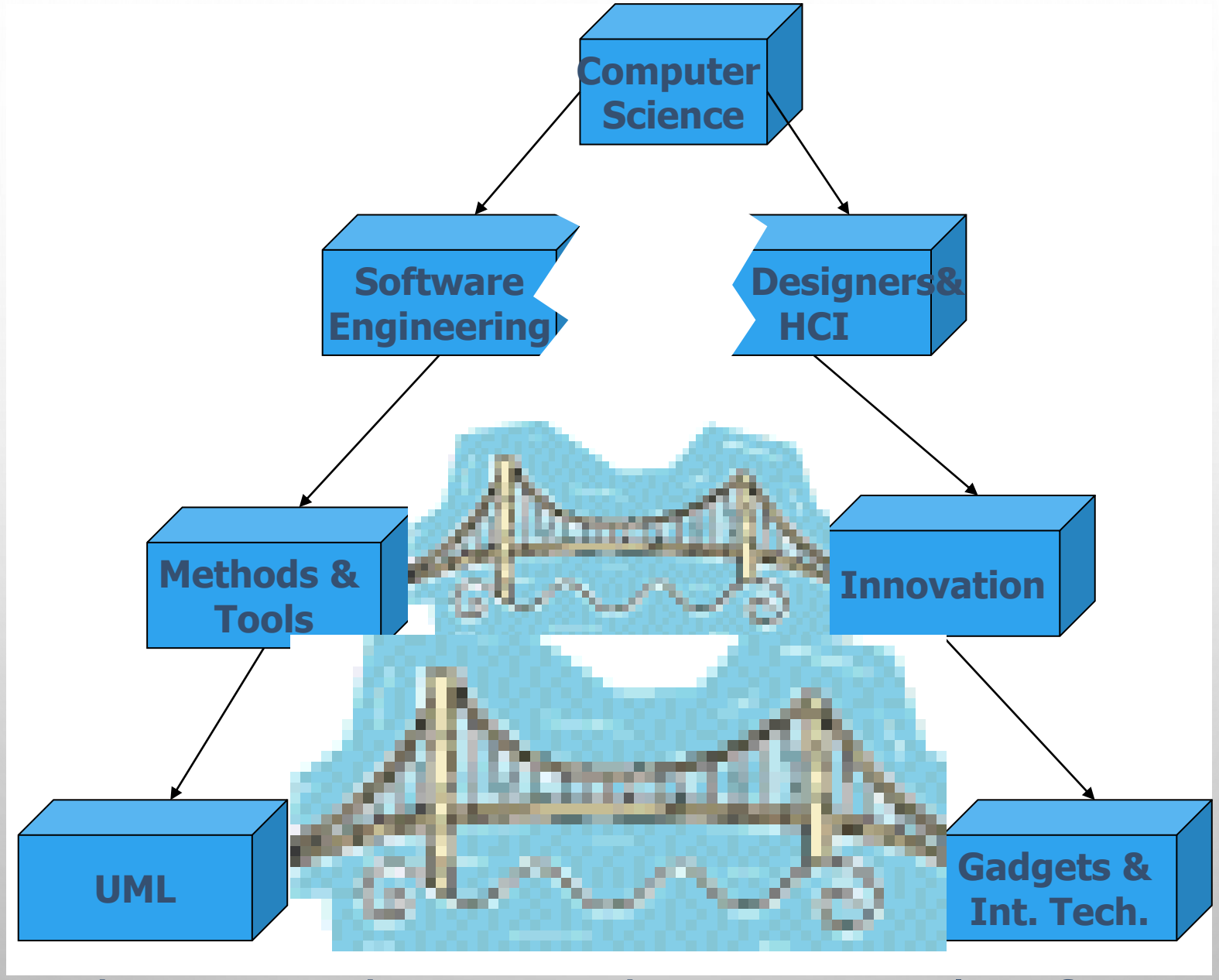


SAFETY CRITICAL INTERACTIVE SYSTEMS

- Safety Critical Systems
  - Software Engineers
  - System centered
  - Reliability
  - Safety requirements (certification)
  - Formal specification
  - Verification / Proof
  - Waterfall model / structured
  - Archaic interaction techniques

- Interactive Systems
  - Usability experts
  - User centered
  - Usability
  - Human factors
  - Task analysis & modeling
  - Evaluation
  - Iterative process / Prototyping
  - Novel Interaction techniques

Reliability vs Usability



Bridges over the Gaps The same applies for HF

BASIC ASSUMPTIONS

- DO NOT trust "HCI designers" (beyond innovation)
  - They will ask for "unfeasible" things
  - They will change their mind faster that one can implement
  - They will ask for trust (when no argument is available)
  - They usually do not care about safety issues
  - They are human being
- DO NOT trust "developers" (beyond hacking)
  - They don't know what to do
  - They "mostly" don’t know how to do it
  - They have a partial view of the problems
  - They usually do not care about usability issues
  - They are human being
- Control over the process and people is required

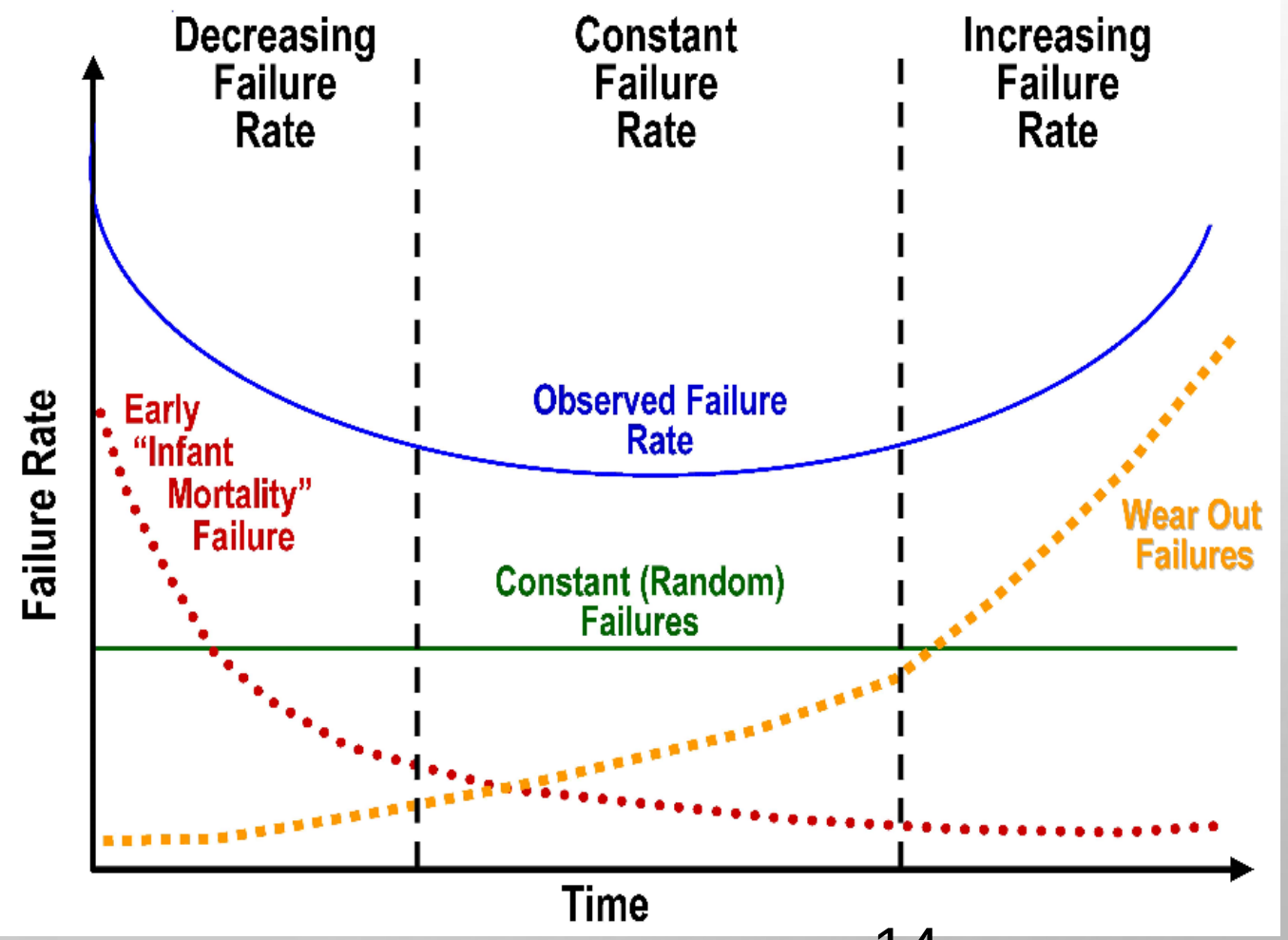
SAFETY, DEPENDABILITY AND RESILIENCE

Standards, properties and processes

RELIABILITY ENGINEERING (PROBABILITY)

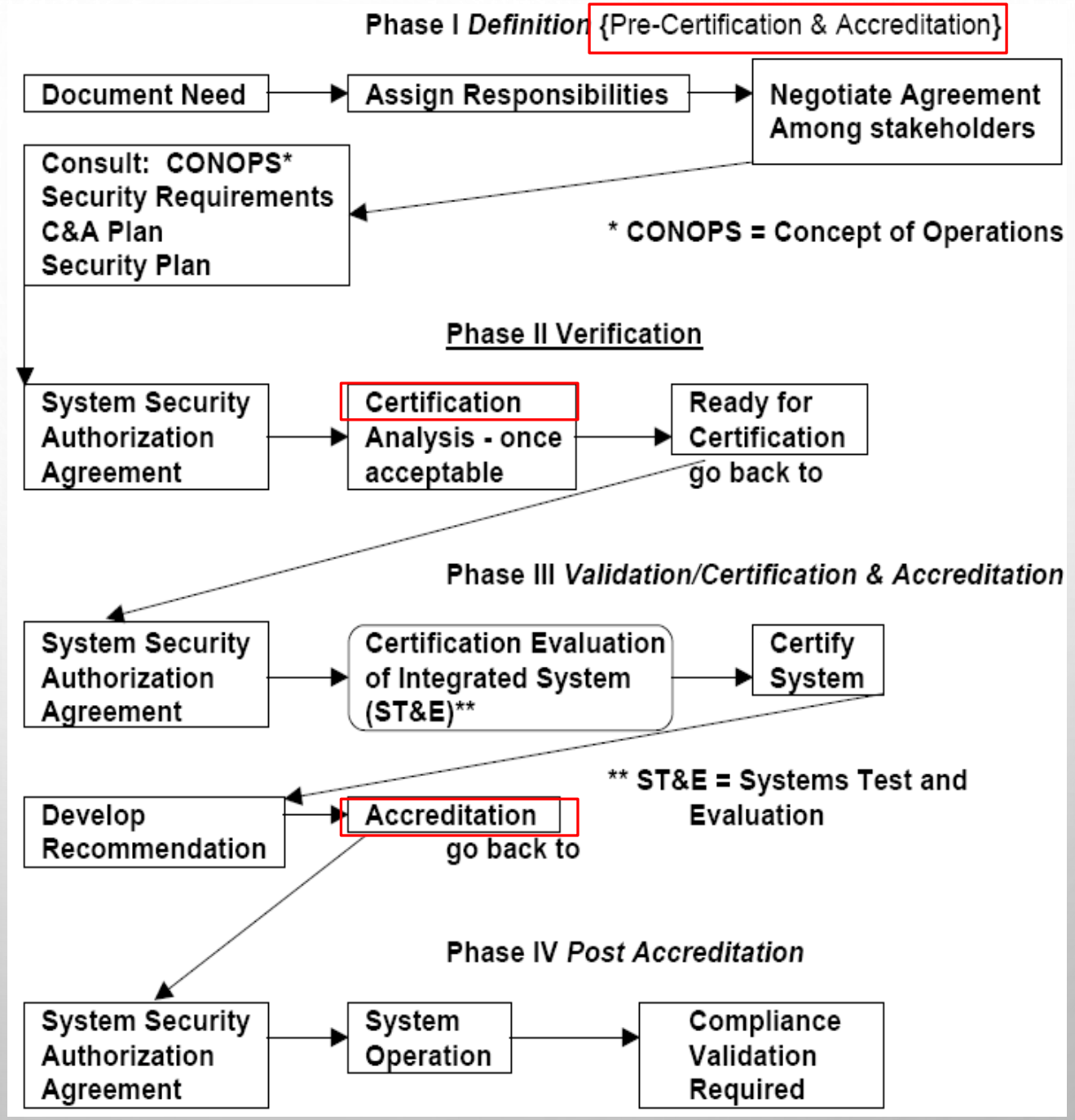
- Reliability engineering is the ability of a system or component to perform its required functions under stated conditions for a specified period of time

- Prevent failures or
- Things will fail well
- Anticipate causes



RELIABILITY ENGINEERING PROCESS

Standard
DO 178-B



SAFETY ENGINEERING (PROBABILITY AGAIN)

- **Safety engineering** assures that a life-critical system behaves as needed even when pieces fail
- IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems"
- Your system will not kill

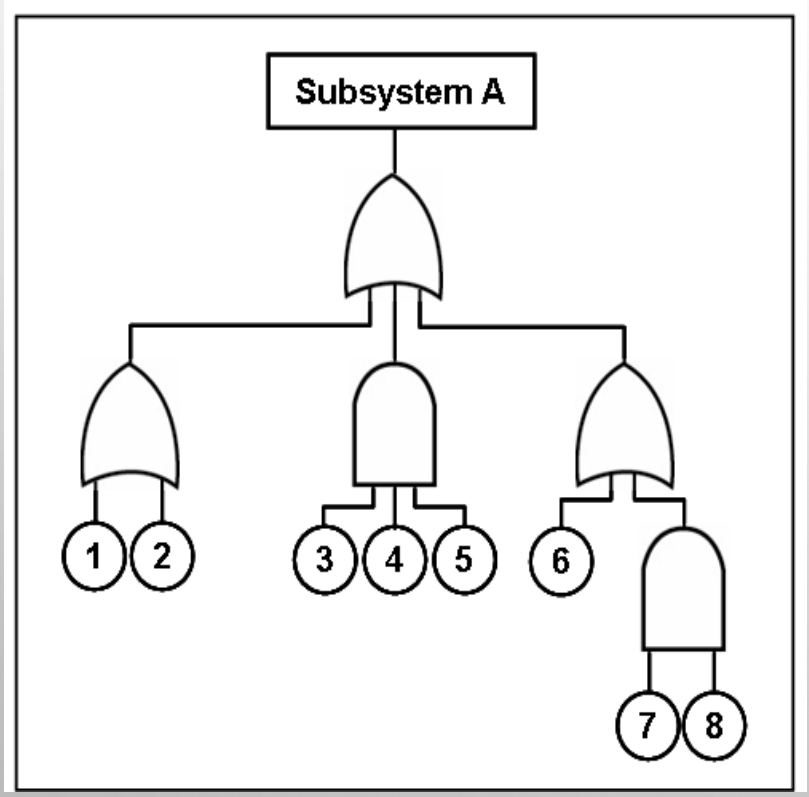
Safety Integrity Level	Probability of dangerous failure per hour (demand mode of operation)
SIL4	$>=10^{-5}$ to $<10^{-4}$
SIL3	$>=10^{-4}$ to $<10^{-3}$
SIL2	$>=10^{-3}$ to $<10^{-2}$
SIL1	$>=10^{-2}$ to $<10^{-1}$

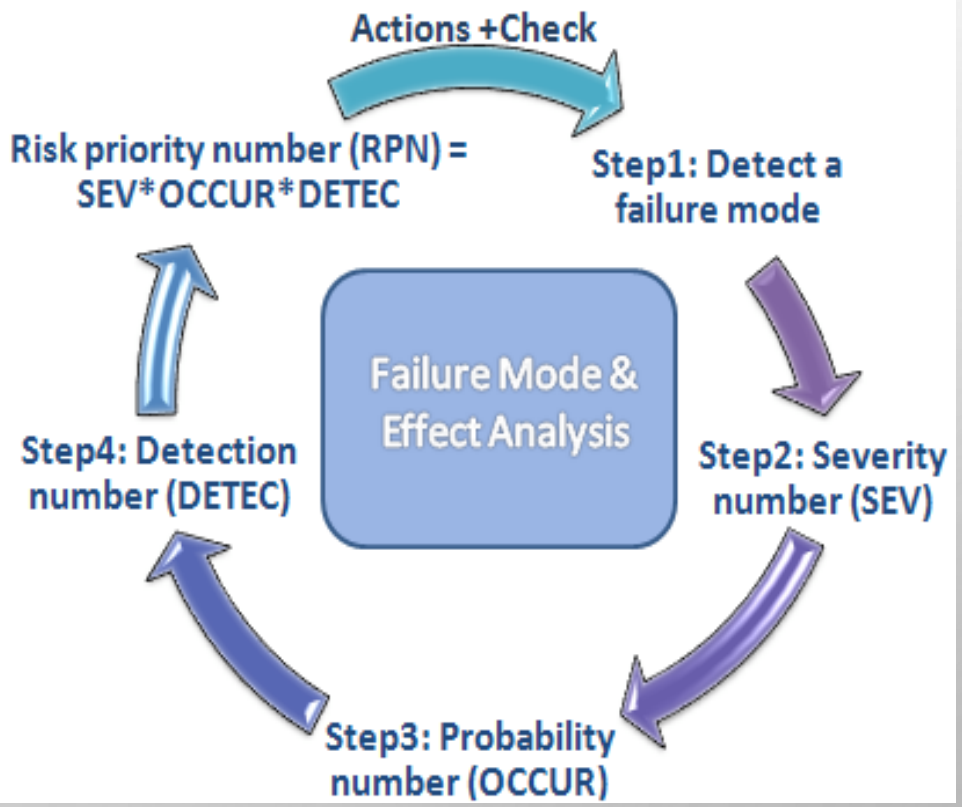
MATRICE D'ACCEPTATION DES RISQUES (EXEMPLE)

Gravité / Occurrence	**Catastrophique** (plusieurs morts)	**Critique/ Hazardous** (1 mort ou plusieurs blessés)	**Marginal/ Major** (1 blessé, dommages à l'environnement)	**Négligeable/ minor**
Fréquent >1E-3 /h	**inacceptable**	**inacceptable**	**inacceptable**	**à examiner**
Occasionnel 1E-3/h à 1E-5/h	**inacceptable**	**inacceptable**	**à examiner**	**acceptable**
Rare 1E-5/h à 1E-7/h	**inacceptable**	**à examiner**	**acceptable**	**acceptable**
Improbable 1E-7/h à 1E-9/h	**à examiner**	**acceptable**	**acceptable**	**acceptable**
Hautement improbable < 1E-9/h	**acceptable**	**acceptable**	**acceptable**	**acceptable**

SAFETY ENGINEERING PROCESSES

- FMEA: Failure Modes and Effects Analysis
- Fault Tree Analysis





SAFETY ENGINEERING PROCESSES

- System Hazard Analysis
- Learning from Accidents/Incidents
- Crash Tests
- Certification

LA NOTION DE RISQUE

RISK = PROBABILITY OF OCCURRENCE x IMPACT

LA NOTION DE RISQUE

RISK = PROBABILITY OF OCCURRENCE x IMPACT

EXERCICE: ÉVALUATION DU RISQUE SANS TRAINING

- Supposons que vous conceviez et développiez un système sans concevoir son training associé
- Réalisez une analyse de risque
- Faites une étude FMECA
  - Failure Modes, Effect and Criticality Analysis
  - (à voir plus tard après Fault-tolerance)

AUTRE EXERCICE POSSIBLE

- Accident Kegworth (wikipedia – section incident)
  https://en.wikipedia.org/wiki/Kegworth_air_disaster
- Décrire la séquence des événements
- Evaluer l'impact positif ou négatif des événements
- Déterminer la personne ou le système responsable de la production des événements/action
- Identifier les responsabilités (HOT)

EXERCISE : ANALYSIS OF KEGWORTH AIRCRAFT ACCIDENT

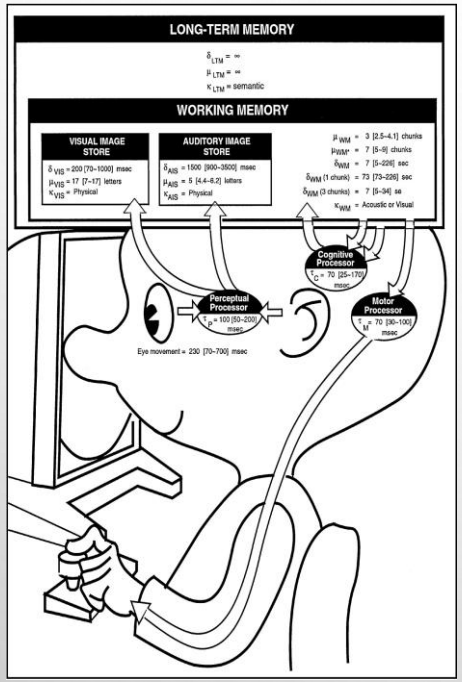
- An industrial example of all this happened in the cockpit of the B737 that crashed at Kegworth in 1989.
- http://en.wikipedia.org/wiki/Kegworth_air_disaster
- The crew throttled back one engine that was suspected to vibrate. Very shortly after that, the vibration level decreased on that engine, leading the crew to believe that they had diagnosed the problem right.

EXERCISE : ANALYSIS OF KEGW AIRCRAFT AC

- An industrial example of the B737 that crashed at Kegworth in 1989.
- http://en.wikipedia.org/
- The crew throttled back vibrate. Very shortly after that, the vibration ading the crew to believe that they had dia

EXERCISE : ANALYSIS OF KEGWORTH AIRCRAFT ACCIDENT

- Hints for the solution

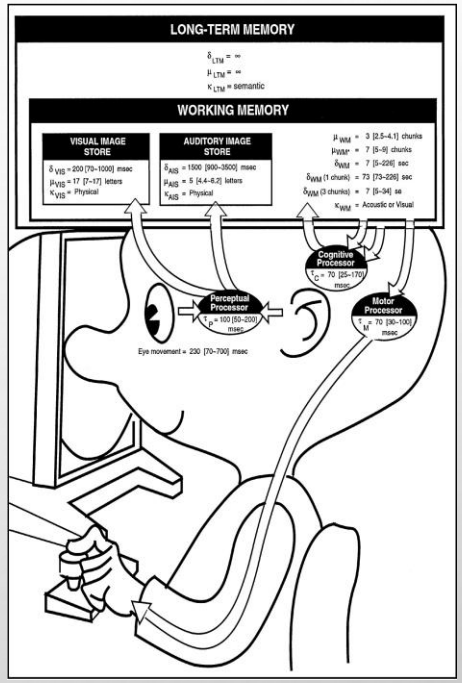


Perception	Proprioception	Smell	Hearing	Taste
	Thermoception	Touch	Vestibular	Sight

EXERCISE : ANALYSIS OF KEGWORTH AIRCRAFT ACCIDENT

- Hints for the solution

- Identification of the root cause
- Identification of barriers (efficient and inefficient)
- Accident attributed to human error (human as a barrier)
  - Cognitive biases
  - Confirmation bias
  - Reinforcing information



SAFETY ENGINEERING PROCESSES

- System Hazard Analysis
- Learning from Accidents/Incidents
- Crash Tests
- Certification

SAFETY ENGINEERING PROCESSES

- System Hazard Analysis
- Learning from Accidents/Incidents
- Cra
- Cer

SAFETY LIFE CYCLE

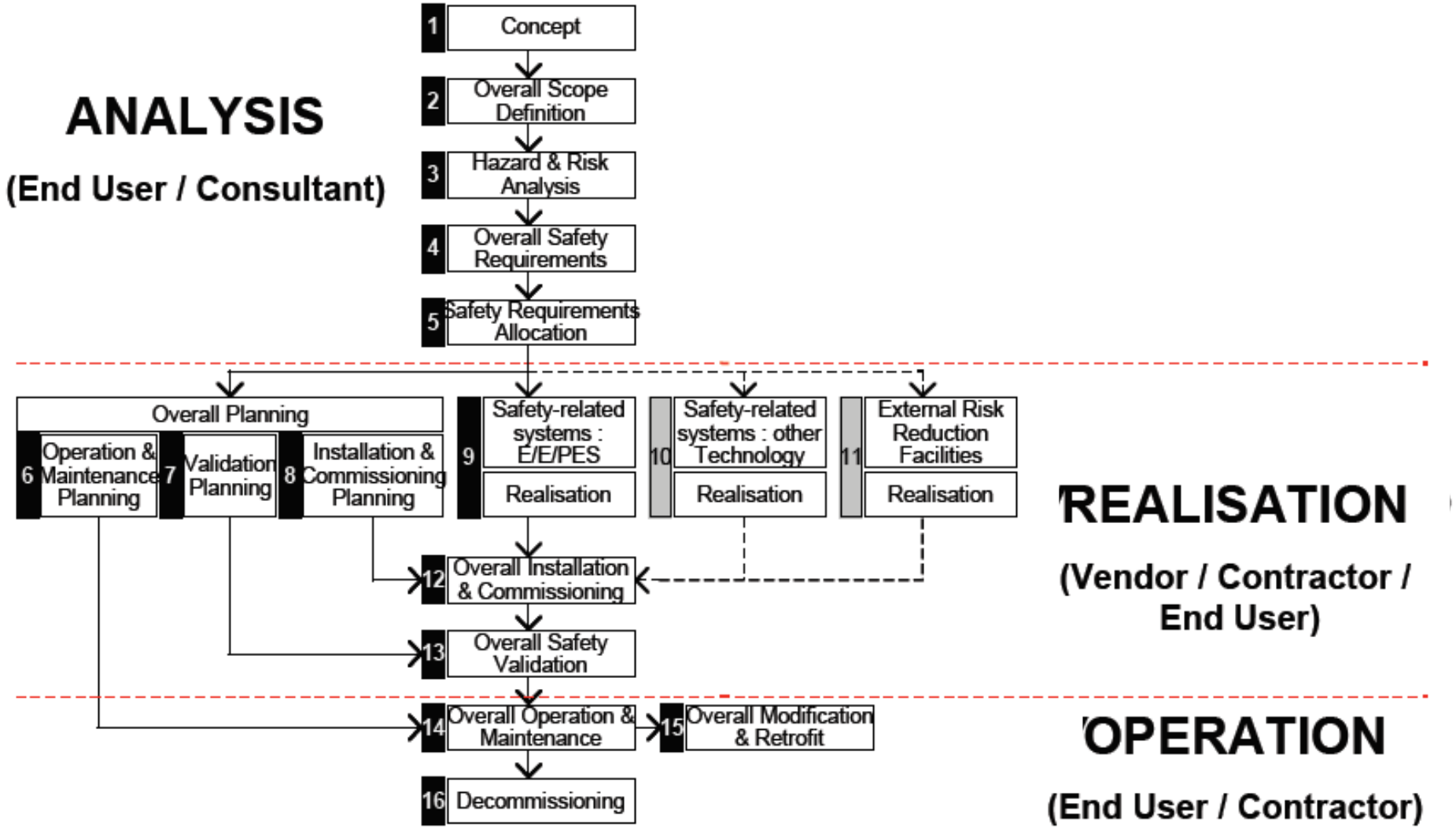


Figure 2: Safety life cycle from IEC 61508.

EXEMPLE ATV CC

- ISO 14620-1 safety requirements for space systems

  "5.4.2 Inadvertant operation

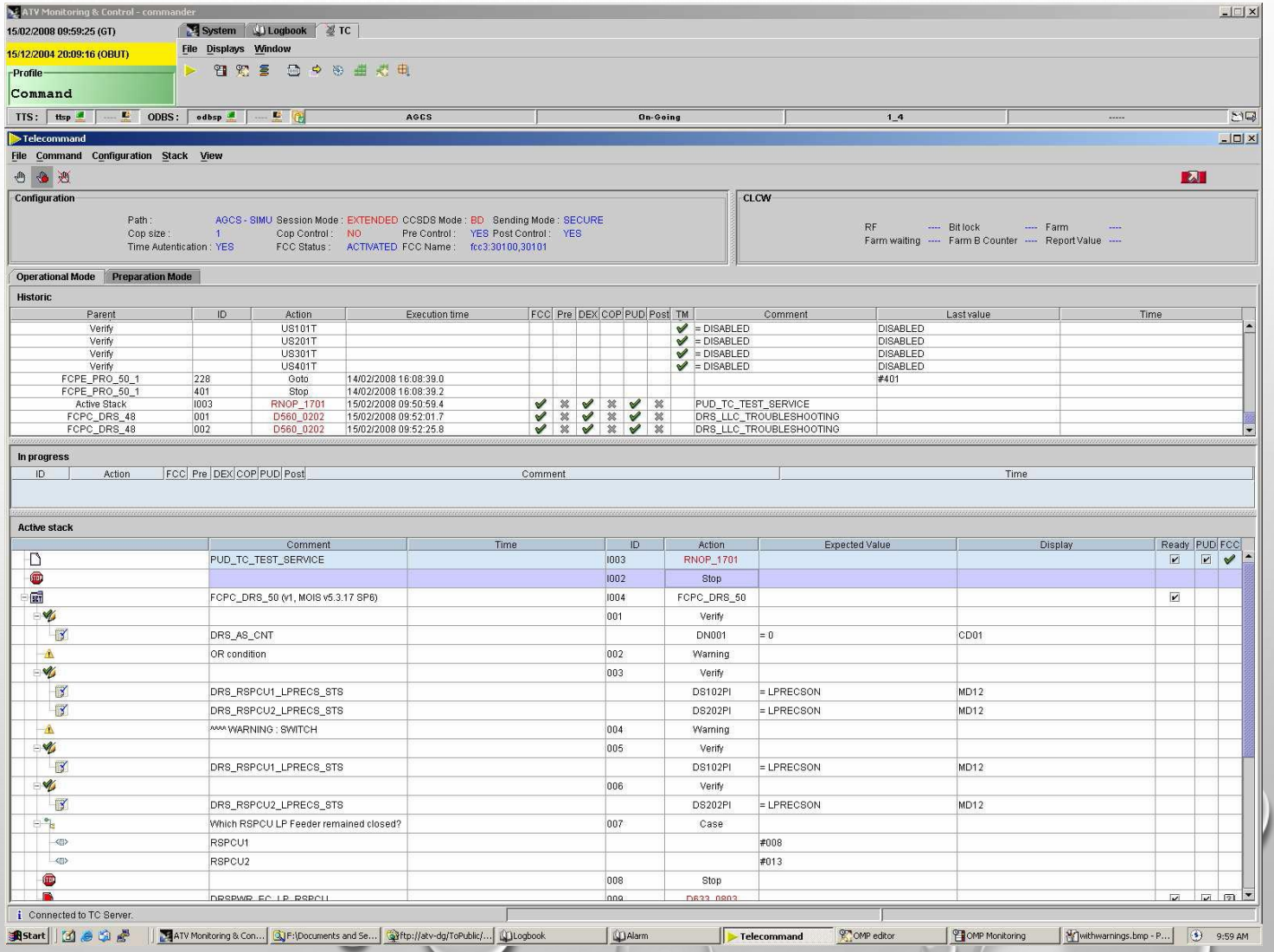
  Inadvertant operation of a safety critical function shall be prevented by:

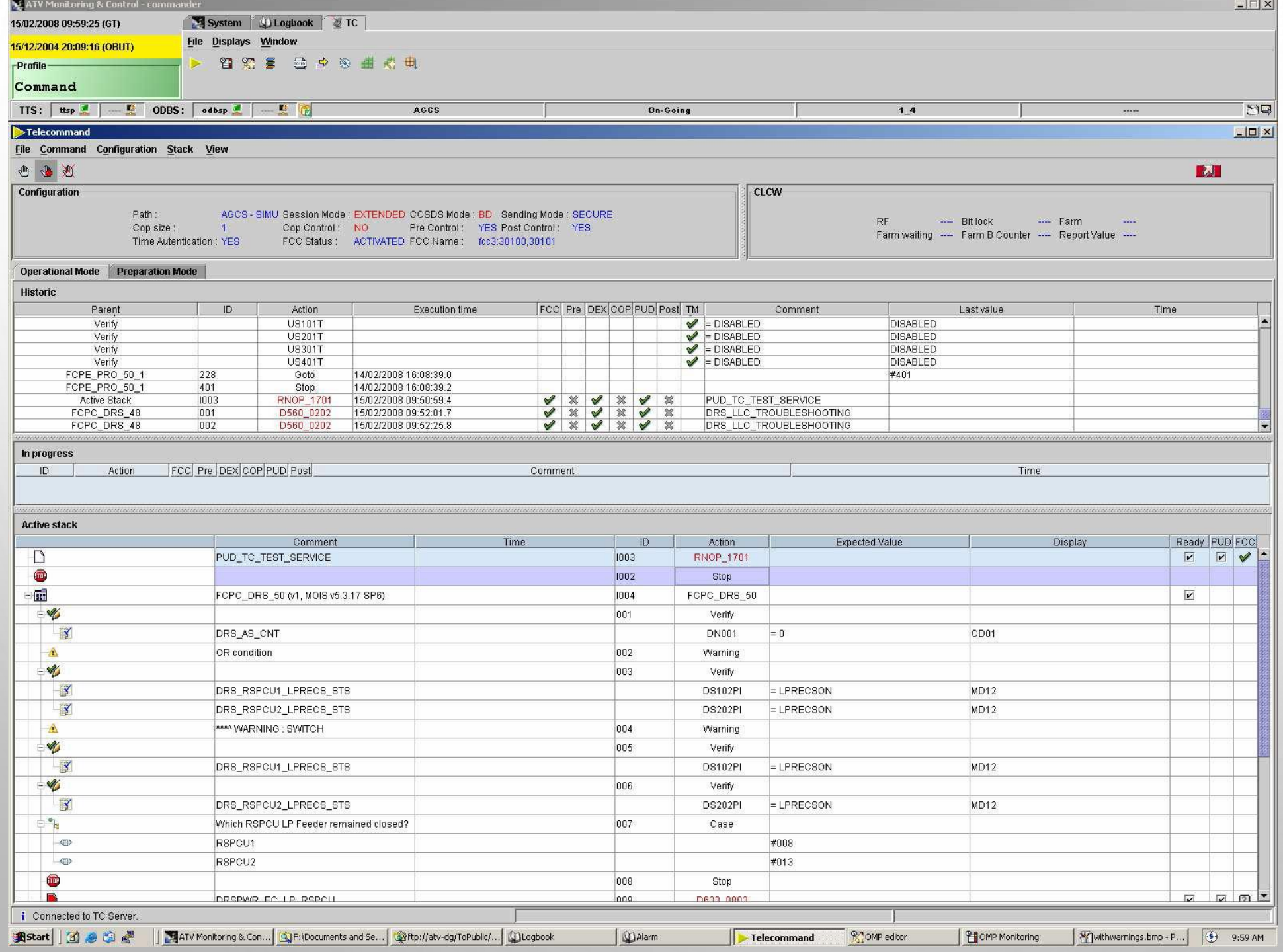
  a) two independant inhibits, if it induces critical consequences

  b) three independant inhibits, if it induces catastrophic consequences"

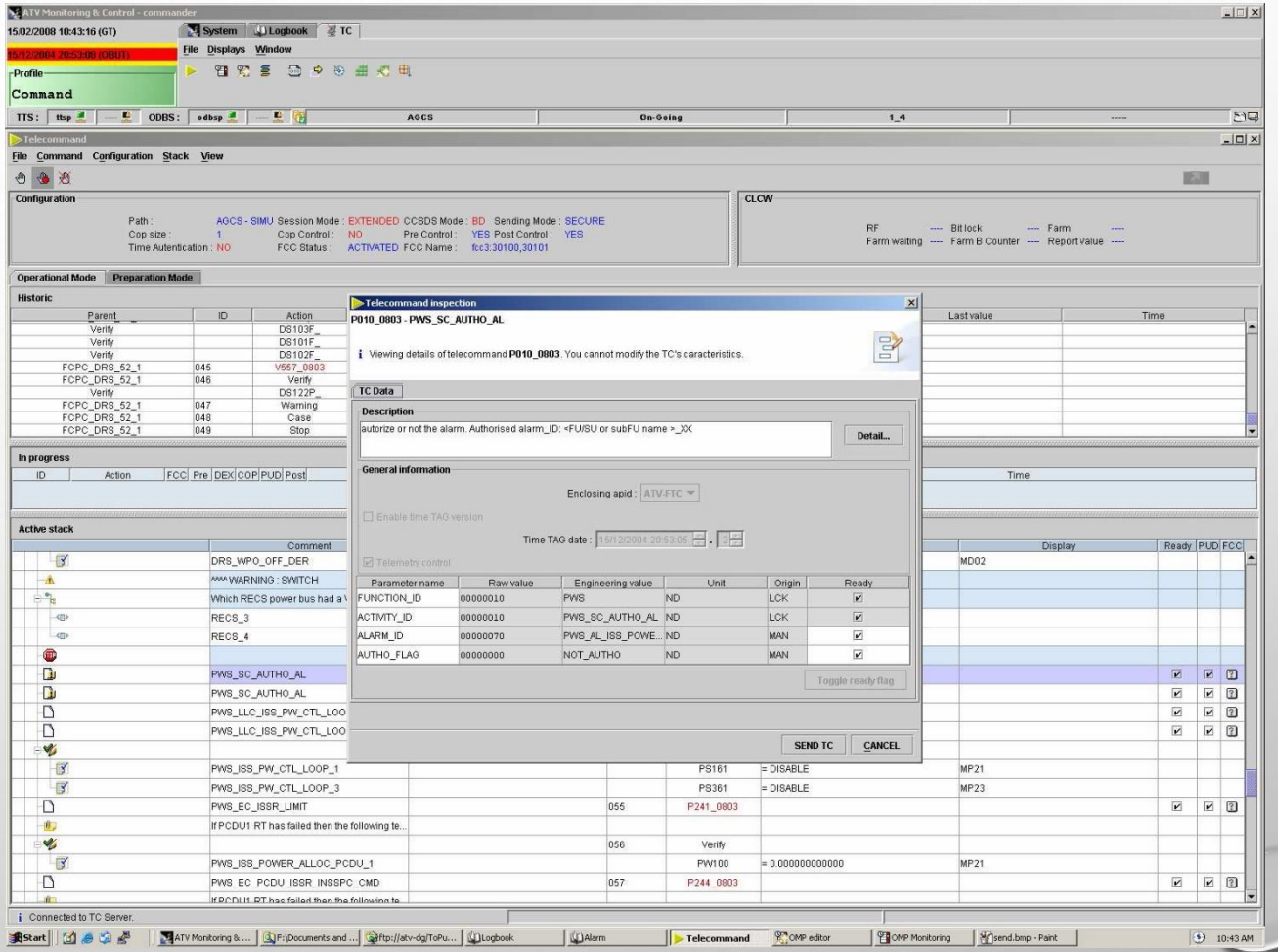
- Implementation of the safety requirements for the ATV
  - "Send CATA telecommand" is confirmed by 3 clicks in 3 different zones of the interface of the ATV control center
  - The protection is likely to be overshooted because its implementation leads to too repetitive human tasks

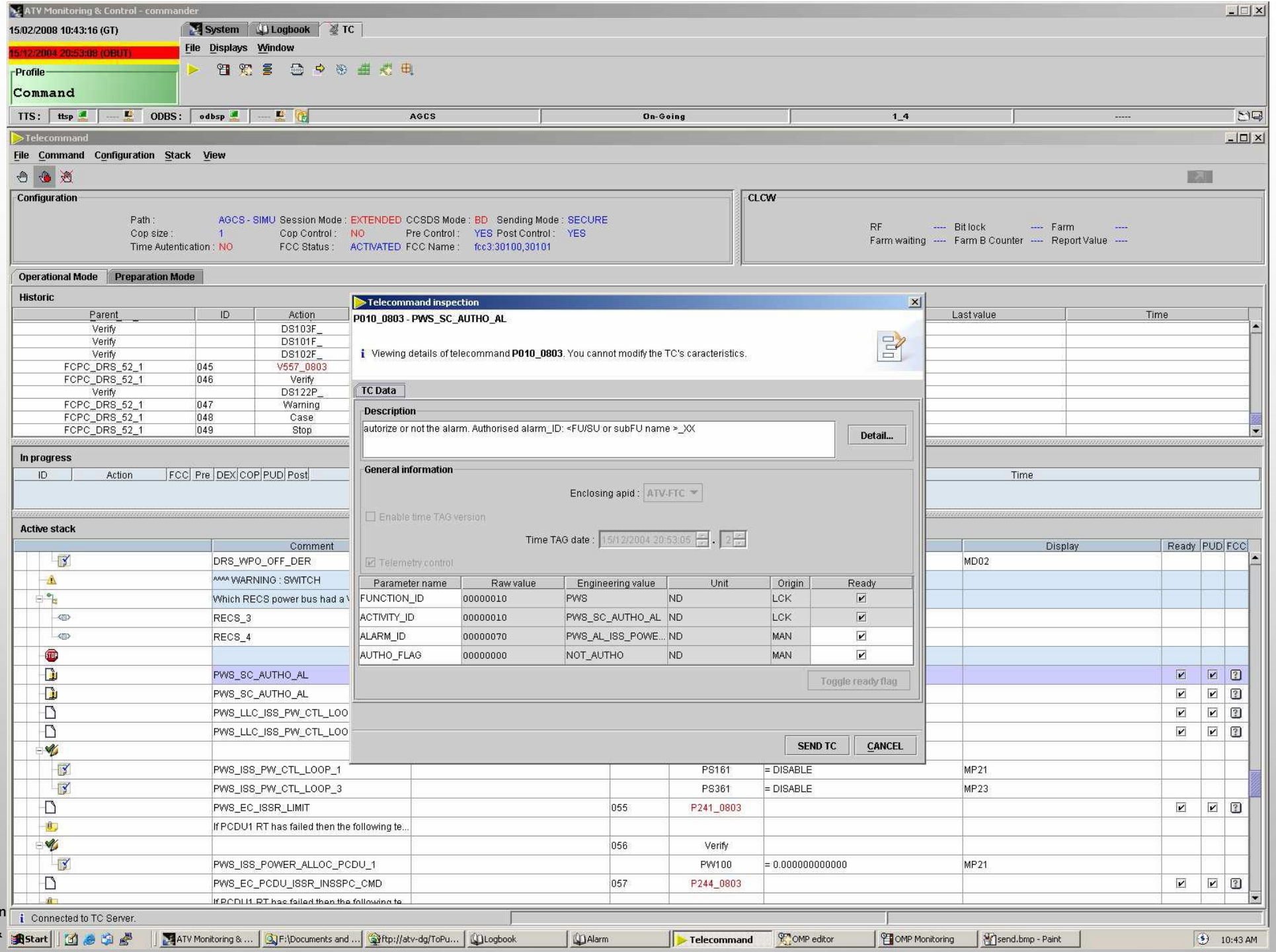
EXEMPLE ATV CC



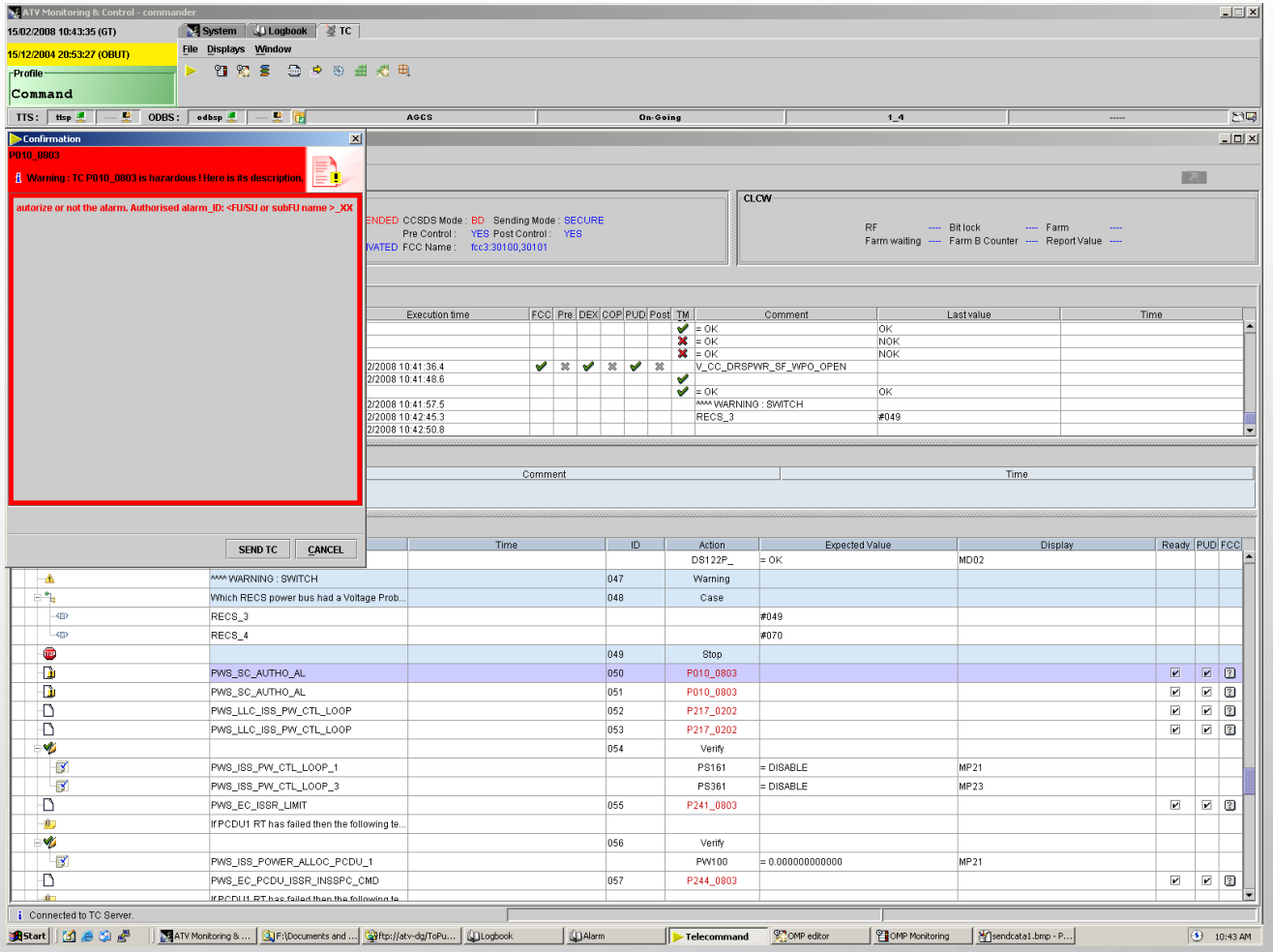

ATV Monitoring & Control - commander
15/02/2008 09:59:25 (GT)
15/12/2004 20:09:16 (OBUT)
Profile
Command
System
Logbook
TC
File Displays Window
TTS: ttsp ODBS: odbsp AGCS On-Going 1_4
Telecommand
File Command Configuration Stack View
Configuration
Path : AGCS - SIMU Session Mode : EXTENDED CCSDS Mode : BD Sending Mode : SECURE
Cop size : 1 Cop Control : NO Pre Control : YES Post Control : YES
Time Autentication : YES FCC Status : ACTIVATED FCC Name : fcc3:30100,30101
CLCW
RF ---- Bit lock ---- Farm ----
Farm waiting ---- Farm B Counter ---- Report Value ----
Operational Mode
Preparation Mode
Historic
Parent | ID | Action | Execution time | FCC | Pre | DEX | COP | PUD | Post | TM | Comment | Last value | Time
Verify | | US101T | | | | | | | | | = DISABLED | DISABLED |
Verify | | US201T | | | | | | | | | = DISABLED | DISABLED |
Verify | | US301T | | | | | | | | | = DISABLED | DISABLED |
Verify | | US401T | | | | | | | | | = DISABLED | DISABLED |
FCPE_PRO_50_1 | 228 | Goto | 14/02/2008 16:08:39.0 | | | | | | | | | #401 |
FCPE_PRO_50_1 | 401 | Stop | 14/02/2008 16:08:39.2 | | | | | | | | | |
Active Stack | I003 | RNOP_1701 | 15/02/2008 09:50:59.4 | | | | | | | | PUD_TC_TEST_SERVICE | |
FCPC_DRS_48 | 001 | D560_0202 | 15/02/2008 09:52:01.7 | | | | | | | | DRS_LLC_TROUBLESHOOTING | |
FCPC_DRS_48 | 002 | D560_0202 | 15/02/2008 09:52:25.8 | | | | | | | | DRS_LLC_TROUBLESHOOTING | |
In progress
ID | Action | FCC | Pre | DEX | COP | PUD | Post | Comment | Time
Active stack
Comment | Time | ID | Action | Expected Value | Display | Ready | PUD | FCC
PUD_TC_TEST_SERVICE | | I003 | RNOP_1701 | | | | | |
| | I002 | Stop | | | | | |
FCPC_DRS_50 (v1, MOIS v5.3.17 SP6) | | I004 | FCPC_DRS_50 | | | | | |
| | 001 | Verify | | | | | |
DRS_AS_CNT | | | DN001 | = 0 | CD01 | | | |
OR condition | | 002 | Warning | | | | | |
| | 003 | Verify | | | | | |
DRS_RSPCU1_LPRECS_STS | | | DS102PI | = LPRECSON | MD12 | | | |
DRS_RSPCU2_LPRECS_STS | | | DS202PI | = LPRECSON | MD12 | | | |
~~~ WARNING : SWITCH | | 004 | Warning | | | | | |
| | 005 | Verify | | | | | |
DRS_RSPCU1_LPRECS_STS | | | DS102PI | = LPRECSON | MD12 | | | |
| | 006 | Verify | | | | | |
DRS_RSPCU2_LPRECS_STS | | | DS202PI | = LPRECSON | MD12 | | | |
Which RSPCU LP Feeder remained closed? | | 007 | Case | | | | | |
RSPCU1 | | | | #008 | | | | |
RSPCU2 | | | | #013 | | | | |
| | 008 | Stop | | | | | |
Connected to TC Server.
Start
ATV Monitoring & Con...
F:\Documents and Se...
ftp://atv-dg/ToPublic/...
Logbook
Alarm
Telecommand
OMP editor
OMP Monitoring
withwarnings.bmp - P...
9:59 AM
~~~

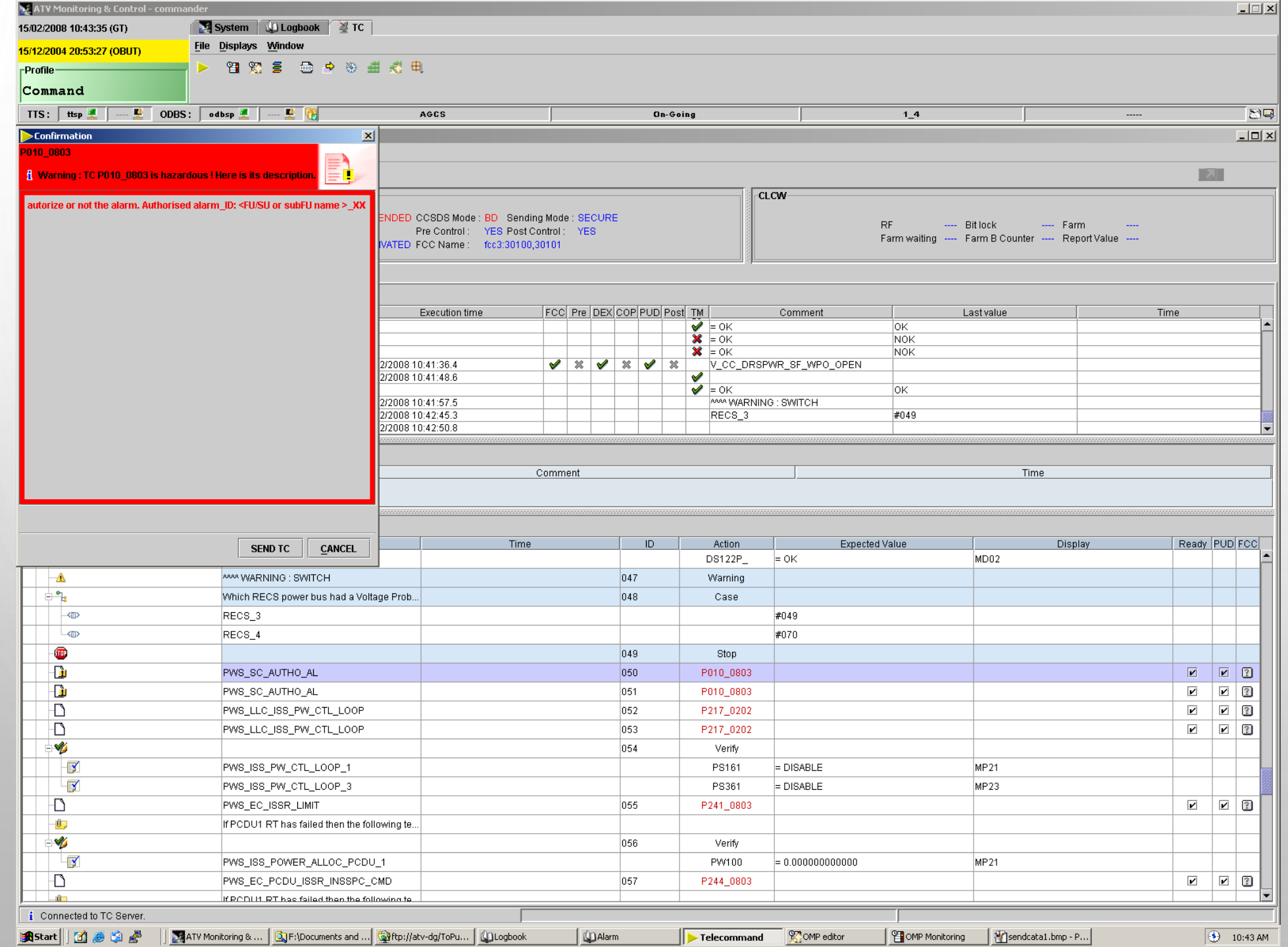
EXEMPLE ATV CC




ATV Monitoring & Control - commander
15/02/2008 10:43:16 (GT)
15/12/2004 20:53:08 (OBUT)
Profile
Command
System
Logbook
TC
File Displays Window
TTS : ttsp
ODBS : odbsp
AGCS
On-Going
1_4
Telecommand
File Command Configuration Stack View
Configuration
Path : AGCS - SIMU Session Mode : EXTENDED CCSDS Mode : BD Sending Mode : SECURE
Cop size : 1 Cop Control : NO Pre Control : YES Post Control : YES
Time Autentication : NO FCC Status : ACTIVATED FCC Name : fcc3:30100,30101
CLCW
RF ---- Bit lock ---- Farm ----
Farm waiting ---- Farm B Counter ---- Report Value ----
Operational Mode
Preparation Mode
Historic
Parent
ID
Action
Verify DS103F_
Verify DS101F_
Verify DS102F_
FCPC_DRS_52_1 045 V557_0803
FCPC_DRS_52_1 046 Verify
Verify DS122P_
FCPC_DRS_52_1 047 Warning
FCPC_DRS_52_1 048 Case
FCPC_DRS_52_1 049 Stop
Last value
Time
In progress
ID
Action
FCC
Pre
DEX
COP
PUD
Post
Time
Active stack
Comment
Display
Ready
PUD
FCC
DRS_WPO_OFF_DER MD02
^^^^ WARNING : SWITCH
Which RECS power bus had a
RECS_3
RECS_4
PWS_SC_AUTHO_AL
PWS_SC_AUTHO_AL
PWS_LLC_ISS_PW_CTL_LOO
PWS_LLC_ISS_PW_CTL_LOO
PWS_ISS_PW_CTL_LOOP_1 PS161 = DISABLE MP21
PWS_ISS_PW_CTL_LOOP_3 PS361 = DISABLE MP23
PWS_EC_ISSR_LIMIT 055 P241_0803
If PCDU1 RT has failed then the following te...
056 Verify
PWS_ISS_POWER_ALLOC_PCDU_1 PW100 = 0.000000000000 MP21
PWS_EC_PCDU_ISSR_INSSPC_CMD 057 P244_0803
Telecommand inspection
P010_0803 - PWS_SC_AUTHO_AL
Viewing details of telecommand P010_0803. You cannot modify the TC's caracteristics.
TC Data
Description
autorize or not the alarm. Authorised alarm_ID: <FU/SU or subFU name >_XX
Detail...
General information
Enclosing apid : ATV-FTC
Enable time TAG version
Time TAG date : 15/12/2004 20:53:05 . 2
Telemetry control
Parameter name
Raw value
Engineering value
Unit
Origin
Ready
FUNCTION_ID 00000010 PWS ND LCK
ACTIVITY_ID 00000010 PWS_SC_AUTHO_AL ND LCK
ALARM_ID 00000070 PWS_AL_ISS_POWE... ND MAN
AUTHO_FLAG 00000000 NOT_AUTHO ND MAN
Toggle ready flag
SEND TC
CANCEL
Connected to TC Server.
Start
ATV Monitoring & ...
F:\Documents and ...
ftp://atv-dg/ToPu...
Logbook
Alarm
Telecommand
OMP editor
OMP Monitoring
send.bmp - Paint
10:43 AM

EXEMPLE ATV CC




ATV Monitoring & Control - commander
15/02/2008 10:43:35 (GT)
15/12/2004 20:53:27 (OBUT)
Profile
Command
System
Logbook
TC
File
Displays
Window
TTS :
ttsp
ODBS :
odbsp
AGCS
On-Going
1_4
Confirmation
P010_0803
Warning : TC P010_0803 is hazardous ! Here is its description.
autorize or not the alarm. Authorised alarm_ID: <FU/SU or subFU name >_XX
SEND TC
CANCEL
CLCW
RF ---- Bit lock ---- Farm ----
Farm waiting ---- Farm B Counter ---- Report Value ----
Execution time
FCC
Pre
DEX
COP
PUD
Post
TM
Comment
Last value
Time
= OK
OK
NOK
V_CC_DRSPWR_SF_WPO_OPEN
^^^^ WARNING : SWITCH
RECS_3
#049
Comment
Time
Time
ID
Action
Expected Value
Display
Ready
PUD
FCC
DS122P_
= OK
MD02
^^^^ WARNING : SWITCH
047
Warning
Which RECS power bus had a Voltage Prob...
048
Case
RECS_3
#049
RECS_4
#070
049
Stop
PWS_SC_AUTHO_AL
050
P010_0803
PWS_SC_AUTHO_AL
051
P010_0803
PWS_LLC_ISS_PW_CTL_LOOP
052
P217_0202
PWS_LLC_ISS_PW_CTL_LOOP
053
P217_0202
054
Verify
PWS_ISS_PW_CTL_LOOP_1
PS161
= DISABLE
MP21
PWS_ISS_PW_CTL_LOOP_3
PS361
= DISABLE
MP23
PWS_EC_ISSR_LIMIT
055
P241_0803
If PCDU1 RT has failed then the following te...
056
Verify
PWS_ISS_POWER_ALLOC_PCDU_1
PW100
= 0.000000000000
MP21
PWS_EC_PCDU_ISSR_INSSPC_CMD
057
P244_0803
Connected to TC Server.
Start
ATV Monitoring & ...
F:\Documents and ...
ftp://atv-dg/ToPu...
Logbook
Alarm
Telecommand
OMP editor
OMP Monitoring
sendcata1.bmp - P...
10:43 AM

RÉCRÉATION

WHY DO I DO THAT? (SAFETY AND RELIABILITY)

- I am not a lucky user!

WHY DO I DO THAT?

- I am not a lucky user!
- Are you usually lucky?

WHY DO I DO THAT?

- I am not a lucky user!
- Are you usually lucky?
- How much can you trust your luck?

SAFETY IMPROVEMENTS?

SAFETY IMPRO-VEMENTS?

SAFETY IMPRO-VEMENTS?

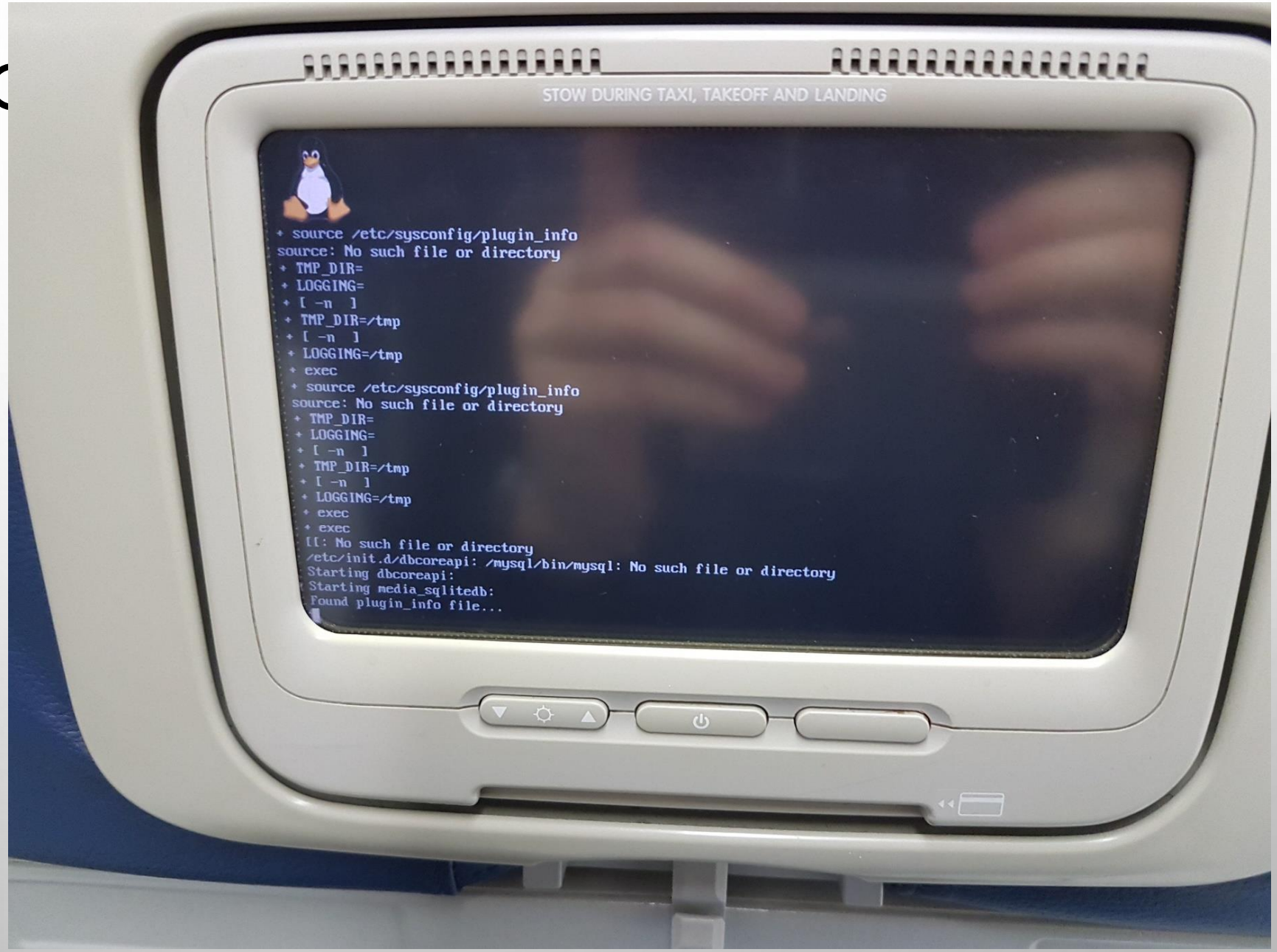
WHY DO I DO THAT?

- I am not a lucky user!
- Are you usually lucky?
- How much can you trust your luck?

User Experience?
Usability?
Reliability?

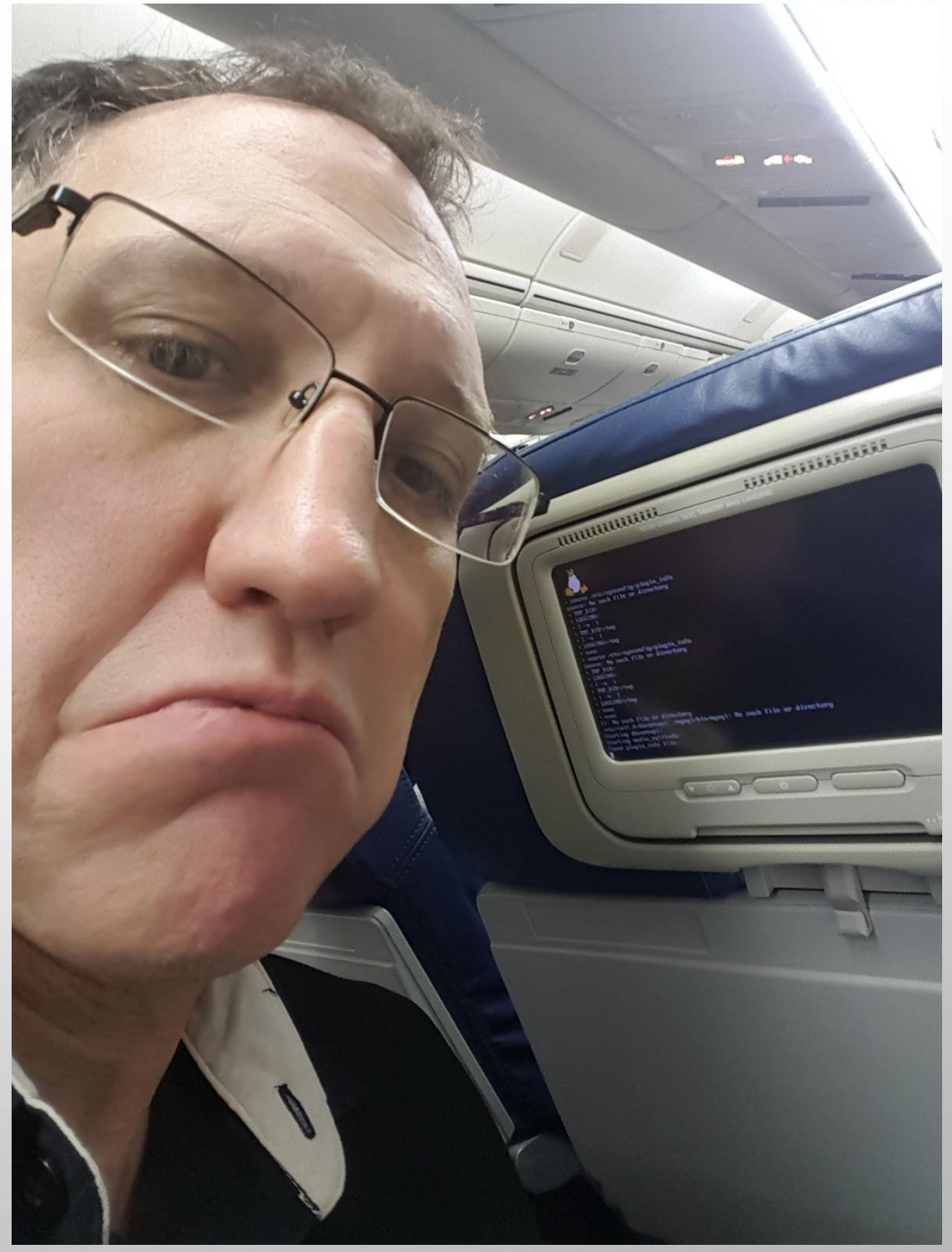
VERY REC

User Experience?
Usability?
Reliability?

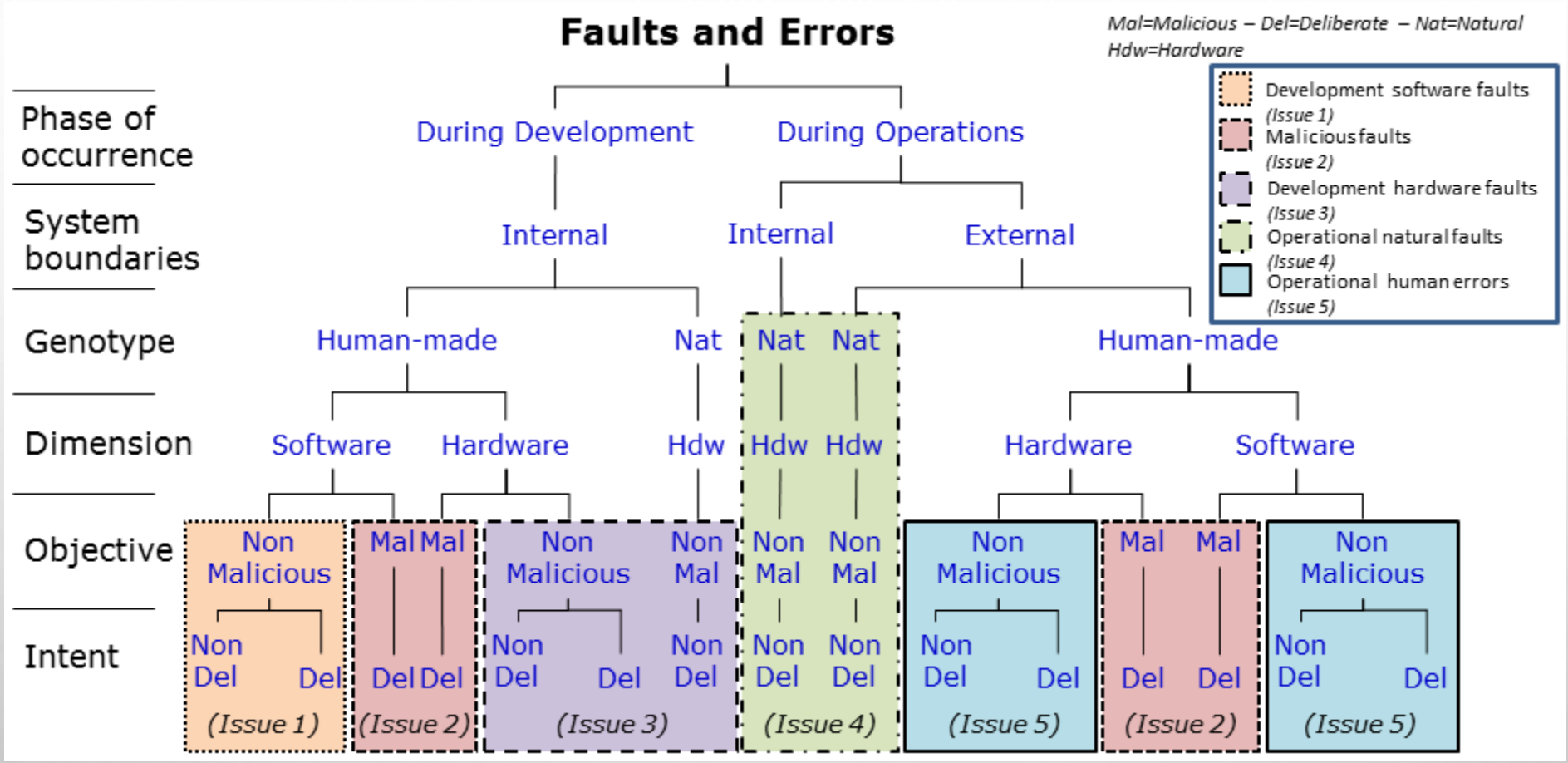


VERY RECENT

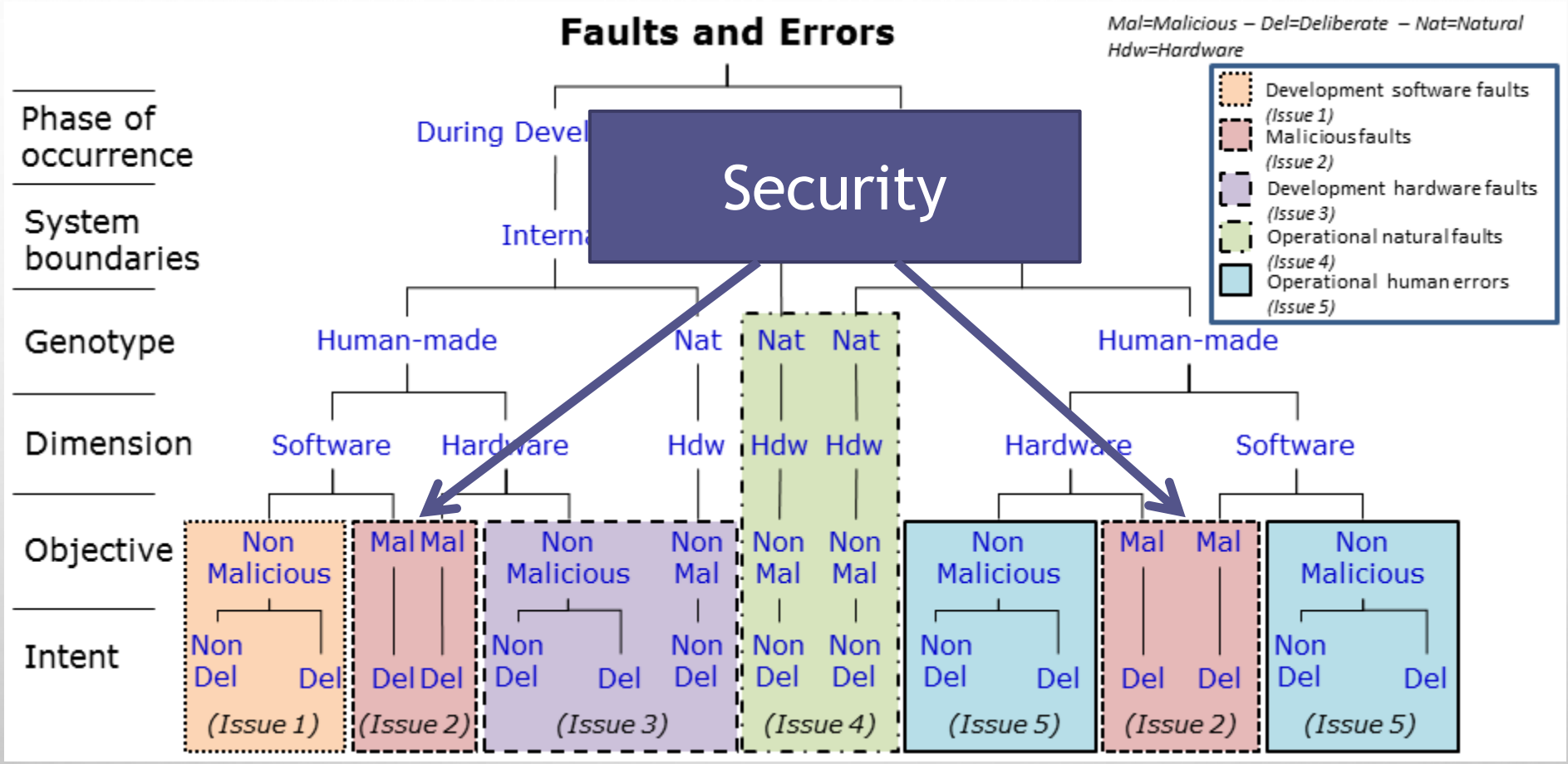
User Experience?
Usability?
Reliability?



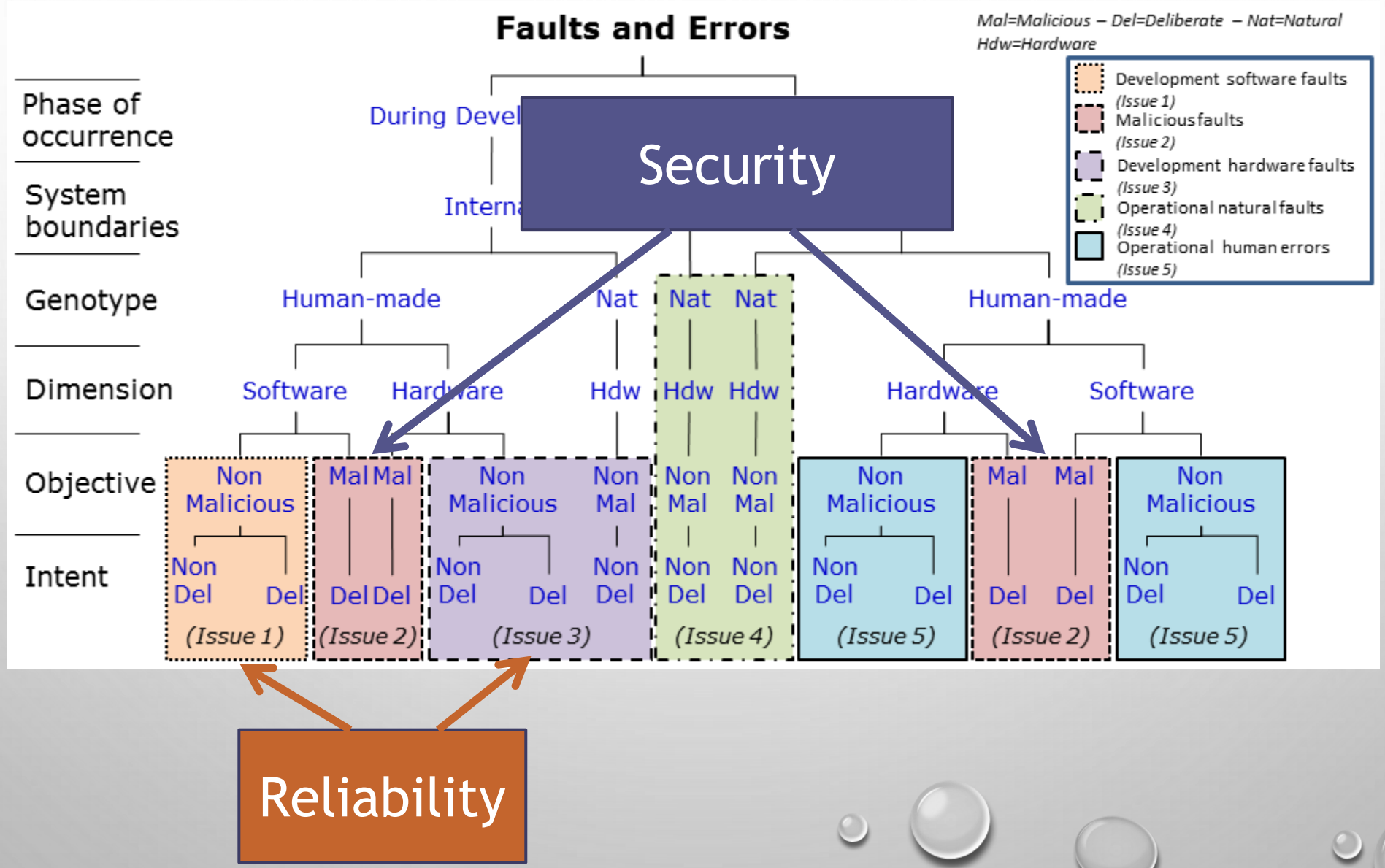
LUCKY IN THE ANALOGIC WORLD

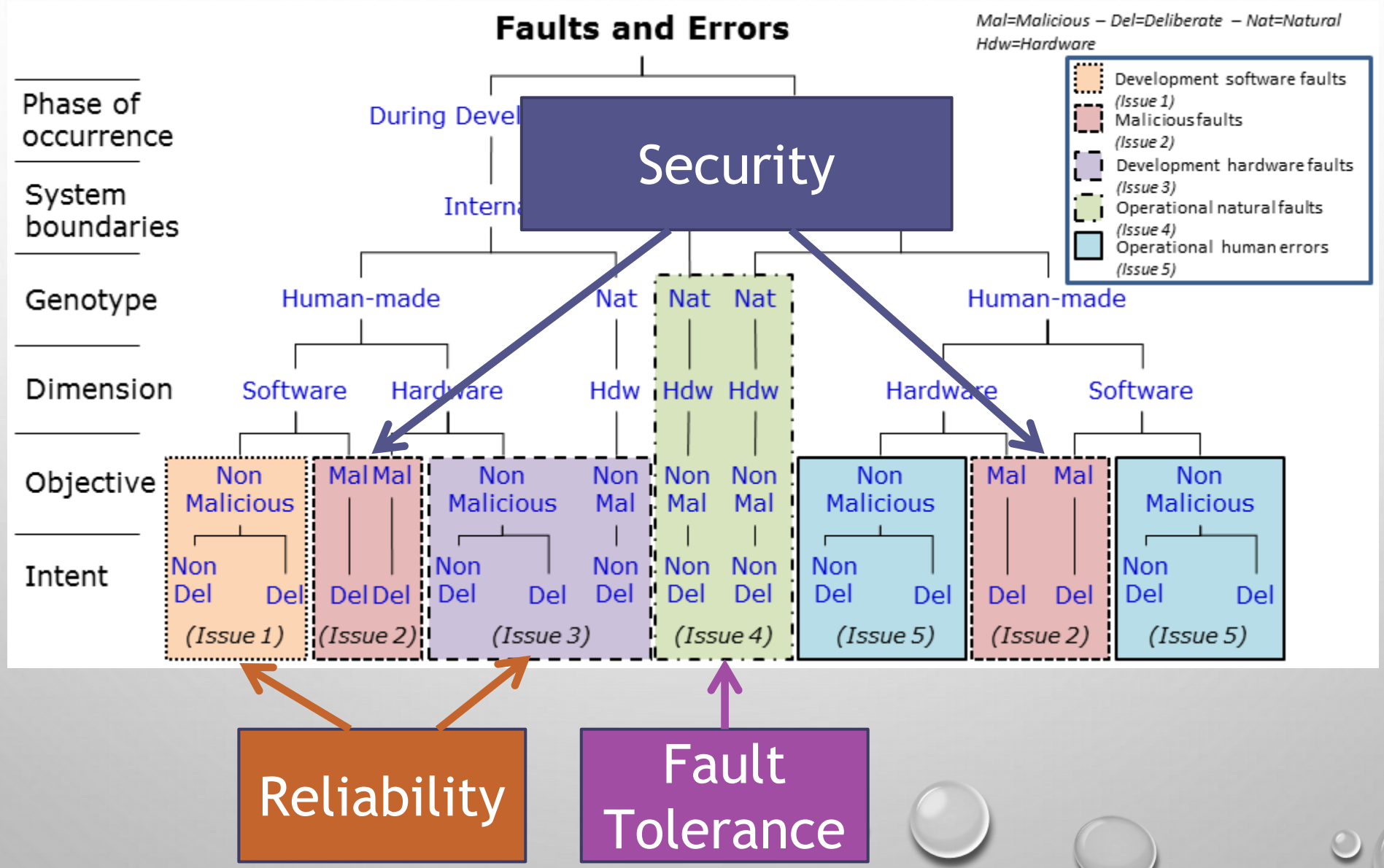


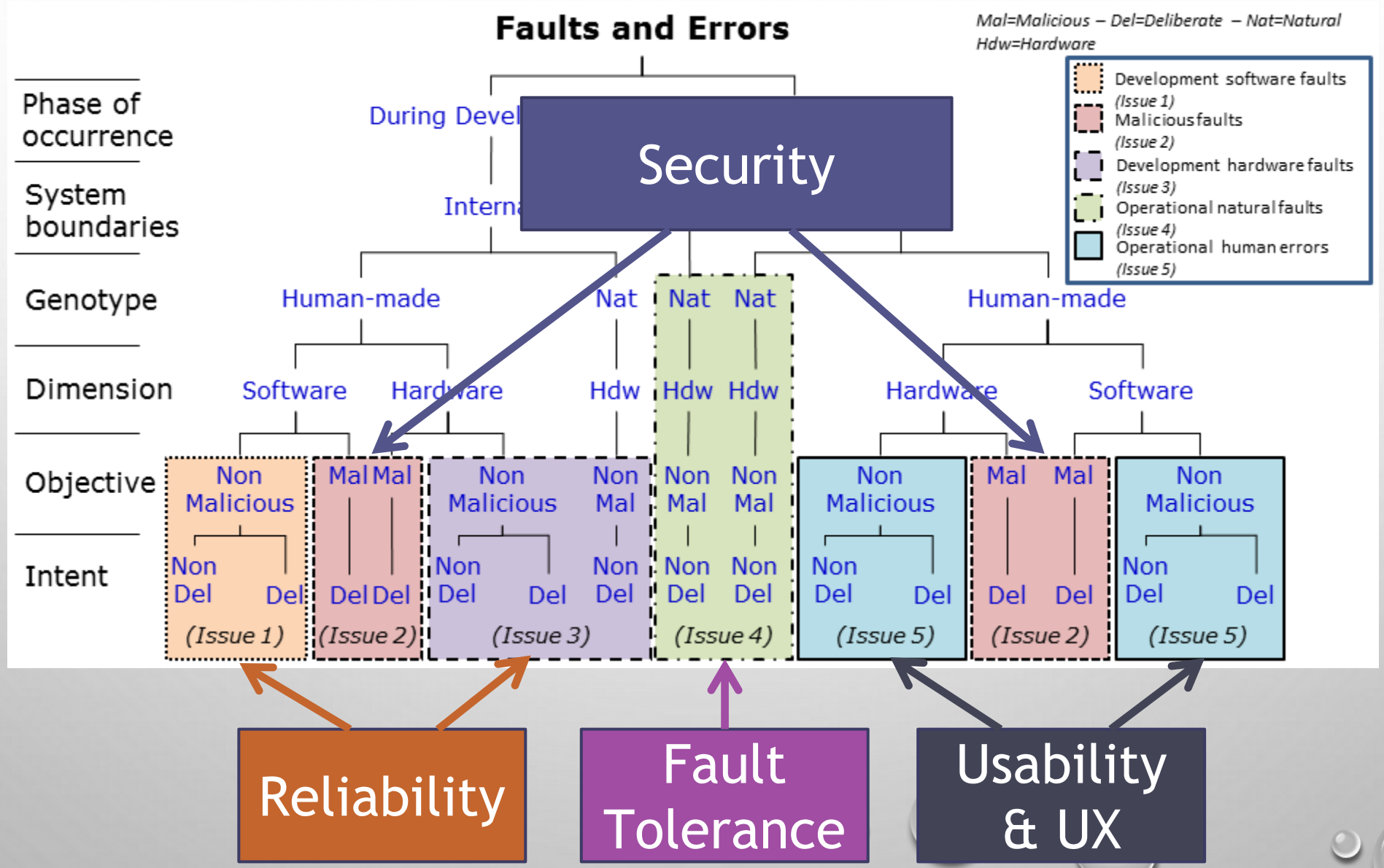
Adapted from: Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C. Basic concepts and taxonomy of dependable and secure computing. In IEEE Trans. on Dependable and Secure Computing, vol.1, no.1, pp. 11- 33, Jan.-March 2004

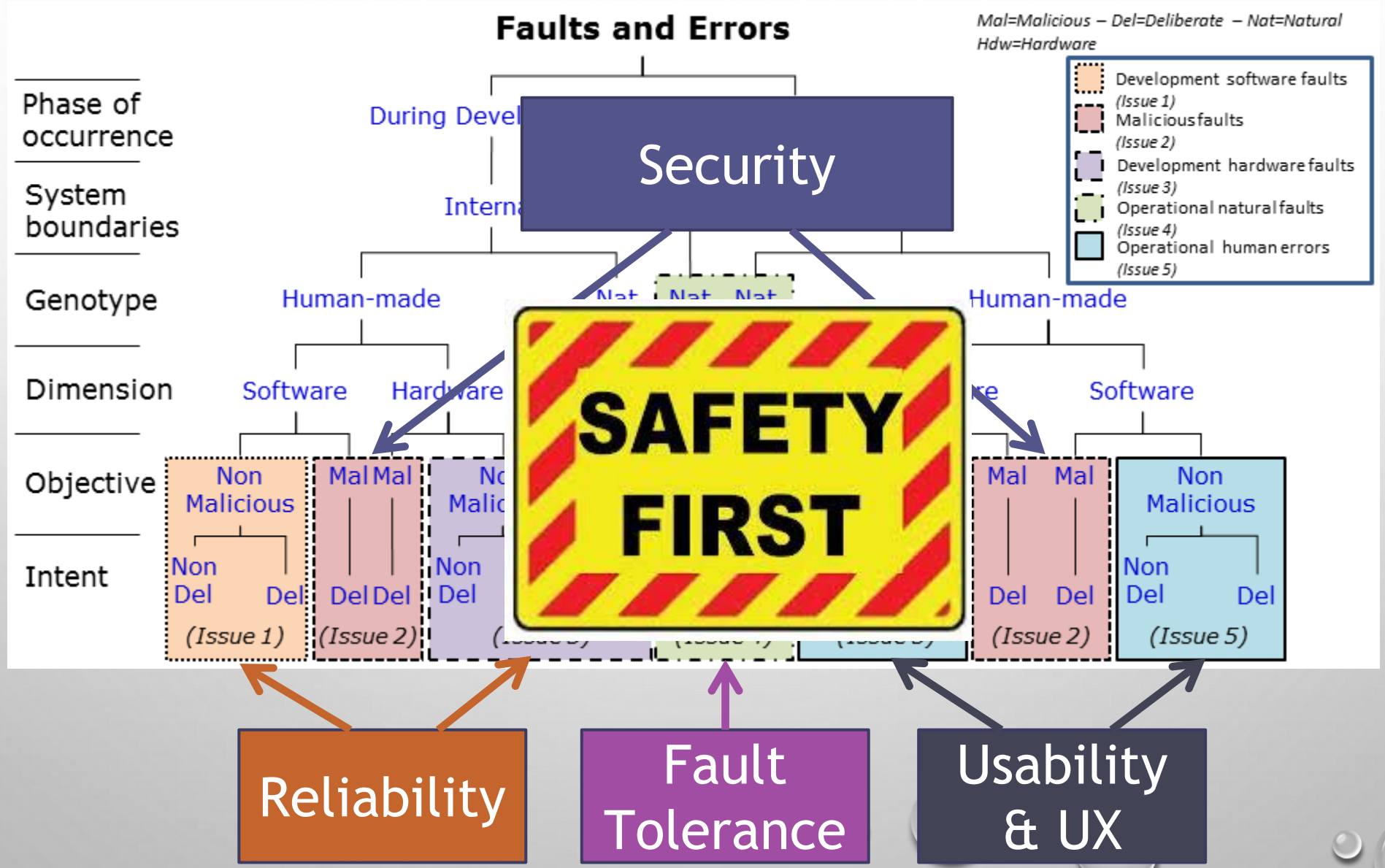


Adapted from: Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C. Basic concepts and taxonomy of dependable and secure computing. In IEEE Trans. on Dependable and Secure Computing, vol.1, no.1, pp. 11- 33, Jan.-March 2004

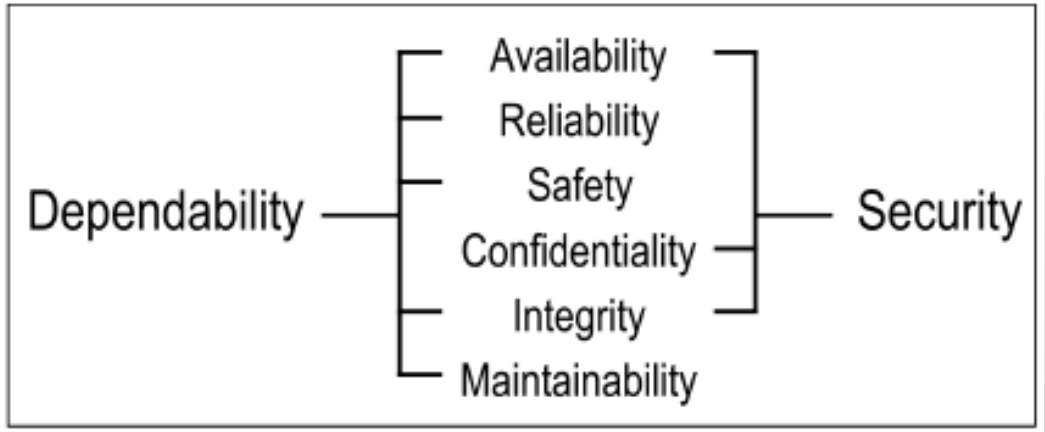

Faults and Errors
Mal=Malicious – Del=Deliberate – Nat=Natural
Hdw=Hardware
Development software faults
(Issue 1)
Malicious faults
(Issue 2)
Development hardware faults
(Issue 3)
Operational natural faults
(Issue 4)
Operational human errors
(Issue 5)
Phase of occurrence
System boundaries
Genotype
Dimension
Objective
Intent
During Devel
Intern
Security
Human-made
Nat
Nat Nat
Human-made
Software
Hardware
Hdw
Hdw Hdw
Hardware
Software
Non Malicious
Mal Mal
Non Malicious
Non Mal
Non Mal
Non Malicious
Mal Mal
Non Malicious
Non Del
Del
Del Del
Non Del
Del
Non Mal
Non Del
Non Mal
Non Del
Non Del
Del
Del Del
Non Del
Del
(Issue 1)
(Issue 2)
(Issue 3)
(Issue 4)
(Issue 5)
(Issue 2)
(Issue 5)
Reliability


Faults and Errors
Mal=Malicious – Del=Deliberate – Nat=Natural
Hdw=Hardware
Development software faults
(Issue 1)
Malicious faults
(Issue 2)
Development hardware faults
(Issue 3)
Operational natural faults
(Issue 4)
Operational human errors
(Issue 5)
Phase of occurrence
System boundaries
Genotype
Dimension
Objective
Intent
During Devel
Intern
Security
Human-made
Nat
Nat Nat
Human-made
Software
Hardware
Hdw
Hdw Hdw
Hardware
Software
Non Malicious
Mal Mal
Non Malicious
Non Mal
Non Mal
Non Malicious
Mal Mal
Non Malicious
Non Del
Del
Del Del
Non Del
Del
Non Del
Non Del
Non Del
Non Del
Del
Del Del
Non Del
Del
(Issue 1)
(Issue 2)
(Issue 3)
(Issue 4)
(Issue 5)
(Issue 2)
(Issue 5)
Reliability
Fault Tolerance


Faults and Errors
Mal=Malicious – Del=Deliberate – Nat=Natural
Hdw=Hardware
Development software faults
(Issue 1)
Malicious faults
(Issue 2)
Development hardware faults
(Issue 3)
Operational natural faults
(Issue 4)
Operational human errors
(Issue 5)
Phase of occurrence
System boundaries
Genotype
Dimension
Objective
Intent
During Devel
Intern
Security
Human-made
Nat
Nat
Nat
Human-made
Software
Hardware
Hdw
Hdw
Hdw
Hardware
Software
Non Malicious
Mal
Mal
Non Malicious
Non Mal
Non Mal
Non Malicious
Mal
Mal
Non Malicious
Non Del
Del
Del
Del
Non Del
Del
Non Del
Non Del
Non Del
Non Del
Del
Del
Del
Non Del
Del
(Issue 1)
(Issue 2)
(Issue 3)
(Issue 4)
(Issue 5)
(Issue 2)
(Issue 5)
Reliability
Fault Tolerance
Usability & UX


Faults and Errors
Mal=Malicious – Del=Deliberate – Nat=Natural
Hdw=Hardware
Development software faults
(Issue 1)
Malicious faults
(Issue 2)
Development hardware faults
(Issue 3)
Operational natural faults
(Issue 4)
Operational human errors
(Issue 5)
Phase of occurrence
System boundaries
Genotype
Dimension
Objective
Intent
Security
During Devel
Interna
Human-made
Human-made
Software
Hardware
Software
Non Malicious
Mal Mal
Non Malicious
Non Del
Del
Del Del
Non Del
Mal Mal
Non Malicious
Del Del
Non Del
Del
(Issue 1)
(Issue 2)
(Issue 2)
(Issue 5)
SAFETY FIRST
Reliability
Fault Tolerance
Usability & UX

DEPENDABILITY ET SECURITY

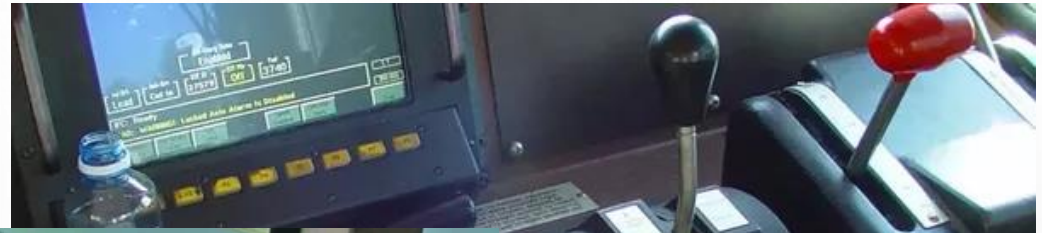


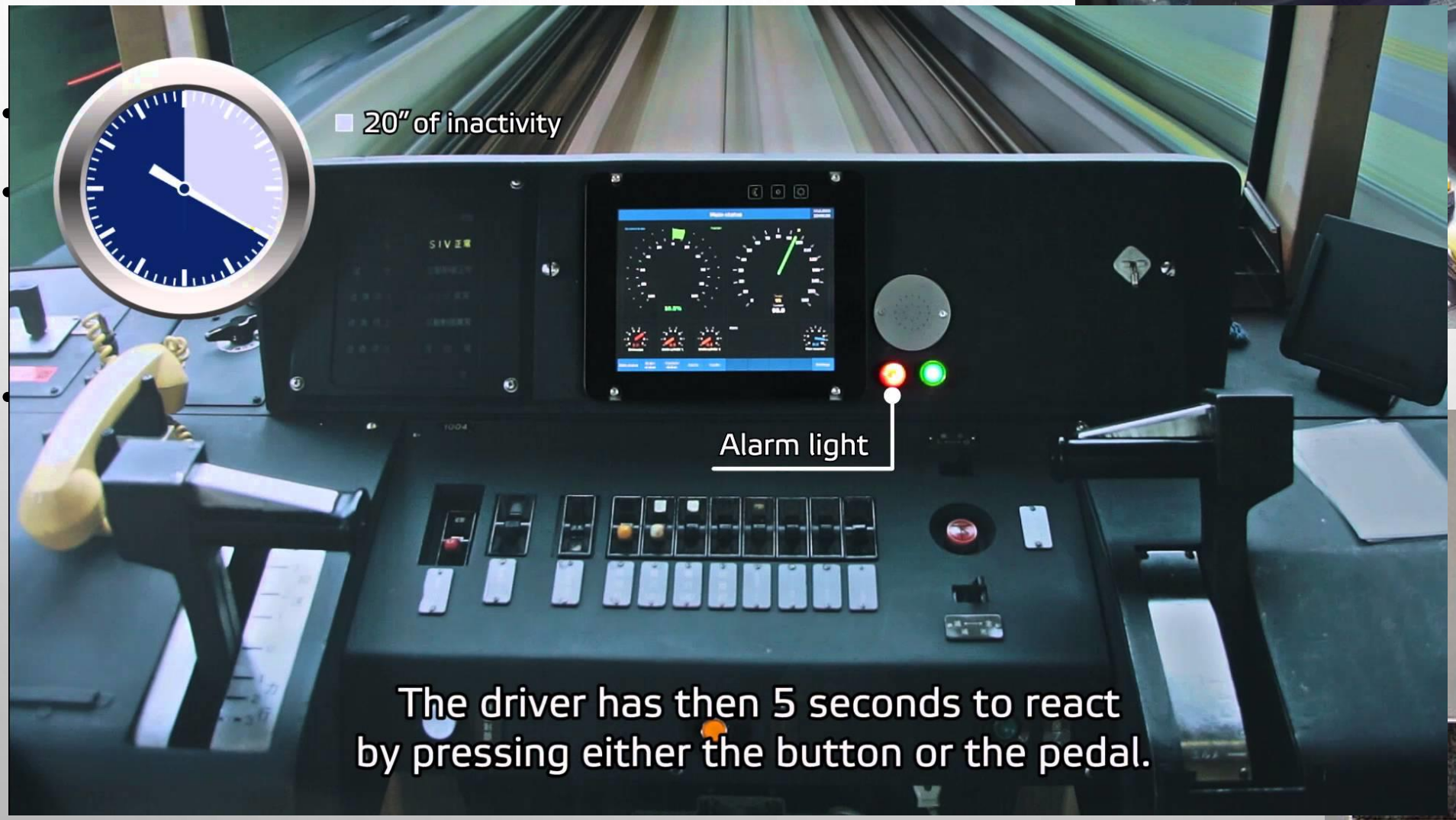
- **availability**: readiness for correct service.
- **reliability**: continuity of correct service.
- **safety**: absence of catastrophic consequences on the user(s) and the environment.
- **integrity**: absence of improper system alterations.
- **maintainability**: ability to undergo modifications and repairs

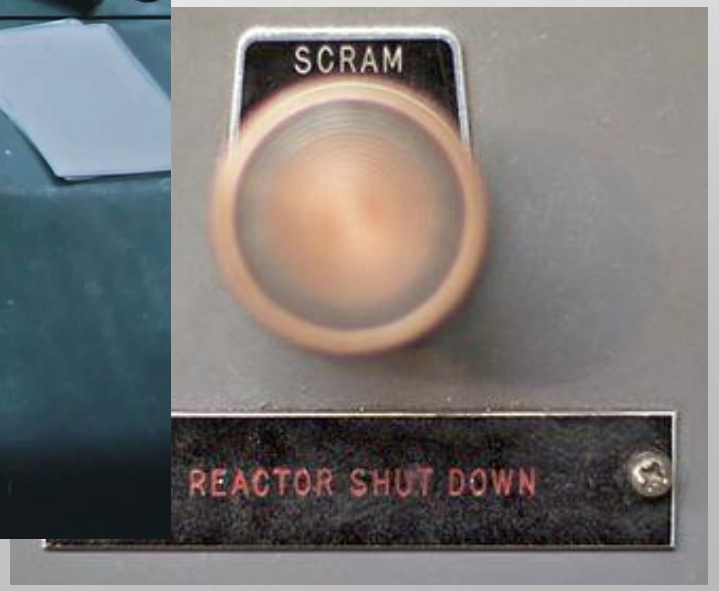
FAIL-SAFE SYSTEMS

- A system that is still safe when it fails
- Exemple 1: trains
  - Anciens systèmes
  - Nouveaux systèmes (dead man switch- alerter)
- Exemple 2: centrales nucléaires

FAIL-SAFE SYSTEMS







BEYOND USABILITY – SAFE, USABLE, RELIABLE AND EVOLVABLE

- Usability
  - Ease of use, ease of learn, reduce errors, user satisfaction, efficiency, effectiveness, …
- Reliability
  - No default, one use does not impact future uses e.g. no wearability (reliability over time = resilience)
- Safety
  - Not put human life (or mission life) at stake
- Evolvability
  - Users needs are always incomplete (or even wrong)
  - User needs evolve through time
  - User needs evolve with tools provided

CONCEPTION CENTRÉE UTILISATEUR (NORME ISO 13407 NOW 9241-PART 210)

- Comment impliquer l'utilisateur dans le processus de conception ?
  - Insertion de l'utilisateur dans l'équipe de développement
  - Implication périodique
- Problèmes
  - Incohérences
  - Incomplétudes

STANDARDS RELIABILITY

- ESARR (ATM)
- DO 178-B software considerations in airborne systems and equipment certification (Aéronautique)
- EASA Certif. Specs. 25 (Aéronautique)
- IEC 62508 (Electrotechnique)
- DO 254 (design assurance guidance for airborne electronic hardware)

European Aviation Safety Agency

Certification Specifications for Large Aeroplanes

CS-25

Amendment 4
27 December 2007

CS–25 BOOK 1

SUBPART F – EQUIPMENT

GENERAL

CS 25.1301 Function and installation
(See AMC 25.1301)

Each item of installed equipment must –

(a) Be of a kind and design appropriate to its intended function;

(b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See AMC 25.1301(b).)

(c) Be installed according to limitations specified for that equipment.

[Amdt. No.:25/2]

CS 25.1302 Installed systems and equipment for use by the flight crew
(See AMC 25.1302)

This paragraph applies to installed equipment intended for flight-crew members' use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can safely perform their tasks associated with its intended function by meeting the following requirements:

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

(b) Flight deck controls and information intended for flight crew use must:

(1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

(2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

(3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(c) Operationally-relevant behaviour of the installed equipment must be:

(1) Predictable and unambiguous, and

(2) Designed to enable the flight crew to intervene in a manner appropriate to the task.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight crew is acting in good faith. This sub-paragraph (d) does not apply to skill-related errors associated with manual control of the aeroplane.

[Amdt. No.:25/3]

CS 25.1303 Flight and navigation instruments

(a) The following flight and navigation instruments must be installed so that the instrument is visible from each pilot station:

(1) A free-air temperature indicator or an air-temperature indicator which provides indications that are convertible to free-air temperature.

(2) A clock displaying hours, minutes, and seconds with a sweep-second pointer or digital presentation.

(3) A direction indicator (non-stabilised magnetic compass).

(b) The following flight and navigation instruments must be installed at each pilot station:

(1) An airspeed indicator. If airspeed limitations vary with altitude, the indicator must have a maximum allowable airspeed indicator showing the variation of V_{MO} with altitude.

(2) An altimeter (sensitive).

(3) A rate-of-climb indicator (vertical speed).

(4) A gyroscopic rate of turn indicator combined with an integral slip-skid indicator (turn-and-bank indicator) except that only a slip-skid indicator is required on aeroplanes with a third attitude instrument system usable through flight attitudes of 360° of pitch and roll, which is powered from a source independent of the electrical generating system and continues reliable operation for a minimum of 30 minutes after total failure of the electrical generating system, and is installed in accordance with CS 25.1321 (a).

CS–25 BOOK 1

SUBPART F – EQUIPMENT

GENERAL

CS 25.1301 Function and installation
(See AMC 25.1301)

Each item of installed equipment must –

(a) Be of a kind and design appropriate to its intended function;

(b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See AMC 25.1301(b).)

(c) Be installed according to limitations specified for that equipment.

[Amdt. No.:25/2]

CS 25.1302 Installed systems and equipment for use by the flight crew
(See AMC 25.1302)

This paragraph applies to installed equipment intended for flight-crew members' use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can safely perform their tasks associated with its intended function by meeting the following requirements:

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

(b) Flight deck controls and information intended for flight crew use must:

(1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

(2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

(3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(c) Operationally-relevant behaviour of the installed equipment must be:

(1) Predictable and unambiguous, and

(2) Designed to enable the flight crew to intervene in a manner appropriate to the task.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight

indications that are convertible to free-air temperature.

(2) A clock displaying hours, minutes, and seconds with a sweep-second pointer or digital presentation.

(3) A direction
magnetic compass).

(b) The following
instruments must be insta

(1) An airspe
limitations vary with
have a maximum all
showing the variation

(2) An altimet

(3) A rate-of-
speed).

(4) A gyrosco
combined with an i
(turn-and-bank indicat
skid indicator is requ
third attitude instrum
flight attitudes of 360°
powered from a so
electrical generating
reliable operation for a minimum of 30 minutes after total failure of the electrical generating system, and is installed in accordance with CS 25.1321 (a).

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight crew is acting in good faith. This sub-paragraph (d) does not apply to skill-related errors associated with manual control of the aeroplane.

CS–25 BOO

SUBPART F – EQU

GENERAL

CS 25.1301 Function and installation
(See AMC 25.1301)

Each item of installed equipment must –

(a) Be of a kind and design appropriate to its intended function;

(b) Be labelled as to its identification, function, or operating limitations, or any applicable combination of these factors. (See AMC 25.1301(b).)

(c) Be installed according to limitations specified for that equipment.

[Amdt. No.:25/2]

CS 25.1302 Installed systems and equipment for use by the flight crew
(See AMC 25.1302)

This paragraph applies to installed equipment intended for flight-crew members' use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can safely perform their tasks associated with its intended function by meeting the following requirements:

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

(b) Flight deck controls and information intended for flight crew use must:

(1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

(2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

(3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(c) Operationally-relevant behaviour of the installed equipment must be:

(1) Predictable and unambiguous, and

errors resulting from tl
interactions with the
reasonably expected in se
crew is acting in good fai
does not apply to skill-rel
manual control of the aero

[Amdt. No.:25/3]

CS 25.1303 Flig
instr

(a) The following
instruments must be insta
is visible from each pilot s

(1) A free-air t
air-temperature indi
indications that are
temperature.

(2) A clock d
and seconds with a
digital presentation.

(3) A direction
magnetic compass).

(b) The following
instruments must be instal

(1) An airspee
limitations vary with a
have a maximum allo
showing the variation c

(2) An altimete

(3) A rate-of-clim
speed).

(4) A gyroscopic
combined with an integr
(turn-and-bank indicator)
skid indicator is required
third attitude instrument
flight attitudes of 360° of p
powered from a source
electrical generating sy
reliable operation for a m
after total failure of the
system, and is installed i
25.1321 (a).

1–F–1

(a) Flight deck controls must be installed to allow accomplishment of these tasks and information necessary to accomplish these tasks must be provided.

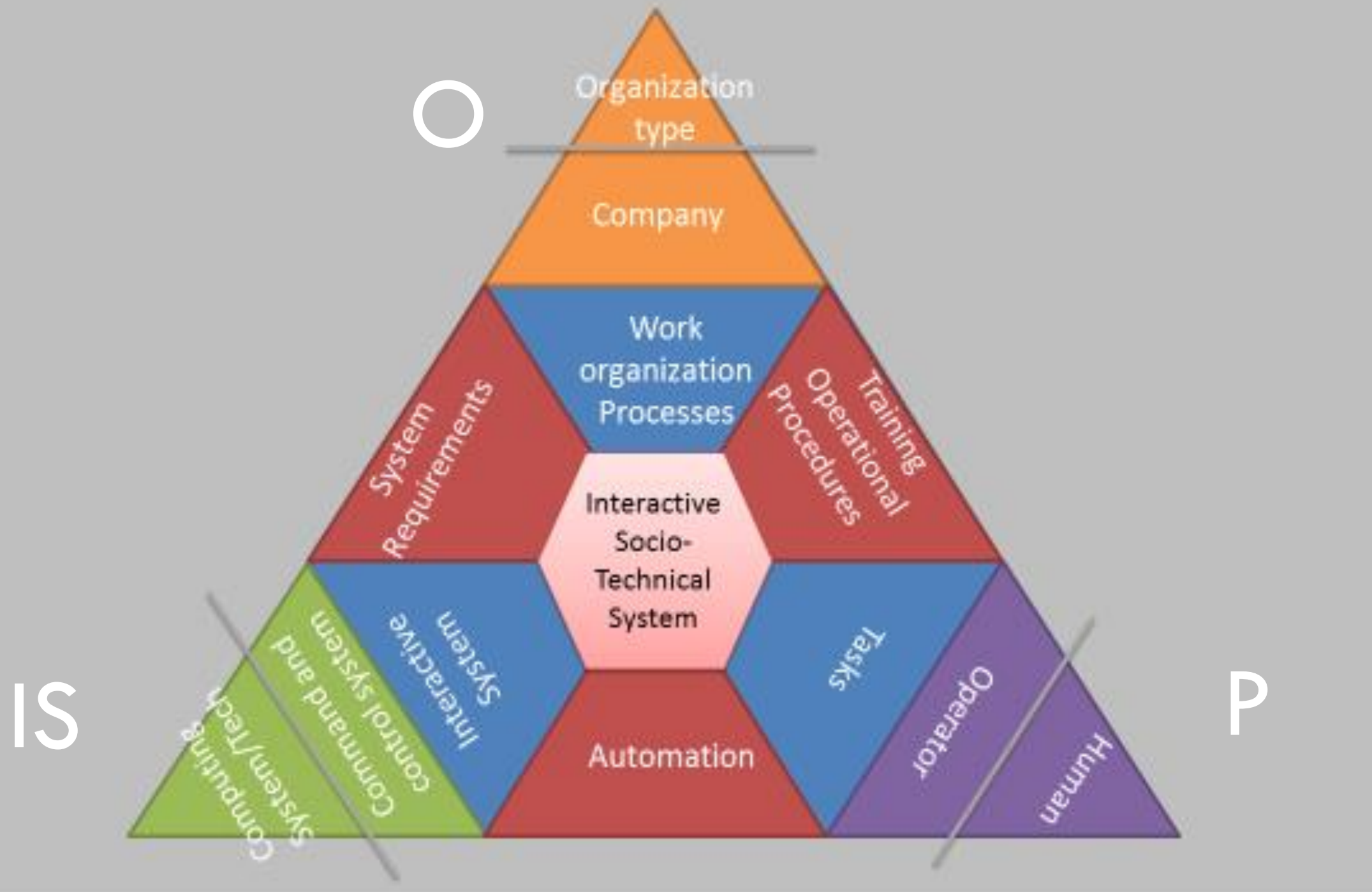
(b) Flight deck controls and information intended for flight crew use must:

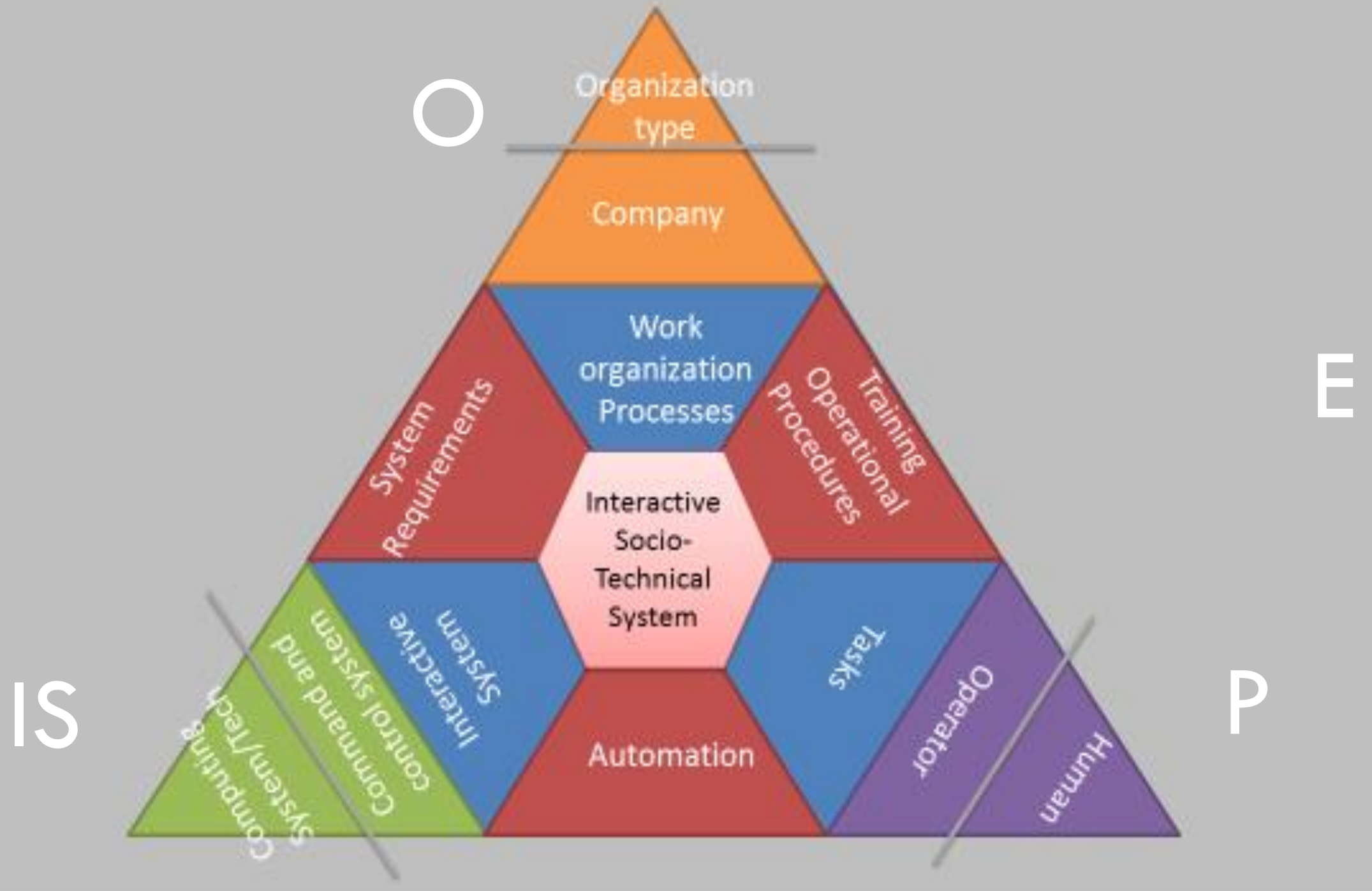
(1) Be presented in a clear and unambiguous form, at resolution and precision appropriate to the task.

(2) Be accessible and usable by the flight crew in a manner consistent with the urgency, frequency, and duration of their tasks, and

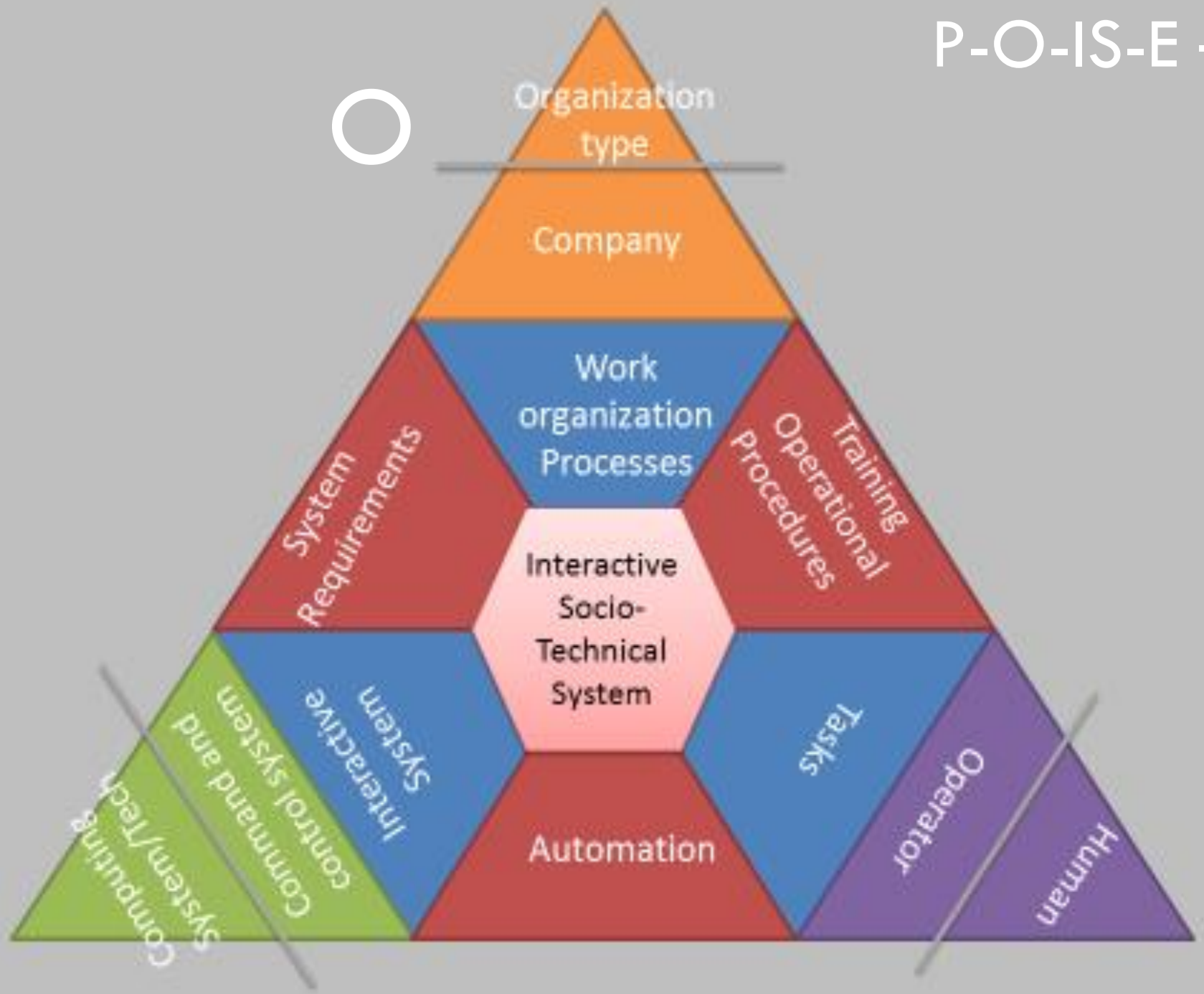
(3) Enable flight crew awareness, if awareness is required for safe operation, of the effects on the aeroplane or systems resulting from flight crew actions.

(d) To the extent practicable, installed equipment must enable the flight crew to manage errors resulting from the kinds of flight crew interactions with the equipment that can be reasonably expected in service, assuming the flight crew is acting in good faith. This sub-paragraph (d) does not apply to skill-related errors associated with


O
Organization type
Company
Work organization Processes
System Requirements
Training Operational Procedures
Interactive Socio-Technical System
Interactive System
Tasks
Automation
Computing System/Tech
Command and control system
Operator
Human
IS
P

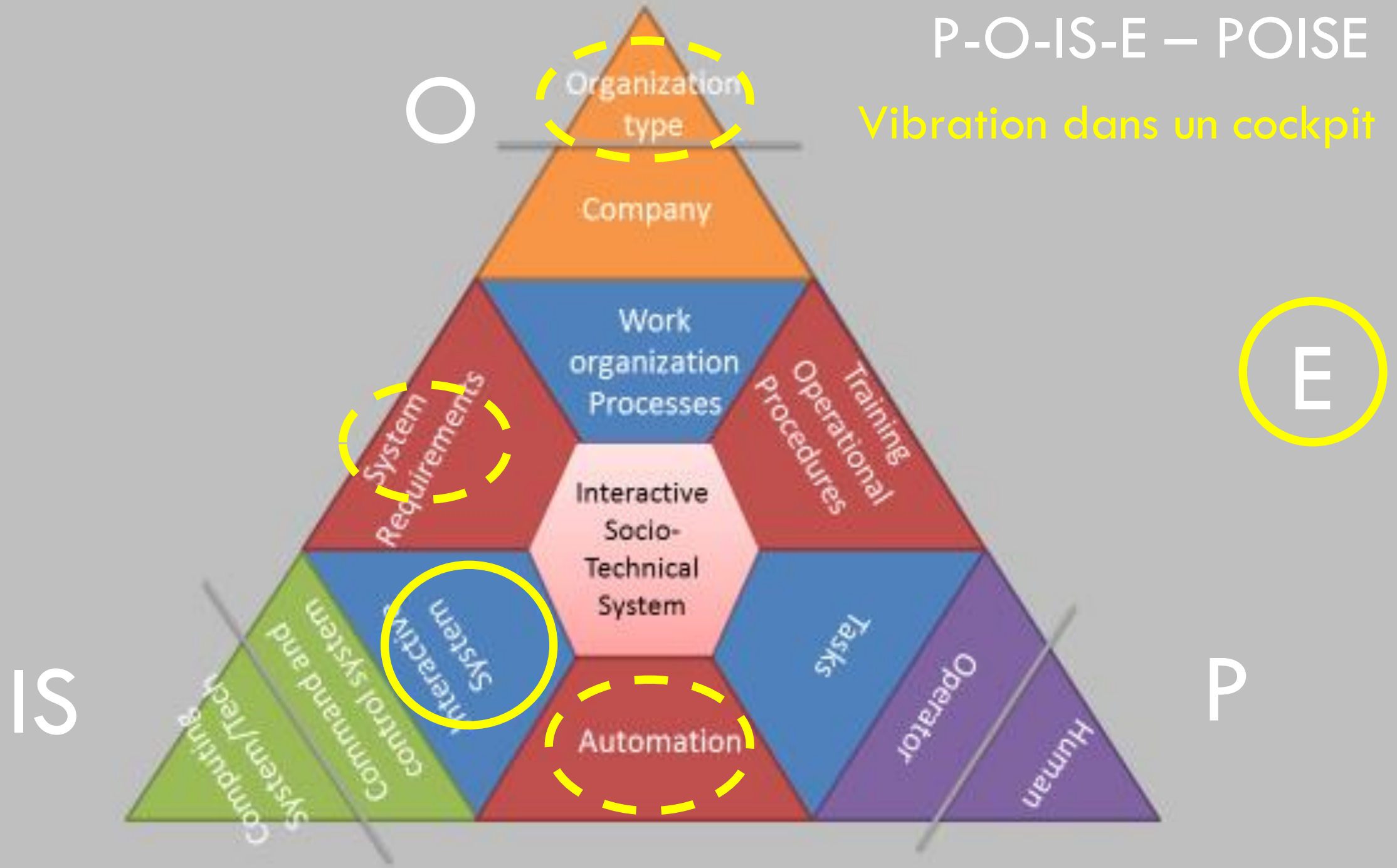

O
Organization type
Company
Work organization Processes
System Requirements
Training Operational Procedures
Interactive Socio-Technical System
Interactive System
Computing System/Tech
Command and control system
Automation
Tasks
Operator
Human
E
IS
P

P-O-IS-E – POISE


O
Organization type
Company
Work organization Processes
System Requirements
Training Operational Procedures
Interactive Socio-Technical System
Interactive System
Computing System/Tech
Command and control system
Automation
Tasks
Operator
Human
E
IS
P

P-O-IS-E – POISE

Vibration dans un cockpit



P-O-IS-E – POISE

A unified framework integrating People, Organizations, Interactive Systems and the **E**nvironment

poise

noun [U] • **UK** /pɔɪz/ **US** /pɔɪz/ APPROVING

calm confidence in a person's way of behaving, or a quality of grace (= moving in an attractive way) and balance in the way a person holds or moves their body:

He looked embarrassed for a moment, then quickly regained his poise.

Her confidence and poise show that she is a top model.

IEC 62508

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 62508

October 2010

ICS 03.120.01

English version

Guidance on human aspects of dependability
(IEC 62508:2010)

Lignes directrices relatives aux facteurs humains dans la sûreté de fonctionnement
(CEI 62508:2010)

Leitlinien zu den menschlichen Aspekten der Zuverlässigkeit
(IEC 62508:2010)

CENELEC
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Ref. No. EN 62508:2010 E

IEC 62508

ÖVE/ÖNORM EN 62508:2011

IEC 62508

Edition 1.0 2010-06

INTERNATIONAL STANDARD

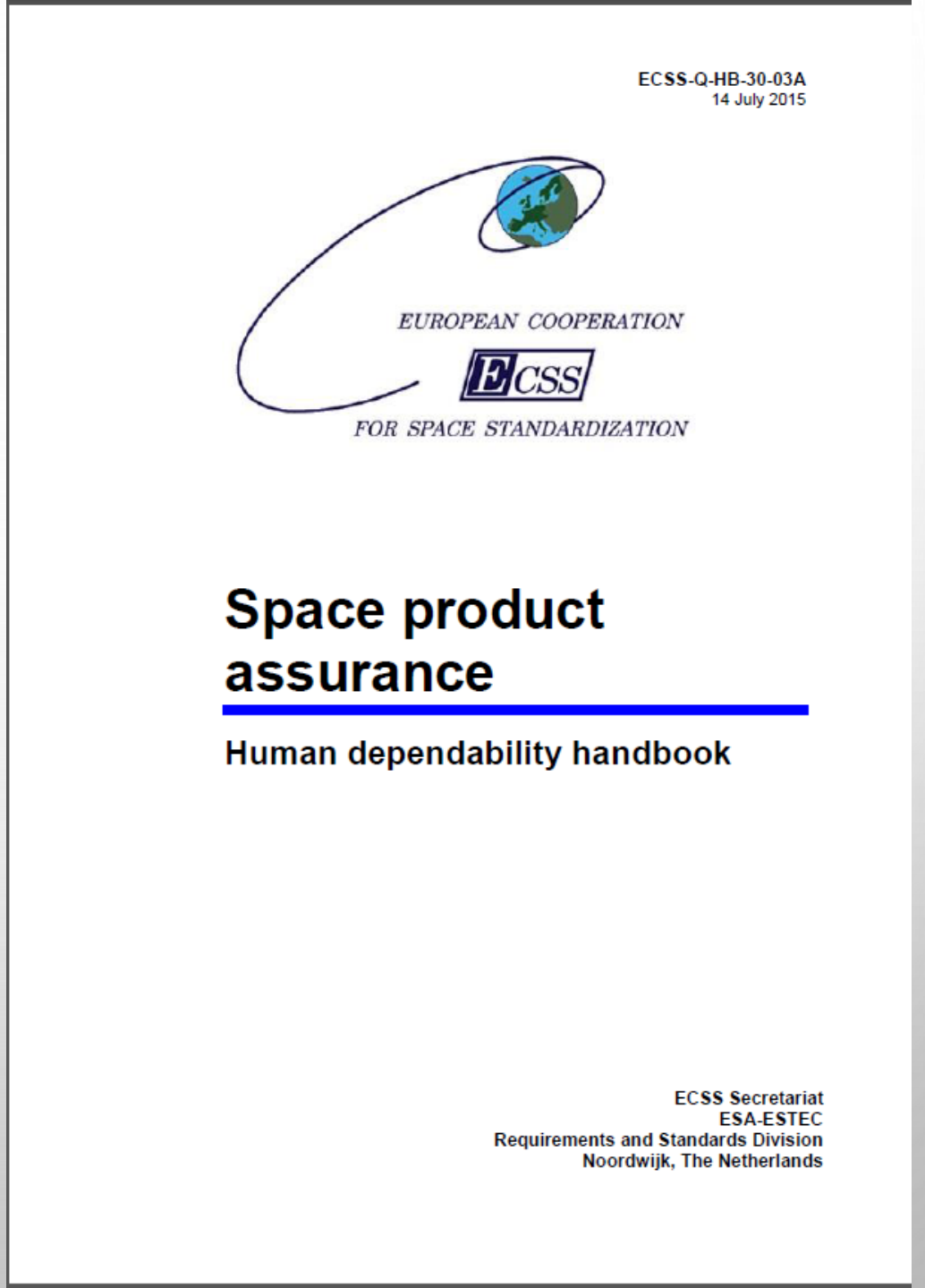
NORME INTERNATIONALE

colour inside

Guidance on human aspects of dependability

Lignes directrices relatives aux facteurs humains dans la sûreté de fonctionnement

ESA HUDEP HANDBOOK



EXEMPLES DE PROBLÈME (UCD VERSUS SAFETY)

- The customer defines System-A to be used as "Advisory only". This fact defines what we call the "Intended Use" of the system
- This Intended Use is the basis of System-A safety analysis, resulting with few hazards marked with MAJOR severity
- The operator of System-X is quite clever to use the system FAR BEYOND the Intendent Use
- If you analyze this "Extra-usage", you find hazards typed as CATASTROPHIC severity, and the mitigation of those hazards is quite expensive
- We do wish to protect the operator activities. However, the customer will not pay the price of FAR BEYOND the Intendent Use mitigation

EXEMPLES DE PROBLÈME (UCD VERSUS SAFETY)

- The customer defines System-A to be used as "Advi… …ll the "Intended Use" of the system
- This Intended Use is the basis of … …with MAJOR severity
- The operat…
- If you … …severity, and the mitigatio…
- We do wis… …he customer will not pay the price of FAR BEYOND the…

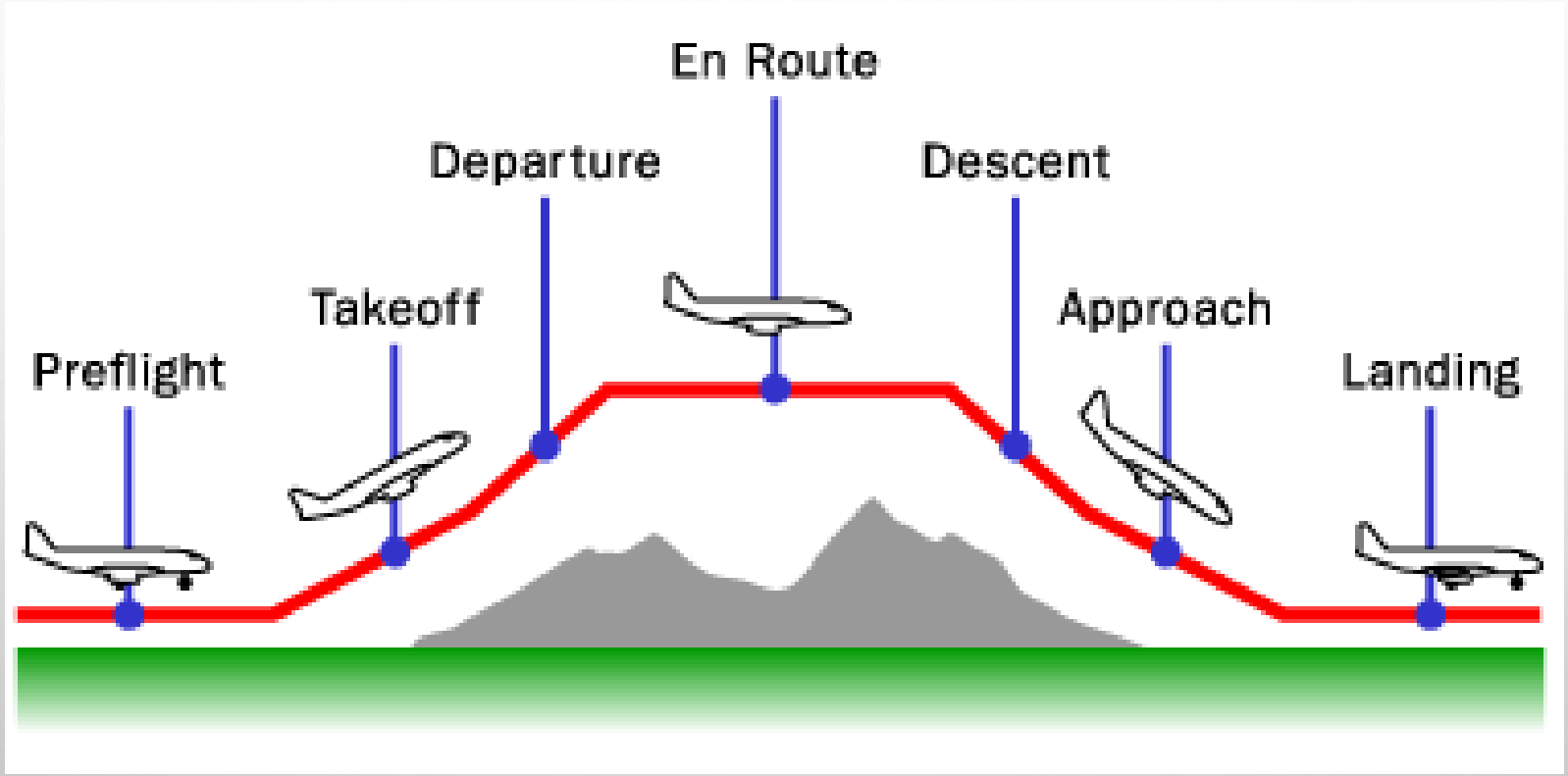
Exemple: conduire la voiture à 40km/h
Je veux conduire la voiture à 175km/h
Même véhicule, l'utilisation change (changement par rapport à l'utilisation attendue)

EXEMPLES DE PROBLÈME (UCD VERSUS SAFETY)

- The customer defines System-A to be used as "Advi… …ll the "Intended Use" of the system
- This Intended Use is the basis of … …with MAJOR severity
- The operat…
- If you … …severity, and the mitigatio…
- We do wis… …e customer will not pay the price of FAR BEYOND the…

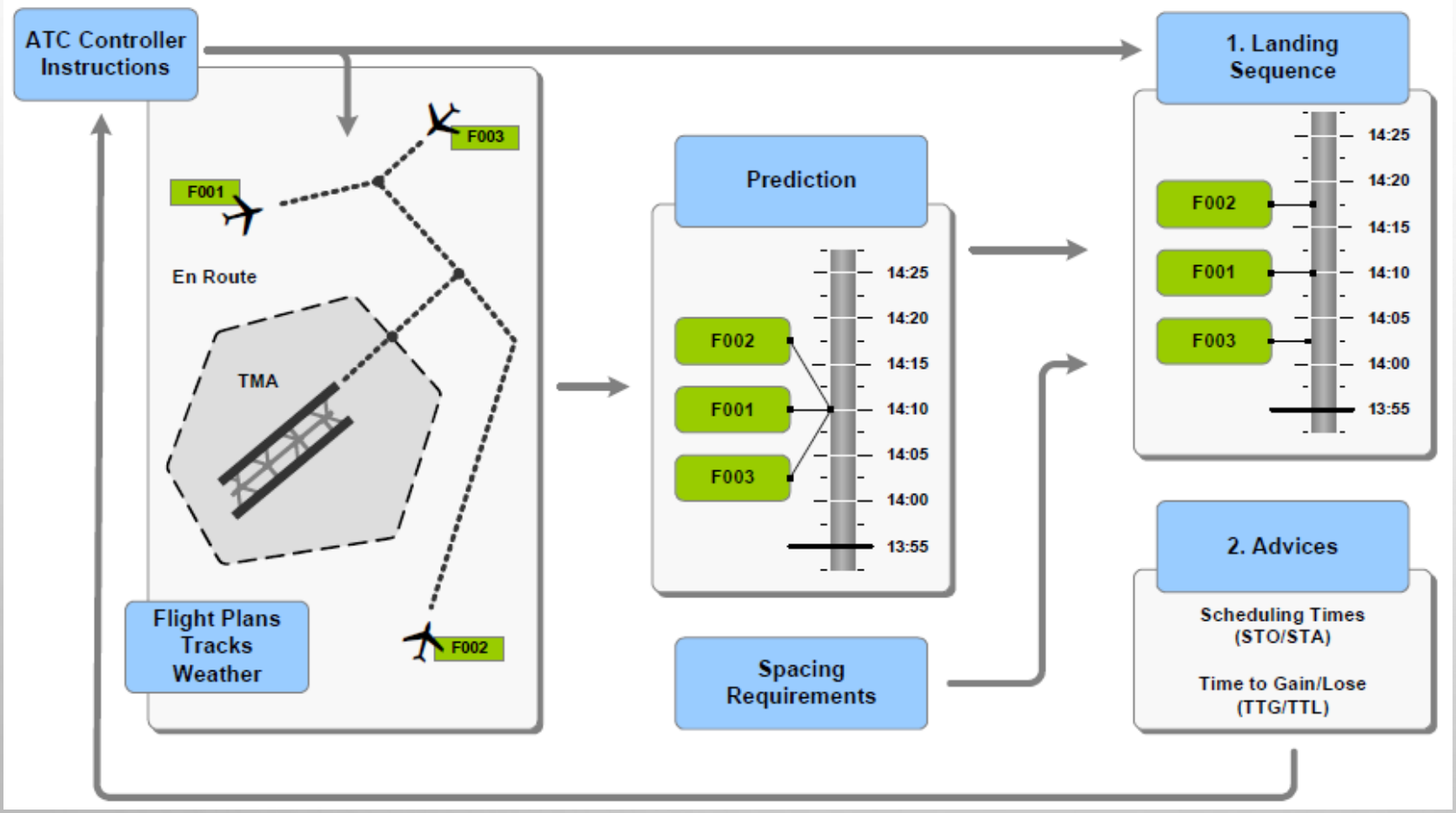
Trouvez un autre exemple récent

AIR TRAFFIC MANAGEMENT

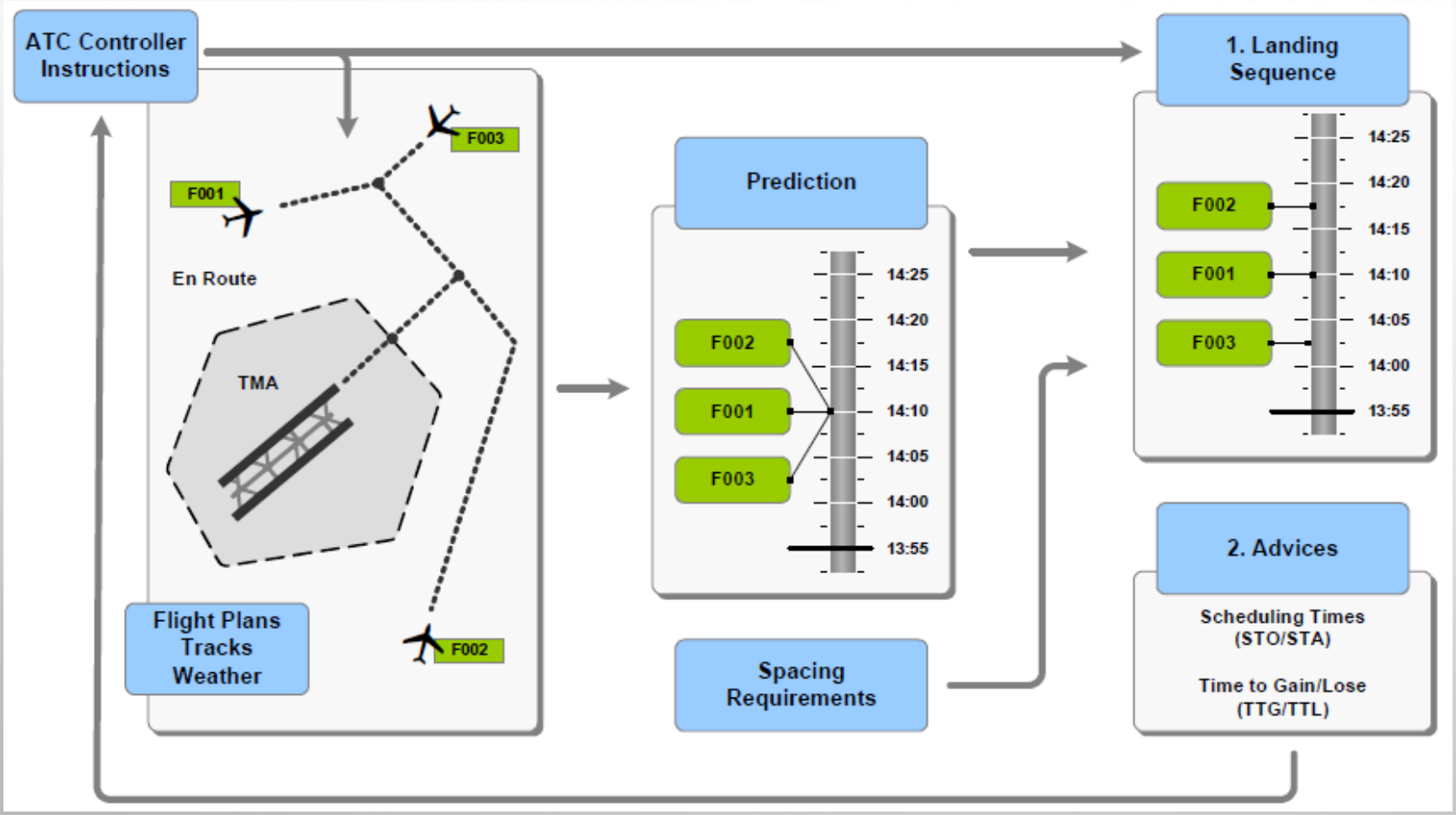


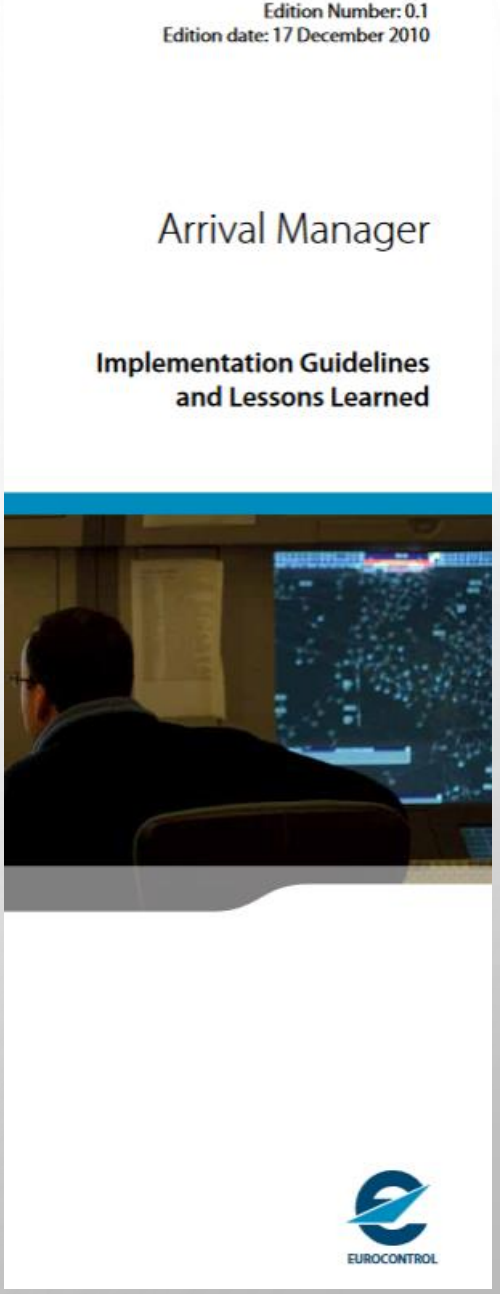
Tower Controler | Approach Controler | En route Controler | Approach Controler | Tower Controler

ARRIVAL MANAGER - AMAN



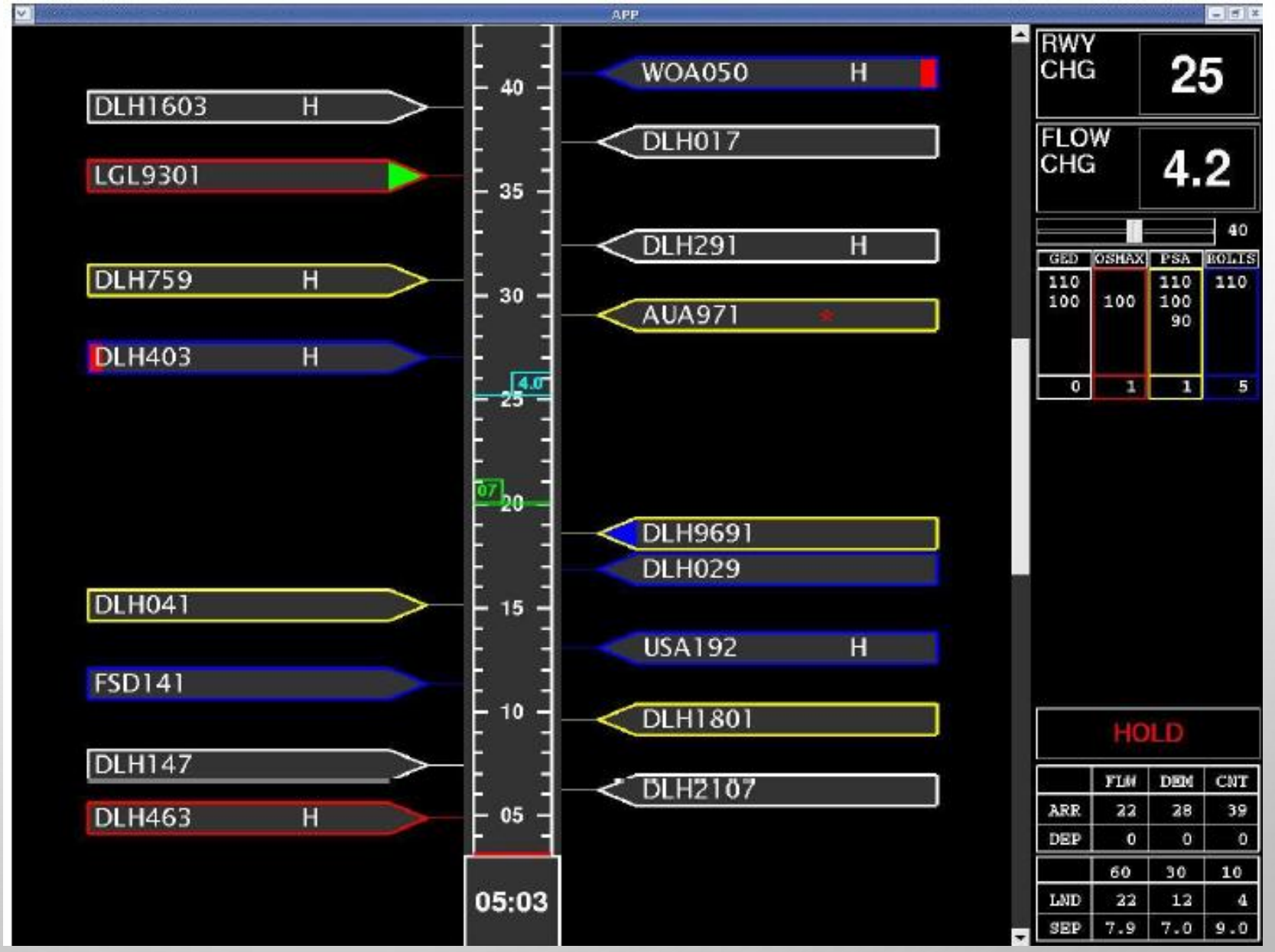
ARRIVAL MANAGER - AMAN





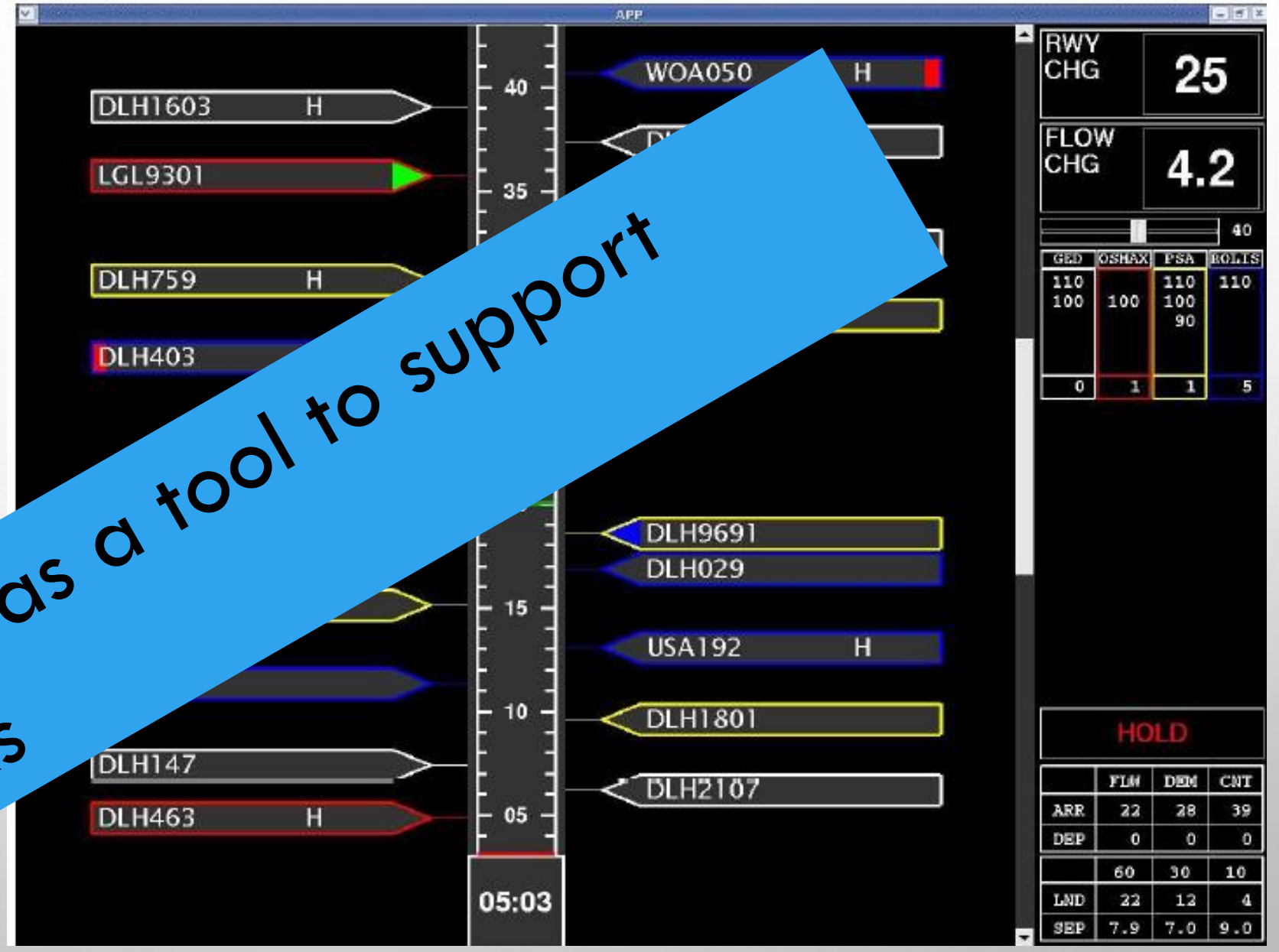
AMAN

- First used as a tool to support users (advisory)
- Now a system that must be followed



AMAN

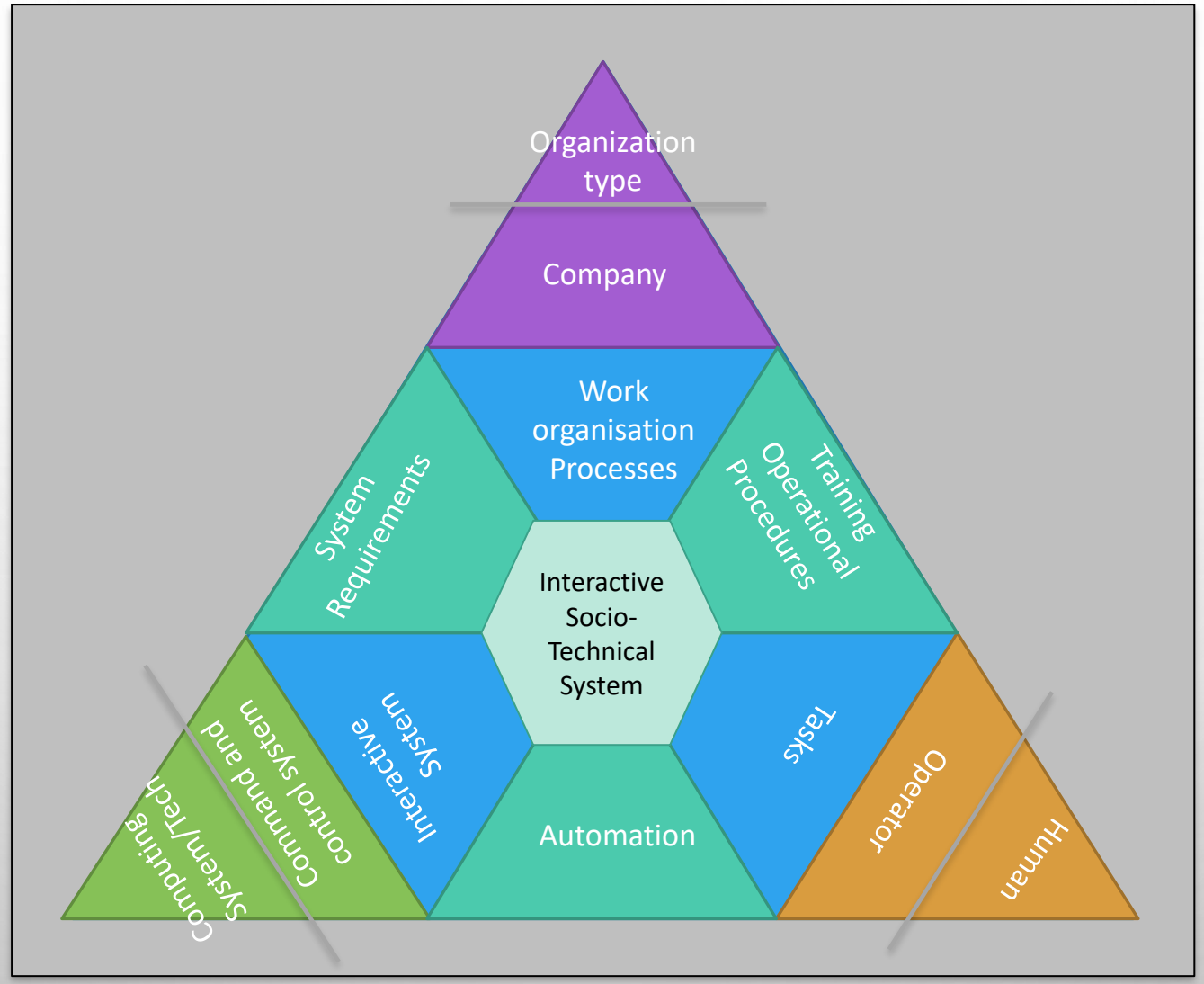
- First used as a tool to support users (advisory)
- Now a system that must be followed



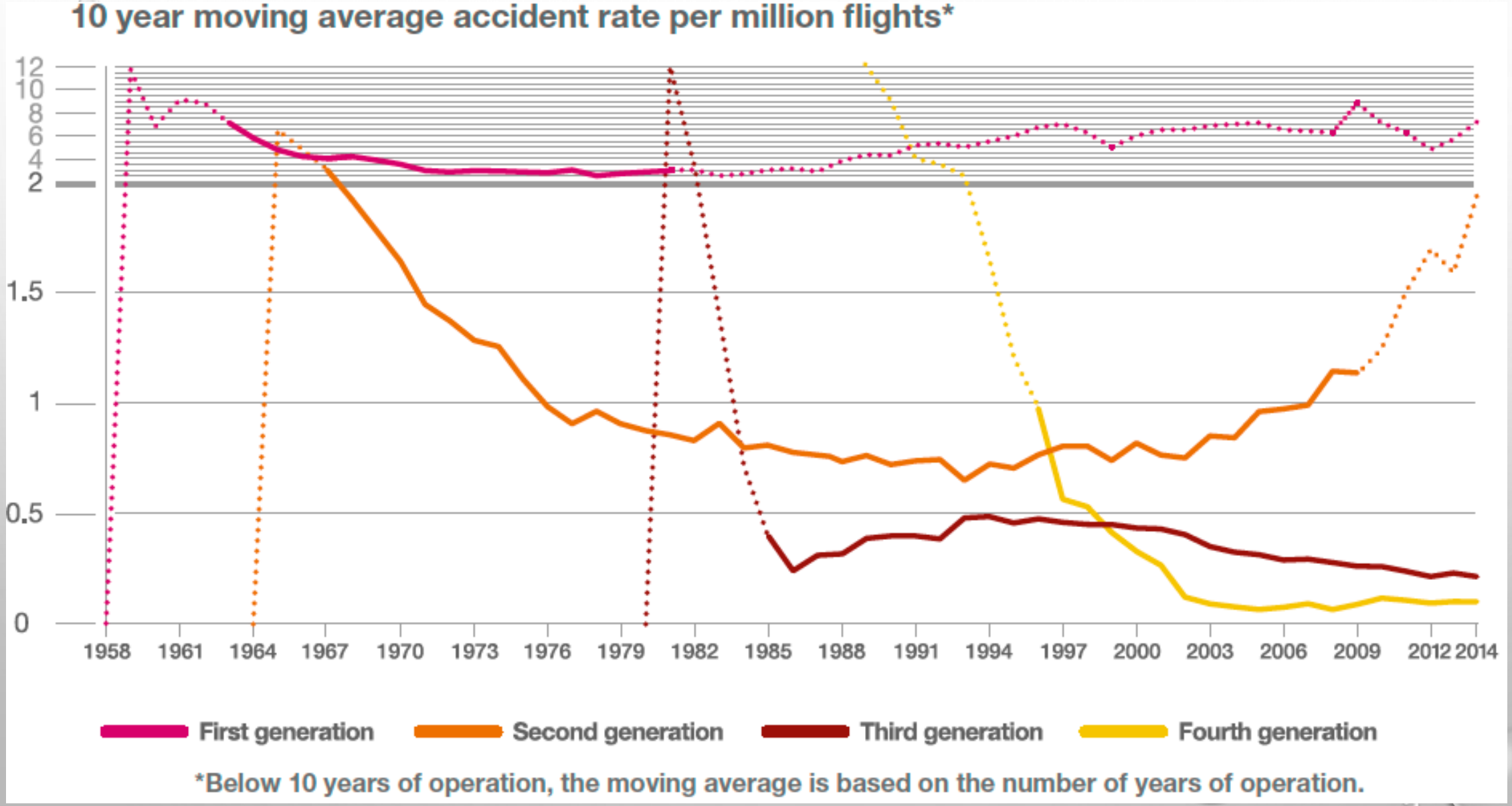
Automation as a tool to support users' tasks

EXEMPLE DU SYSTÈME AMAN EN ATM

- Advisory system for ATC
- Moved to primary system by making mandatory for the ATC to follow AMAN
- Impact of Failures
  - Impact (nowadays) – all is fine
  - Impact (future) – when ATC don't control manually anymore


Organization type
Company
Work organisation Processes
System Requirements
Training Operational Procedures
Interactive Socio-Technical System
Interactive System
Computing System/Tech
Command and control system
Automation
Tasks
Operator
Human

AVIATION FATAL ACCIDENT RATE



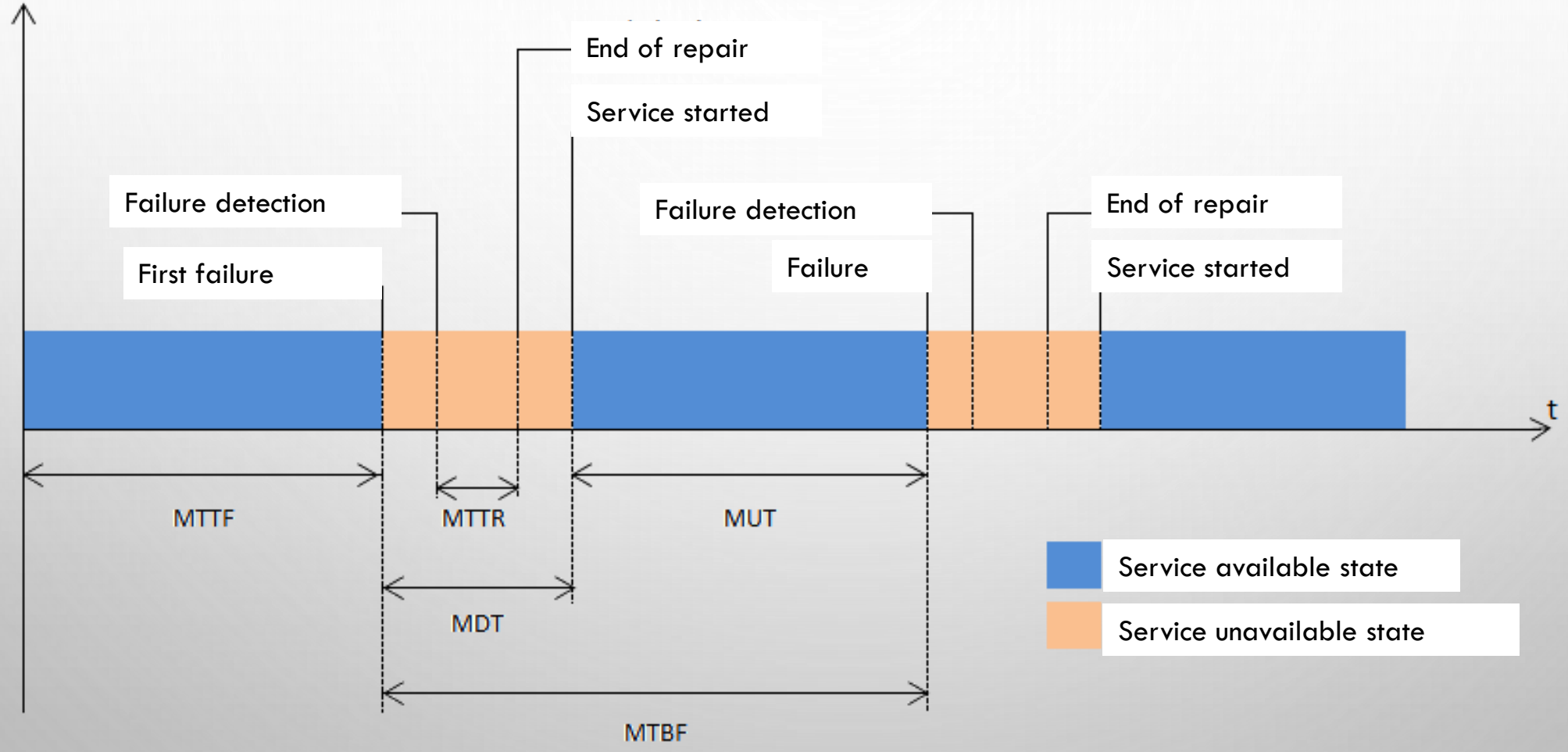
FAILURES

MTTF: Mean Time To Fail

MTTR: Mean Time To Recover

MDT: Mean Down Time

MUT: Mean Up Time



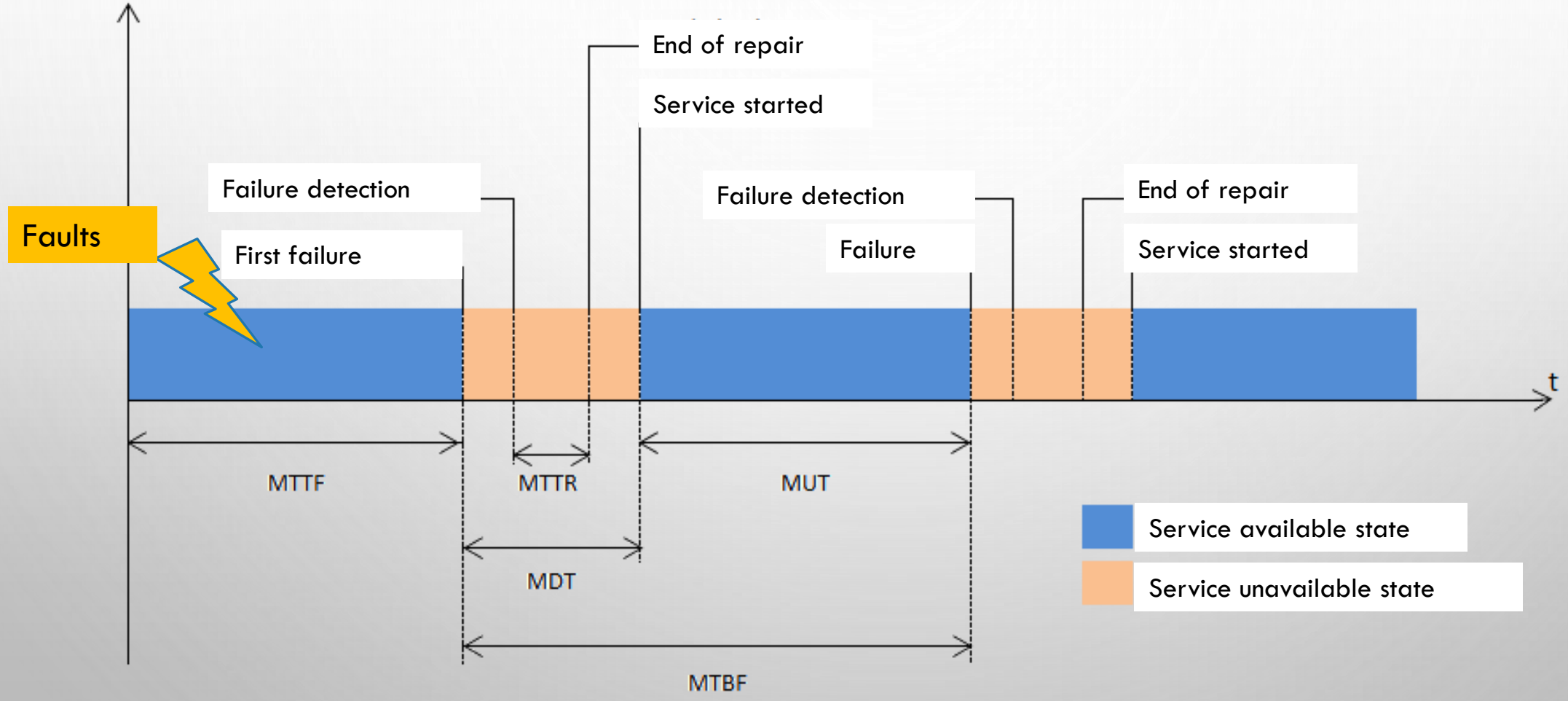
FAILURES

MTTF: Mean Time To Fail

MTTR: Mean Time To Recover

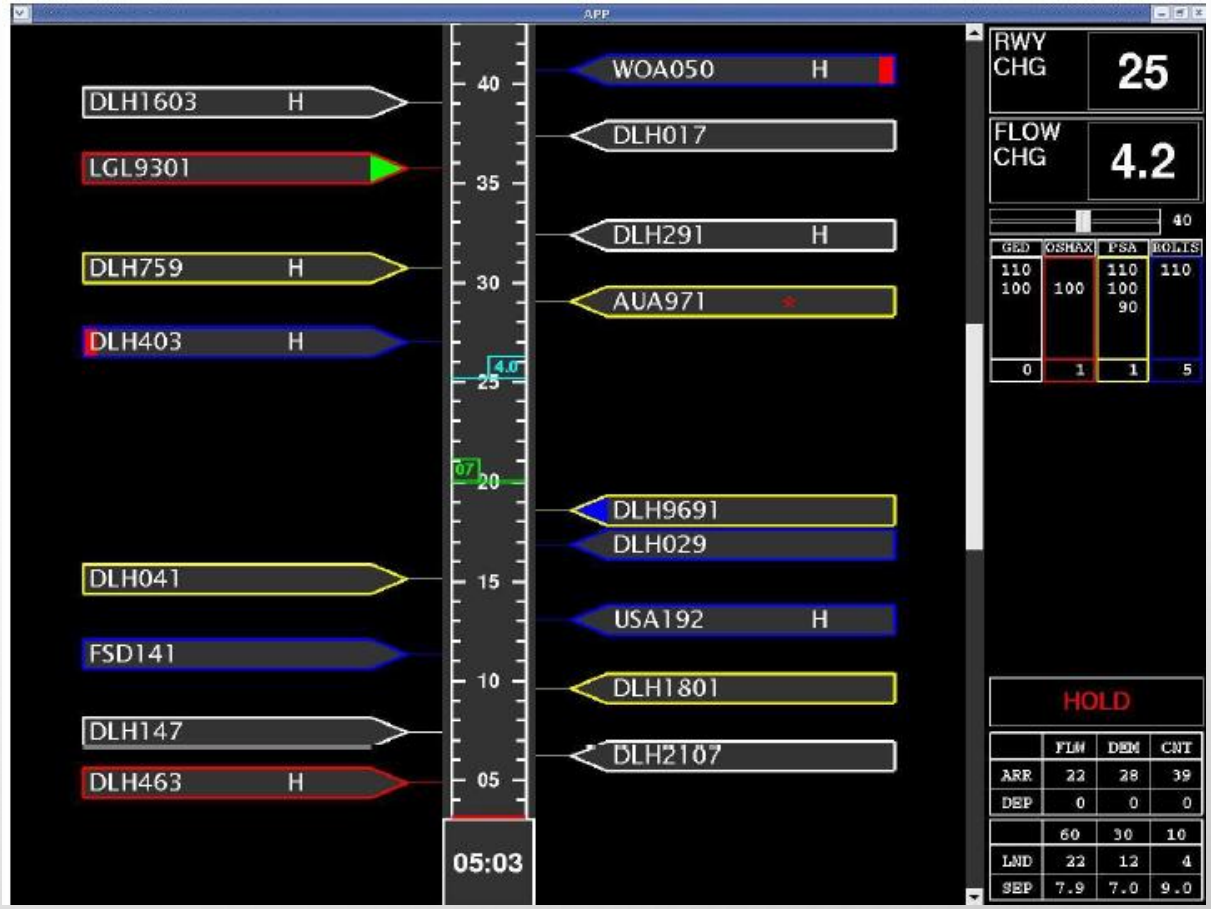
MDT: Mean Down Time

MUT: Mean Up Time



DEALING WITH FAULTS (AT OPERATION TIME)

- Fault prevention
- Fault detection
- Fault removal
- Fault tolerance
  - Redundancy
  - Diversity
  - Segregation
- Fault mitigation



Fayollas C., et al. **Interactive cockpits as critical applications: a model-based and a fault-tolerant approach**. Int. Journal on Critical Component Based Systems 4(3): 202-226 (2013)

Tankeu-Choitat A. et al. **Self-Checking Components for Dependable Interactive Cockpits Using Formal Description Techniques**. IEEE Pacific Rim Dependable Computing conference 2011: 164-173

Fayollas C. et al. **An Approach for Assessing the Impact of Dependability on Usability: Application to Interactive Cockpits.** European Dependable Computing Conferences 2014: 198-209

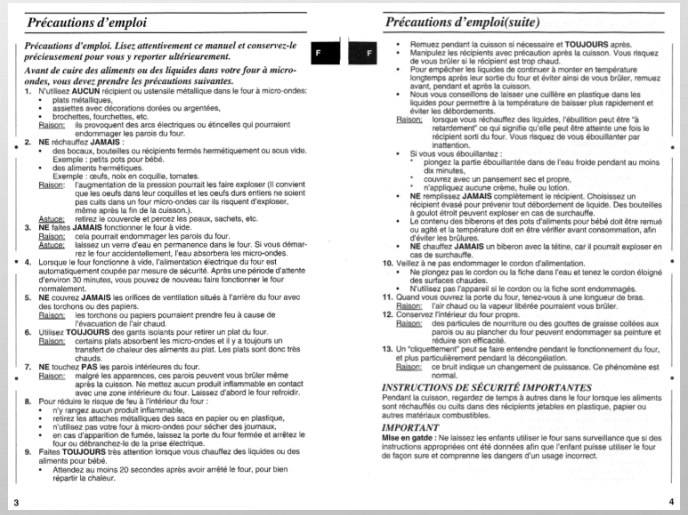
EXEMPLE DE SOLUTION

- Firstly, be explicit about the permitted limits of use within which the product is warranted or certified to be safe
- Secondly, be explicit about the critical risks if the product is used outside these limits - and state clearly that the warranty and any safety certification is invalidated by such use
- Thirdly, where a particular and dangerous misuse is foreseeable, design the product so that it prevents or detects such misuse and fails safely

CAS DU MICRO ONDE + CHAT

CAS DU MICRO ONDE + CHAT

- Documentation utilisateur pour avertir du danger pour les chats: description + raison



Précautions d'emploi

Précautions d'emploi. Lisez attentivement ce manuel et conservez-le précieusement pour vous y reporter ultérieurement.

Avant de cuire des aliments ou des liquides dans votre four à micro-ondes, vous devez prendre les précautions suivantes.

1. N'utilisez **AUCUN** récipient ou ustensile métallique dans le four à micro-ondes:
   - plats métalliques,
   - assiettes avec décorations dorées ou argentées,
   - brochettes, fourchettes, etc.

   Raison: ils provoquent des arcs électriques ou étincelles qui pourraient endommager les parois du four.
2. **NE** réchauffez **JAMAIS** :
   - des bocaux, bouteilles ou récipients fermés hermétiquement ou sous vide. Exemple : petits pots pour bébé.
   - des aliments hermétiques. Exemple : œufs, noix en coquille, tomates.

   Raison: l'augmentation de la pression pourrait les faire exploser (il convient que les œufs dans leur coquilles et les œufs durs entiers ne soient pas cuits dans un four micro-ondes car ils risquent d'exploser, même après la fin de la cuisson.).

   Astuce: retirez le couvercle et percez les peaux, sachets, etc.
3. **NE** faites **JAMAIS** fonctionner le four à vide.

   Raison: cela pourrait endommager les parois du four.

   Astuce: laissez un verre d'eau en permanence dans le four. Si vous démarrez le four accidentellement, l'eau absorbera les micro-ondes.
4. Lorsque le four fonctionne à vide, l'alimentation électrique du four est automatiquement coupée par mesure de sécurité. Après une période d'attente d'environ 30 minutes, vous pouvez de nouveau faire fonctionner le four normalement.
5. **NE** couvrez **JAMAIS** les orifices de ventilation situés à l'arrière du four avec des torchons ou des papiers.

   Raison: les torchons ou papiers pourraient prendre feu à cause de l'évacuation de l'air chaud.
6. Utilisez **TOUJOURS** des gants isolants pour retirer un plat du four.

   Raison: certains plats absorbent les micro-ondes et il y a toujours un transfert de chaleur des aliments au plat. Les plats sont donc très chauds.
7. **NE** touchez **PAS** les parois intérieures du four.

   Raison: malgré les apparences, ces parois peuvent vous brûler même après la cuisson. Ne mettez aucun produit inflammable en contact avec une zone intérieure du four. Laissez d'abord le four refroidir.
8. Pour réduire le risque de feu à l'intérieur du four :
   - n'y rangez aucun produit inflammable,
   - retirez les attaches métalliques des sacs en papier ou en plastique,
   - n'utilisez pas votre four à micro-ondes pour sécher des journaux,
   - en cas d'apparition de fumée, laissez la porte du four fermée et arrêtez le four ou débranchez-le de la prise électrique.
9. Faites **TOUJOURS** très attention lorsque vous chauffez des liquides ou des aliments pour bébé.
   - Attendez au moins 20 secondes après avoir arrêté le four, pour bien répartir la chaleur.

3

Précautions d'emploi(suite)

- Remuez pendant la cuisson si nécessaire et **TOUJOURS** après.
- Manipulez les récipients avec précaution après la cuisson. Vous risquez de vous brûler si le récipient est trop chaud.
- Pour empêcher les liquides de continuer à monter en température longtemps après leur sortie du four et éviter ainsi de vous brûler, remuez avant, pendant et après la cuisson.
- Nous vous conseillons de laisser une cuillère en plastique dans les liquides pour permettre à la température de baisser plus rapidement et éviter les débordements.

  Raison: lorsque vous réchauffez des liquides, l'ébullition peut être "à retardement" ce qui signifie qu'elle peut être atteinte une fois le récipient sorti du four. Vous risquez de vous ébouillanter par inattention.
- Si vous vous ébouillantez :
  - plongez la partie ébouillantée dans de l'eau froide pendant au moins dix minutes,
  - couvrez avec un pansement sec et propre,
  - n'appliquez aucune crème, huile ou lotion.
- **NE** remplissez **JAMAIS** complètement le récipient. Choisissez un récipient évasé pour prévenir tout débordement de liquide. Des bouteilles à goulot étroit peuvent exploser en cas de surchauffe.
- Le contenu des biberons et des pots d'aliments pour bébé doit être remué ou agité et la température doit en être vérifier avant consommation, afin d'éviter les brûlures.
- **NE** chauffez **JAMAIS** un biberon avec la tétine, car il pourrait exploser en cas de surchauffe.

10. Veillez à ne pas endommager le cordon d'alimentation.
    - Ne plongez pas le cordon ou la fiche dans l'eau et tenez le cordon éloigné des surfaces chaudes.
    - N'utilisez pas l'appareil si le cordon ou la fiche sont endommagés.
11. Quand vous ouvrez la porte du four, tenez-vous à une longueur de bras.

    Raison: l'air chaud ou la vapeur libérée pourraient vous brûler.
12. Conservez l'intérieur du four propre.

    Raison: des particules de nourriture ou des gouttes de graisse collées aux parois ou au plancher du four peuvent endommager sa peinture et réduire son efficacité.
13. Un "cliquettement" peut se faire entendre pendant le fonctionnement du four, et plus particulièrement pendant la décongélation.

    Raison: ce bruit indique un changement de puissance. Ce phénomène est normal.

INSTRUCTIONS DE SÉCURITÉ IMPORTANTES

Pendant la cuisson, regardez de temps à autres dans le four lorsque les aliments sont réchauffés ou cuits dans des récipients jetables en plastique, papier ou autres matériaux combustibles.

IMPORTANT

Mise en garde : Ne laissez les enfants utiliser le four sans surveillance que si des instructions appropriées ont été données afin que l'enfant puisse utiliser le four de façon sure et comprenne les dangers d'un usage incorrect.

4

CAS DU MICRO ONDE + CHAT

Précautions d'emploi

Précautions d'emploi. Lisez attentivement ce manuel et conservez-le précieusement pour vous y reporter ultérieurement.

Avant de cuire des aliments ou des liquides dans votre four à micro-ondes, vous devez prendre les précautions suivantes.

1. N'utilisez **AUCUN** récipient ou ustensile métallique dans le four à micro-ondes:
   - plats métalliques,
   - assiettes avec décorations dorées ou argentées,
   - brochettes, fourchettes, etc.

   Raison: ils provoquent des arcs électriques ou étincelles qui pourraient endommager les parois du four.
2. **NE** réchauffez **JAMAIS** :
   - des bocaux, bouteilles ou récipients fermés hermétiquement ou sous vide.
     Exemple : petits pots pour bébé.
   - des aliments hermétiques.
     Exemple : oeufs, noix en coquille, tomates.

   Raison: l'augmentation de la pression pourrait les faire exploser (Il convient que les oeufs dans leur coquilles et les oeufs durs entiers ne soient pas cuits dans un four micro-ondes car ils risquent d'exploser, même après la fin de la cuisson.).

   Astuce: retirez le couvercle et percez les peaux, sachets, etc.

PRÉVENTION DU RISQUE

- Alerte détecteur de chat dans le micro-onde
- Blocage du système d'allumage

COOPÉRATION AVEC L'UTILISATEUR

Techniques de "Collaborative Design":
Au départ, utilisées en Scandinavie pour faire accepter l'informatisation aux syndicats des grandes entreprises.

Se sont avérées très profitables.

-> Les seuls vrais connaisseurs du système, de ses limites et de ses capacités sont :
- les programmeurs (+ encore que les concepteurs)
- les utilisateurs

Technique:
Les futurs utilisateurs et les concepteurs partagent un plan de travail. Les concepteurs exposent leur projet, les utilisateurs annotent le projet, avec de petites notes, chacune relatant un point précis qui semble poser problème.

USER-CENTERED SYSTEMS DESIGN IN PRACTICE

EXERCICE : INTERFACE DE VOITURE

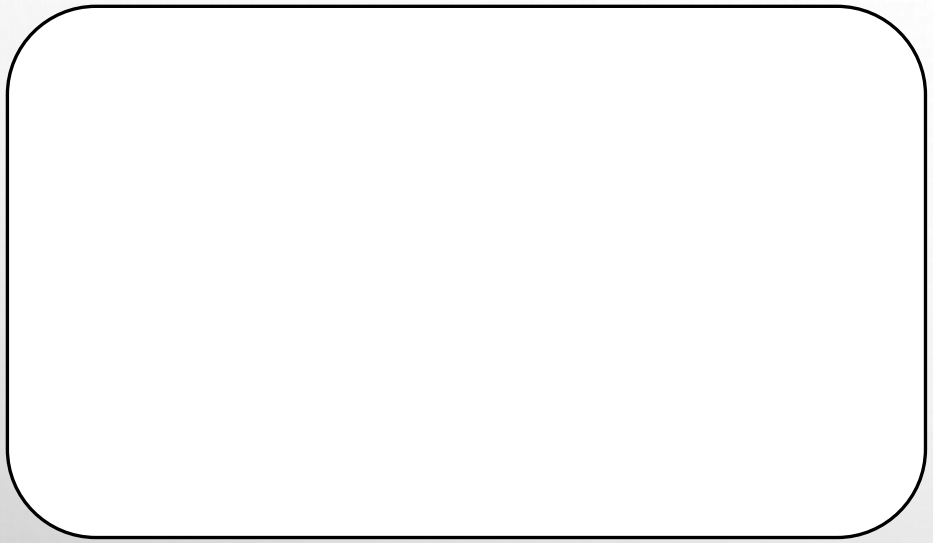
Faire l'interface d'utilisation d'une voiture en simulant son utilisation par un ordinateur. Il faudra représenter graphiquement les deux principales tâches:

- le pilotage (gestion des interactions temps réel)
  - afficher les compteurs
  - actions offertes au pilote
  - donner du retour d'information
- la navigation : on peut avoir besoin d'informations du genre (gestion de la planification à long et moyen terme)
  - proposer une route en fonction de la durée de voyage
  - proposer une route en fonction de son taux de chargement
  - proposer une route en fonction du départ et de la destination
  - communication entre véhicules
- on suppose que la manipulation du volant, pédales et le levier de vitesse restent telles que offertes dans les véhicules actuels

Faites cet exercice par binôme. Chacun des éléments du binôme est expert soit dans la tâche de pilotage soit dans la tâche de navigation.

EXERCISE 1 – PAIR WORKING (2 USERS)

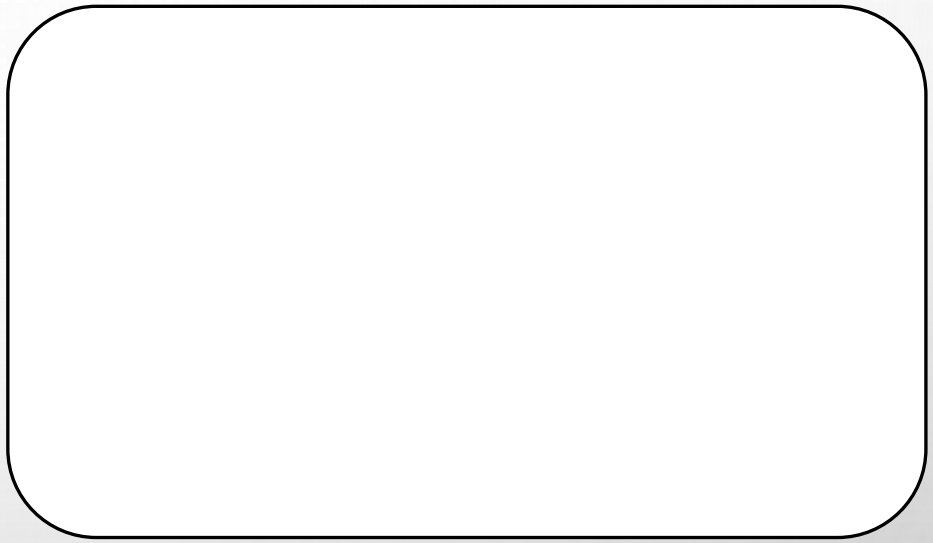
Driver



User interface for driving

Navigator

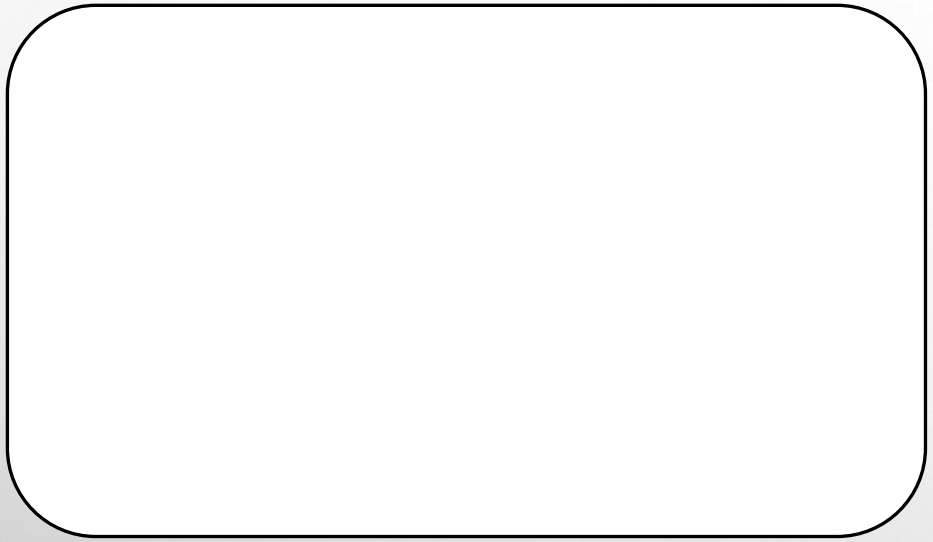
(front seat passenger)



User interface for navigation

EXERCISE 1 – PAIR WORKING (2 USERS)

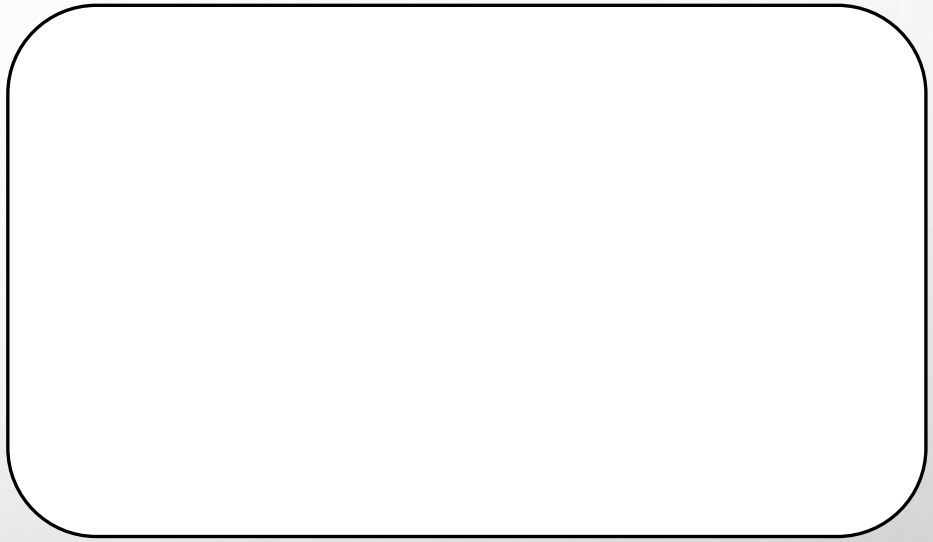
User 1: Driver



User interface for driving

Job: Designer of the navigation UI

User 2: Navigator (front seat passenger)



User interface for navigation

Job: Designer of the driving UI

EXERCISE 1 – MAIN STEPS

1	Informal discussion with user	20 min
2	Prototyping version 1	20 min
3	Presentation of the v1 of the prototype to the user	20 min
4	Prototyping version 2	10 min
5	Presentation of the v2 of the prototype to the user	15 min

EXERCISE 1 STEP 1: INFORMAL DISCUSSION WITH USER

- The designer of the Driving UI discusses with the Driver – 10 min
  - What does the driver needs/wants to do? How?
  - Which data/information does the user need? Why?
  - Which preferences has the user to perform her/his tasks?
- The designer of the Navigation UI discusses with the Navigator – 10 min
  - What does the driver needs/wants to do? How?
  - Which data/information does the user need? Why?
  - Which preferences has the user to perform her/his tasks?

SAFETY ANALYSIS

- Intended use
- Failure modes
- Hazard analysis
- Fault tolerance work
- Impact on system design

EXERCICE : AÉROPORT

- Vous êtes dans un avion en train d'atterrir
- Vous avez une correspondance très courte
- Vous avez faim
- Déterminez un système de distribution de nourriture permettant d'obtenir la nourriture le + rapidement possible

MISE EN ŒUVRE

• Création d'un modèle conceptuel:

- définir les objets utilisés
- les opérations à effectuer
- les concepts et les enchaînements logiques

• Représentation du Contrôle

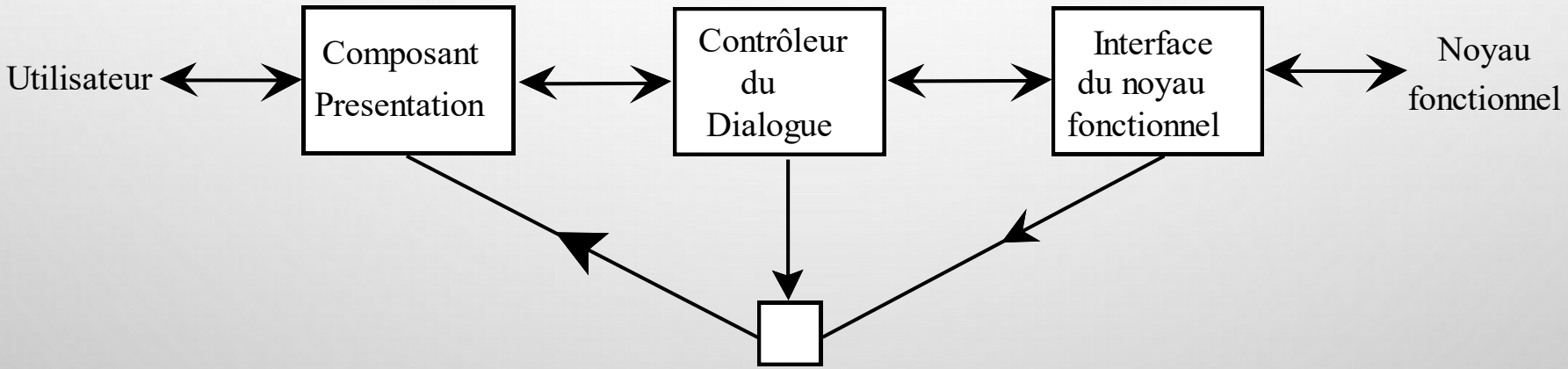
• Choix et Spécification du Modèle d'Interaction

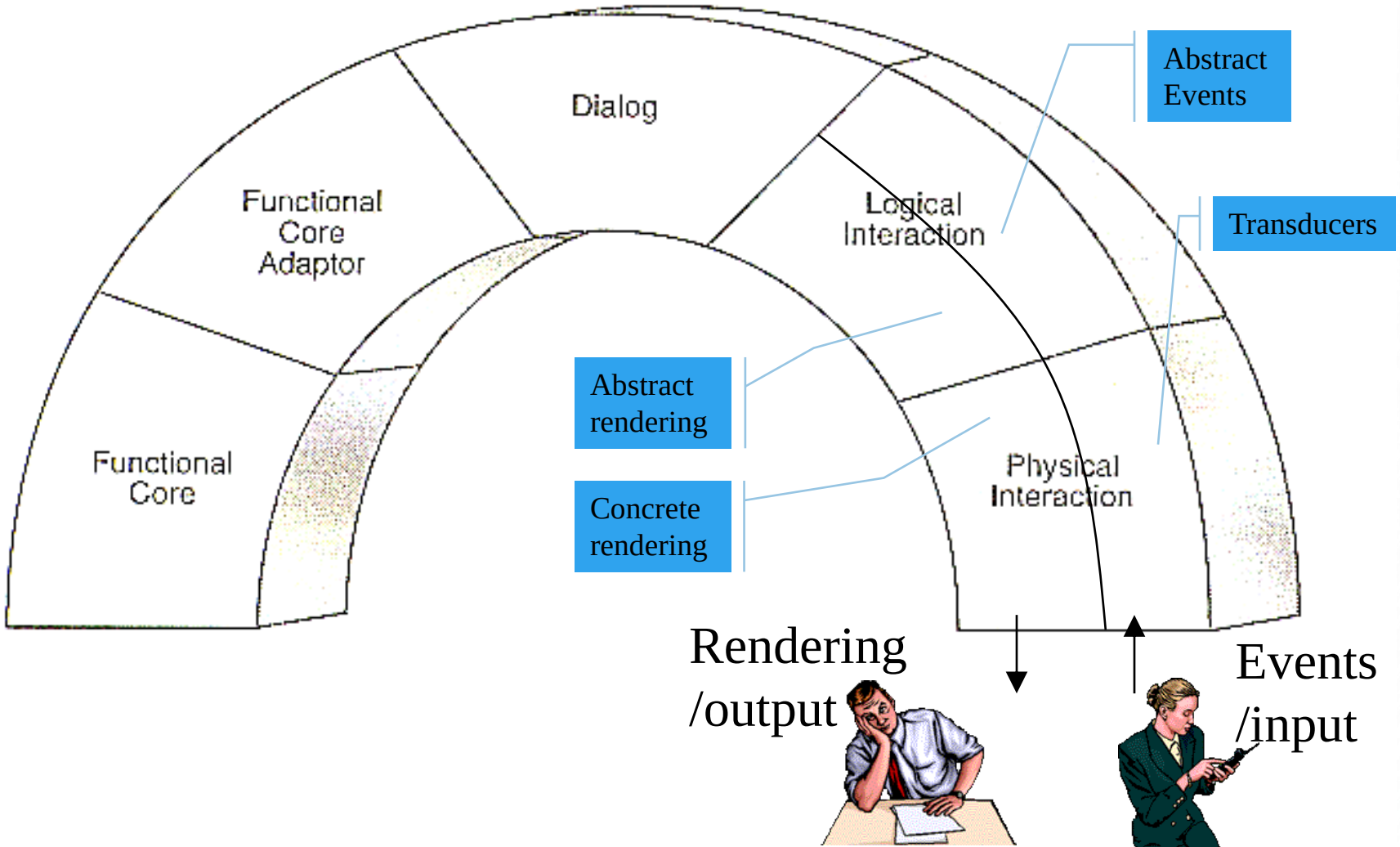
-> utiliser les dessins et schémas sur papier pour
- la représentation des données
- la formulation des actions

-> penser aux règles de cohérence et concision.

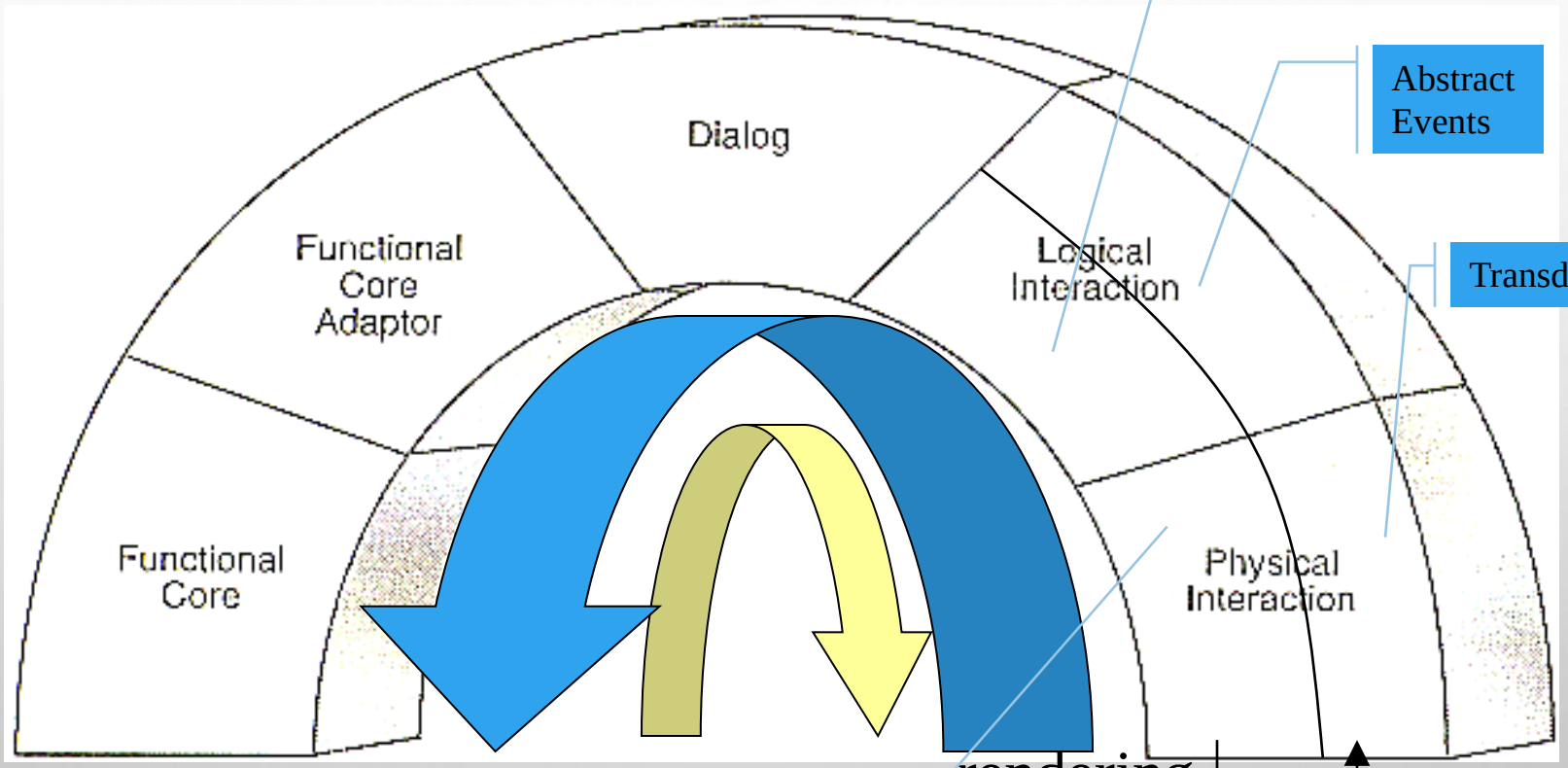
DU DESIGN À LA CONSTRUCTION

MODÈLE DE SEEHEIM 83



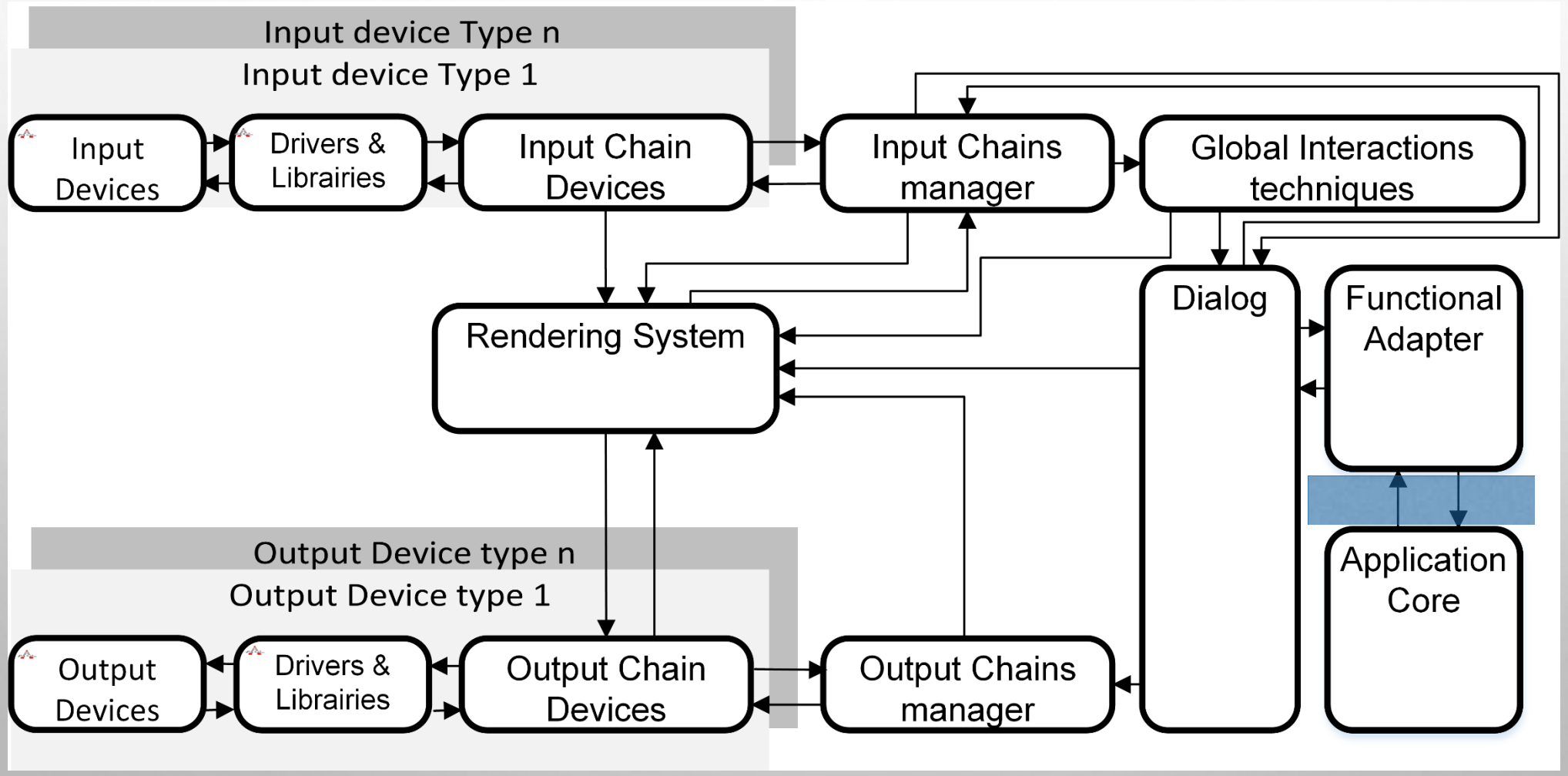

Dialog
Functional Core Adaptor
Functional Core
Logical Interaction
Physical Interaction
Abstract Events
Transducers
Abstract rendering
Concrete rendering
Rendering /output
Events /input

MODÈLE ARCHITECTURAL



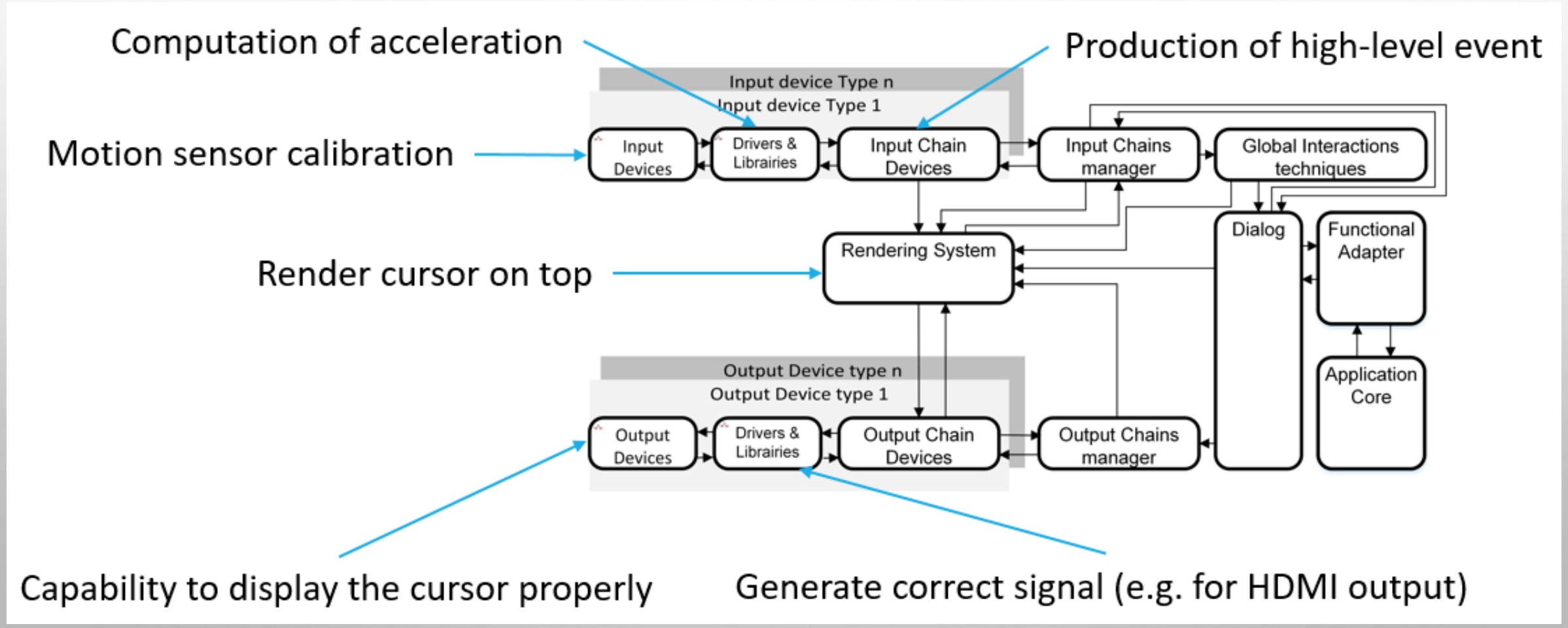
MIODMIT: A GENERIC ARCHITECTURE FOR DYNAMIC INTERACTIVE SYSTEMS

Cronel M. et al. **MIODMIT: A Generic Architecture for Dynamic Multimodal Interactive Systems.** Human-Centered Software Engineering. HCSE 2018. LNCS vol 11262. Springer,



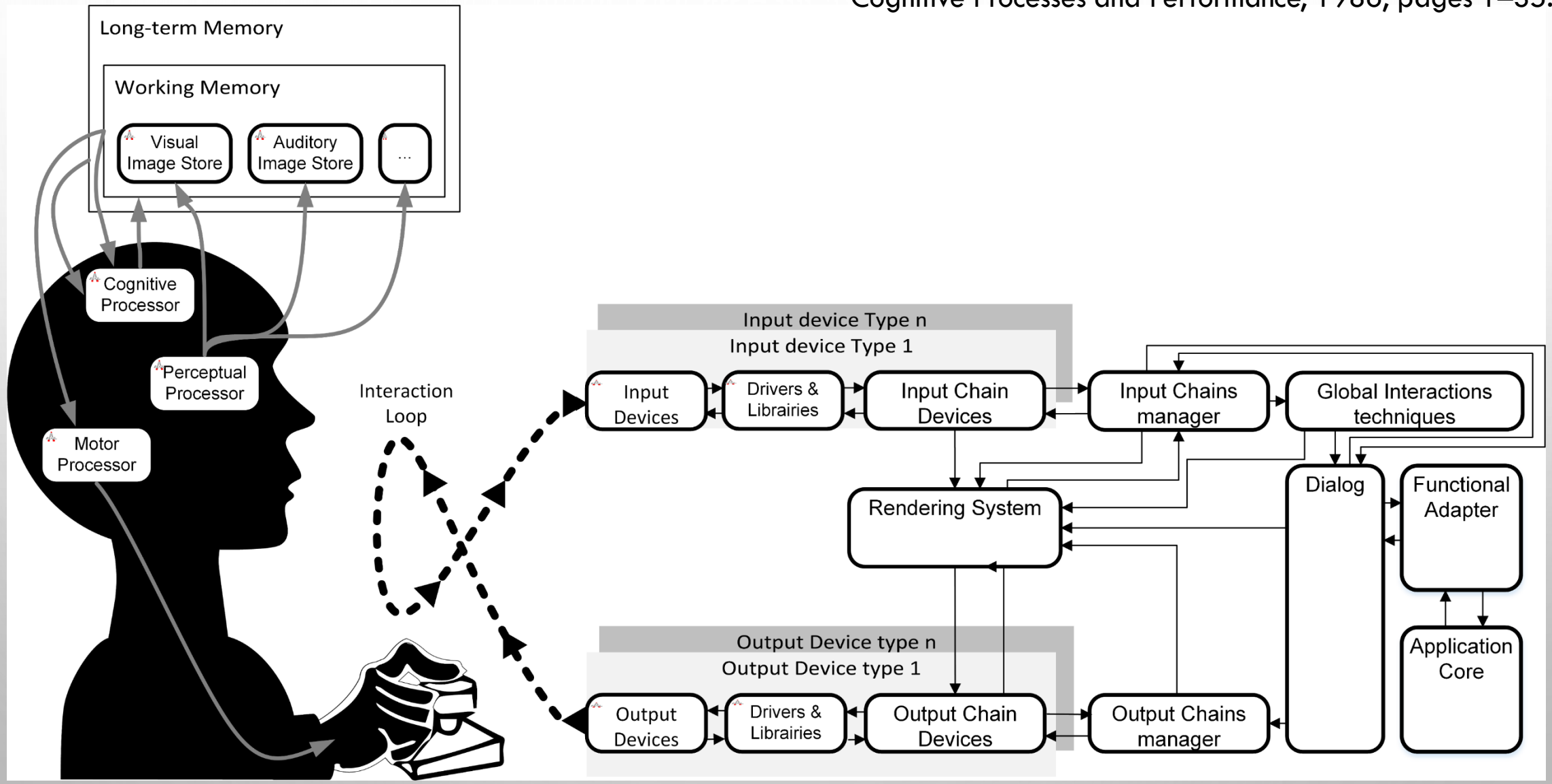
MIODMIT AND IMMEDIATE FEEDBACK

REQ: « The position of the manipulator of an input device (e.g. mouse cursor) should evolve in accordance with user action on that device »



THE INTERACTING HUMAN

Card, S.K; Moran, T. P; and Newell, A. *The Model Human Processor: An Engineering Model of Human Performance.* **Handbook of Perception and Human Performance.** Vol. 2: Cognitive Processes and Performance, 1986, pages 1–35.



CONCEPTION DES IHM

Il faut concevoir les trois parties du modèle de Seeheim:

La présentation : ce que l'utilisateur voit de l'application

Le dialogue :

- qu'est-ce que l'utilisateur a la possibilité de faire
- comment l'utilisateur agit sur la présentation
- l'influence de son action sur ce qu'il pourra faire ensuite

Le noyau fonctionnel :

- les fonctions réalisées par l'application
- les données manipulées par l'application

FONCTIONNEMENT DES SYSTÈMES PAR ÉVÉNEMENTS

STRUCTURE D'UNE APPLI. CLASSIQUE

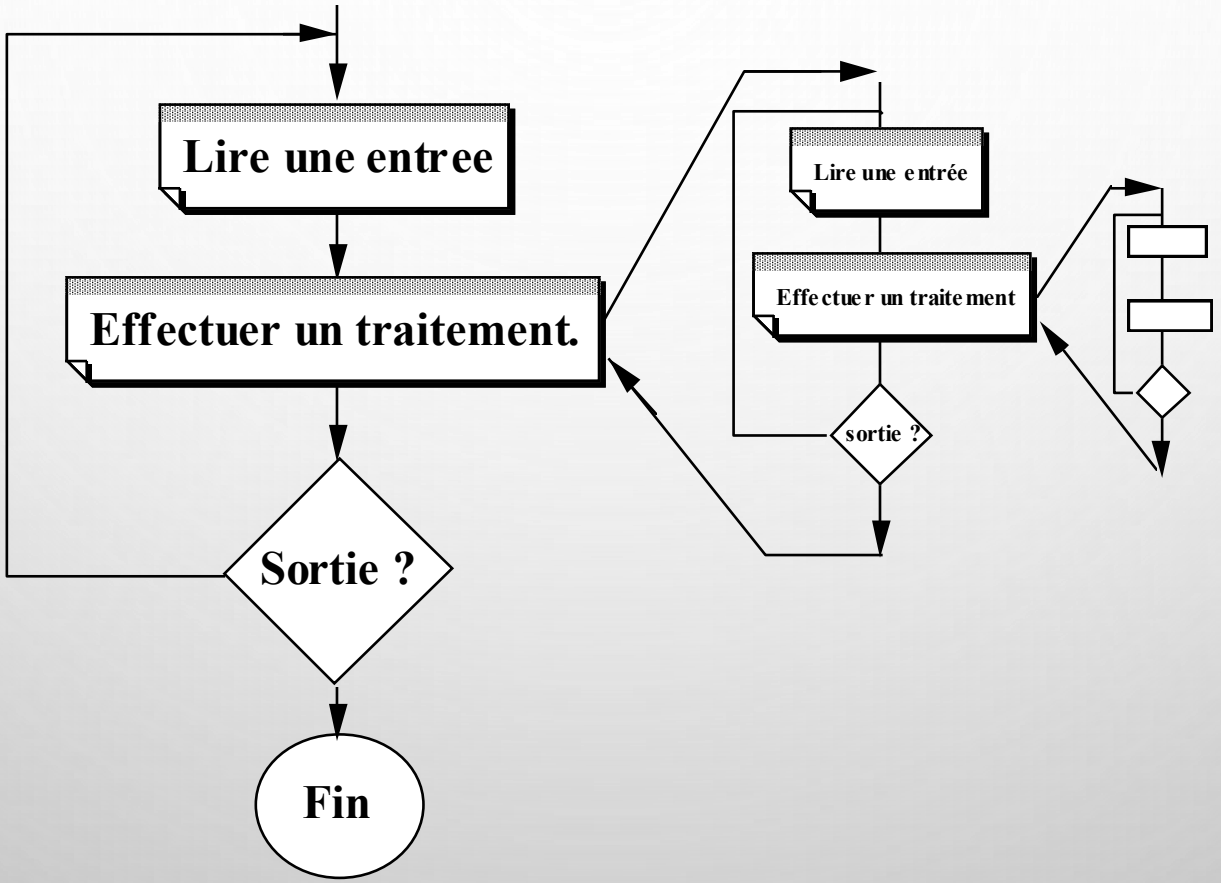
A tout instant l'application est en attente d'une entrée de l'utilisateur.

Dans les autres cas elle fait des calculs et l'utilisateur doit attendre

```
Début
choix = '1';
Tantque choix <> '9' faire
    affiche-menu;
    lire(choix);
    case choix of
    1 : ajouter;
    2 : modifier;
    3 : supprimer;
    9 : Quitter;
    Fin Case
FintTantque
Fin
```

```
Procédure Ajouter;
début
rep = 'o';
Tantque rep <> 'n';
    dessin-écran;
    lire(nom);
    lire(prenom);
    ...
    écrire('voulez-vous
    continuer ?');
    lire(rep)
FinTantque
Fin
```

FONCTIONNEMENT CLASSIQUE



STRUCTURE APPLI. PAR EVT.

- La boucle d'événement (main event loop) : reçoit chaque événement produit par l'utilisateur
- Les gestionnaires d'événements : sont des procédures associées à chaque couple (widget, action sur un widget) et appelées par la main event loop dès que une action a été réalisée.

Tous les event handlers ont la même structure :

EH1;

Précondition;

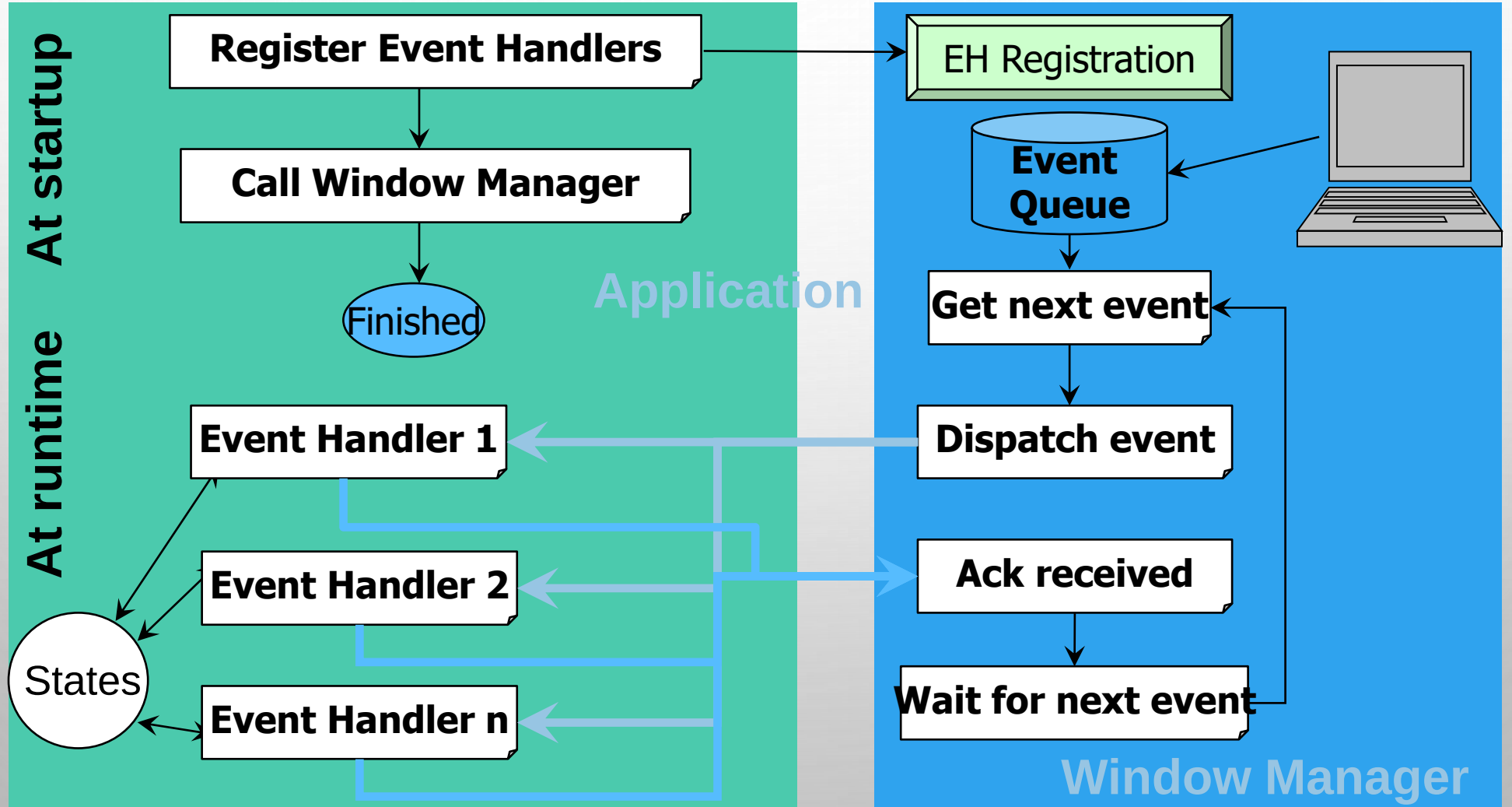
Action;

Modification de l'état du dialogue;

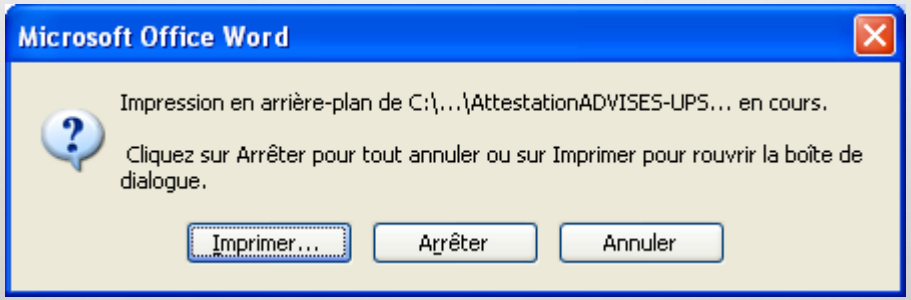
Rétroaction graphique;

Fin EH1;

EVENT-BASED FUNCTIONING



COMPORTEMENT BASÉ SUR LES ÉTATS



DEUXIÈME COUCHE

Principe Télétubbies

Principe Gillette

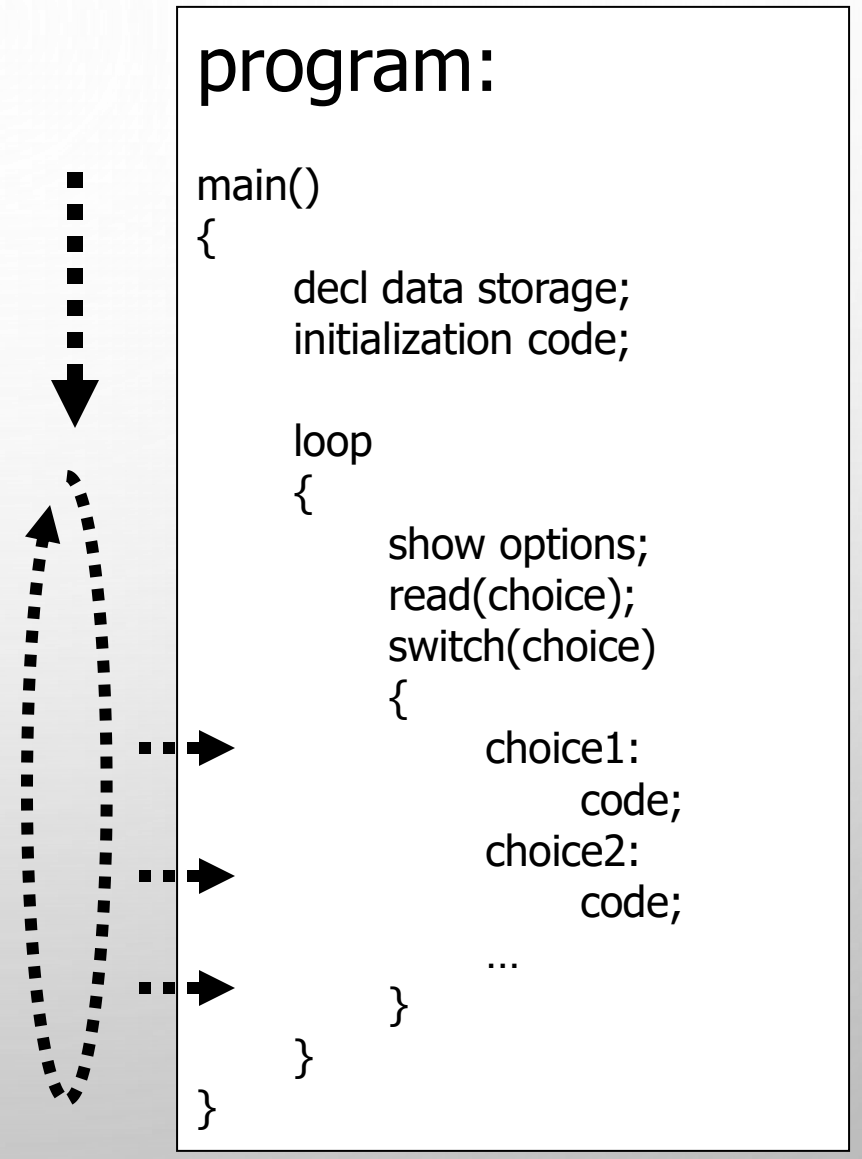
EXÉCUTION TYPIQUE D'UN PROGRAMME

- Non-interactive
- Exécution linéaire
- Processus d'automatisation (automatique)
- Ne prends pas en compte les capacités de l'humain versus ordinateur

```
program:

main()
{
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
        code;
}
```

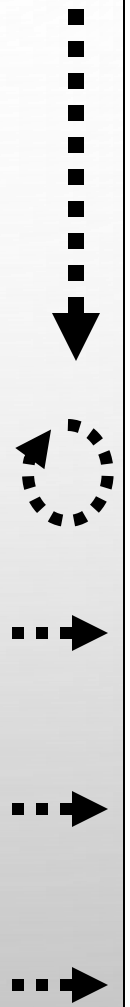
PROGRAMME INTERACTIF À CHOIX MULTIPLES

- L'utilisateur choisit les options
- Exécution non-linéaire ("branching")
- Ordre imprédictible
- Système arrêté sur instruction de lecture
- Possibilité de continuer



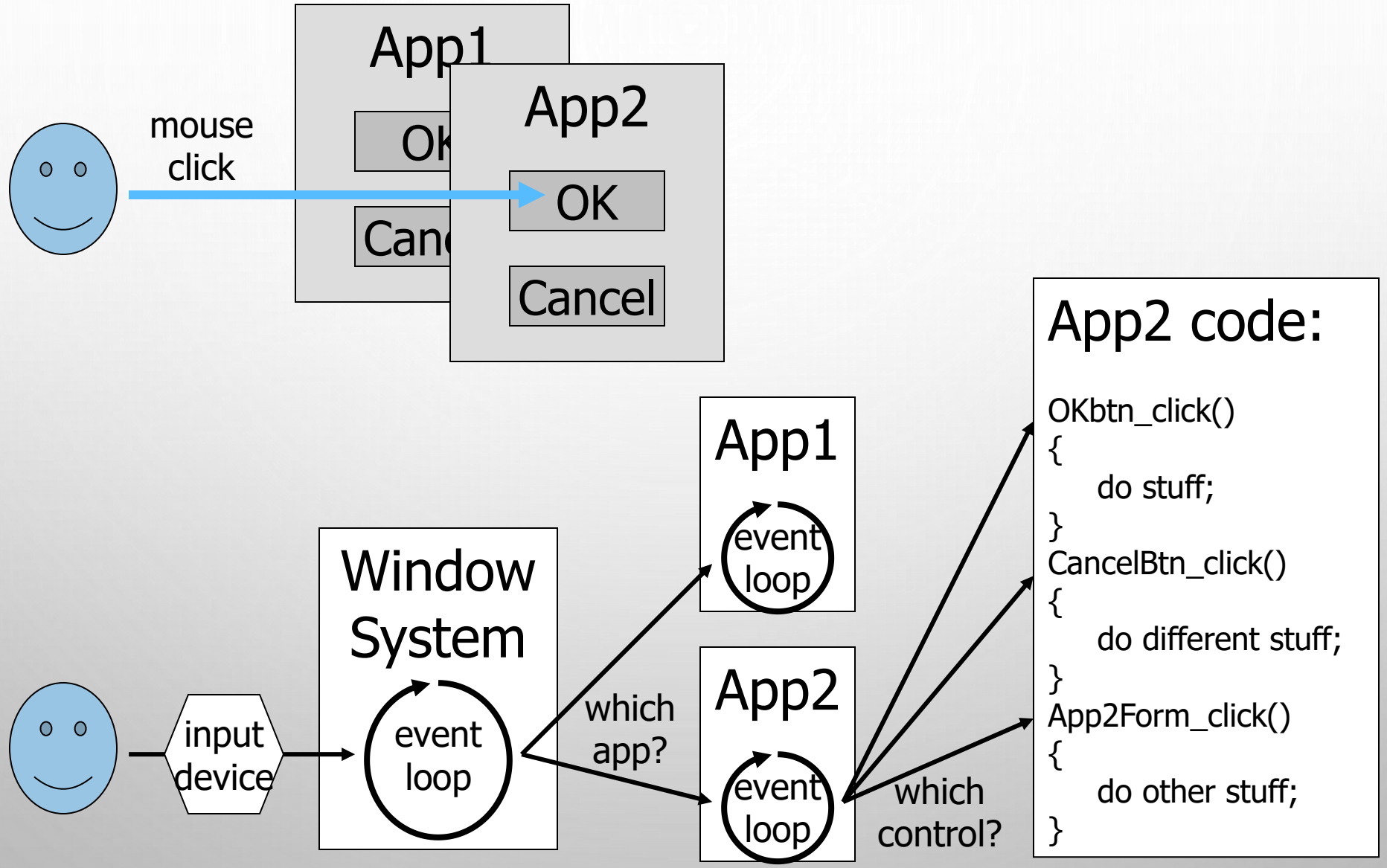
INTERFACE DIRIGÉE PAR L'UTILISATEUR

- L'utilisateur déclenche des commandes
- Exécution non linéaire
- Ordre non prédictible
- La plupart du temps le système ne fait rien
- Les procédure de gestion d'événements



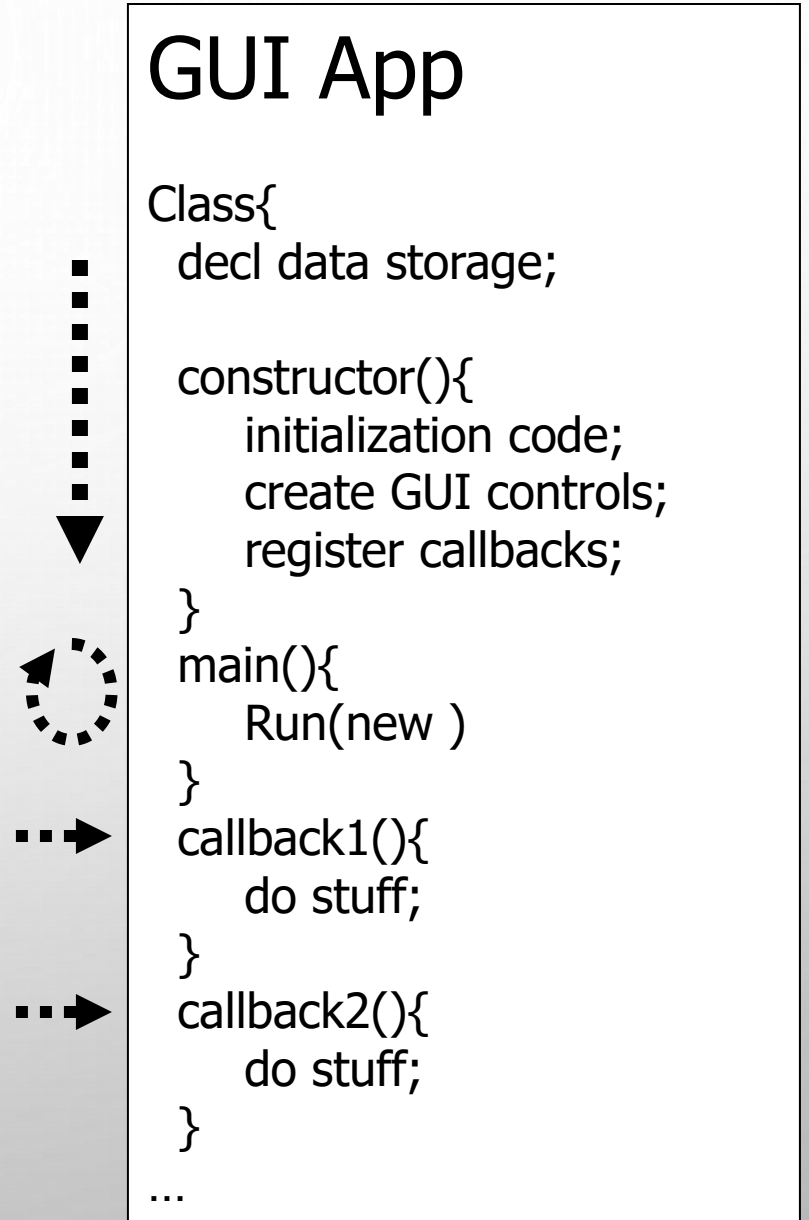
```
GUI program:

main()
{
      decl data storage;
      initialization code;

      create GUI;
      register callbacks;

      main event loop;
}

Callback1()    //button1 press
{     code;
}
Callback2()    //button2 press
{     code;
}
…
```

GUI EVENTS

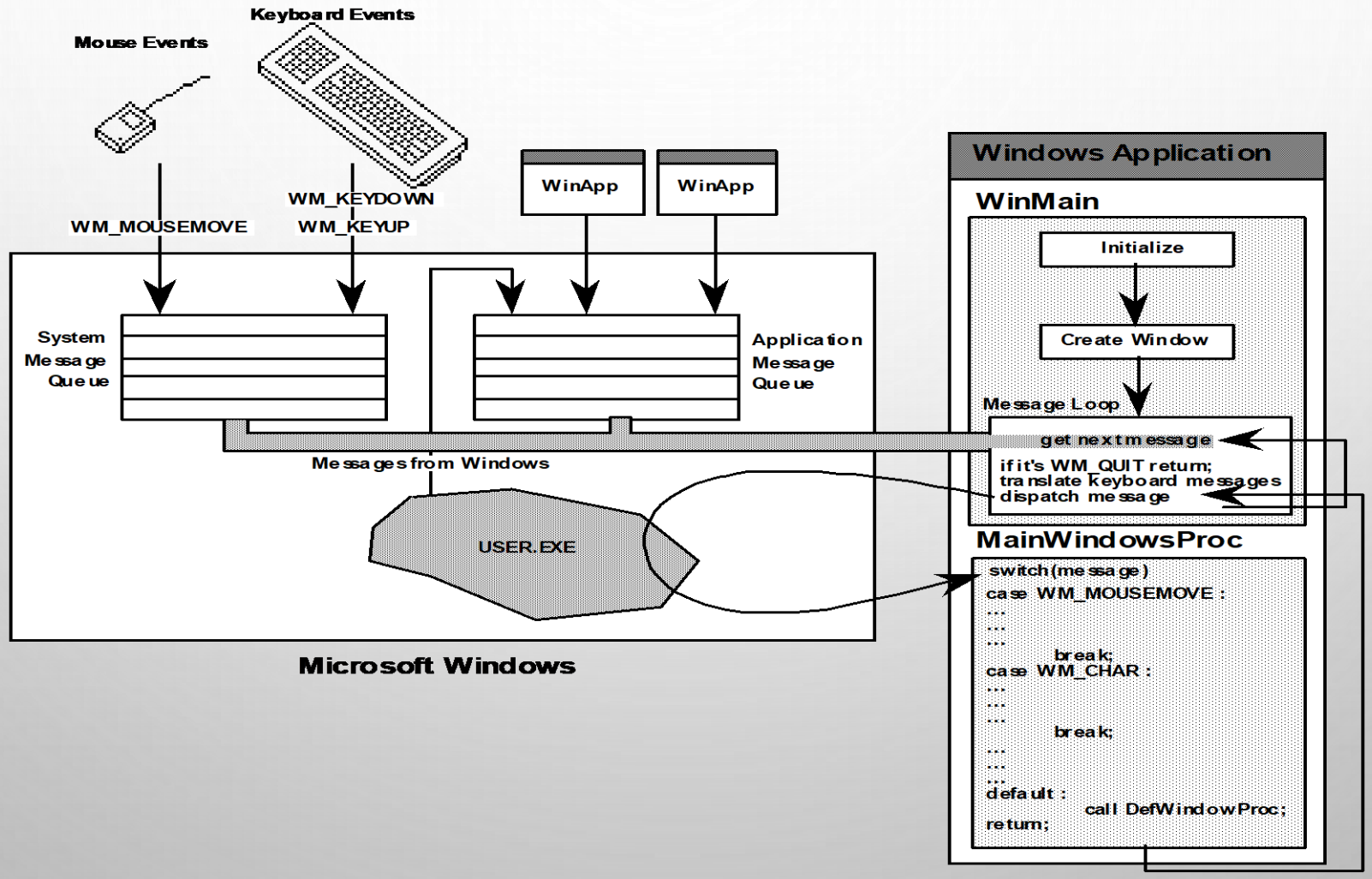

App1
App2
OK
Cancel
mouse
click
Window
System
input
device
event
loop
which
app?
App1
event
loop
App2
event
loop
which
control?
App2 code:
OKbtn_click()
{
do stuff;
}
CancelBtn_click()
{
do different stuff;
}
App2Form_click()
{
do other stuff;
}

FONCTIONNEMENT

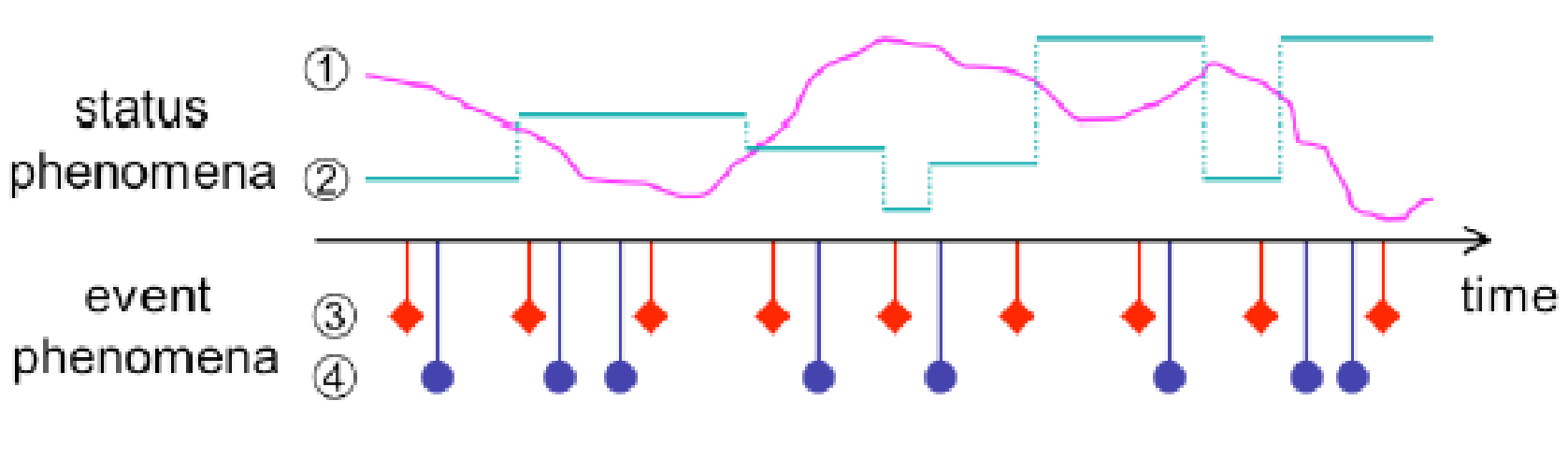
- “delegates” = callbacks
- Java: Listeners



EVÉNÉMENTS DANS WINDOWS



ETATS ET ÉVÉNEMENTS



- 1- le monde réel évolue de façon continue
- 2- les variables représentent des variations par pallier
- 3- les événements peuvent avoir une origine périodique (regarder sa montre toutes les 30s)
- 4- les événements arrivent et ont un impact sur l'état

UNE DÉMARCHE DE CONCEPTION

- Une démarche de conception
- Une notation les automates
- Un processus proche de E/A (conception de bases de données)
- Un cheminement vers le code de l'application
- Pas de fossés à combler intellectuellement

UNE DÉMARCHE DE CONCEPTION

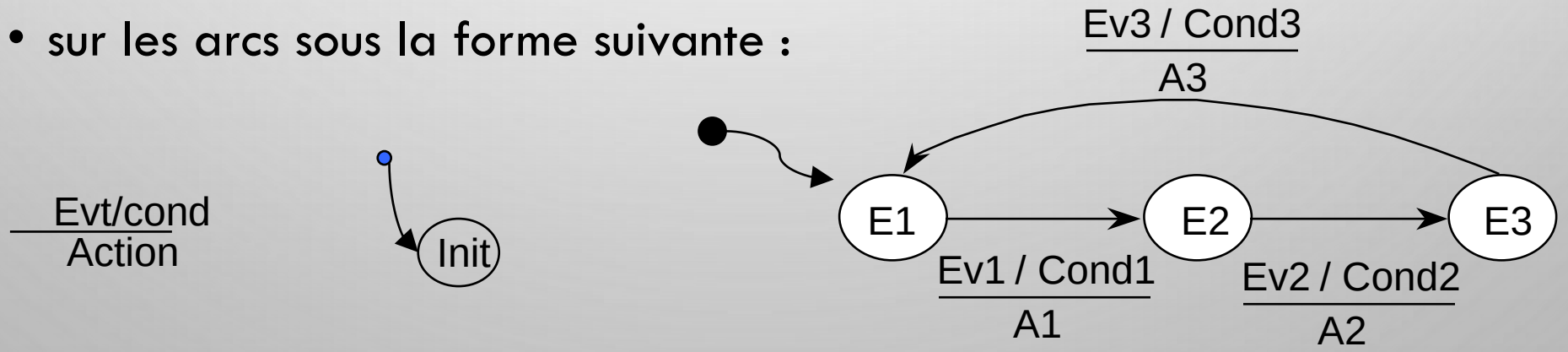
- 1) Analyse
  - a) conception de l'interface (design, choix des objets, …)
  - b) liste des événements
  - c) liste des actions
- 3) automate de comportement
- 4) Matrice états/événements
- 5) Event-handlers

AVANTAGES

- Description complète et non ambiguë
- Analyse de propriétés
  - Comportementales
  - D'utilisabilité
- Génération de code
- Il est plus facile de prouver que de tester

LES AUTOMATES ETENDUS

- Un automate étendu est un automate à états pouvant posséder :
  - des événements déclenchant des actions
  - des conditions de déclenchement des actions
  - des registres (variables d'états supplémentaires)
  - effectuer des actions sur les registres (affectation)
- Les événements, les conditions et les actions sont représentés
- sur les arcs sous la forme suivante :



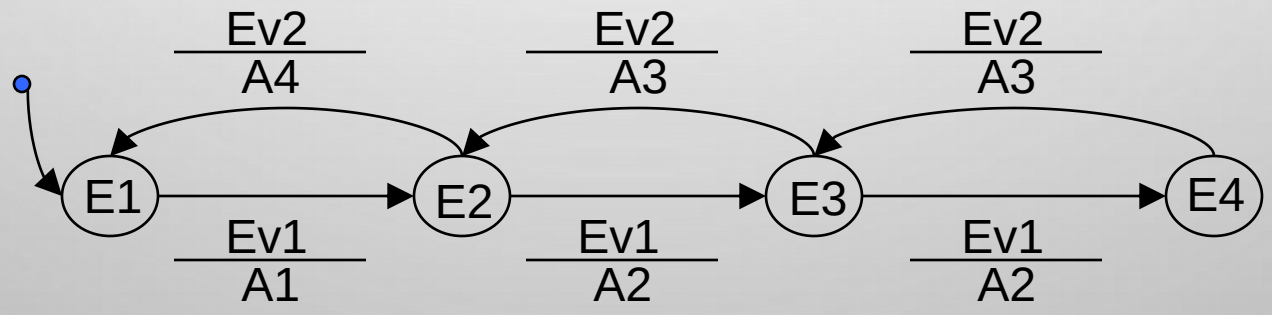
LES AUTOMATES ETENDUS (2)

Exemple : un système simple

- **E = {E1, E2, E3, E4}, s0 = E1**
- **Ev = {Ev1, Ev2},**
- **A = {A1, A2, A3, A4},**

- **f : E x Ev -> A,**
  f(E1,Ev1) = A1,
  f(E2,Ev1) = f(E3,Ev1) = A2,
  f(E4,Ev2) = f(E3,Ev2) = A3,
  f(E2,Ev2) = A4

- **g : E x Ev -> E,**
  g(E1,Ev1) = g(E3,Ev2) = E2,
  g(E2,Ev1) = g(E4,Ev2) = E3,
  g(E3,Ev1) = E4,
  g(E2,Ev2) = E1.



EXEMPLE PAR ÉVÉNEMENT

V = {v}, v0 = 1 and v : integer

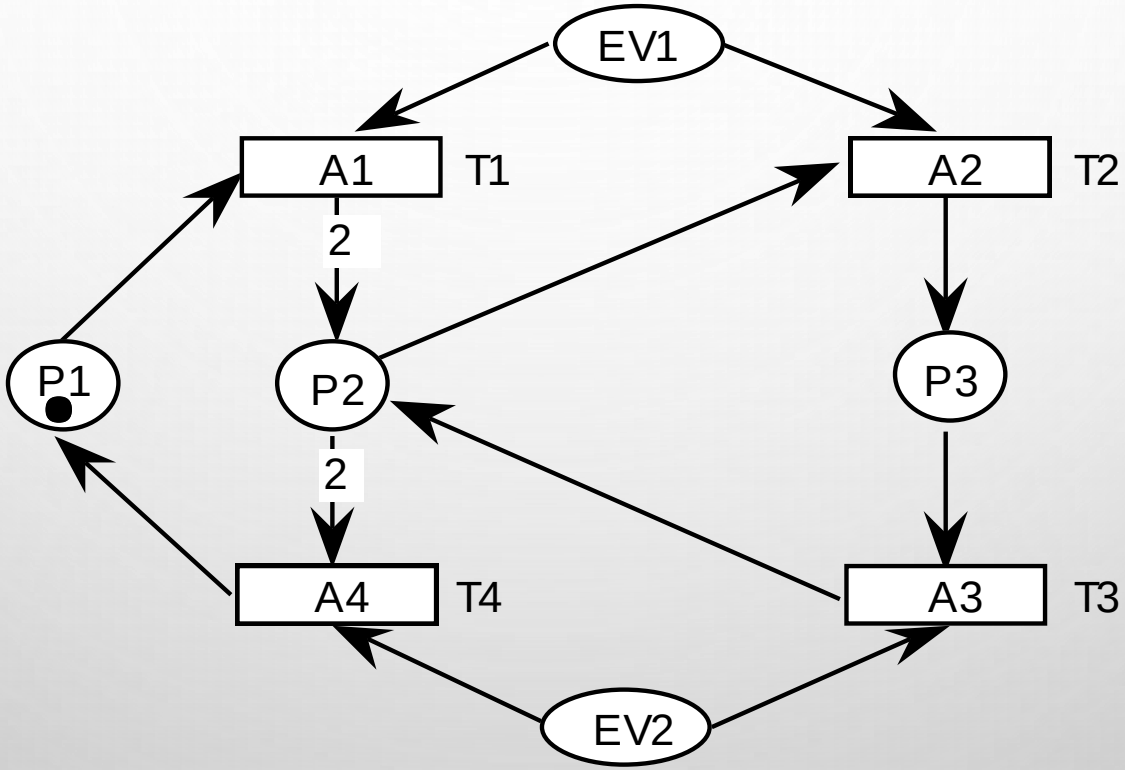
```
Handler Ev1 is begin
 Case v of
  1 : A1; v:=2; ev1 actif ev2 actif
  2 : A2; v:=3; ev1 actif ev2 actif
  3 : A2; v:=4; ev1 inactif ev2 actif
  4 : 'Interdit
 Endcase
EndHandler;
```

```
Handler Ev2 is begin
 Case v of
  1 : 'Interdit
  2 : A4; v:=1; ev1 actif ev2 inactif
  3 : A3; v:=2; ev1 actif ev2 actif
  4 : A3; v:=3; ev1 actif ev2 actif
 Endcase
EndHandler;
```

TABLEAU RECCAPITULATIF

Protocole de communication (appli. - utilisateur)	Application = client Utilisateur = serveur	Application = serveur Utilisateur = client
Nature de l'appli.	Transformationnelle	Inter(ré)active
Contrôle	Impératif	Déclaratif
Etat du dialogue (de l'interaction)	Historique	Valeur des variables d'état

DES RÉSEAUX DE PETRI ???



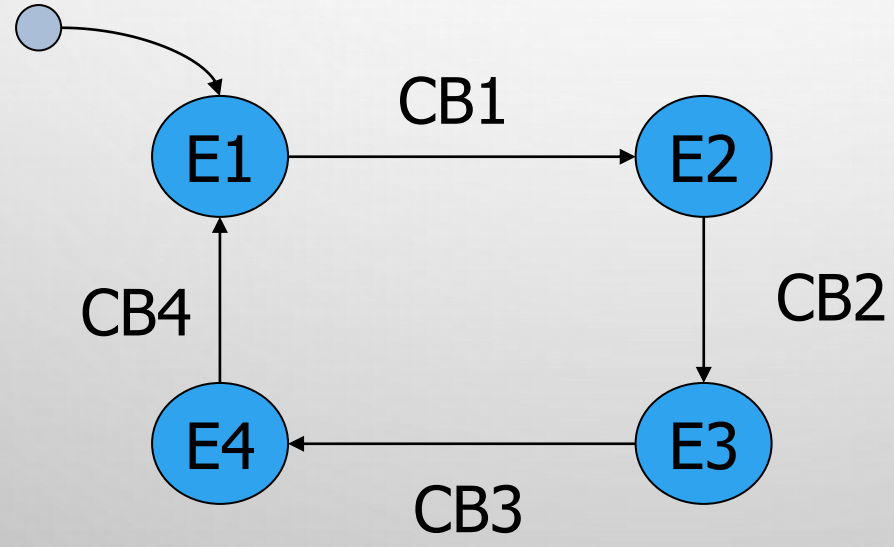
EXEMPLE: LES 4 BOUTONS

- Spécification du comportement d'une application avec 4 boutons cycliques

Application

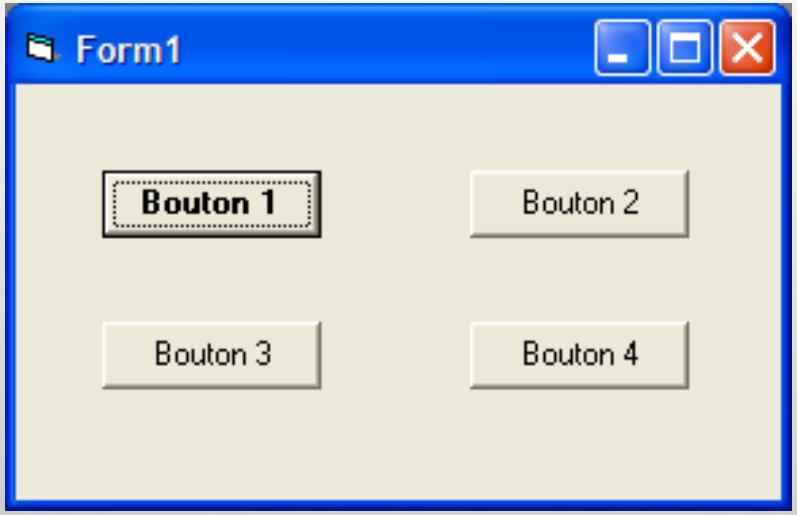
AUTOMATE EXERCICE 1

- 4 événements CB1, CB2, CB3, CB4



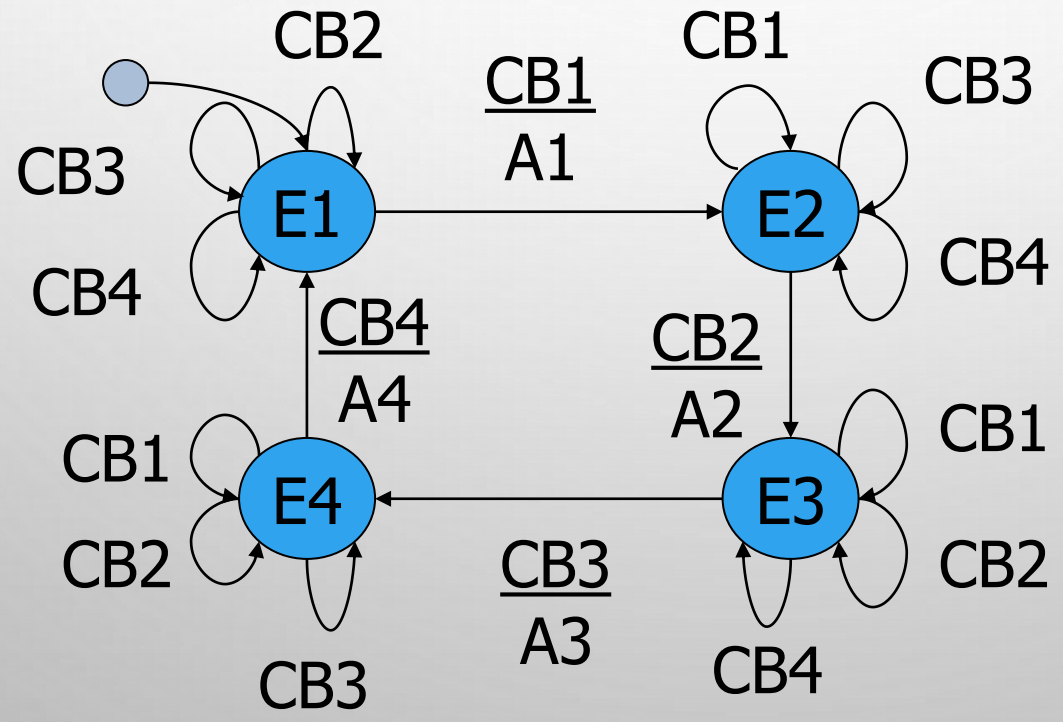
EXEMPLE: LES 4 BOUTONS CYCLIQUES

- Spécification du comportement d'une application avec 4 boutons cycliques toujours actifs



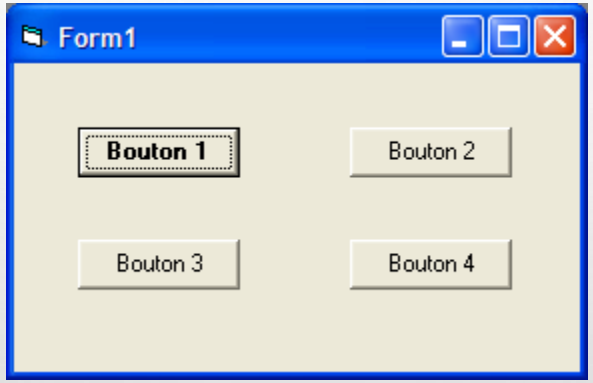
AUTOMATE EXERCICE 1 (2/3)

- 4 événements CB1, CB2, CB3, CB4
- 4 actions (chgt apparence boutons)



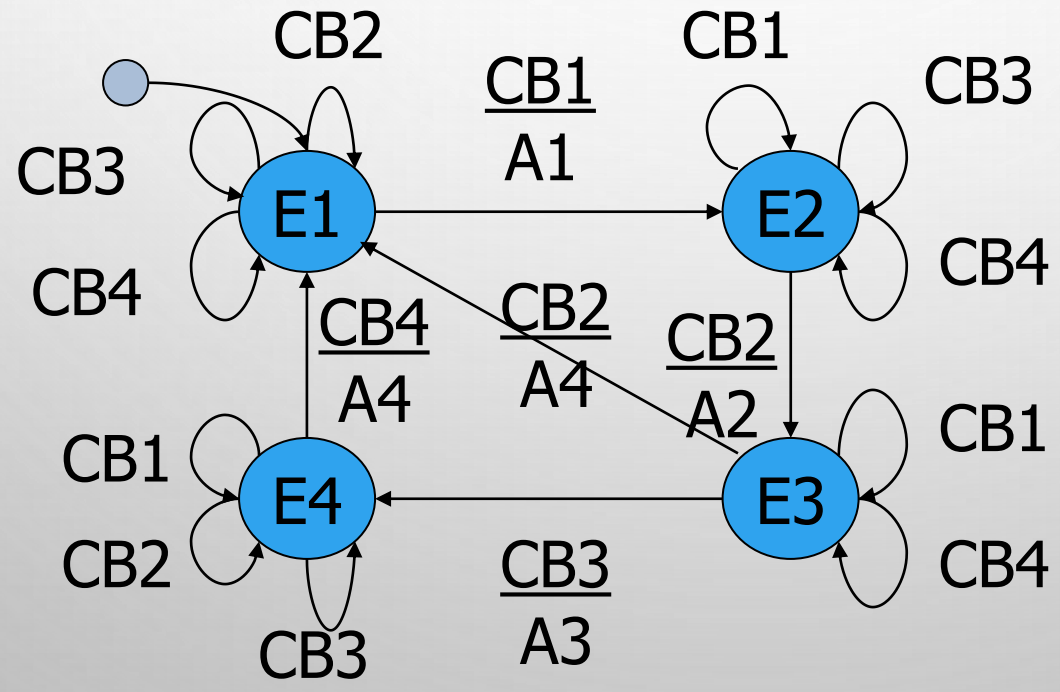
EXEMPLE: LES 4 BOUTONS CYCLIQUES (3/3)

- Spécification du comportement d'une application avec 4 boutons cycliques toujours actifs et en ajoutant un raccourci (dans l'état où B3 est en gras, on peut cliquer sur B2 ce qui conduit dans l'état initial)



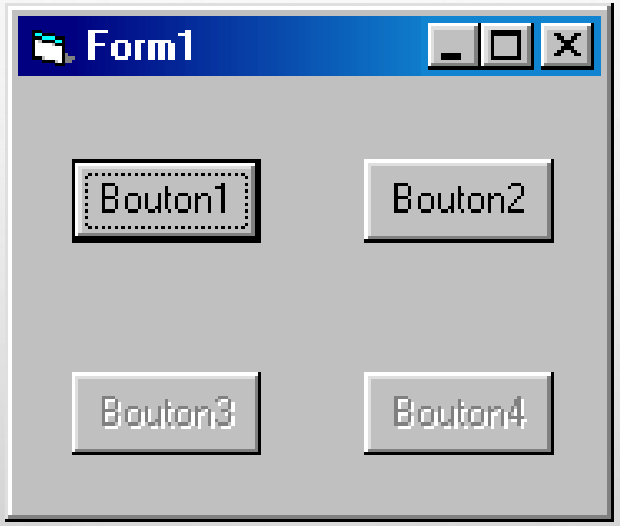
AUTOMATE EXERCICE 1 (3/3)

- 4 événements CB1, CB2, CB3, CB4
- 4 actions (chgt apparence boutons)



EXEMPLE: LES 4 BOUTONS

- Spécification du comportement d'une application avec 4 boutons alternatifs



Application

Ou pas

VÉRIFICATION DE PROPRIÉTÉS

- P1: Au **moins** 2 boutons sont toujours actifs
- P2: Au **plus** 2 boutons sont toujours actifs
- P3: Chaque bouton peut redevenir actif à partir de n'importe quel état
  - P3.1: Quelque soit l'état il est toujours possible de trouver un chemin qui rende Bouton1 actif
  - P3.2: Quelque soit l'état il est toujours possible de trouver un chemin qui rende Bouton2 actif
  - P3.3: Quelque soit l'état il est toujours possible de trouver un chemin qui rende Bouton3 actif
  - P3.4: Quelque soit l'état il est toujours possible de trouver un chemin qui rende Bouton4 actif

VÉRIFICATION DE PROPRIÉTÉS

- P4: exclusion mutuelle des boutons 2 à 2
  - P4.1: Jamais le bouton 1 et le bouton 3 ne sont actifs en même temps
  - P4.2: Jamais le bouton 1 et le bouton 4 ne sont actifs en même temps
  - P4.3: Jamais le bouton 2 et le bouton 3 ne sont actifs en même temps
  - P4.4: Jamais le bouton 2 et le bouton 4 ne sont actifs en même temps
- P5: fonctionnement par paire
  - P5.1: Si le bouton 1 est actif, alors le bouton 2 est actif
  - P5.2: Si le bouton 2 est actif, alors le bouton 1 est actif
  - P5.3: Si le bouton 3 est actif, alors le bouton 4 est actif
  - P5.1: Si le bouton 4 est actif, alors le bouton 3 est actif
- P6: initialisation
  - P.6.1: à l'initialisation les boutons 1 et 2 sont actifs
  - P.6.2: à l'initialisation les boutons 3 et 4 sont inactifs

MODÉLISATION EN LOGIQUE TEMPORELLE

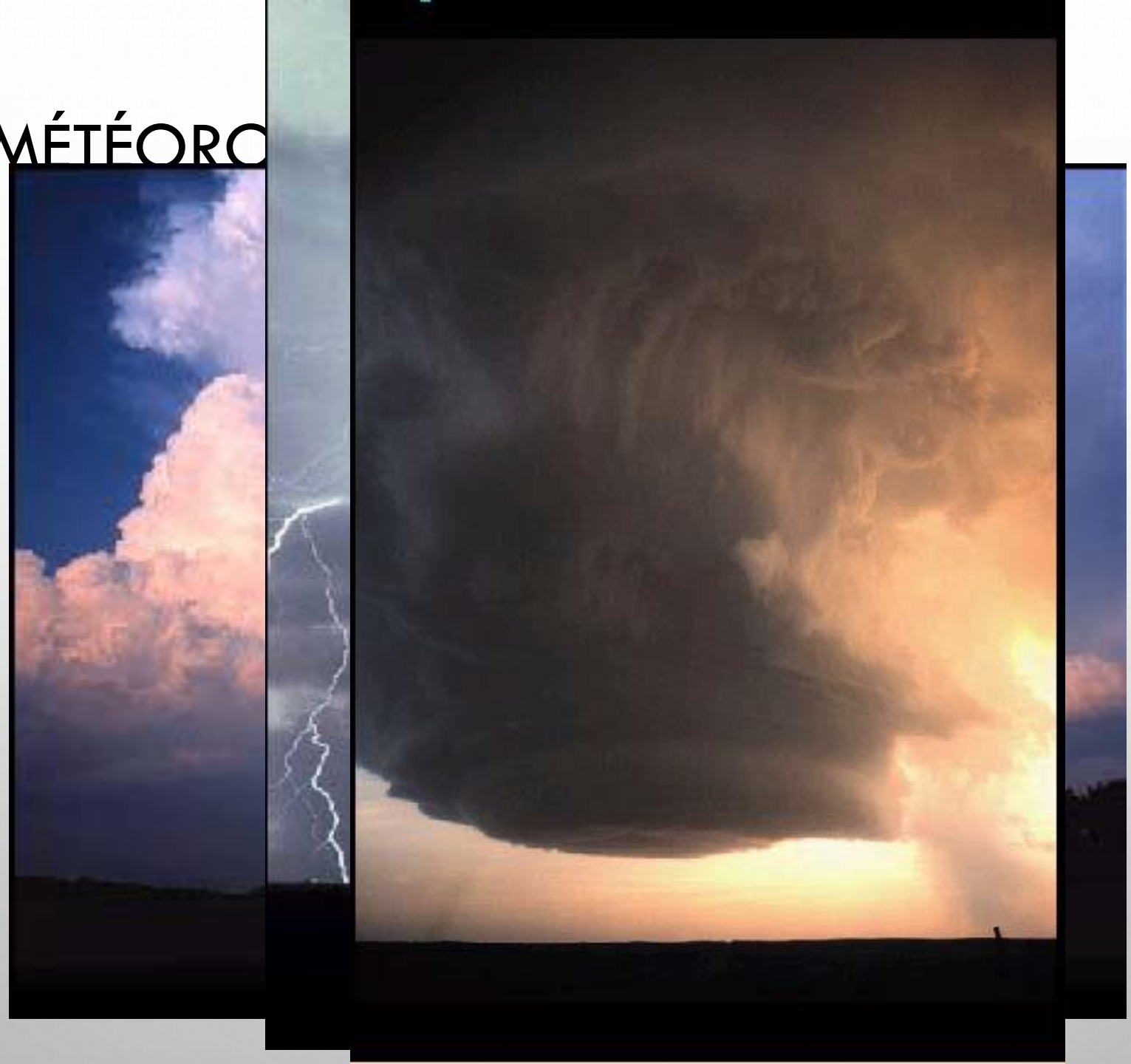
- si |= AG button1.enabled => button2.enabled
- si |= AG button2.enabled => button1.enabled
- si |= (AG button1.enabled)=False
- si |= AF button1.enabled
- si |= AF (not button3.enabled)
- si |= AG button1.enabled U button3.enabled
- si |= [AF (button1.enabled $\wedge$ button3.enabled)] = false
- si |= [EG (button1.enabled $\wedge$ button2.enabled) $\vee$ (button3.enabled $\wedge$ button4.enabled)])
- si |= [AG (button1.enabled $\wedge$ button2.enabled) $\vee$ (button3.enabled $\wedge$ button4.enabled)])

$\wedge$ (et); $\vee$ (ou); $\neg$ (non); $\Rightarrow$ (implication)

UNE APPLICATION RÉELLE

LE CONTEXTE

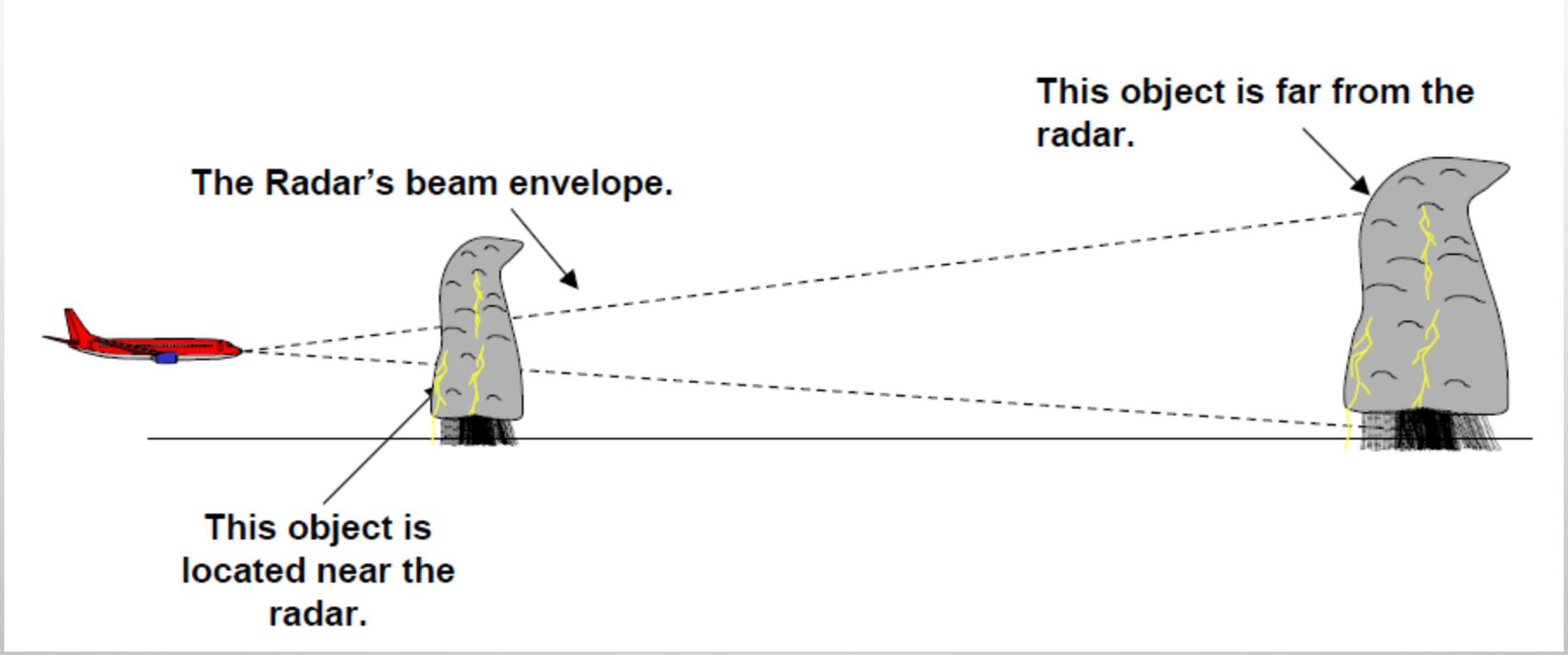
RISQUE MÉTÉORO



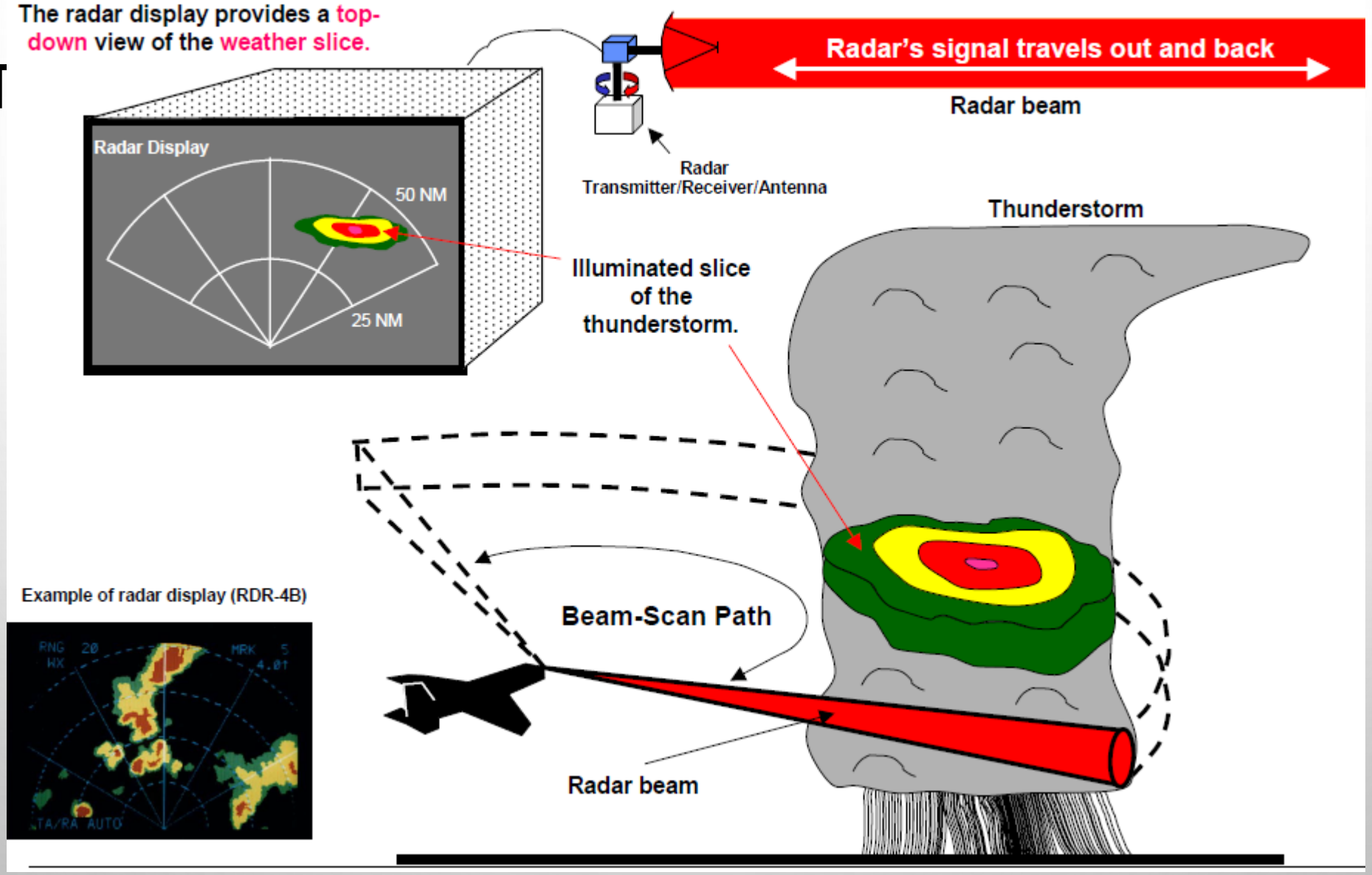
GESTION DU RISQUE MÉTÉO DANS UN COCKPIT D'AVION

- Bulletin météo
- Communication avec les contrôleurs de traffic aérien
- Radar météo (affichage + contrôle)

RADAR MÉTÉO

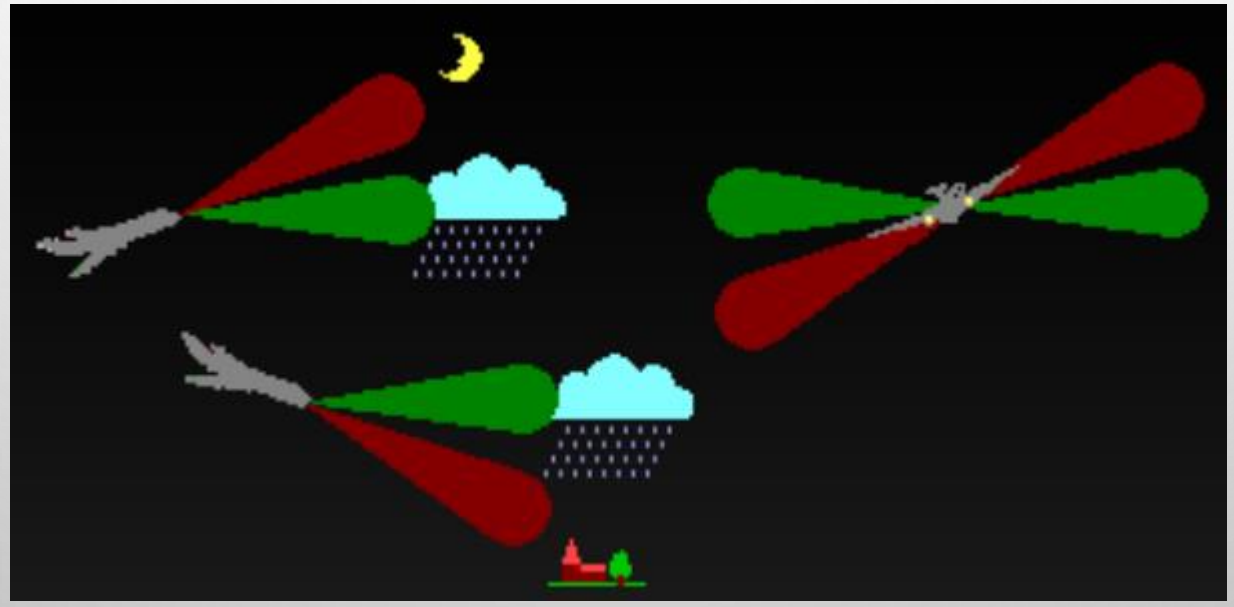

This object is far from the radar.
The Radar's beam envelope.
This object is located near the radar.

L'INT


The radar display provides a top-down view of the weather slice.
Radar's signal travels out and back
Radar beam
Radar Display
50 NM
25 NM
Radar Transmitter/Receiver/Antenna
Thunderstorm
Illuminated slice of the thunderstorm.
Example of radar display (RDR-4B)
Beam-Scan Path
Radar beam

STABILISATION

- En cas de turbulences




BARO
1013
is Hg
hPa
GSTR
WPT
VOR.D
NDB
ARPT
ROSE
NAV
VOR
ARC
ILS
PLAN
20
40
80
10
160

PH1


PH1

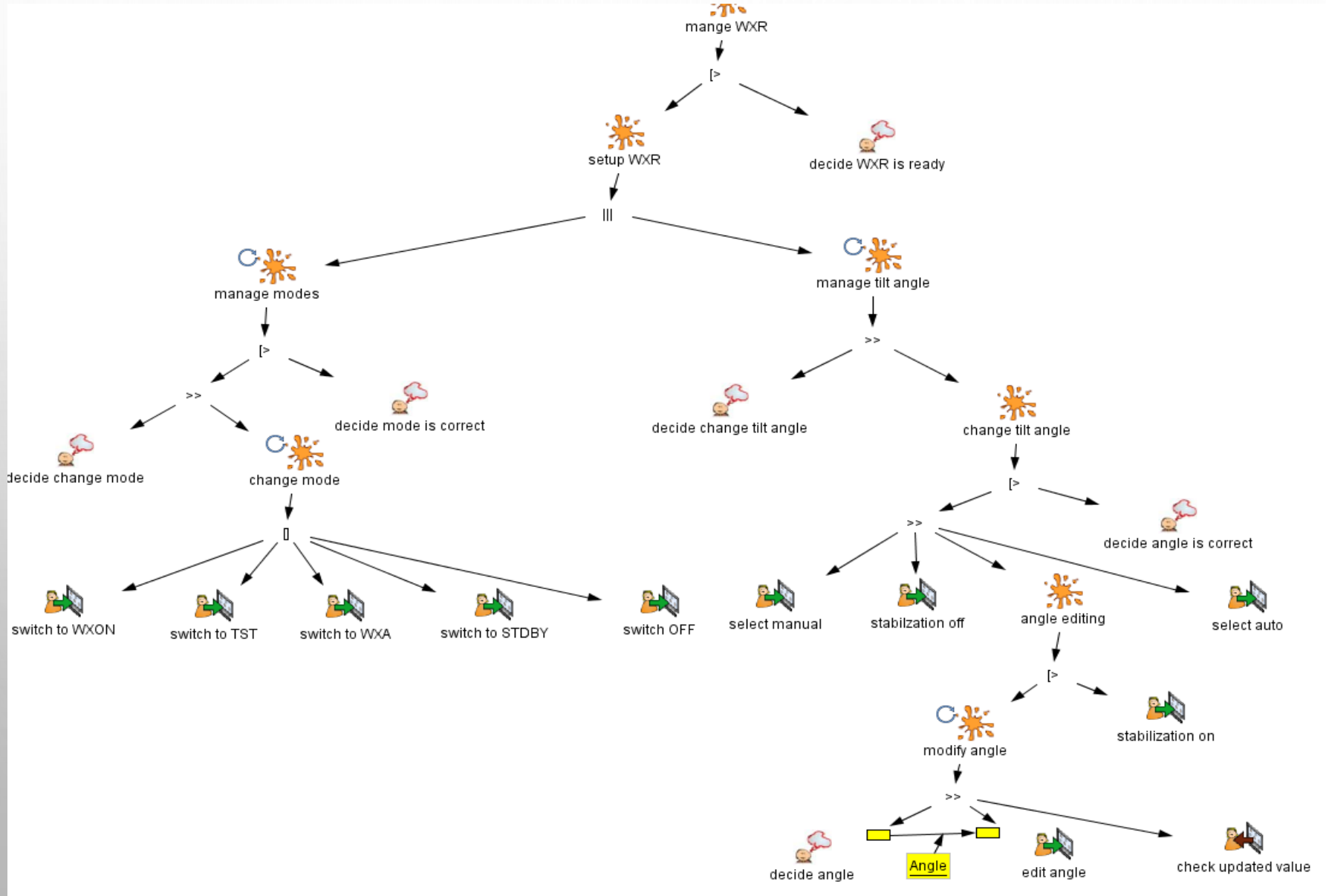
CONTRÔLE DU RADAR MÉTÉOROLOGIQUE

- Disponible dans plusieurs types de cockpit
  - Sélection du mode :
    - OFF, SDBY, TEST, WXON, WXA
  - Direction du faisceau:
    angle tilt [-15°; 15°]
  - En mode AUTO, tilt non modifiable
- Exemple simple mais réaliste
- Activités simples mais conséquences d'une erreur de manipulation et/ou d'interprétation potentiellement dramatiques

MODÉLISATION DES TÂCHES

- Interaction avec le systèmes
- Buts de haut niveau
- Intégration tâches/système

MODÉLISATION DES TÂCHES



DESCRIPTION FORMELLE DE SYSTÈMES INTERACTIFS

RÉFÉRENCES (LIVRES)

- Formal Methods in HCI, Harrison & Thimbleby 90, Cambridge University Press
- Formal Methods for Interactive Systems, Dix 91, Cambridge University Press
- Formal Methods in HCI, Palanque & Paterno 97, Springer Verlag

RÉFÉRENCES (CONFÉRENCES)

- Eurographics workshops DSV-IS (Design, Specification and Verification of Interactive Systems) depuis 94 publiés par Springer Verlag (LNCS depuis 2000)
- CHI 96 workshop on Formal Methods and HCI
- CHI 98 workshop on designing user interfaces for safety critical systems
- Mini-track de FM 99
- SUCA 2000 workshop (Safety and Usability Concerns in Aeronautics)

RÉFÉRENCES (JOURNAUX)

- Pas de revue dédiée
- Des "special issues"
  - Journal of Visual Language and Computing (vol 10, n°3, 1999)
  - ACM Transactions on Computer Human Interaction (début 2000)
  - Interacting with computers (BCS HCI group journal)

GROUPES DE TRAVAIL

- Pas de groupe dédiés
- Groupe IFIP 2.7 (13.4)
- Groupe FLASHI du GDR-PRC-CHM (de 96 à 98)
- Groupe ALF du GDR-PRC-I3 depuis 98

- Marginalement
  - Groupe IFIP 13.5 Human Error
  - Groupe IFIP 13.2 Methodologies

VÉRIFICATION DE LA COMPATIBILITÉ DES MODÈLES

- Tous les objets du modèle de tâches existent dans le modèle de données du système
- Toutes les actions du modèle de tâche sont offertes par le système
- Toutes les actions du système existent dans le modèle de tâche
- Toutes les séquences d'action du modèle de tâche sont "licites" dans le modèle du système
- Toutes les contraintes (requirements) exprimés sont vrai sur le modèle du système et/ou sur le modèle de tâche