ulm university universität uulm

STUDIENBRIEF

MANAGEMENT ASPECTS OF SYSTEMS ENGINEERING I

Weiterbildender Masterstudiengang „Sensorsystemtechnik“
der Fakultät für Ingenieurwissenschaften und Informatik
mit dem Abschluss „Master of Science (M. Sc.)“
an der Universität Ulm

5 Management-Aspects of Systems Engineering I

Kürzel / Nummer:	SyE
Englischer Titel:	Management-Aspects of Systems Engineering I
Leistungspunkte:	6 ECTS
Sprache:	Deutsch/Englisch
Turnus / Dauer:	jedes Wintersemester / 1 Semester
Modulverantwortlicher:	Michael Leute (Fa. Airbus Defence and Space)
Dozenten:	Michael Leute Sascha Ackva
Einordnung des Moduls in Studiengänge:	Sensorsystemtechnik, M.Sc., Wahlpflichtmodul, Management-Aspekte
Voraussetzungen (inhaltlich):	None
Lernziele:	Today, starting at household applications providing automation services up-to highly complex systems as the International Space Station a lot of functionality is integrated and thus systems get more and more complex. In addition, the environment a system is installed imposes constraints increasing the complexity. Those constraints could be raised by a competitive market, by already installed systems or are given by laws and regulation. Systems Engineering has developed over the decades from a capability of a few experienced senior engineers to an acknowledged profession. Systems Engineering has proven to be a key enabling factor for companies to handle and manage complexity and to ensure effective communication among all stakeholders allowing developing successful systems. The course provides the rationale and foundation for applying Systems Engineering and an introduction into the thinking of Systems Engineering. The student explains the difference of a system in comparison to its system elements. He describes the role of a systems engineer within a project organisation and his work areas . The student understands the importance of upfront analysis in order to avoid unplanned costs in later life cycle stages. He recognizes that communication, e.g. with specialist disciplines, is a fundamental element of the SE tasks. The student analyses a system completely on a specific abstraction level considering the complete life cycle before advancing into deeper details towards implementation. The student works with a wide scope overlooking all technical aspects of the whole system instead of focusing solely on a single detail. The student elicits requirements starting from the stakeholder needs, identifies possible solution architectures, performs trade studies and decomposes and allocates the requirements down the system hierarchy. He applies different techniques to validate architecture, design and requirements. The student manages the technical aspects of a system and coordinates with the specialist engineering disciplines as well as project management.

Inhalt:	- Definition of a System and Systems Engineering, System Hierarchy and Development Lifecycle - Justification and Rationale for the Systems Engineering discipline i.e. managing complexity, effective communication and common understanding - The role of the Systems Engineer - Systems Engineering Processes - Typical Reviews (Quality Gates) in the development life cycle, and their objectives - Requirements Engineering - System Architecture - Integration and Testing - Project life cycle and development models
Literatur:	- INCOSE Systems Engineering Handbook (ISBN 978-1-937076-02-3) - Systems Engineering, Principles and Practice, Kossiakoff et.al.
Lehrveranstaltungen und Lehrformen:	Präsenzveranstaltungen: Einführungsveranstaltung: 2 h (Präsenztag zu Beginn des Moduls) Seminar zur Prüfungsvorbereitung: 4 h Modulprüfung: 2 h (schriftliche Prüfung) E-Learning: Webinar: 4 h Online-Vortrag: 16 h Selbststudium: 140 h Zwischentests: 8 h (Multiple-Choice) Chat zur Prüfungsvorbereitung: 4 h
Abschätzung des Arbeitsaufwands:	Vermittlung des Unterrichtsstoffs: 38 h Vor- und Nachbereitung, Übungen, Anwendung: 140 h Modulprüfung: 2 h Summe: 180 h
Leistungsnachweis und Prüfungen:	Zur Modulprüfung wird zugelassen, wer die Aufgaben aus den Übungen erfolgreich bearbeitet hat. Die Modulprüfung erfolgt schriftlich (multiple choice).
Voraussetzungen (formal):	None
Notenbildung:	Die Modulnote ergibt sich aus der Modulprüfung

1. Definitions

1.1 What is a System? 11
1.2 What is Systems Engineering (SE)? 12
1.3 Who is INCOSE / GfSE? 13
1.4 System Life Cycle Stages 13
1.5 Generic System Hierarchy (INCOSE) 15
1.6 Systems of systems 20

2. Justification and Rationale 22

2.1 The Mission of Systems Engineering 22
2.2 Systems Engineering Challenges and Cost of Failures (Examples) 23
2.2.1 Heathrow, Sun Mar 30, 2008 23
2.2.2 Airbus A380, June,13, 2006 24
2.2.3 Ariane 5, June 4, 1996 25
2.2.4 Hubble space telescope, 1990, 1993 26
2.2.5 USS Yorktown, 1997 27
2.2.6 Sea Launch Zenit 3SL. March 12, 2000 27
2.2.7 EADS GAST, EADS Loses Military Contract 28
2.2.8 Conclusion 28
2.2.9 Jules Verne, Friday, April 3, 2008 29
2.2.10 Eugen Maersk 30
2.3 Chaos Report, Standish Group International, Inc. 31
2.3.1 Reasons for a Successful Project 32
2.3.2 Factors that cause projects to be Challenged 32
2.3.3 Factors that Cause Projects to Be Canceled 33
2.3.4 Conclusion from Standish 33
2.4 Cost to Fix 36
2.5 Life Cycle Cost 37
2.6 Value of Systems Engineering 38
2.6.1 SE Effort – Actual Cost 38
2.6.2 SE Effort – Schedule Overrun 39
2.6.3 Conclusion 39
2.7 Myths 40
2.8 I need a Systems Engineer 41

3. The role of the Systems Engineer 42

3.1 SE Network 42
3.1.1 Authorities 43
3.1.2 Customer 43
3.1.3 HW & SW Development 43
3.1.4 Other Systems 43
3.1.5 Procurement & Suppliers 44

3.1.6 Product Support 44
3.1.7 Production 44
3.1.8 Program and Product Management 44
3.1.9 Project Management 44
3.1.10 QA and CM 45
3.1.11 Sales and Marketing 45
3.1.12 Speciality Engineering 45
3.1.13 Stakeholder 45
3.1.14 System provider 45
3.1.15 Systems Engineer 45
3.1.16 User 46
3.1.17 V&V 46
3.2 Customers, Users and Stakeholders 46
3.2.1 The Architect's Role 46
3.2.2 Customer Involvement 47
3.3 Project Management 48
3.3.1 Project Planning 48
3.3.2 Project Control 51
3.4 HW/SW Development 55
3.4.1 Interface between SE, HW and SW Development 57
3.4.2 Information Provided by SE 57
3.4.3 Relation of HW / SW development to Safety Processes (e.g. ARP4754, ISO 26262) 61
3.5 Speciality Engineering 62
3.5.1 ILS Responsibilities 63
3.5.2 Acquisition Logistics Focus on Design Requirements Criteria 63
3.5.3 Interactions with ILS Example RAMS 64
3.6 "-itilities" Analysis Methods - Availability, Reliability, Maintainability 65
3.6.1 FMEA / FMECA 67
3.6.2 Packaging, Handling, Storage and Transportation (PHS&T) 67
3.6.3 Level of Repair Analysis 68
3.6.4 Logistic Support Analysis (LSA) 69
3.6.5 Survivability Analysis 69
3.6.6 System Security Analysis 70
3.6.7 Electromagnetic Compatibility Analysis 70
3.6.8 Environmental Impact Analysis 71
3.7 Human Systems Integration (HSI) 72
3.7.1 Human Performance and Human Engineering Design Requirements 73
3.7.2 Mass Properties Engineering (MPE) 73
3.7.3 Safety & Health Hazard Analysis 74
3.7.4 Sustainment Engineering Analysis 75
3.7.5 Training Needs Analysis 76
3.8 Production 77
3.8.1 Manufacturability 77
3.8.2 Stabilize Manufacting Process 77
3.8.3 Design simplification 78
3.8.4 Testing 79
3.8.5 Integration 79
3.9 Configuration Management 80

3.9.1 Overview of Configuration Management (CM) 80
3.9.2 Baselines 81
3.9.3 Baselines in the Life Cycle 83
3.9.4 Change Control 84
3.9.5 The Systems Engineer's role in the CM Context 86
3.9.6 The Importance of the Product Baseline 87

4. Processes

4.1 Need for Processes 88
4.2 Standards 89
4.3 Common Understanding on Systems Engineering Processes 91
4.4 Examples of Process Models 92
4.4.1 Waterfall Model (see Fig. 28) 92
4.4.2 V-Model 93
4.4.3 Spiral Model (see Fig. 30 on page 95) 94
4.4.4 Incremental Approach 96
4.4.5 The V-Model with Iterative and Incremental Development 97
4.5 Milestones and Reviews 98
4.6 Generic Systems Engineering Process 100
4.6.1 Requirements Elicitation Process (P1) 101
4.6.2 Requirements Elicitation Process Objectives 101
4.6.3 System Design Process (P2) 111
4.6.4 SE Activities during Implementation on Next-Lower Level 139
4.6.5 System Integration, Verification and Qualification 140

5. Models and Prototypes 159

5.1 Abstract 159
5.2 Motivation 159
5.3 What is a model? 160
5.4 Types of models 161
5.4.1 Analytical models 161
5.4.2 Physical Models 162
5.4.3 Textual Models 163
5.4.4 Statistical Models 163
5.4.5 Visual Models 163
5.5 Prototypes 164

6. Model-Based Systems Engineering 165

6.1 Models in Engineering 165
6.2 What Makes Models so Important for the Systems Engineer's Job? 166
6.3 What Does a Model-Based Approach Need to Provide in Order to Provide Benefit? 170

7. Reviews

7.1 Why Reviews? 174

7.2 Milestone Reviews 176
7.3 Technical Reviews 176
7.4 Definitions and Terms 177
7.5 Usage of Checklists 178
7.6 Document Preparation and Proofreading Process 178
7.7 Requirements Reviews 180
7.8 Plans Review 181
7.9 System Requirements Review (SyRR) 183
7.10 Preliminary Design Review (PDR) 185
7.11 Critical Design Review (CDR) 190
7.12 System Integration Review (SIR) 195
7.13 Test Readiness Review (TRR) 197
7.14 Integration and Verification Review (IVR) 199
7.15 System Acceptance Review 201
7.16 Technical Peer Reviews 203
7.17 Functional and Physical Configuration Audits (FCA/PCA) 203
7.18 Production Readiness Review (PRR) 204
7.19 Operational Readiness Review (ORR) 204
7.20 Flight Readiness Review (FRR) 204

8. Requirements Management 206

8.1 What Is a Requirement? 206
8.2 Requirements Engineering 208
8.3 Requirements Management 208
8.4 Requirements Development 209
8.5 Classification of Requirements (FURPS) 209
8.6 Example Content of a Requirements Specification 211
8.7 Requirements Flow 213
8.7.1 Requirements Flow Down 215
8.7.2 Step by Step 217
8.8 Requirements Based Engineering 221
8.9 Validation & Verification 224
8.9.1 Validation and Verification in the Life Cycle 224
8.9.2 Validation of Requirements 225
8.9.3 Verification 226
8.9.4 Product Validation 227
8.9.5 Product Verification Strategy for Current Level 227
8.10 Traceability 230
8.11 Quality of Requirements 231
8.11.1 Characteristics of Individual Requirements 231
8.11.2 Characteristics of a Requirements Specification (a Set of Requirements) 233
8.12 Users, Customers and Stakeholders 235
8.13 Difference between Needs and Requirements 237
8.14 Technical Requirements on Different Abstraction Levels 243
8.15 Performance Requirements and Their Allocation 246

9. Architecture and Design

9.1 About the Difference between Architecture and Design 248
9.2 The Role of Standards and Patterns in the Architectural Blueprint 252
9.3 Architectural Design Decisions 254
9.4 Reference Architectures 255
9.5 Logical and Physical Architecture 260
9.5.1 Logical Architecture 260
9.5.2 The Physical Architecture 262
9.5.3 Examples 263

10. Structuring the System 267

10.1 Organizational 267
10.2 Re-use and COTS 273
10.3 Providing Structure in a Timely Manner 278
10.3.1 Functional Baselines 278
10.3.2 Functional Scope 278
10.3.3 Procedural Scope 280
10.3.4 Coordinating the Scope 281

1. Definitions

After working through this course you will be able to explain the essential characteristics of a system compared to a set of elements. You will be able to decompose the hierarchical system and to describe "systems engineering" and the system (product) life cycle.

1.1 What is a System?

INCOSE, Systems Engineering Handbook, v3.1, August 2007 defines a system as an integrated set of elements that accomplishes a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements.

SE Handbook
www.incose.org

Hitchins defines a system as a set of complementary, interacting parts with properties, capabilities and behaviors emerging both from the parts and from their interactions to synthesize a unified whole. Hitchins emphasizes that perhaps the key feature of the definition is the word „interactions."
He further explains what distinguishes a system from a pile of bits is simply that, in a system, the parts interact with each other to produce „properties, capabilities and behaviors." So, while a pile of bits has properties - mass, volume, etc. - none of them emerges because of interactions between the bits. He points out that this is important: Because of the interactions, a system is complex, systems may be able to do things, to exhibit behavior and to have system capabilities that none of their parts have.
(hitchins.net)

WCSE
www.hitchins.net

ISO/IEC 15288 defines a system as an integrated set of complementary, interacting elements organized to achieve one or more stated purposes.

Definition 1 – System: A system is an integrated set of complementary and interacting elements that accomplishes one or more stated purposes. These elements may include products (hardware, software, and firmware), processes, people, information, techniques, facilities, services, and other support elements. Because of the interactions between its elements, a system is able to do things, to exhibit behavior, to have capabilities that none of its elements have.

What is a system? Describe in your own words and find examples.

1.2 What is Systems Engineering (SE)?

The definition of INCOSE, Systems Engineering Handbook, v3.2, August 2007 is most complete as it addresses the most important issues. Therefore, we could follow their definition:

Definition 2 – Systems engineering: Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.

What is important?

Systems engineering is an interdisciplinary approach, which means that several specialist engineering disciplines and domain experts are involved in the system realization. Systems engineers establish an interaction and communication network with customers, users and stakeholders and other organizational units of the enterprise.

interdisciplinary approach

It is commonly agreed that the overall goal of engineering is the realization of successful systems. Therefore, it is necessary to consider both the business needs of one's own company and its interest in making profit, but also the needs of the customers. The customer's satisfaction is supported by providing a quality product that meets the user needs, while considering and balancing the interests of all stakeholders.
Of equal importance is the understanding that a system is successful not only if it is realized on time and on cost, but also if it meets customer expectations during operation, and finally allows retirement according to applicable regulations. Therefore, the scope of systems engineering is not limited to the implementation phase but also considers the complete life cycle of a system.

To accomplish these goals, systems engineering focuses on understanding the user domain, defining user needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis, system validation and system verification while considering the complete problem (see system life cycle).

Systems engineering is a discipline that concentrates on the whole (system) as distinct from its parts. It involves looking at a problem entirely, taking into account all the facets and variables and relating the social to the technical aspect.

To achieve good results, systems engineers involve themselves in nearly every aspect of a project. Systems engineers pay close attention to interfaces where two or more systems or system elements work together and they apply technical processes to reduce the risk of project failures to an acceptable level.

All of these issues are discussed in more detail in the following chapters.

What is systems engineering? Describe in your own words.

4. Processes

By working through the following issue you will be able to understands the process as an enabler for repetitive project success and a container for collecting lessons learned in order to avoid procedural failures. As well you will be able to use the provided description of a generic SE process to guide your SE activities and to apply the different aspects of development life cycles according to the project needs.

4.1 Need for Processes

need for processes
www. gurteen.com/gurteen/gurteen.nsf/id/ksculture

What are processes? Processes can be understood as a kind of knowledge management; a way of capturing procedural know-how in a written document, enabling others to share in the experience of the process author or the organization that has established the process.
The process of baking a cake is examined here for the sake of example.

A chemical analysis provides the cake's molecular constituents. It is not possible to bake the cake based solely on this data; it cannot even be determined if it is a cake or something else. Such data is not very useful for most purposes.
A list of ingredients is information that is more useful. An experienced baker could probably make the cake. Therefore, the data has been given a meaning. But what about less experienced bakers? Would they be able to make the cake? The result may be some sort of a lottery.
The recipe would be knowledge - it tells you how to make the cake. It is generally a written recipe, a process capturing explicit knowledge. However, even with the recipe, an inexperienced baker might not make a very good cake.
After several more-or-less successful attempts, the baker will certainly make a much better cake. He will have gathered some experience, allowing him to reproduce a better result with each attempt. This relevant knowledge acquired by applying the recipe is not always easily written down. It is tacit (hidden) knowledge.
Ultimately, wisdom is about knowing which cake to make!
Clearly, some sort of recipe is beneficial, especially for inexperienced engineers. These recipes are the processes, telling them how to perform and structure their work and describing the activities to be performed in detail. It is a challenge for an organization to add the experience – the lessons learned - to the process, making it more valuable.
For an organization, it is essential that engineers and workers accept and apply the processes in order to assure a repeatable quality for their products. It is in the interest of the organization to become as independent as possible of the experience and knowledge of the individual staff. The problems an organization faces in dealing with highly complex products and production methods are massive if the most experienced engineers and workers leave the company, without their procedural knowledge having been documented.
It is also a matter of efficiency. Processes integrate lessons learned, i.e. they help to improve development activities and avoid future failures, which are often expensive to fix. Good process documentation answers the questions people have often raised in the past. It avoids situations in which the individual staff members need to search for and ask people with the necessary experience when they have a question.

However, it is important to note that while processes may be good for reproducing a high-quality work product, it is the responsibility of the engineer to know how to apply the process in an acceptable manner and adapted to the specific problem when faced with different scenarios and conditions. Such process tailoring should be performed and documented before the respective development phase is started.
It is therefore necessary and essential for the engineer to know what purpose a specific process is intended for. Otherwise, the engineer will not be able to provide the correct contents in a usable manner.

Describe in your own words the purpose of processes. Do you know processes from your own experience? Are your processes updated regularly with lessons learned? Do you think that those processes are helpful for you in your daily work?

4.2 Standards

standards

A standard is a document that provides rules, guidelines or characteristics for activities of specific (engineering) domains or their results. Standards are either internal within an organization or publicly approved and controlled by an officially recognized body (e.g. DIN, ISO, ANSI, IEEE). Standards are intended for common and repeated use and deal with actual or potential problems in a given context or scope. Standards express either recommendations as guidance (should) or requirements as a prescription (shall). They are aimed at achieving an optimum degree of satisfaction, establishing the same basic minimum requirements for everyone.
Since the use of a standard is voluntary, it is quite different from the mandatory regulations that are drawn up by public authorities. However, public authorities often require the application of standards e.g. in order to gain certification for a specific purpose.
Standards are documents established by consensus of different parties and stakeholders and are therefore the result of a compromise. They usually do not provide the absolute optimum, which could be achieved when considering the domain and environment in which they are applied. Therefore, if the intention is to apply a procedural standard in one's own organization, there is enough room for optimization and integration of lessons learned and experience gathered in one's own domain. Standards provide a sound starting point.

7. Reviews

Within this chapter you will learn how to apply reviews within the project life cycle and to adapt the technical contents to your needs.

7.1 Why Reviews?

At certain points in the project development, there are some synchronization points and milestones: the technical reviews. Usually, they are scheduled after a certain phase of work has been completed and a new phase starts. A primary outcome of each technical review is the technical assessment on the readiness to proceed to the next phase of work. There are different reasons why a review is conducted:

To align the team

to align the team

Usually a team is working on a project with all the different stakeholders, but only one or a few engineers are working on a certain item, where other stakeholders are not involved. A review is used to present and communicate the working result and to ask for agreement and discuss the selected design solutions.

For example

The systems engineer is working on the system architecture of an avionics computer. The people responsible for testing the equipment need to know which specific means of testing are available to plan their tests and which test interfaces they can use to connect the test equipment. The people responsible for the logistic support want to know the reliability of the equipment, which maintenance features are provided and which interfaces are used for connecting the maintenance equipment. The product manager wants to know the life cycle and the production costs. The production people are interested in knowing if the equipment can be produced and which tools are necessary. There are other people with other interests. Finally, there is the project manager with a limited budget.
Of course, the systems engineer may have discussed with all the people in advance and run trade-off studies to find the best solutions for all the requirements. However, the full overall picture should be presented to all the stakeholders in a joint meeting. Such a meeting allows the stakeholders to align their expectations on the system to be developed and to come to an agreement on the systems engineer's decisions.

To provide confidence

to provide confidence

The review also provides confidence that the project is moving in the right direction and that all requirements as well as business needs are met and nothing has been forgotten.

To avoid additional cost

to avoid additional cost

The reviews are scheduled when one project phase has been completed and a new phase starts. Usually reviews are scheduled before considerable costs are generated, when commencing work. Project management has the viable interest of starting these activities on a mature basis, to avoid additional costs due to redesigns and repetition of activities or the risk that efforts already expended are lost.

For example
The system architecture has been generated and different components have been identified to be purchased. Before starting contractual negotiations and supplier selection or even starting the development activities on supplier side, it is a good idea to have a clear and mature specification available. Once the contracts have been signed, it is rather difficult and costly to change the contracted requirements or even to select a different supplier.

Another example:
a qualification of the equipment needs to be performed. All the test facilities (e.g. temperature chamber, vibration lab) are booked and the technicians are waiting for the units to be tested and the necessary test equipment. Now it is established that some cables that are necessary for connecting the test equipment are missing, or some software for automating the tests is not available in time. The qualification campaign must be shifted. Nevertheless, the costs for the test facilities need to be paid by the project. A test readiness review aims to avoid such situations.

To identify failures

to identify failures

Other items are prepared by less experienced engineers who do not have the background of a senior engineer. They probably do not know about failures and mistakes other engineers have made in the past and therefore they run the risk of making the same mistakes again. Reviews are therefore a method of knowledge insertion. Experienced engineers examine the work performed by the less experienced engineers and provide guidance and support.

To seek acceptance from the customer

to seek acceptance from the customer

Usually reviews with customer participation are also payment milestones. In this case, a successful review will trigger the next partial payment. It is important to provide transparency for the customer's requirements, the implemented technical solutions and the rationale for the design decisions.

> What is the purpose of reviews? What is most important for you as a systems engineer? Describe in your own words.

9. Architecture and Design

After working through this issue you will be able to differentiate between design and architecture and to apply standards and architectural patterns. Furthermore you will be able to structure the system of interest considering architectural as well as business constraints. At least you will be able to generate a system architecture supporting re-use while considering appropriate interface standards and to use the system architecture to manage information and to make decisions at the right system hierarchy level.

9.1 About the Difference between Architecture and Design

Developing the system architecture is one of the most important responsibilities of the systems engineer. It is a creative process, and there is not just one solution to satisfy user requirements.

Many process descriptions talk about architecture and design. There are documents to be delivered for both: an architecture document describing the architectural concepts and a design document describing the details of the design. To fill in the correct contents, it is important to know the difference between architecture and design.
There are different approaches. Some of them say that it is a matter of detail, e.g. talking about a preliminary design and later on a detailed design. See added example from the NASA homepage.

NASA
www.hq.nasa.gov

Develop the Preliminary Design
The role of the complex electronics engineer during the preliminary design phase (also called the conceptual or architectural phase) is to come up with a high-level design that meets the requirements. At this level, the design engineer has to keep a systems point of view, because the complex electronics interfaces with other elements of the system. It's also common that the functions to be performed are still fluid, and the allocation to software, complex electronics, and other electronic elements may change.
The architectural (high-level) design converts the requirements into functions to be performed and the interfaces between those functional blocks. There is often a certain amount of trade-off during this phase, as various requirements or goals come into conflict. The design engineer needs to be aware when design decisions for the complex electronics affect other elements of the system, especially by imposing constraints on those elements.
The initial high-level design may be made using functional block diagrams, design and architecture descriptions, or sketches. As the design is solidified, it is usually implemented in a hardware description language (HDL), such as VHDL, Verilog, or SystemC. At this stage of development, the complex electronics is defined as black boxes, with specified behavior and interfaces. The specific details are added later in the design life cycle.
During the preliminary design phase, the architecture of the chip is defined, verified and documented. The architecture is defined down to the level of basic blocks that implement all intended functions. In addition, the interfaces and interactions between the blocks are specified. Important decisions for the implementation of the chip are made at this phase, including the selection of COTS, third-party, or re-used IP (Intellectual Property) modules or cores.

development of preliminary design

Does this statement make clear what a preliminary design is and what an architecture is? There is one sentence that indicates the right direction: "There is often a certain amount of trade-off during this phase, as various requirements or goals come into conflict." But what are the conflicting requirements and what are the criteria to drive the decision in a trade-off?

Let's first explain the difference between architecture and design with the help of a simple example, a bookshelf (see Fig. 93 on page 249).

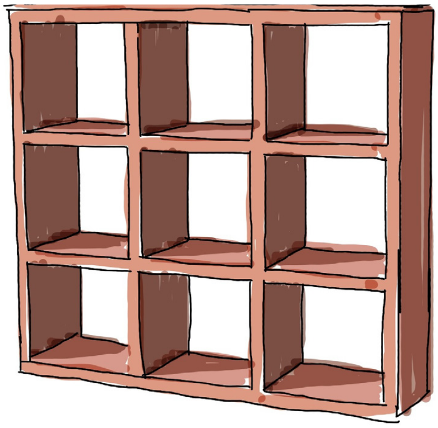

Fig. 93: bookshelf

The bookshelf is able to hold books as well as other decorative items. It is divided into squares. In designing such a rack, there are basically some functional requirements to be considered, which are: height, width, depth and number of shelves, as well as requirements for the look and feel, e.g. it has a dark wooden appearance with solid boards.

So far, these are all design requirements. None of it is architecture.

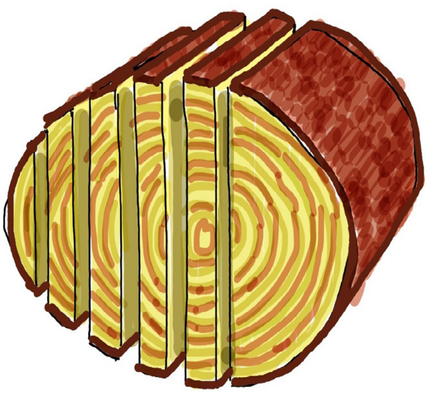

Fig. 94: trunk of a tree, sawn into boards

But now, the rack needs to be produced. Further questions will come up. For example, could the rack be fully built from a single large trunk of a tree? Certainly, that is a waste of resources, inefficient and would be expensive. It would be more elegant to saw the tree trunk into boards (see Fig. 94 on page 249), which can be used as sidewalls as well

Ansprechpartner

Mod:Master
Sensorsystemtechnik

Dr. Gabriele Gröger
Albert-Einstein-Allee 45
89081 Ulm

Tel 0049 731 – 5 03 24 00
Fax 0049 731 – 5 03 24 09

gabriele.groeger@uni-ulm.de
www.uni-ulm.de/saps

Postanschrift

Universität Ulm
School of Advanced Professional Studies
Albert-Einstein-Allee 45
89081 Ulm

Das Studienangebot „Sensorsystemtechnik" wurde entwickelt im Projekt Mod:Master, das aus Mitteln des Bundesministeriums für Bildung und Forschung gefördert und aus dem Europäischen Sozialfonds der Europäischen Union kofinanziert wird (Förderkennzeichen: 16OH11027, Projektnummer WOH11012). Dabei handelt es sich um ein Vorhaben im Programm „Aufstieg durch Bildung: offene Hochschulen".

AUFSTIEG DURCH BILDUNG OFFENE HOCHSCHULEN

EUROPÄISCHE UNION

ESF Europäischer Sozialfonds für Deutschland

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung