AARHUS
UNIVERSITET

18. FEBRUARY 2014

TISYE1 – Lecture 4

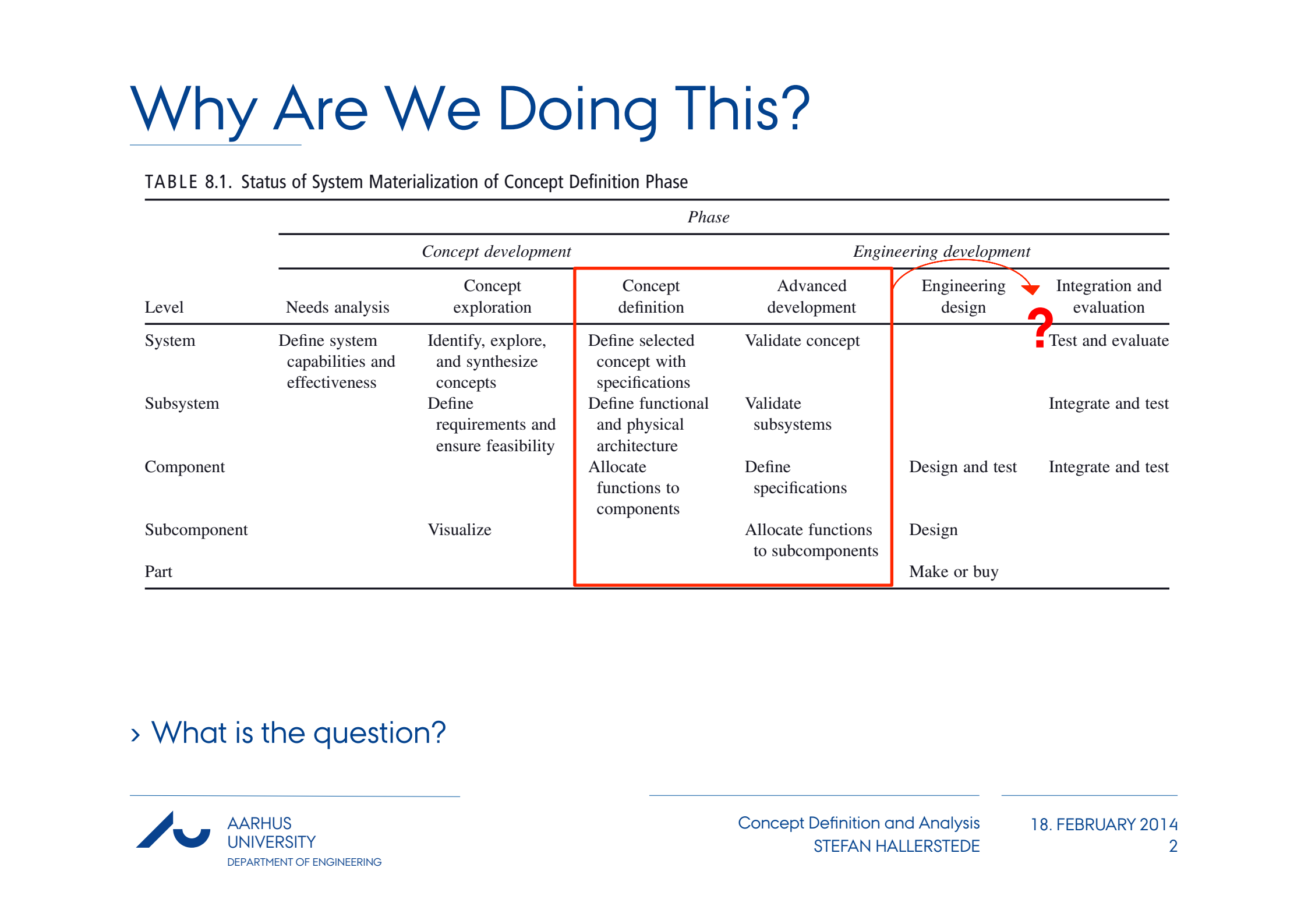
Concept Definition and Analysis

STEFAN HALLERSTEDE (SHA@ENG.AU.DK)
ASSOCIATE PROFESSOR

Why Are We Doing This?

TABLE 8.1. Status of System Materialization of Concept Definition Phase

	Phase					
	Concept development			*Engineering development*		
Level	Needs analysis	Concept exploration	Concept definition	Advanced development	Engineering design	Integration and evaluation
System	Define system capabilities and effectiveness	Identify, explore, and synthesize concepts	Define selected concept with specifications	Validate concept		Test and evaluate
Subsystem		Define requirements and ensure feasibility	Define functional and physical architecture	Validate subsystems		Integrate and test
Component			Allocate functions to components	Define specifications	Design and test	Integrate and test
Subcomponent		Visualize		Allocate functions to subcomponents	Design	
Part					Make or buy	



› What is the question?

Before We Start Building a System

› … we should ask this question

Are we building the right system?

› … and, maybe, this one:

Are we making the right choices?

On Not Asking Questions

- The “right system”:
  - Is a system that satisfies the requirements
- “Right choices”:
  - Are those choices that achieve
    - Performance
    - Reliability
    - Safety
    - Maintainability
    - …

Plan for the Lecture

› Concept Definition and Analysis:

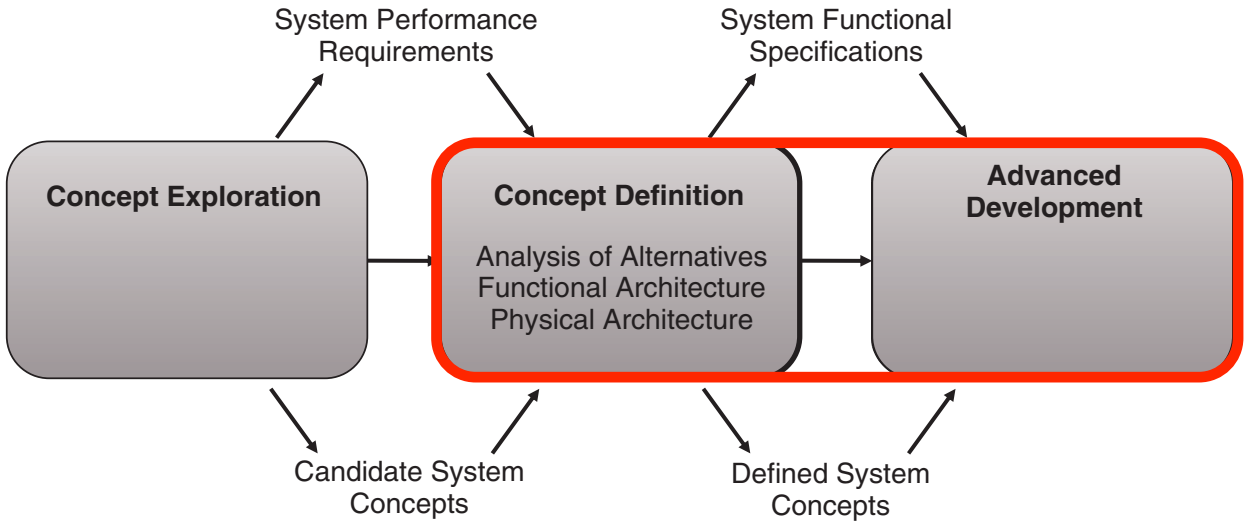


Figure 8.1. Concept definition phase in system life cycle.

Overview of Tasks

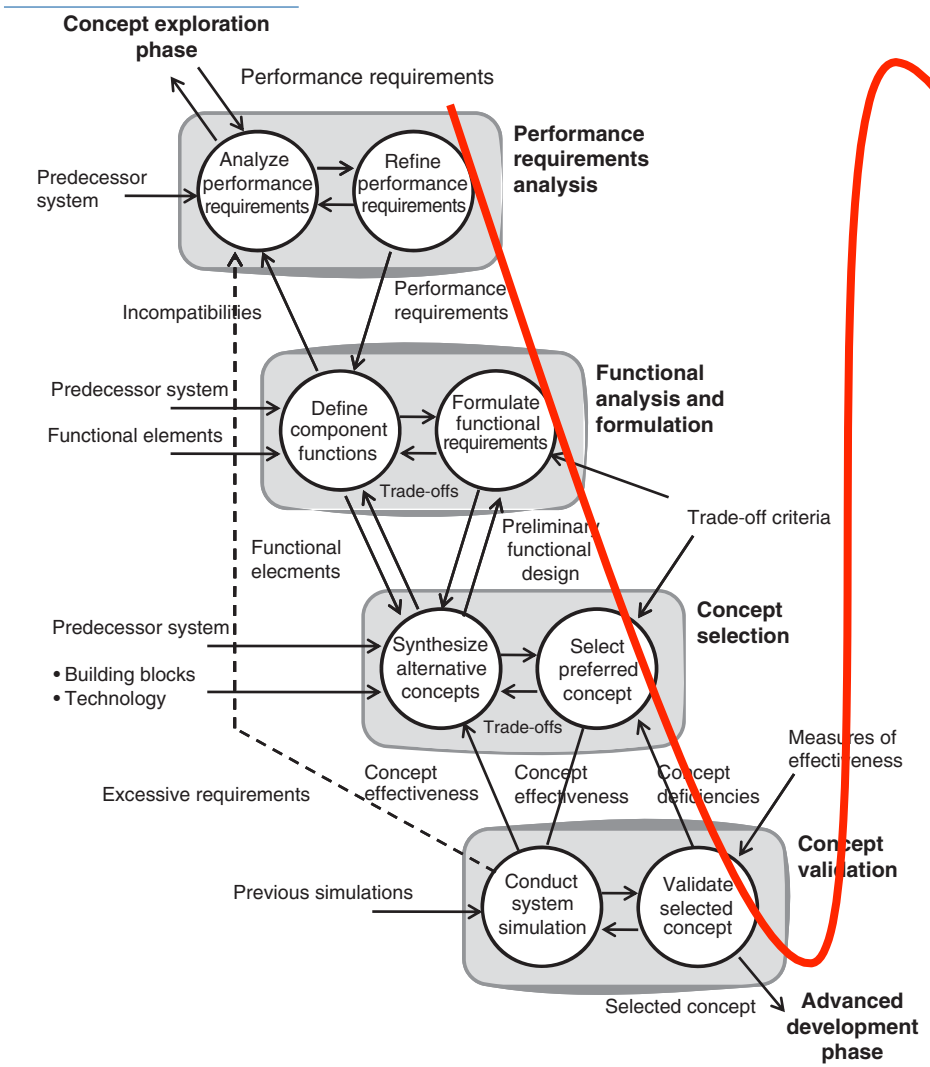


Figure 8.2. Concept definition phase flow diagram.

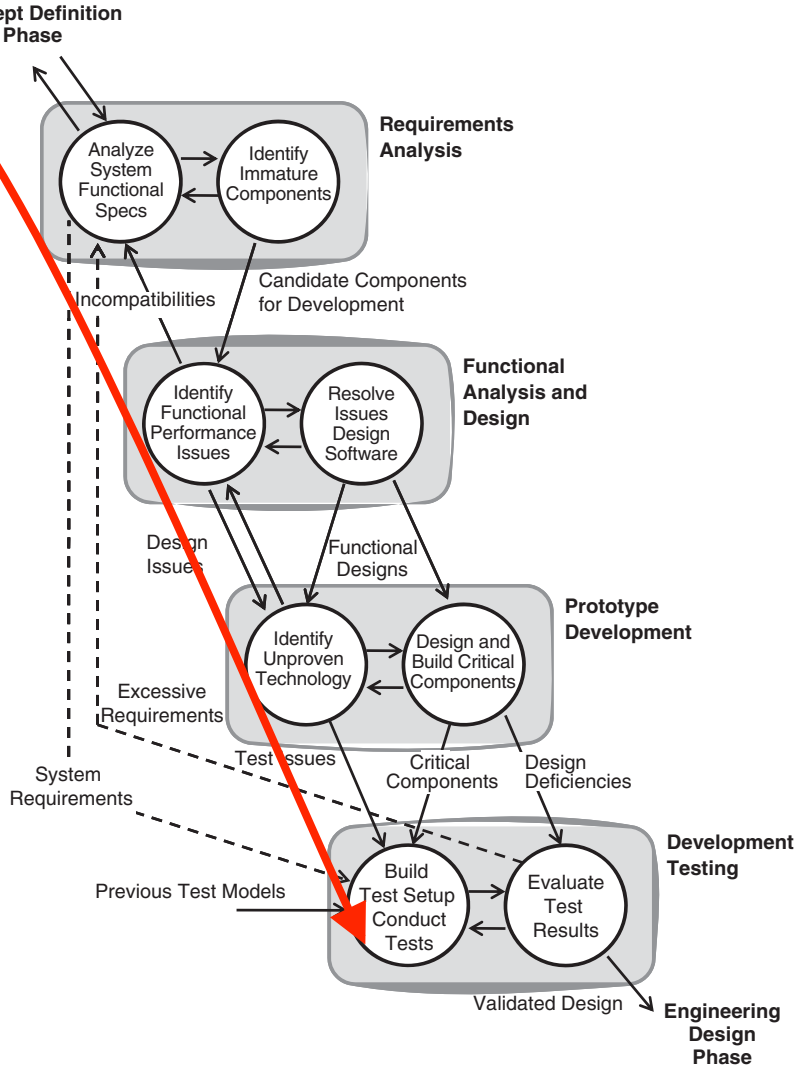
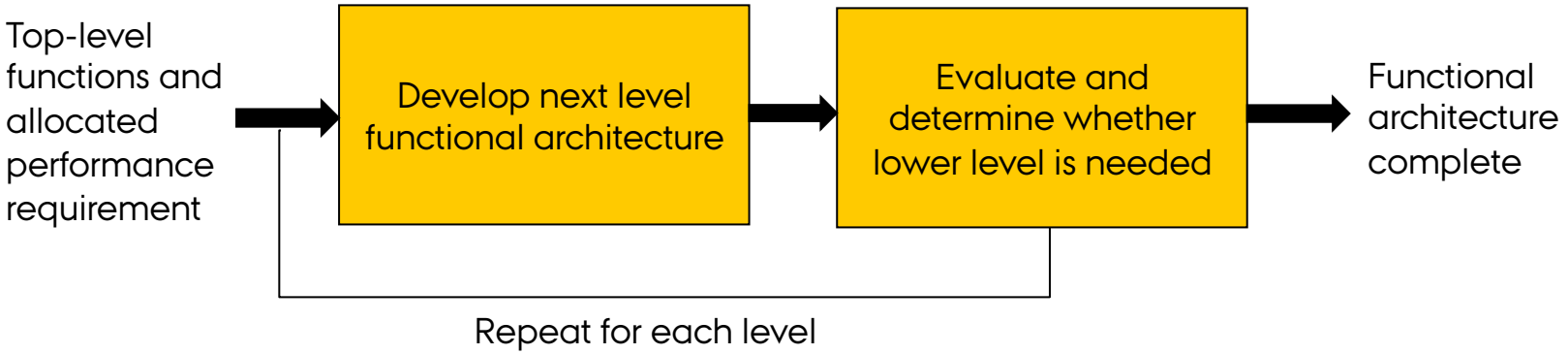


Figure 10.2. Advanced development phase flow diagram.

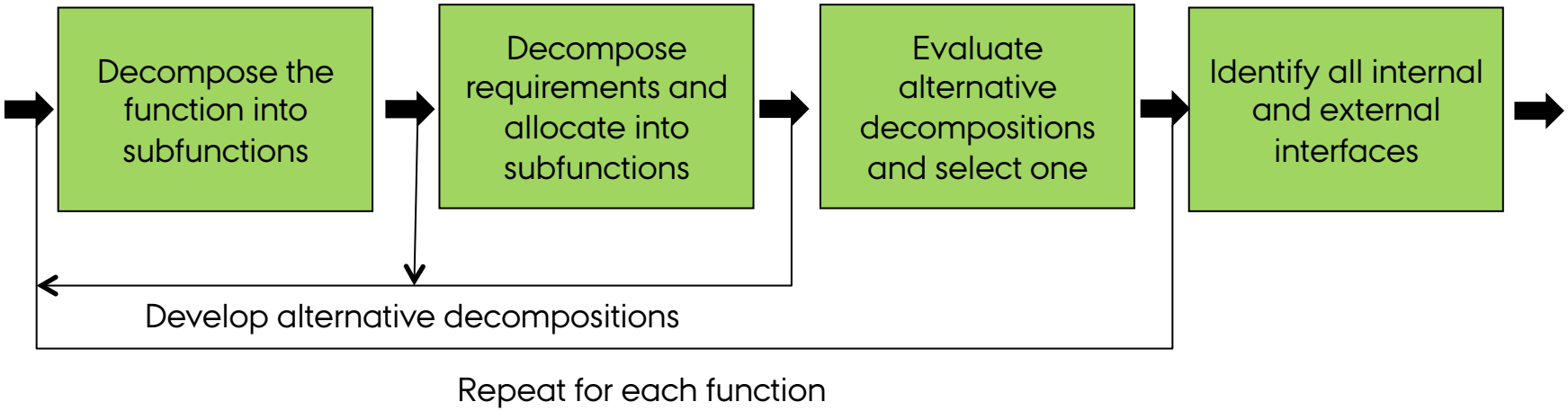
Consider System Life-Cycle

- storage of the system and/or its components,
- transportation of the system to its operational site,
- assembly and readying the system for operation,
- extended deployment in the field,
- operation of the system,
- routine and emergency maintenance,
- system modification and upgrading, and
- system disposition.

Functional Analysis/Allocation



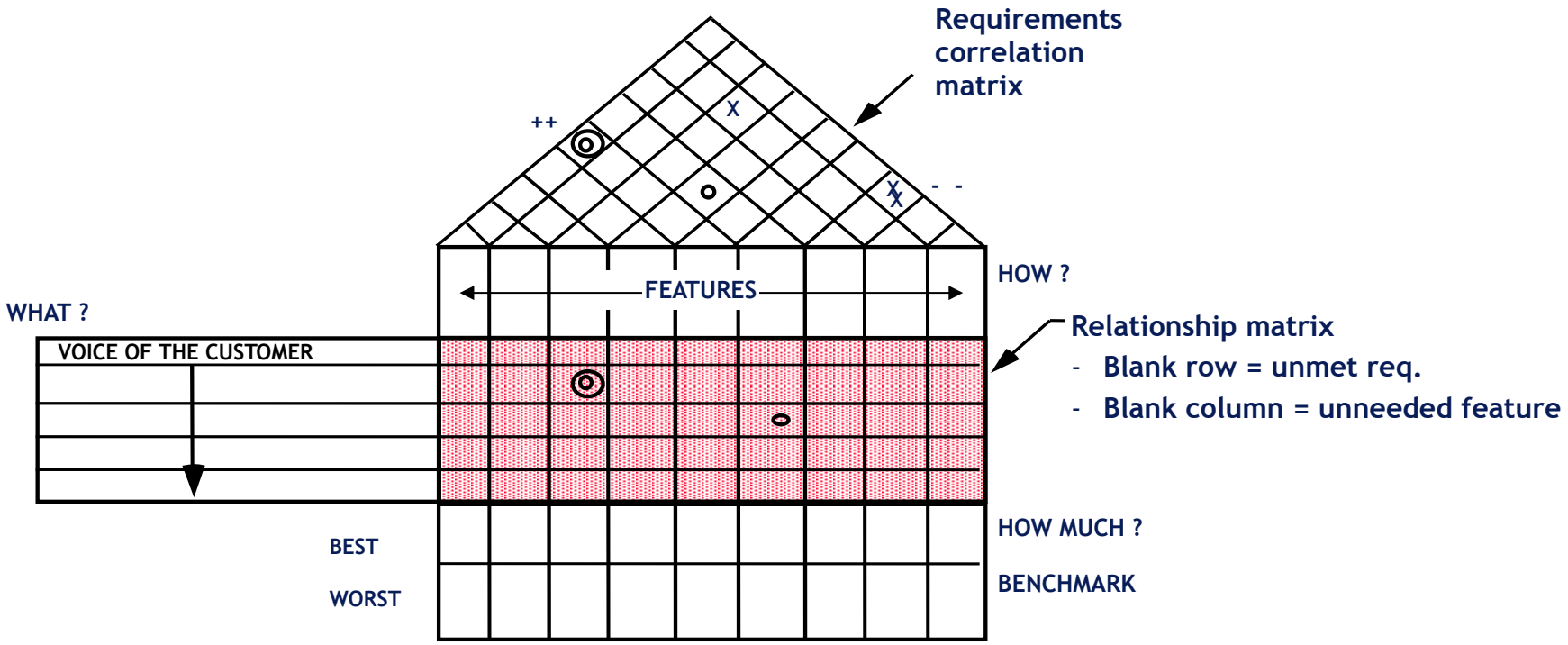
› For each level:



Decomposition Methods and Models

- The defined technical requirements can be analysed and decomposed by:
  - Function; Time; Behaviors; Data flow; Objects; States and Modes; Failure Modes and Effect
- The models of decomposition may include:
  - QFD, Function Flow Block Diagram (FFBD); time lines; data control flow; behaviour diagrams; operator task sequence etc.
- Analysis of decomposition and requirement allocation is based on cost, schedule, safety and risk analysis etc.
  - **Which safety are we talking about?**
  - **What kinds of risk do we consider?**

Quality Function Deployment (QFD)

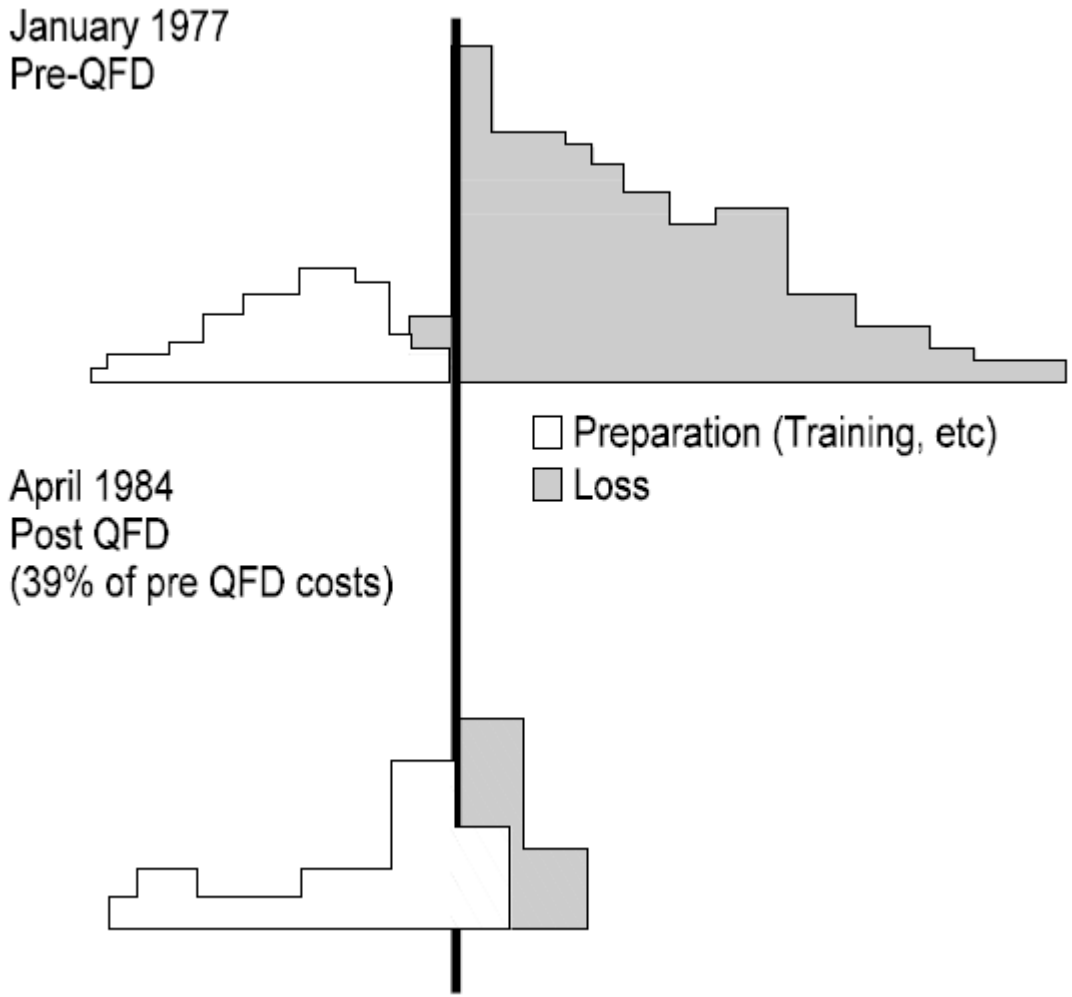


› A planning tool for translating customer requirements into specifications.
› Deploys the “voice of the customer”
› A fast, systematic way to derive & flow down requirements to design, parts, manufacturing, and productions

Startup and preproduction cost at Toyota

› Before QFD

› After QFD



› Source: Stevens Institute of Technology, Course 625

Functional Block Diagram (FBD)

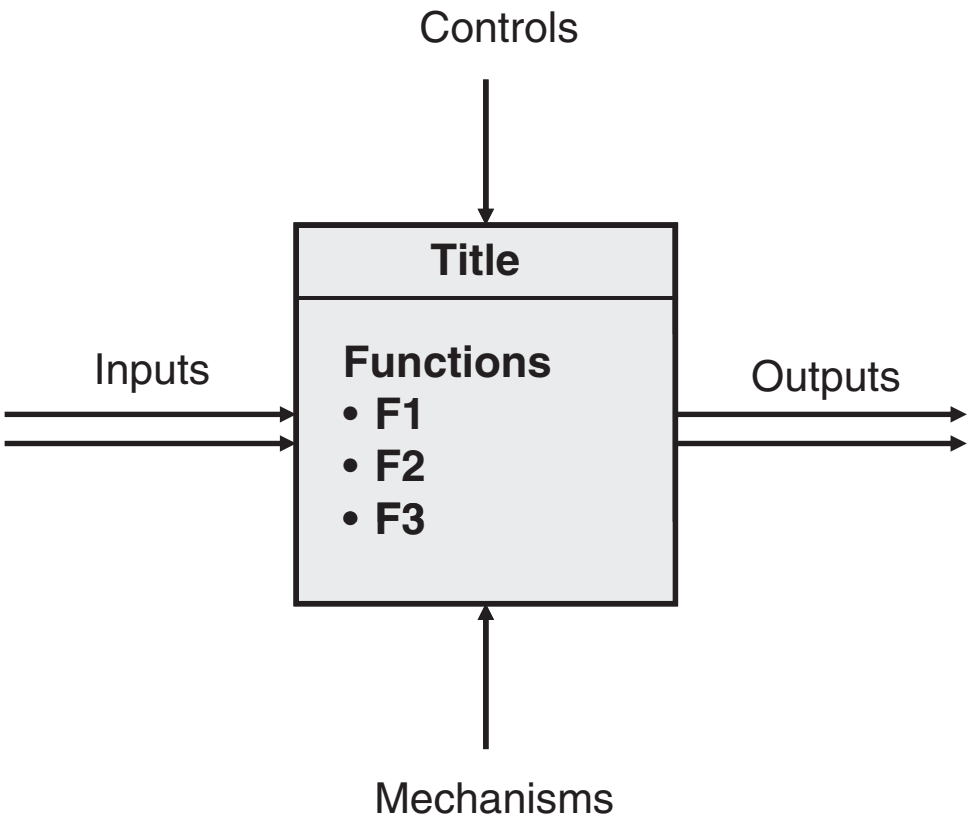
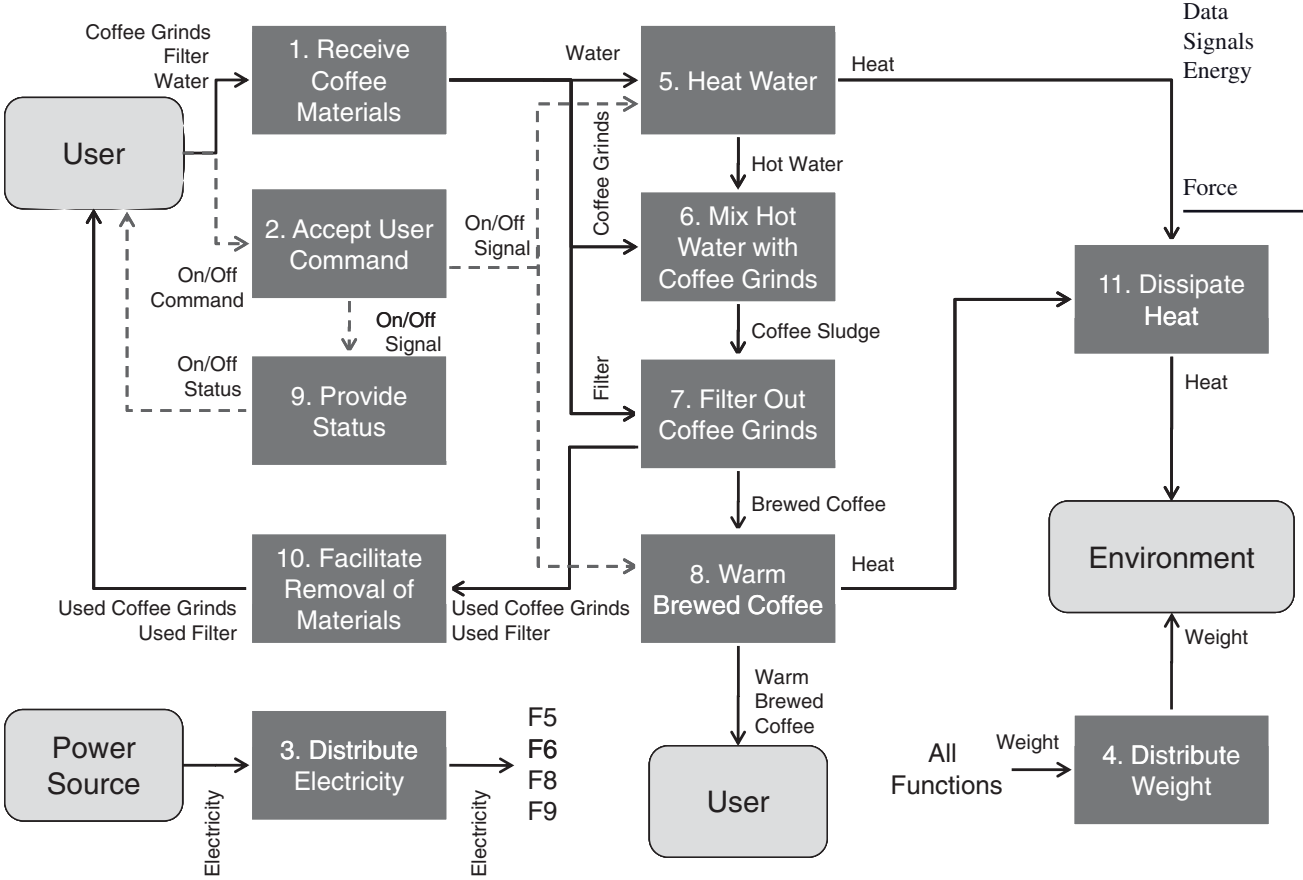


Figure 8.3. IDEF0 functional model structure.

Functional Analysis of a Coffeemaker



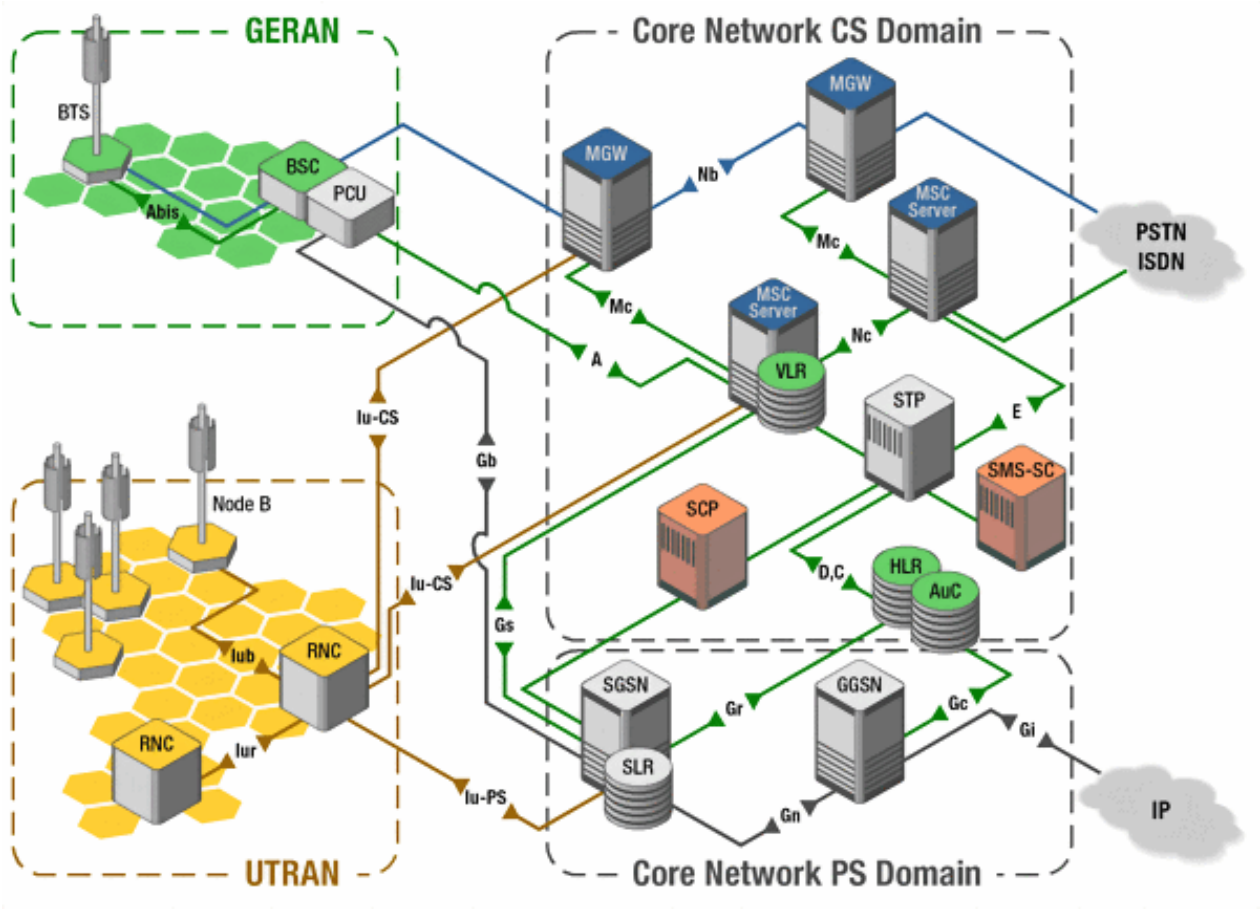
Materials	Receive coffee materials
	Mix hot water with coffee grinds
	Filter out coffee grinds
	Facilitate removal of materials
Data	Provide status
Signals	Accept user commands
Energy	Distribute electricity
	Heat water
	Warm brewed coffee
	Dissipate heat
Force	Distribute weight

What is this useful for?

Figure 8.4. Functional block diagram of a standard coffeemaker.

Example

› UMTS network functional architecture



AuC	Authentication Server
BSC	Base Station Controller
BTS	Base Transceiver Station
CS	Circuit Switched
GERAN	GSM/EDGE RAN
GGSN	Gateway GSN
GPRS	General Packet Radio Service
GSN	GPRS Support Node
HLR	Home Location Register
IP	Internet Protocol
ISDN	Integrated Service Digital Network
MGW	Media GateWay
MSC	Mobile Switching Center
PCU	Packet Control Unit
PS	Packet Switched
PSTN	Public Switched Telephone Network
RAN	Radio Access Network
RNC	Radio Network Controller
SGSN	Serving GSN
SCP	Service Control Point
SLR	SGSN Location Register
SMS SC	SMS Service Center
STP	Signaling Transfer Point
UTRAN	UMTS RAN
VLR	Visitor Location Register

Hardware-Software-Mechanics Allocation

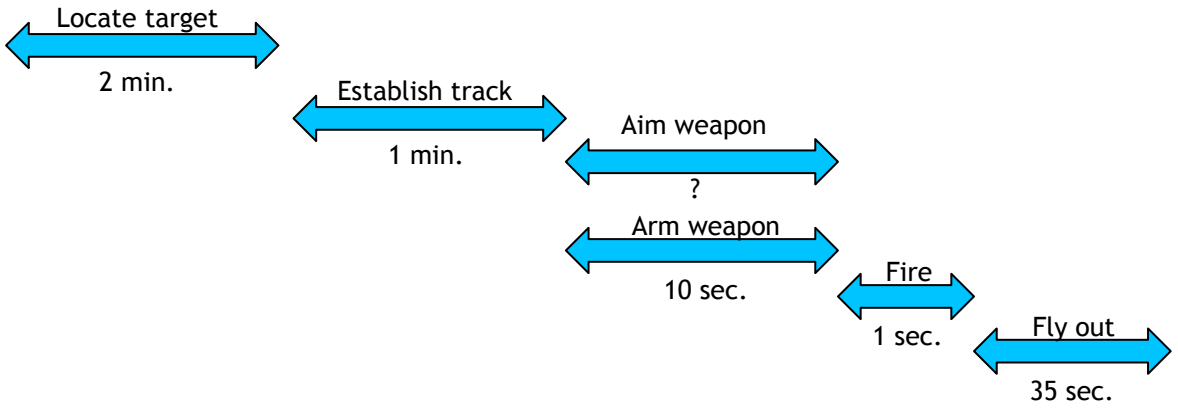
› The issue whether a function should be realized in hardware, software or by mechanics depends on system level decision (often driven by non-functional requirements).

› The functional definition at component level should include the allocation of all functional processing functions to either hardware, software or mechanics.

FPGA

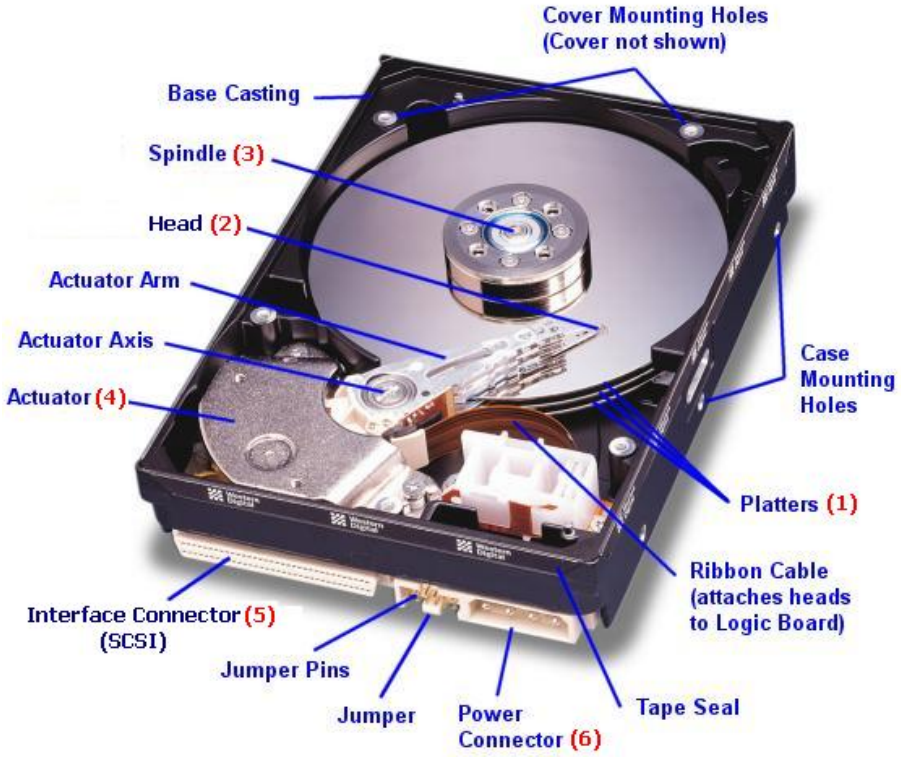
Example: Response-Time Analysis

(Non-functional requirement analysis)



- The system shall destroy a target within 5 minutes of receipt of order.
- The system shall locate the target within 2 minutes of receipt of order
- The system shall establish track within 1 minutes of locating the target
- The system shall arm the weapon within in 10 sec. of establishing track
- The system shall fire the weapon within 1 second of completing the aim of the weapon
- The weapon shall fly out to the target within 35 seconds of being fired

Architecture, Architects and Architecting

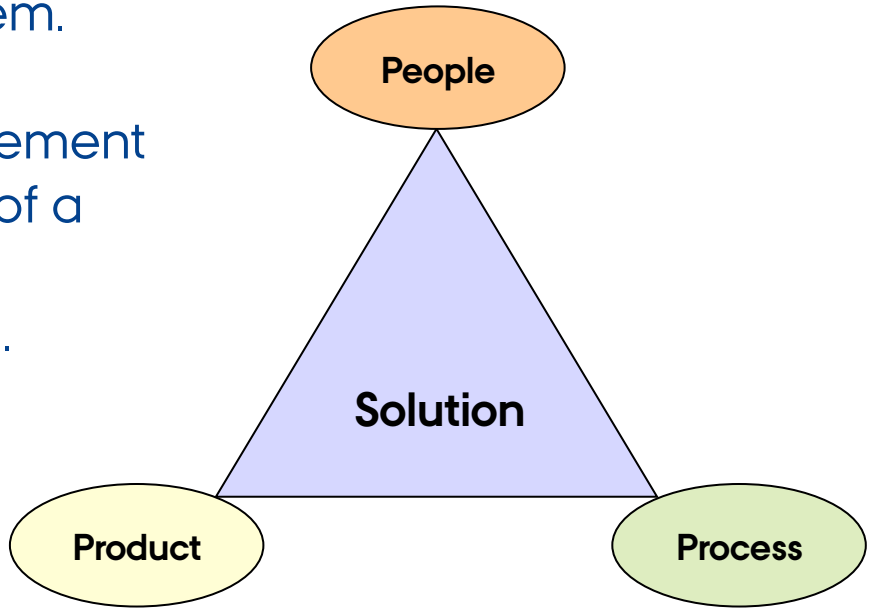


- Lock-in to older hardware architecture.
- Unable to see the new vision.

› Developing (and maintaining) the system architecture is one of the most important responsibilities of the system engineer

System Architecture Synthesis

› Steps in synthesizing a possible solution:
  › Selecting the types of system elements that comprise the system.
  › Assessing their characteristics.
  › Determining an effective arrangement that fits within the design space of a possible system architecture arrangements of those elements.



› Architectural design (architecture synthesis) is not an exact science. It requires insight, outlook and experience.

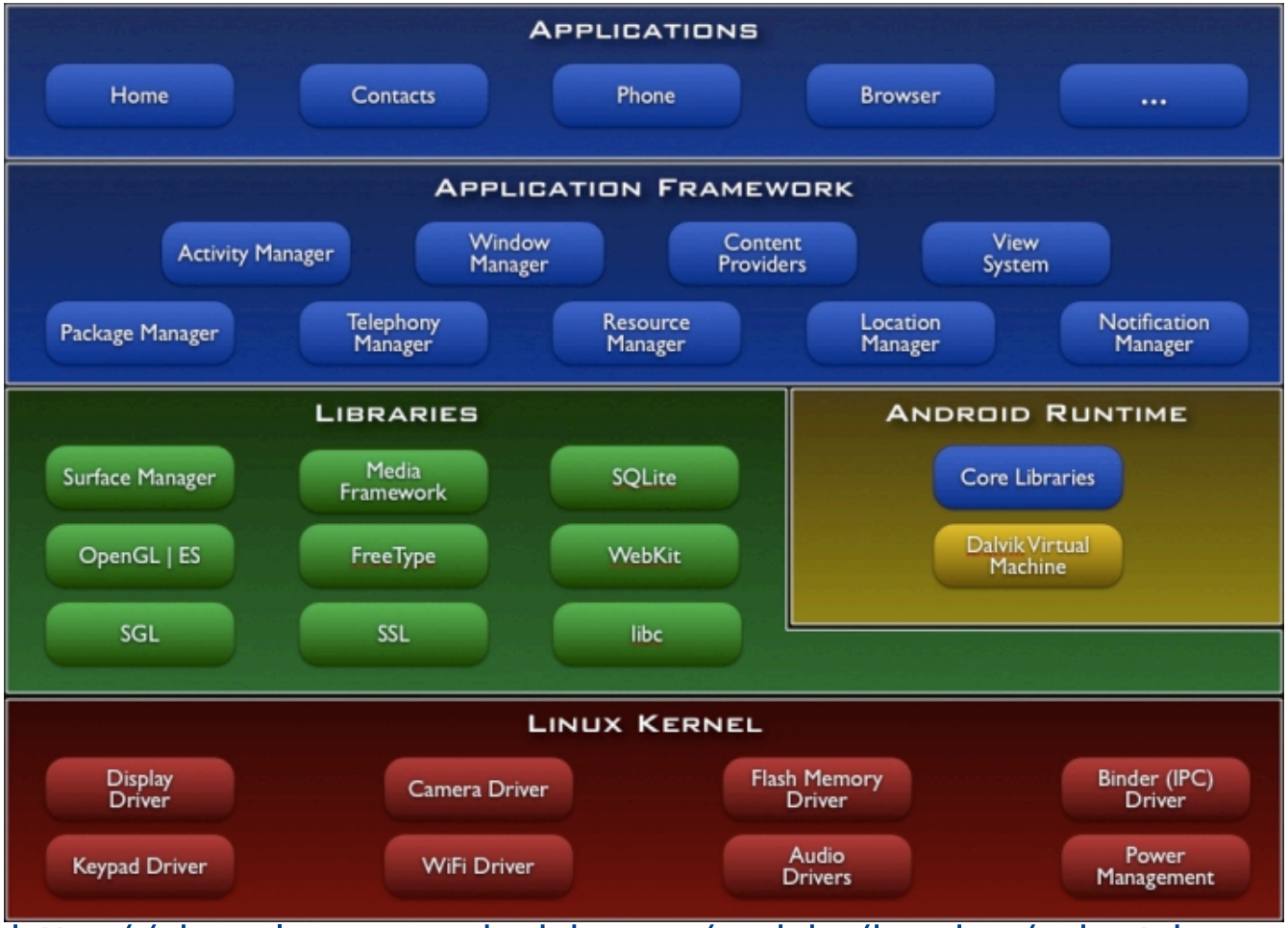
Verify the Architecture

The synthesis of an architecture solutions is an iterative process that requires feedback from subsequent steps.

- Specific questions to verify the architecture are:
  - Does the architecture perform all functions of the system?
  - Is it capable of meeting requirements and satisfies all constraints?
  - Is resource usage within acceptable limits?
  - Are elements compatible?
  - Are all interfaces satisfied?

Android Architecture

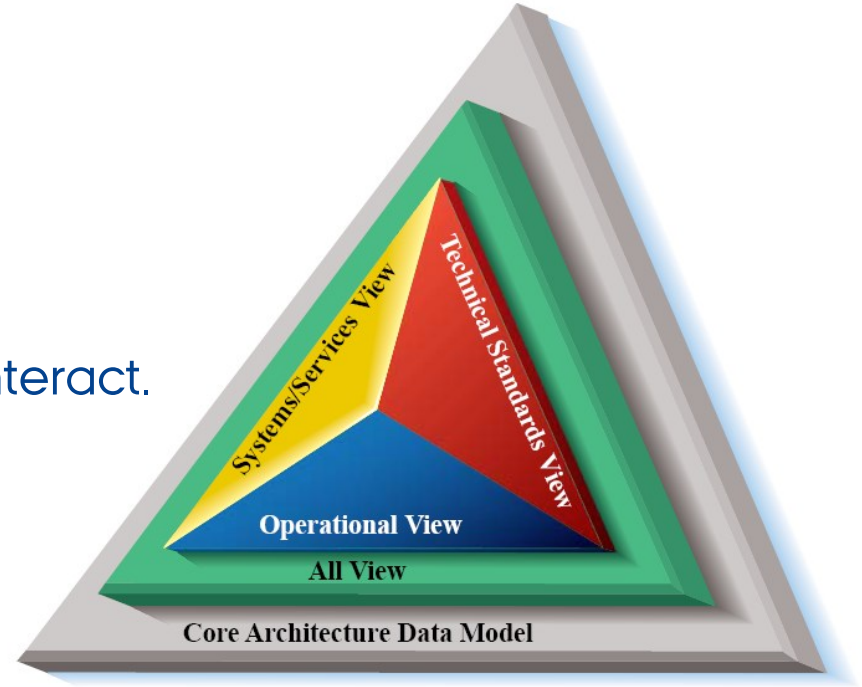
› (major components of the Android operating system)



› From: http://developer.android.com/guide/basics/what-is-android.html

Architecture Views

- Operational view
  - How it is used.
- Logic view
  - How functions interrelate and interact.
- Physical view
  - How it is build.



- Example: Architectural view framework (DoDAF)
  http://en.wikipedia.org/wiki/Department_of_Defense_Architecture_Framework
  http://dodcio.defense.gov/dodaf20/dodaf20_viewpoints.aspx

DoDAF 2.0 Architectural Views

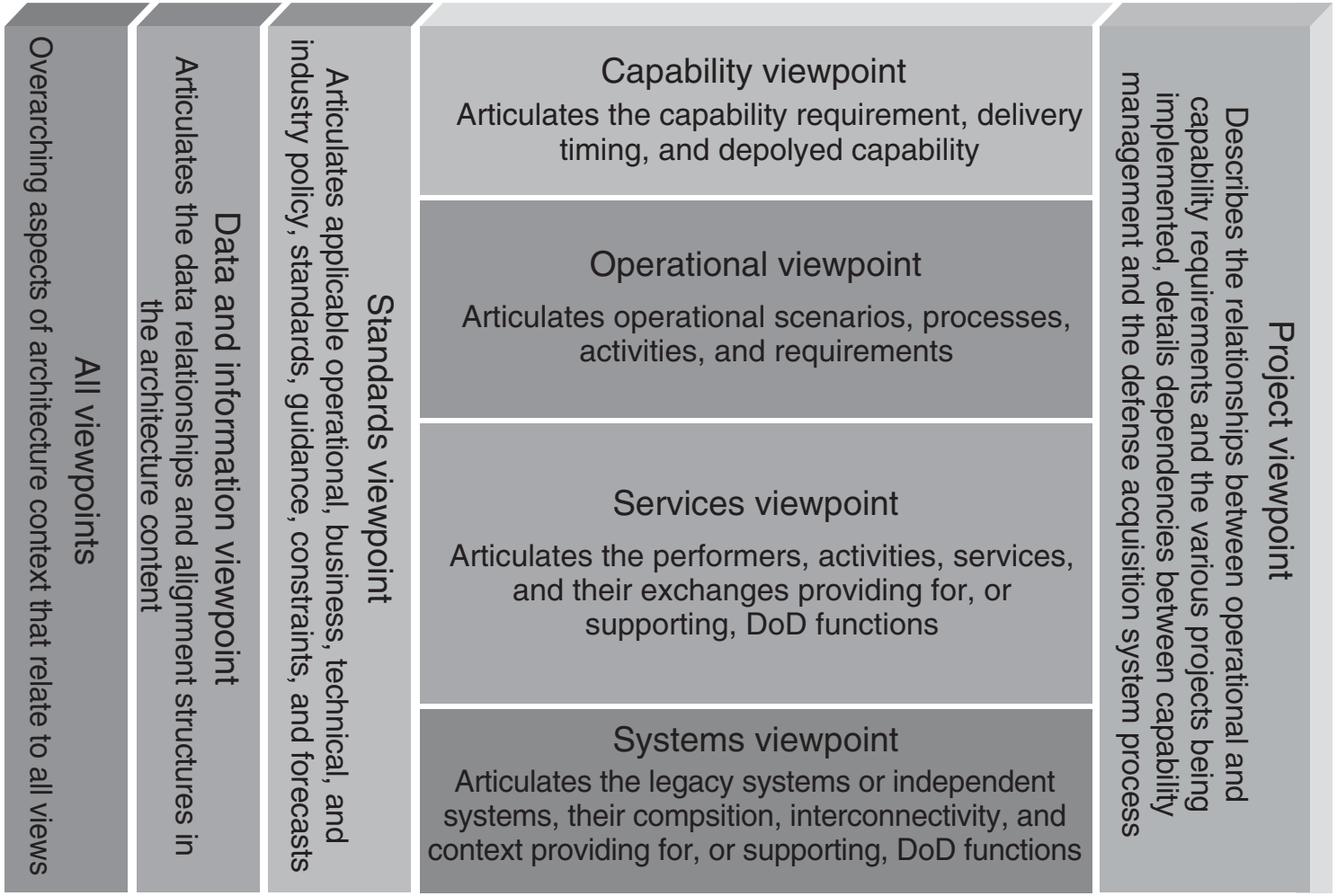
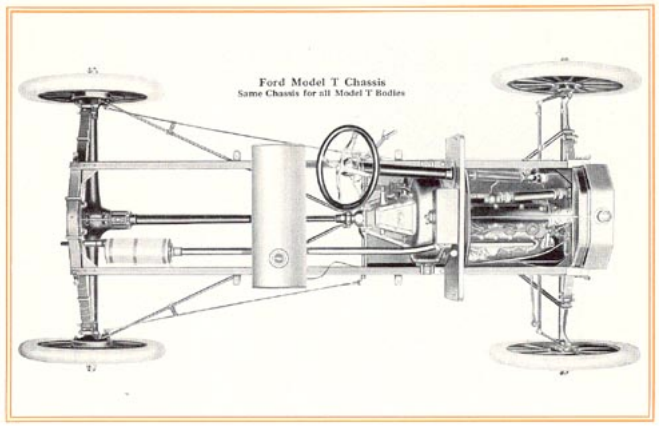


Figure 8.6. DODAF version 2.0 viewpoints.

Example: Natural-Gas Automobile

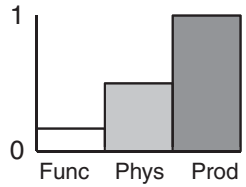
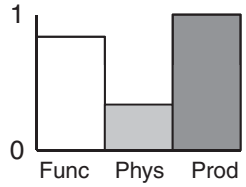
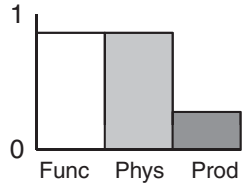
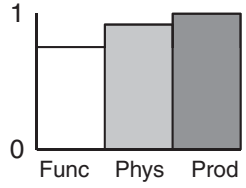
› Current technology:



› Well, we wait for a couple of years…

Evaluation of Novelty in Design

TABLE 10.2. Development of New Components

Design approach	Maturity	Development	Validation
New function Proven physical medium and production method		Design, build, and test rapid prototype	Functional performance
New implementation Proven function and production method		Design, build, and test rapid development model	Engineering design
New production method Proven function and implementation		Perform critical experiments on the production method	Production method
Extended function Proven component		Design and run functional simulation	Functional performance

Maybe...

› ... too small

Or Bigger:

› … well-suited for larger families

OK, Here is a Solution

› Let's have a look in the boot…

Where do my Shoppings go?

Anyway, …

1. We should have specified the needs first
2. Then we should have derived the requirements
3. Only then are we ready to architect and evaluate the architecture!

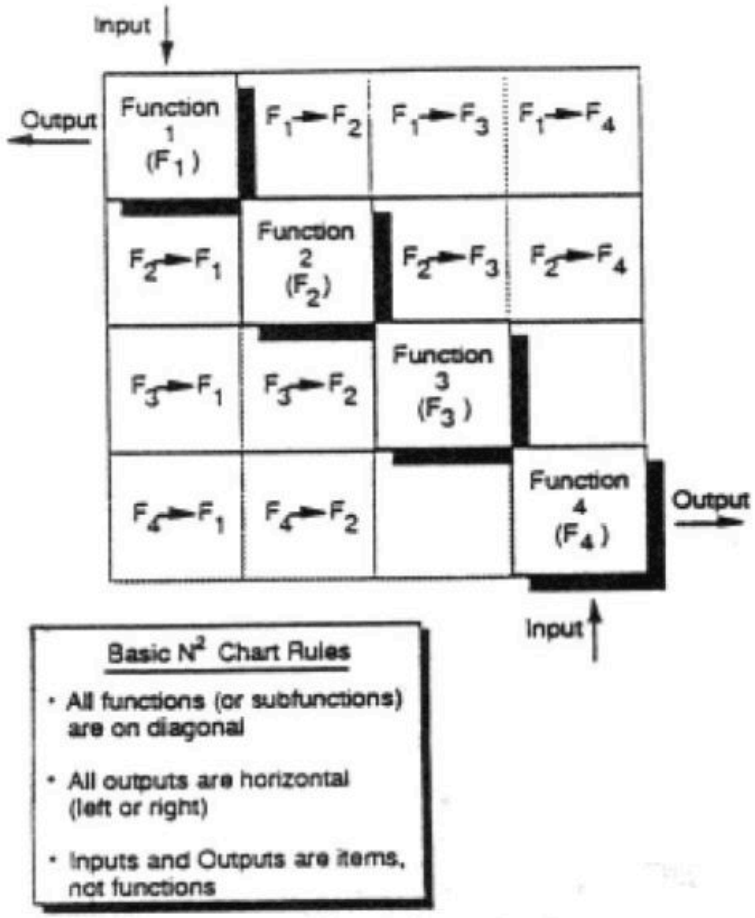
Functional Interface Definitions

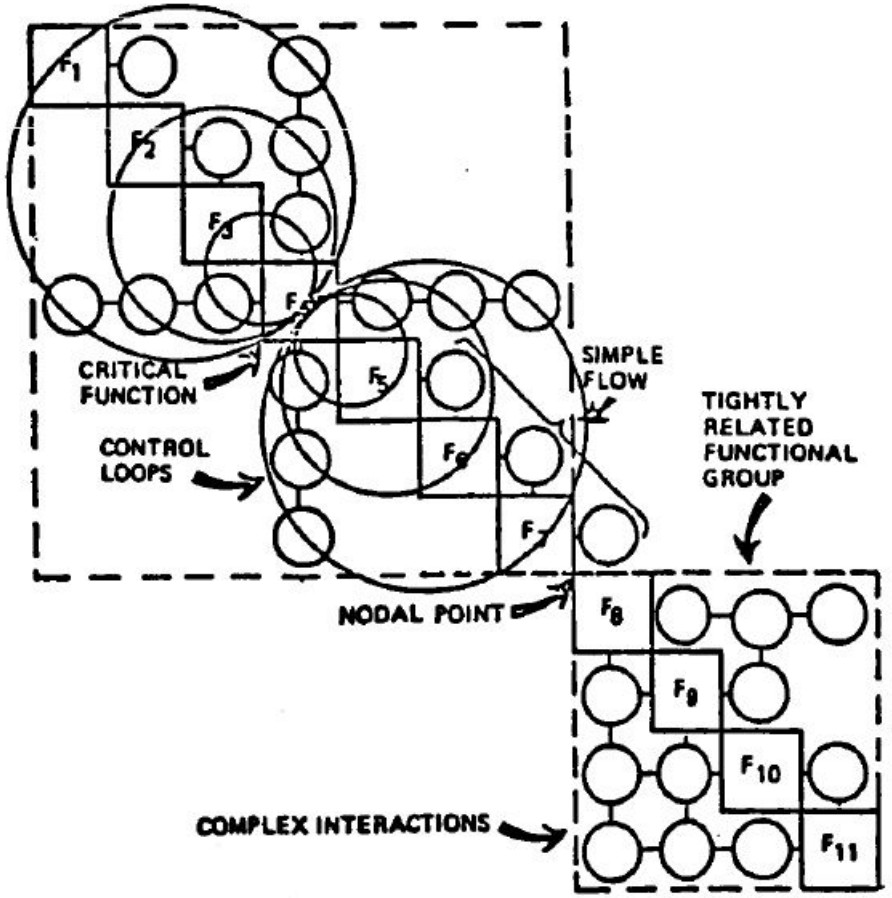
› Functions require inputs to operate and produces outputs

› Two classes of interfaces
› Internal – A connection between elements inside the system
› External – A connection between element of a system to an outside entity

› N^2 – diagrams are used to map input/output relationships associated with interfaces

N^2 – Diagrams


Input
Output
Function 1 (F_1)
$F_1 \rightarrow F_2$
$F_1 \rightarrow F_3$
$F_1 \rightarrow F_4$
$F_2 \rightarrow F_1$
Function 2 (F_2)
$F_2 \rightarrow F_3$
$F_2 \rightarrow F_4$
$F_3 \rightarrow F_1$
$F_3 \rightarrow F_2$
Function 3 (F_3)
$F_4 \rightarrow F_1$
$F_4 \rightarrow F_2$
Function 4 (F_4)
Output
Input
Basic N^2 Chart Rules
• All functions (or subfunctions) are on diagonal
• All outputs are horizontal (left or right)
• Inputs and Outputs are items, not functions


CRITICAL FUNCTION
CONTROL LOOPS
SIMPLE FLOW
TIGHTLY RELATED FUNCTIONAL GROUP
NODAL POINT
COMPLEX INTERACTIONS

Internal Interfaces

› The system engineer is responsible for the overall technical quality of the system,
hence the system engineer is a major stakeholder in defining the interfaces internal to the system.

› Examples
› Between embedded computers in the system.
› Between mission support software and embedded computers.
› Some components are a systems themselves (cf. SoS).

› The system engineer does not have to write these documents, but the system engineer must supervise their creation and participate in reviews.

Interface Control Documents

- Different types of ICDs
  - Software – describing the use of exact bits, frequencies of messages etc.
  - Electrical – describing voltage level, active state, logical meaning
  - Mechanical – Installation drawing description exact measurements, weights location of holes for mounting etc.

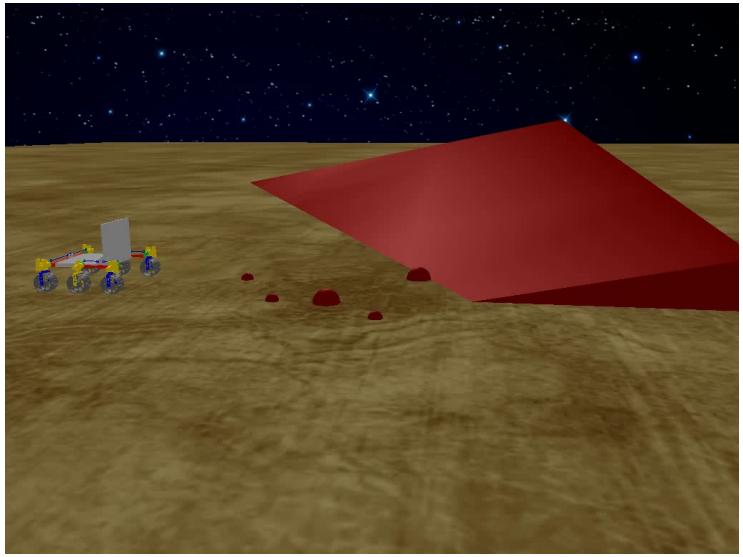
- Defining all the above is the responsibility of the system engineer.

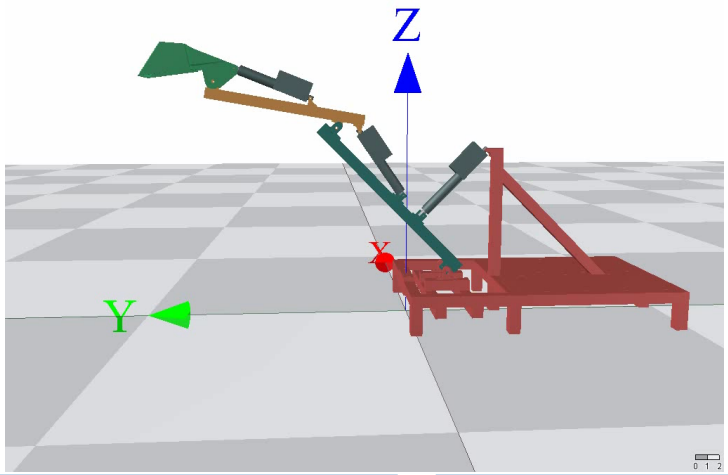
Modelling, simulation and prototyping

- Essential elements in analysing and selecting the preferred system architecture.
- This analysis is best conducted with the aid from subject matter experts.
- Modelling, simulation and prototyping can significantly reduce the risk of failure in the finished system.

Prototypes, models and simulation

- Software Engineering Group
  Dept. of Engineering, AU
  (Peter Gorm Larsen)
- Mars Rover
- River excavation





Validation and Verification

› Testing is planned early

› TEMP:
Test and Evaluation
Master Plan

› All tests are specified and recorded:
› Test scenario
› Data collection
› (Performance) evaluation

PART I	SYSTEM INTRODUCTION
	MISSION DESCRIPTION
	SYSTEM DESCRIPTION
	SYSTEM THREAT ASSESSMENT
	MEASURES OF EFFECTIVENESS AND SUITABILITY
	CRITICAL TECHNICAL PARAMETERS
PART II	INTEGRATED TEST PROGRAM SUMMARY
	INTEGRATED TEST PROGRAM SCHEDULE
	MANAGEMENT
PART III	DEVELOPMENT TEST AND EVALUATION OUTLINE
	DEVELOPMENT TEST AND EVALUATION OVERVIEW
	FUTURE DEVELOPMENTAL TEST AND EVALUATION LIMITATIONS
PART IV	OPERATIONAL TEST AND EVALUATION OUTLINE
	OPERATIONAL TEST AND EVALUATION OVERVIEW
	CRITICAL OPERATIONAL ISSUES
	FUTURE OPERATIONAL TEST AND EVALUATION LIMITATIONS
	LIVE FIRE TEST AND EVALUATION
PART V	TEST AND EVALUATION RESOURCE SUMMARY
	TEST ARTICLES
	TEST SITES AND INSTRUMENTATION
	TEST SUPPORT EQUIPMENT
	THREAT REPRESENTATION
	TEST TARGETS AND EXPENDABLES
	OPERATIONAL FORCE TEST SUPPORT
	SIMULATIONS, MODELS, AND TEST BEDS
	SPECIAL REQUIREMENTS
	TEST AND EVALUATION FUNDING REQUIREMENTS
	MANPOWER/PERSONNEL TRAINING
APPENDIX A	BIBLIOGRAPHY
APPENDIX B	ACRONYMS
APPENDIX C	POINTS OF CONTACT
ATTACHMENTS	(AS APPROPRIATE)

SOURCE: *Defense Acquisition Guidebook*, September 2004.

Testing methodology

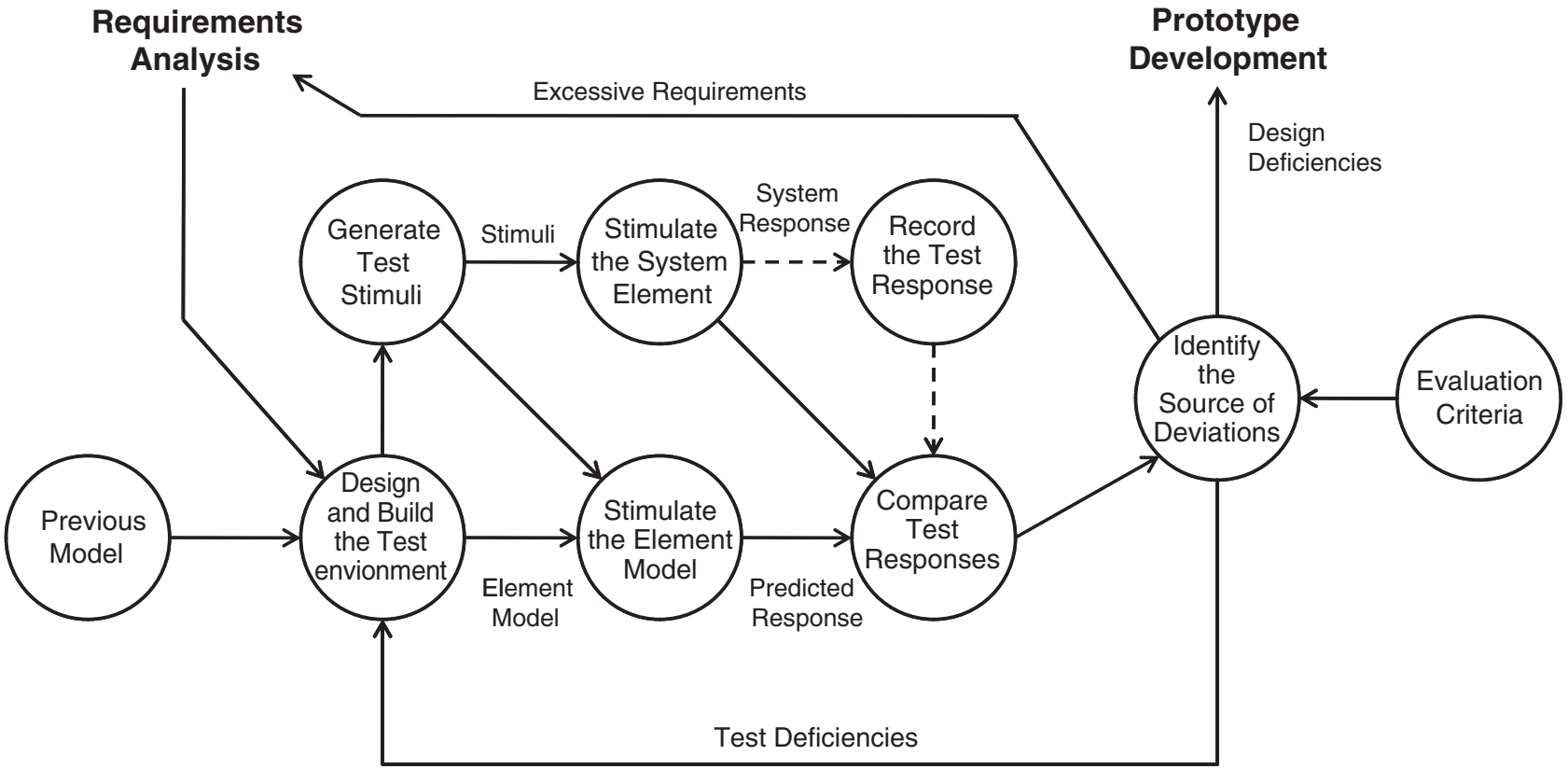


Figure 10.3. Test and evaluation process of a system element.

Speciality Engineering Activities

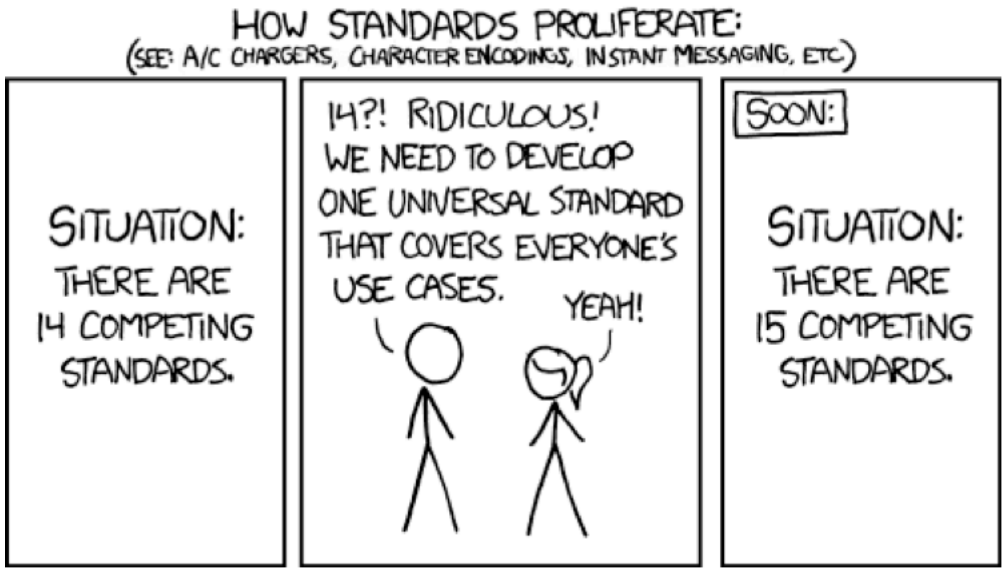
› Integrated Logistics Support (ILS)
› Cost-effectiveness Analysis
› Electromagnetic Compatibility Analysis (EMC)
› Environmental Impact Analysis
› Interoperability Analysis
› Life-cycle-Cost (LCC) Analysis
› Manufacturing and Producibility Analysis
› Mass Properties Engineering Analysis (MPE)
› Safety & Health Hazard Analysis
› Sustainment Engineering Analysis
› Training Needs Analysis
› Usability Analysis/Human Systems Integration
› Value Engineering

Integrated Logistics Support (ILS)

› The scope of ILS includes:

› Acquisition Logistics – activities influencing the design of a system and its readiness for utilization and support.

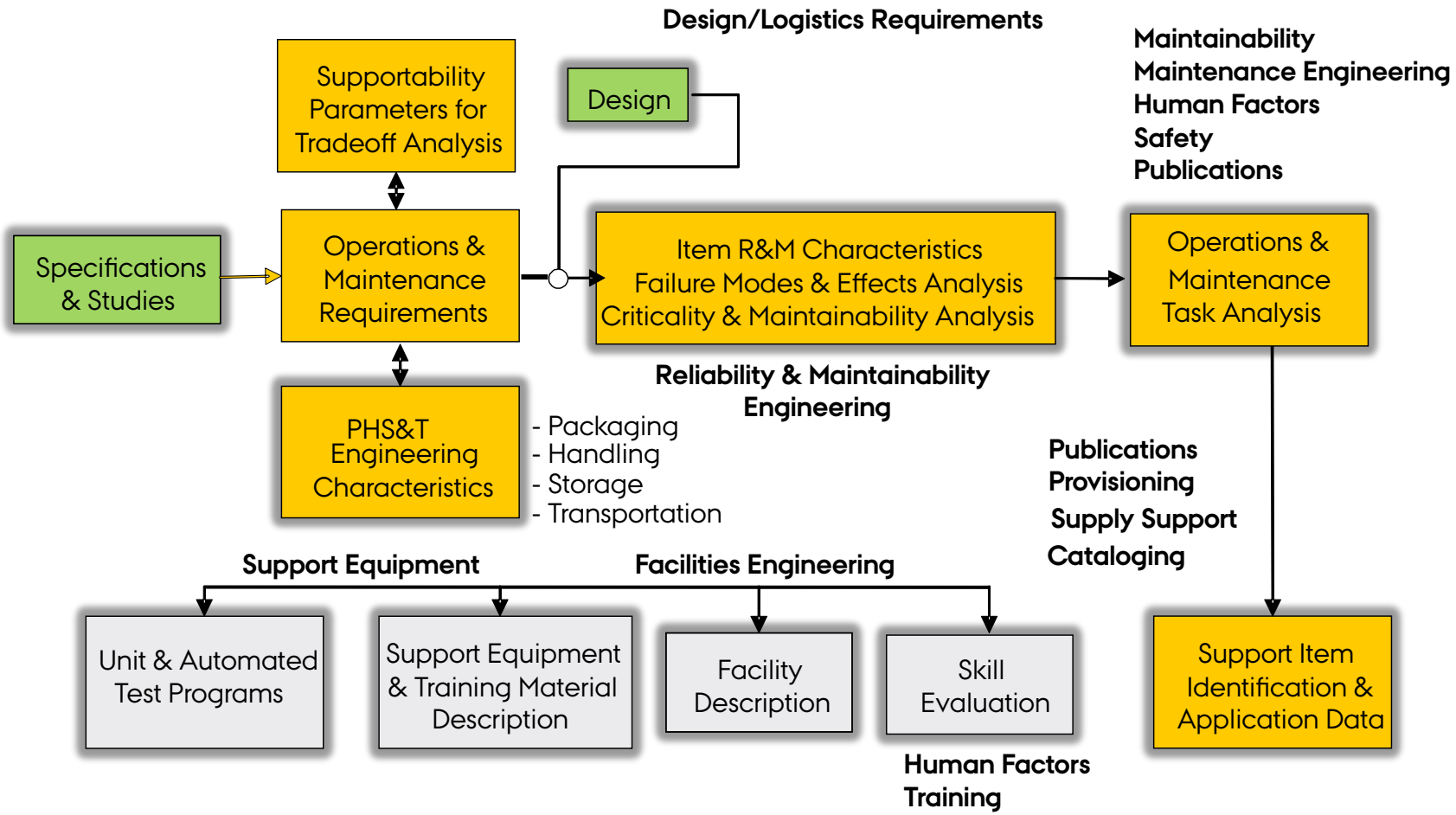
› Operational Logistics – activities which ensure that the right material and resources, in the right quantity and quality, are available at the right place at the right time throughout the utilization and support stages.



› ILS also receives attention under the heading of Supply Chain Management, product support, customer services.

Activities during earlier stages affect the sustainment of the operations and maintenance processes during the utilization and support stages.

Relationship between ILS analysis activities



Quality indicators

- Acquisition Logistics focus on design, requirements, and criteria applicable to all system elements such as
  - Affordability (Life Cycle Cost)
  - Cost/System Effectiveness
  - Disposability (Recycling / Retirement)
  - Maintainability
  - Packaging, Handling, Storage and Transportation (PHS&T)
  - Producibility (Manufacturability)
  - Reconfigurability (Flexibility / Standardization)
  - Reliability
  - Security
  - Supportability (Serviceability)
  - Survivability
  - Vulnerability

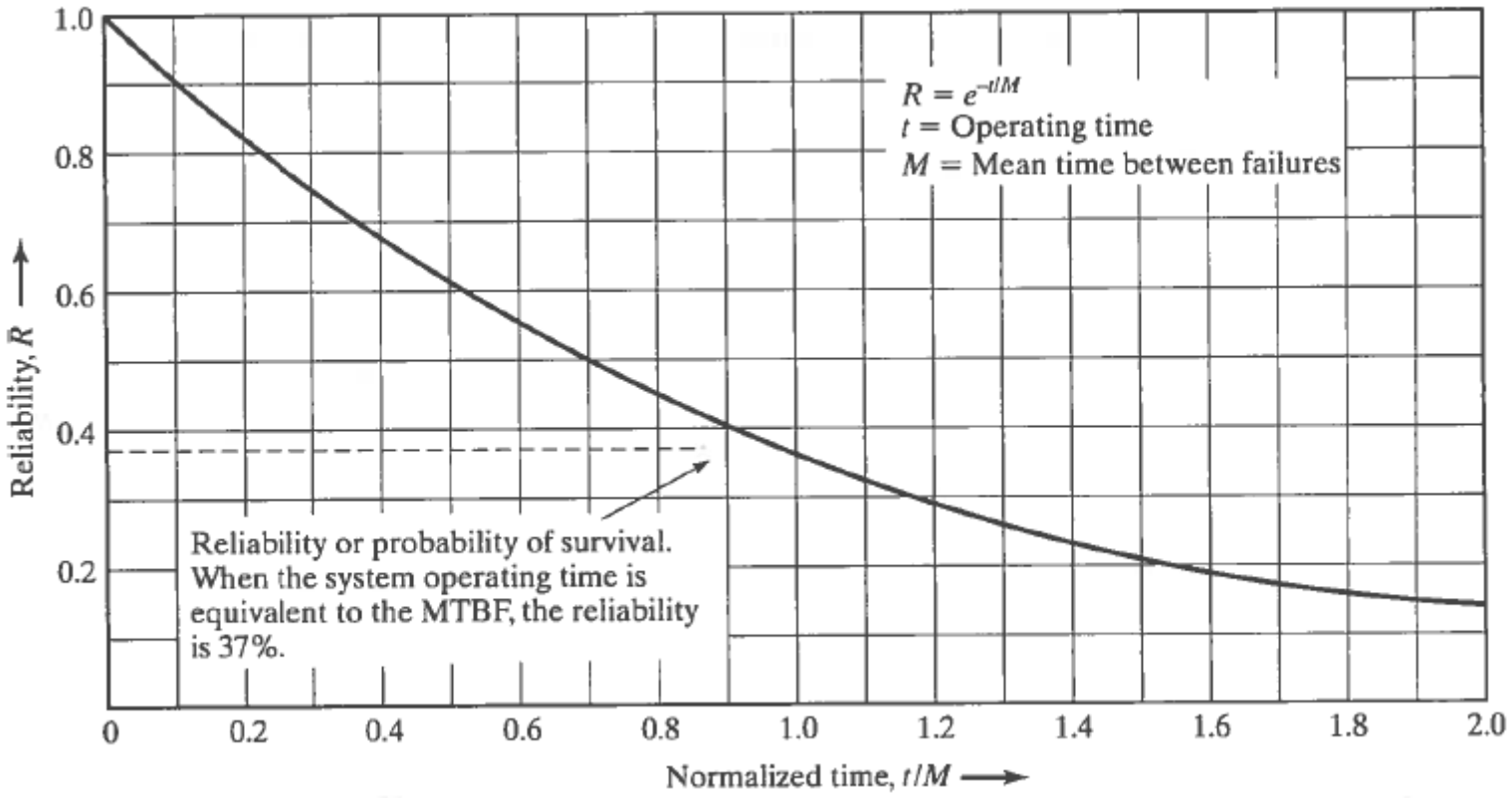
Analysis Methods

- Within reliability, maintainability and supportability, several analyses are performed iteratively and recursively as they are co-dependent on other analyses:
  - Failure Modes Effects and Criticality Analysis (FMECA or FMEA)
  - Level of Repair Analysis (LRA)
  - Logistic Support Analysis (LSA) / Supportability Analysis
  - Reliability Centred Maintenance Analysis (RCM)
  - Survivability analysis
  - System Security Analysis
  - …

Metrics

System attribute	Metric which can be used
Reliability	Mean time to failure
Reliability	Rate of occurrence of failure
Availability	Probability of failure on demand
Performance	Number of transactions to be processed per second
Performance	Response time to user input
Store utilization	Maximum size of system in kilobytes
Usability	Time taken to learn 75% of user facilities
Usability	Average number of errors made by users in a given time period
Robustness	Time to re-start after system failure
Integrity	Maximum permitted data loss after system failure
Maintainability	Mean time to restore (MTTR)

System Reliability Over Time



Maintainability

› A measure of the ease of accomplishing functions required to maintain the system, in a fully operable condition.
  › Repair
  › Schedule periodic servicing

› Metric: Mean time to restore (MTTR)