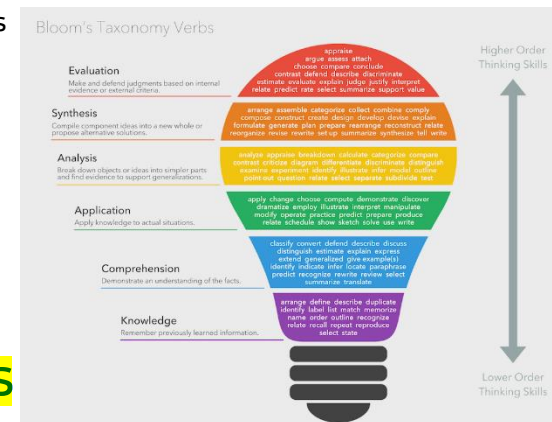


## Master Software Technology **Software Project Management 2 – [02] Building Blocks: SWEBOK & PMBOK**

## Learning Goals

Bloom's Taxonomy Verbs  
by Fractus Learning,  
License: CC-BY-SA 4.0

- ✓ Understand the basic idea behind the guidebooks
- ✓ Understand use of the SWEBOK and PMBOK
- ✓ Apply PMBOK Scope Management by creating Work Breakdown Structures



## Agenda

### SWEBOK

- **Basic definition**: software engineering
- Guide to the Software Engineering Body of Knowledge
  - SWEBOK's objectives
- Development process of SWEBOK
- General organization of SWEBOK
- Selected knowledge areas
  - Software Construction (chapter 4)
  - Software Maintenance (chapter 6)

### PMBOK

- Structure
- Project Lifecycle and Organization
- PMBOK Process Groups
- Selected Knowledge Areas
  - The Scope Management Knowledge Area
    - Work Breakdown Structures

## Definition Software Engineering

1. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
2. The study of approaches as in (1).

IEEE Standard Glossary of Software Engineering Terminology

## **SWEBOK** project objectives

Target of  
SWE/PMBOK

1. To promote a consistent view of software engineering worldwide.
2. To clarify the place – and set the boundary – of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics
3. To characterize the contents of the software engineering discipline
4. To provide a topical access to the Software Engineering Body of Knowledge [by structuring it into knowledge areas]
5. To provide a foundation for curriculum development and for individual certification and licensing material

Specialized Practices used only for certain types of software	Generally Accepted Established traditional practices recommended by many organizations
	Advanced and Research Innovative practices tested and used only by some organizations and concepts still being developed and tested in research organizations

Whole document publicly available at  
<http://www.computer.org/portal/web/swebok>

## SWEBOK development supporters



CANADIAN COUNCIL OF PROFESSIONAL ENGINEERS  
CONSEIL CANADIEN DES INGÉNIEURS



National Research  
Council Canada

Conseil national  
de recherches Canada

Rational®  
the e-development company™

Raytheon

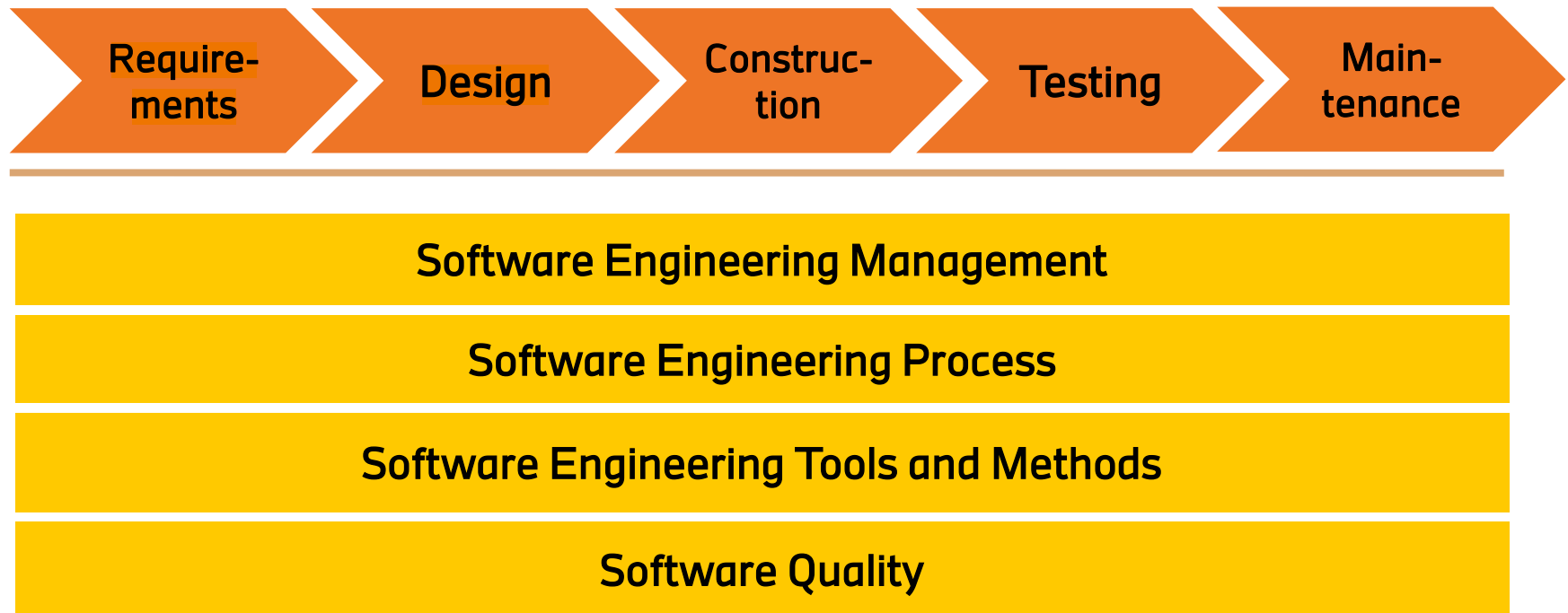


UQÀM



Université du Québec  
École  
de technologie  
supérieure

## SWEBOK Processes



Primary Processes



Supporting Processes

## **SWEBOK V3 (2014) Knowledge Areas and related disciplines**

### Knowledge Areas:

1. SW requirements
2. SW design
3. SW construction
4. SW testing
5. SW maintenance
6. SW configuration mgmt
7. SW engineering management
8. SW engineering process
9. SW engineering tools and methods
10. SW quality
11. SW Engineering Professional Practice
12. SW Engineering Economics

13. Computing Foundations

14. Mathematical Foundations

15. Engineering Foundations

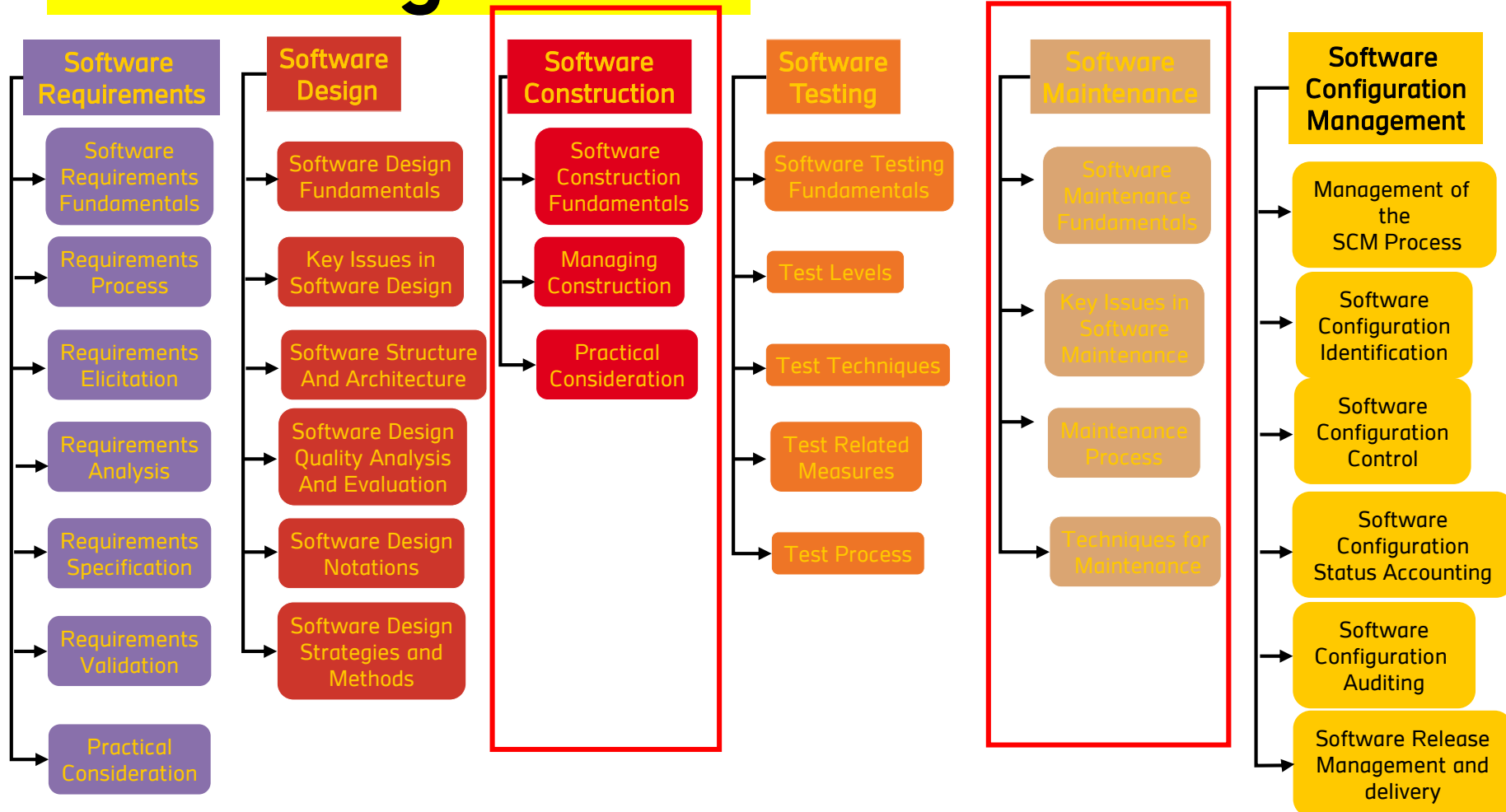
### related disciplines:

- Computer engineering
- Computer science
- Management
- Mathematics
- Project management
- Quality management
- Systems engineering

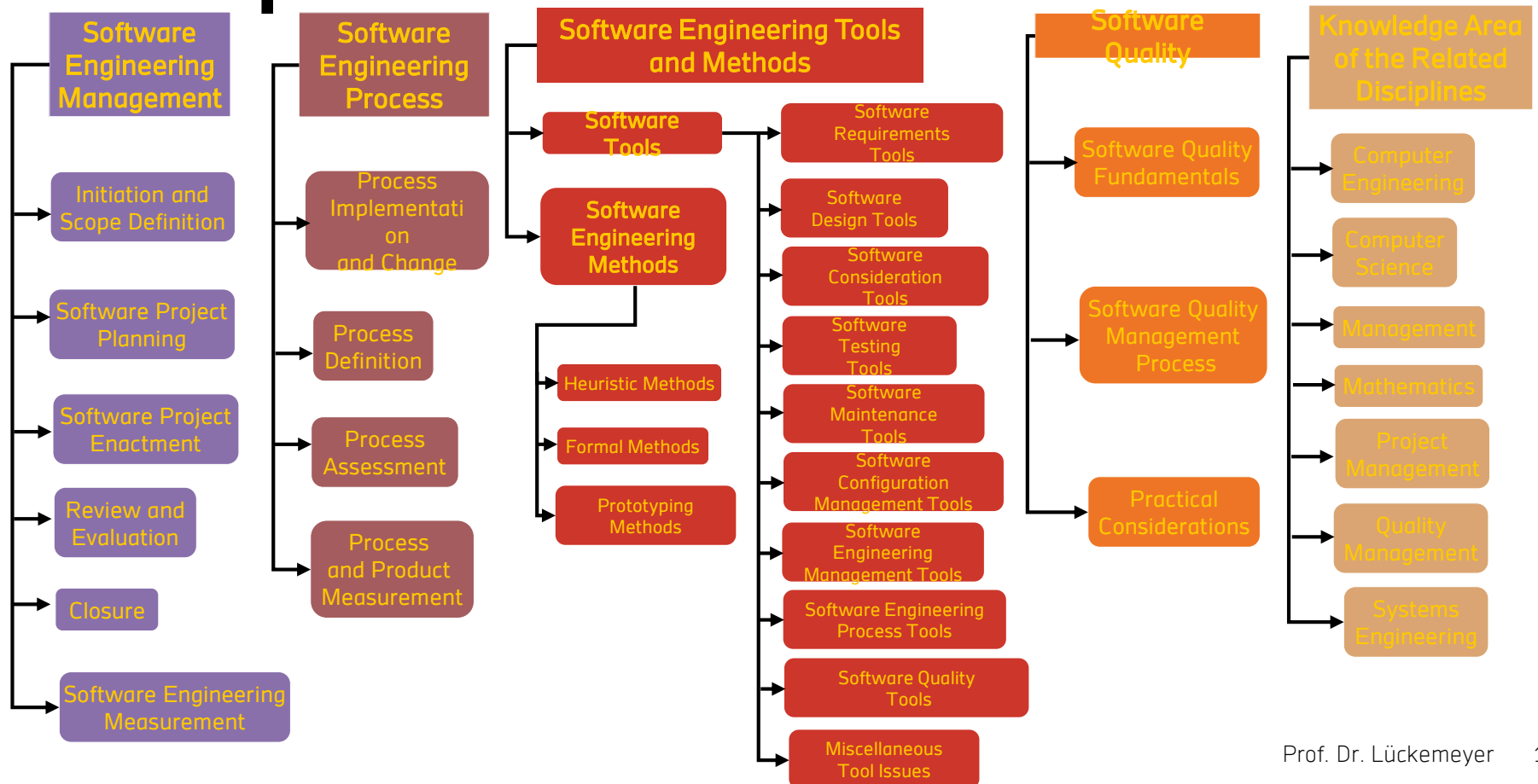
**Where is the project management?**



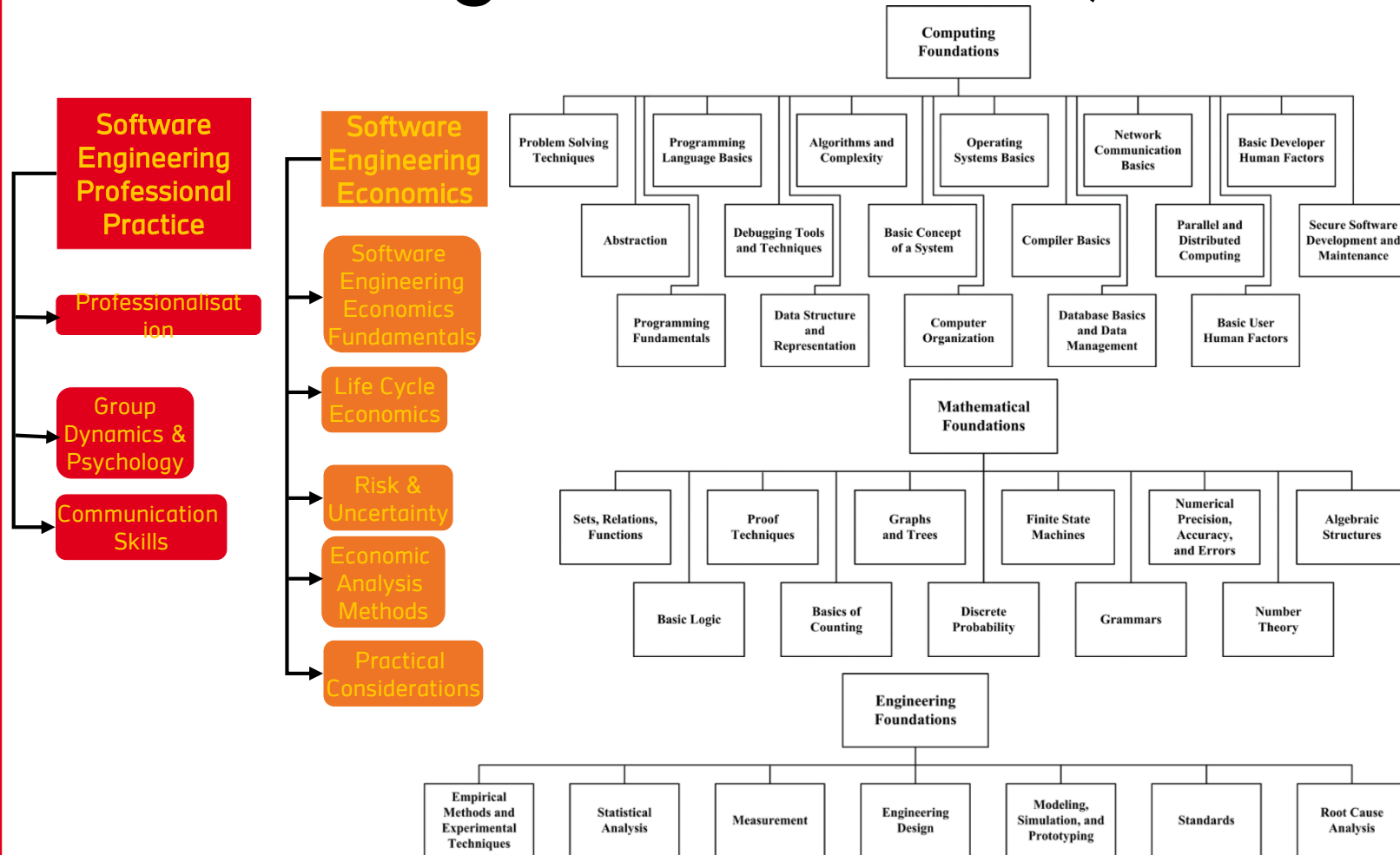
## Knowledge Areas 1-6



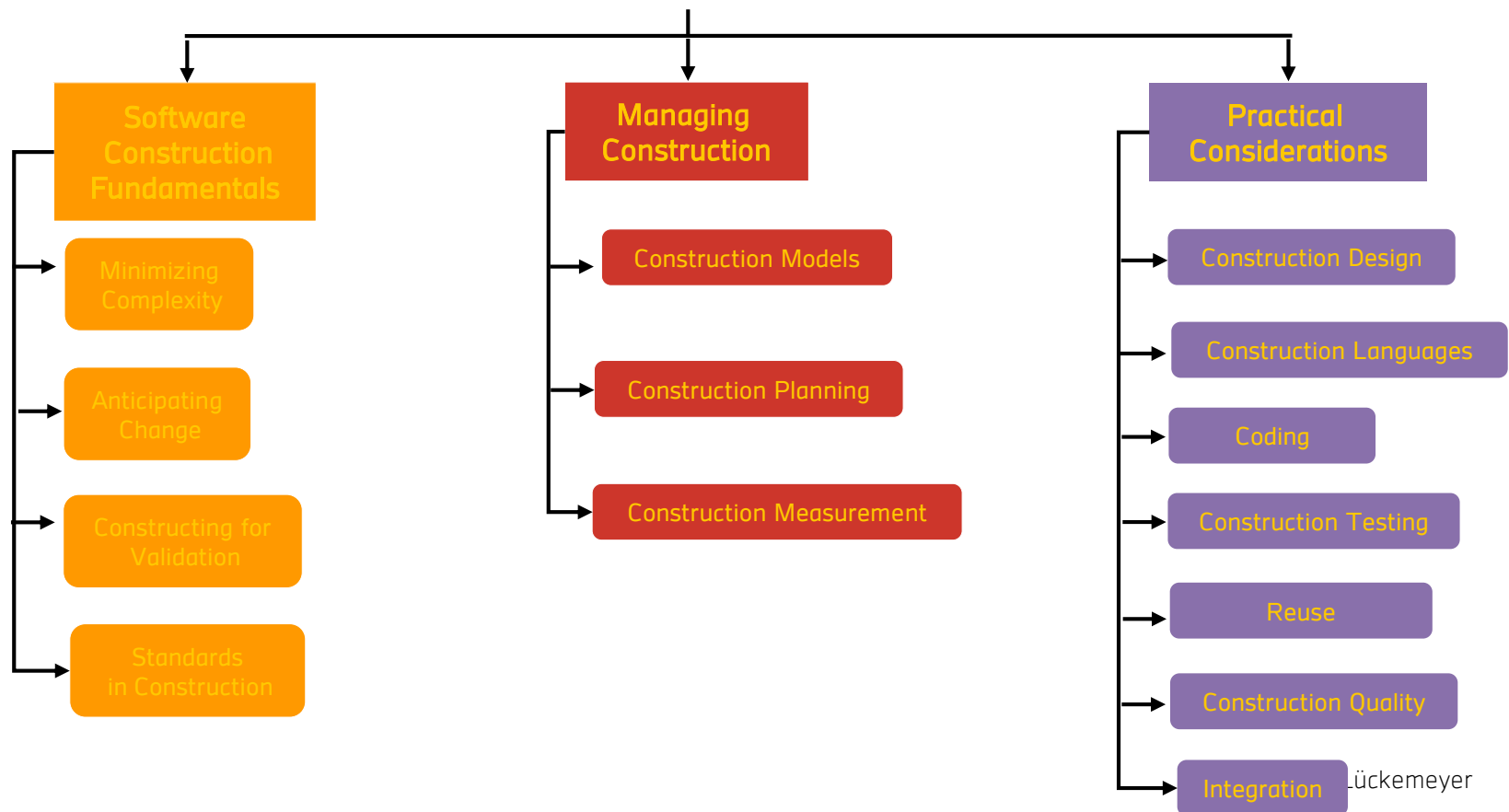
## Knowledge Areas 7-10 & related disciplines



## Knowledge areas 11-15, new to V3



## Knowledge Area 4: Software Construction



## Software Construction: Fundamentals

### Minimizing complexity

- Emphasizing the creation of simple and readable code
- Making use of standards

### Anticipating change

- Anticipation of change drives many aspects of software construction

### Constructing for verification

- Building software in the way that faults can be ferreted out

### Standards in construction

- Programming languages
- Communication methods
- Platforms
- Tools

## Software Construction: Management

Construction models

- Waterfall model, spiral model, V-model, RUP, XP

Construction planning

- Construction methods
- Defining the order in which components are created and integrated

Construction measurement

- Code complexity
- Code inspection statistics
- Fault-fix and fault-find rates

## Software Construction: **practical** **issues**

Construction design

- Real-world problem addressed by the software

Construction languages

- Configuration languages
- Toolkit languages
- Programming languages

Coding

- Source code organization: Creating understandable source code
- Use of classes, variables and other similar entities
- Use of control structures
- Handling error conditions
- Prevention of code-level security breaches
- Resource usage
- Code documentation
- Code tuning

## Software Construction: practical issues (2)

### Construction testing

- Unit testing
- Integration testing

### Reuse

- The selection of the reusable units, databases, test procedures or test data
- The evaluation of code of test reusability
- The reporting of reuse information on new code, test procedures or test data

### Construction quality

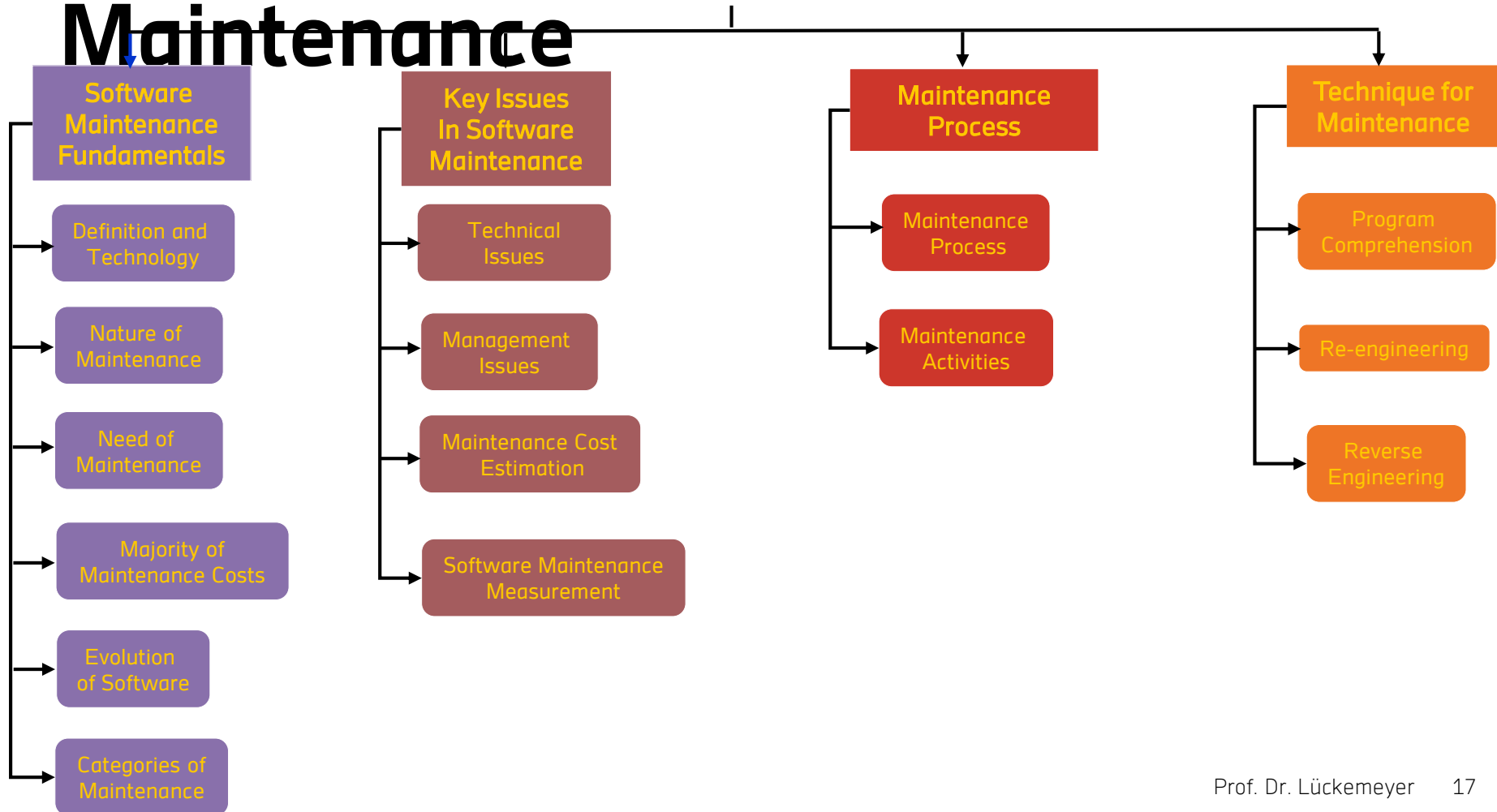
- Unit testing and integration testing
- Test-first development
- Debugging and Code stepping
- Use of assertions
- Technical reviews
- Static analysis

### Integration

- Integration of separately constructed routines, classes, components and subsystems



## Knowledge Area 6: Software Maintenance



## Software Maintenance: Fundamentals

### Definitions and terminology

- IEEE 1219, IEEE/EIA 12207, ISO/IEC 14764

Category	Correction	Enhancement
Proactive	Preventive	Perfective
Reactive	Corrective	Adaptive

### Nature of maintenance

- Maintenance has a broader scope, with more to track and control

### Need for maintenance

- Correct faults, improve the design, interface with other system
- Adapt program so that different hardware or system features can be used

### Majority of maintenance costs

- Application type
- Software maintenance staff availability, software life span
- Hardware characteristics
- Quality of software design, construction, documentation and testing

## Software Maintenance: **Key Issues**

### Technical Issues

- Limited understanding
- Testing
- Impact analysis

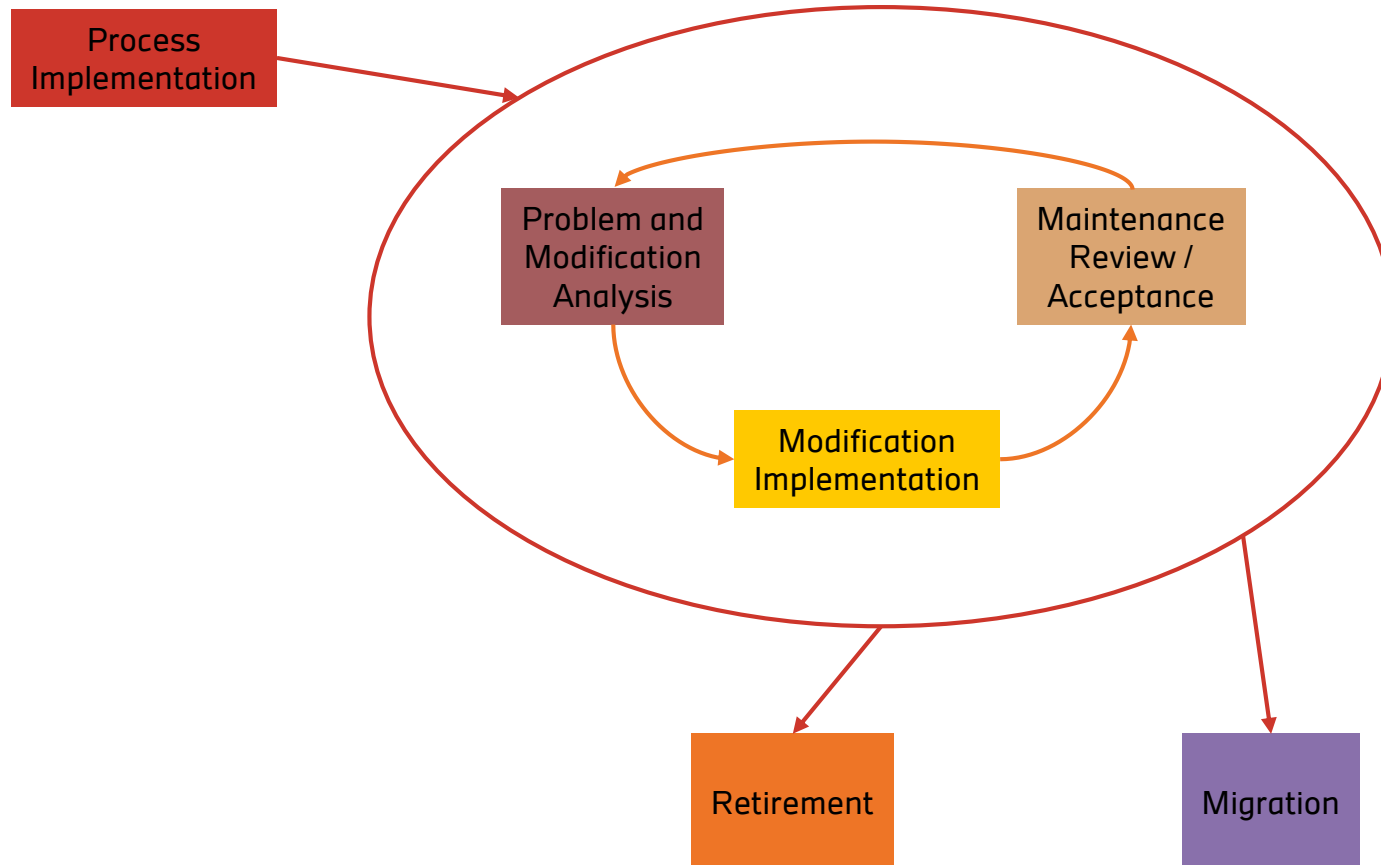
### Management Issues

- Alignment with organizational objectives
- Process
- Outsourcing

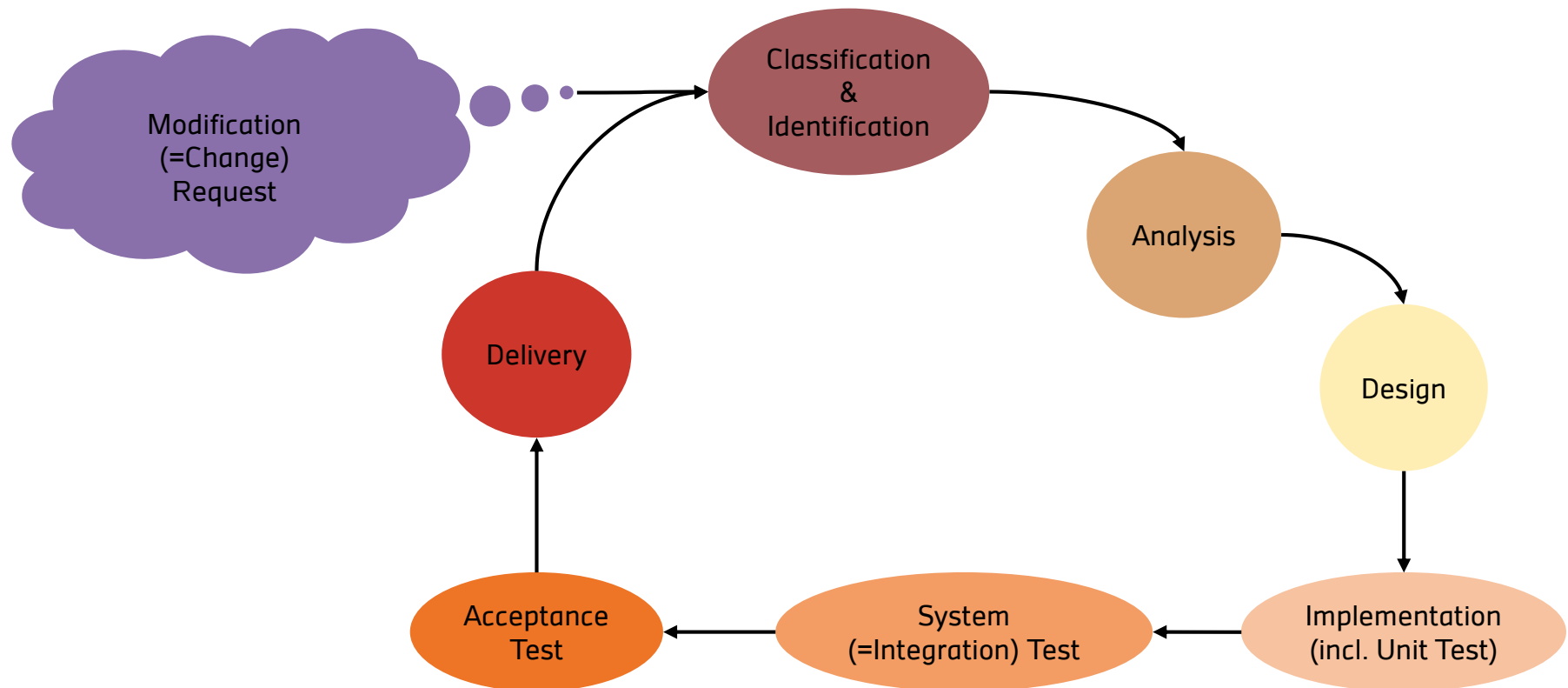
### Maintenance Cost Estimation

- Cost estimation
- Parametric models
- Experience

## Software Maintenance: Process



## Software Maintenance: Activities



## PMBOK V5 (2013): Areas & Structure

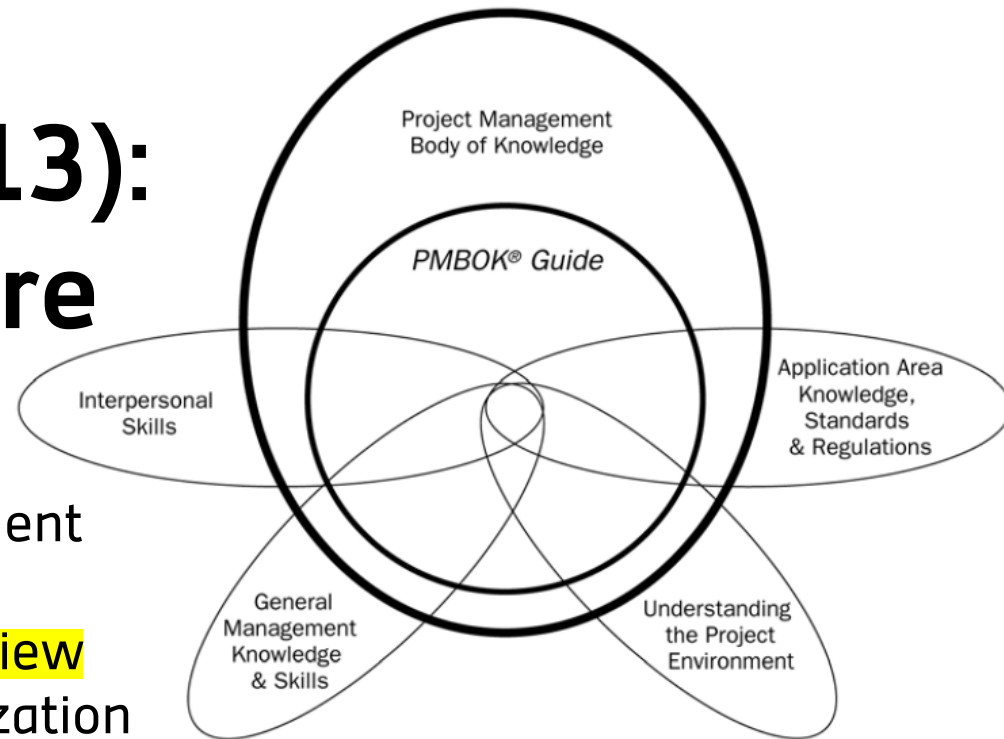
### Section I: The Project Management Framework

- Key terms and general overview
- Project Lifecycle and Organization

### Section II: The Standard for Project Management of a Project

- Project Management Processes for a Project

### Section III: The Project Management Knowledge Areas



## PMBOK: Major Project Documents

### Project Charter

- Formally authorizes the project
- Contains project **objectives**
- Created in, derived from project portfolio management

### Project Scope Statement

- States what work should be accomplished and what deliverables need to be produced (the project content)

### Project Management Plan

- States how the work will be performed

### SMART goals

- **Specific**
  - Objectives must be precisely and defined unmistakably
- **Measurable**
  - Both quality and quantity
- **Attainable** (action-oriented)
  - Objectives must be challenging and pretentious
- **Realistic**
  - Objectives must as well be achievable
- **Time and resource constrained**
  - Starting dates and ending dates; milestones

## PMBOK v6 (2017) Updates

The Sixth Edition incorporates terminology and practices that reflect the larger, more inclusive spectrum of project management practices. This is especially present in Part 1 - Section 1, where the project and development life cycles are discussed, as well as the various Predictive, Adaptive, Agile, Iterative and Incremental, and Hybrid project approaches, which are referred to throughout the Knowledge Areas.

Part 1 - Section 3 discusses the Role of the Project Manager, including topics on how project managers operate in various organizational environments and the skills and competencies that they need to be effective.

Part 1 – Sections 4 – 13 provide the detail for each knowledge area and processes.

Part 2 becomes The Standard for Project Management including an Introduction and sections for each of the five Process Groups.

Part 3 includes Appendices, Glossary and the Index. Of special note are:

- Appendix X3 – Agile, Iterative, Adaptive, and Hybrid Project Environments
- Appendix X4 – Summary of Key Concepts for Knowledge Areas
- Appendix X5 – Summary of Tailoring Considerations for Knowledge Areas
- Appendix X6 – Tools and Techniques



## PMBOK: Project Organisation

Huge projects divided further into sub projects



Problem:

Combining projectized organisation with the other forms of structural organisation

- Process
- Matrix
- Functional

## PMBOK: Project Lifecycle

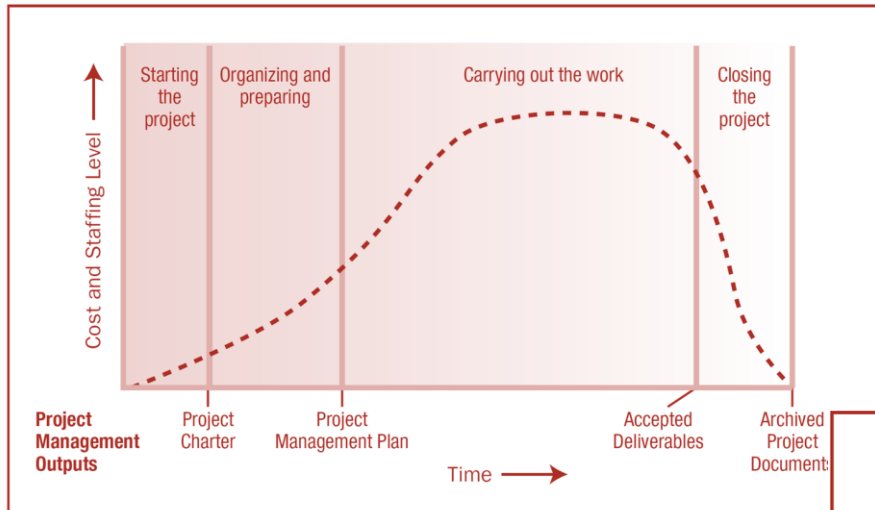


Figure 2-1. Typical Cost and Staffing Levels Across the Project Life Cycle

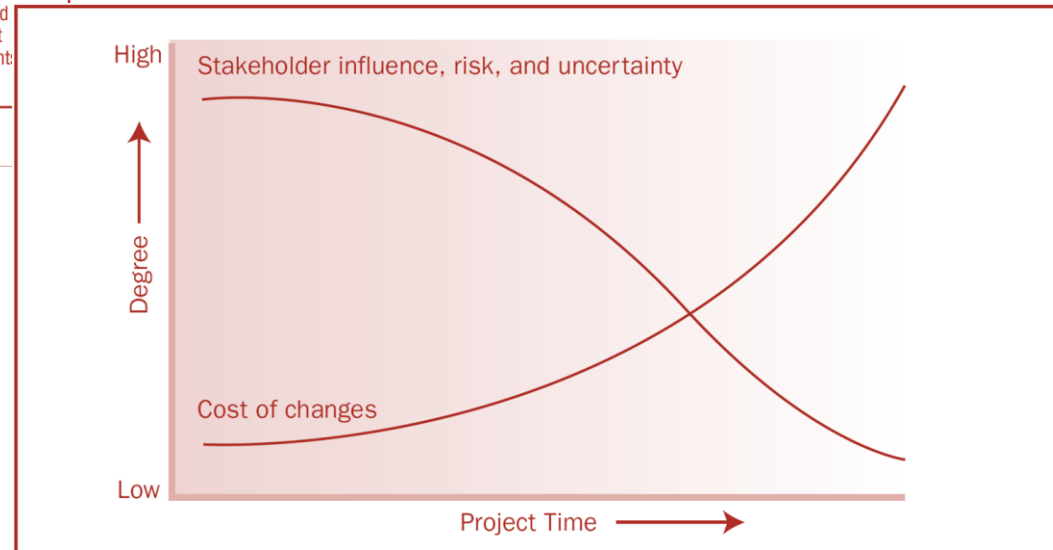


Figure 2-2. Impact of Variable Based on Project Time

## PMBOK: Process Groups

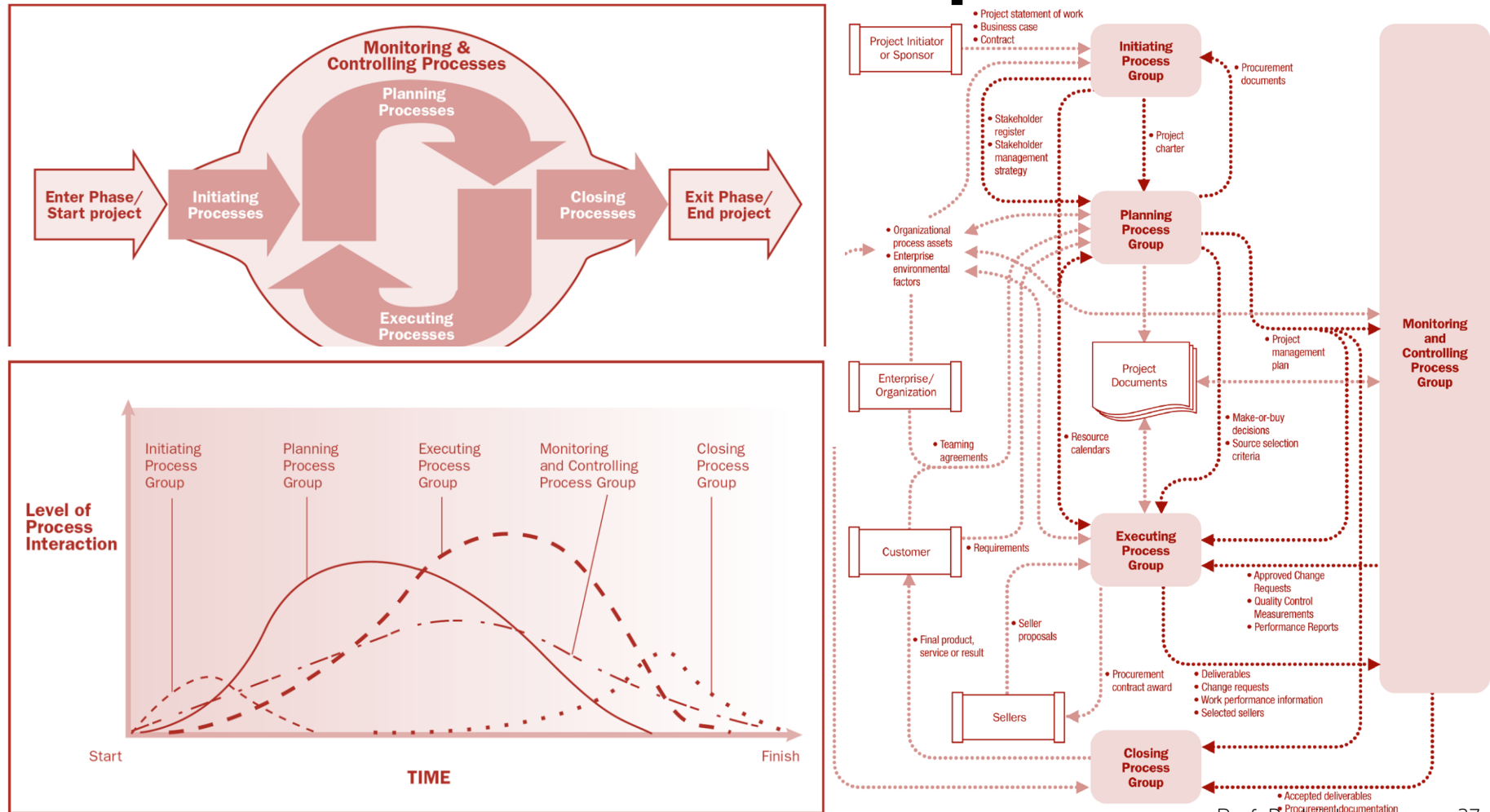


Figure 3-2. Process Groups Interact in a Phase or Project

## PMBOK v6 Processes

Knowledge Areas (49 processes)	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring and Controlling Process Group	Closing Process Group
Project Integration Management	.1 Develop Project Charter	.2 Develop Project Management Plan	.3 Direct and Manage Project Work  .4 <b>Manage Project Knowledge</b>	.5 Monitor and Control Project Work  .6 Perform Integrated Change Control	.7 Close Project or Phase
Project Scope Management		.1 Plan Scope Management  .2 Collect Requirements  .3 Define Scope  .4 Create WBS		.5 Validate Scope  .6 Control Scope	
Project <b>Schedule</b> Management		.1 Plan <b>Schedule</b> Management  .2 Define Activities  .3 Sequence Activities  .4 Estimate Activity Durations  .5 Develop Schedule		.6 Control Schedule	
Project Cost Management		.1 Plan Cost Management  .2 Estimate Costs  .3 Determine Budget		.4 Control Costs	
Project Quality Management		.1 Plan Quality Management	.2 <b>Manage Quality</b>	.3 Control Quality	
Project Resource Management		.1 Plan Resource Management  .2 <b>Estimate Activity Resources</b>	.3 Acquire <b>Resources</b>  .4 Develop <b>Team</b>  .5 <b>Manage Team</b>	.6 <b>Control Resources</b>	
Project Communications Management		.1 Plan Communications Management	.2 <b>Manage</b> Communications	.3 <b>Monitor</b> Communications	
Project Risk Management		.1 Plan Risk Management  .2 Identify Risks  .3 Perform Qualitative Risk Analysis  .4 Perform Quantitative Risk Analysis  .5 Plan Risk Responses	.6 <b>Implement Risk Responses</b>	.7 <b>Monitor Risks</b>	
Project Procurement Management		.1 Plan Procurement Management	.2 Conduct Procurements	.3 Control Procurements	
Project Stakeholder Management	.1 Identify Stakeholders	.2 Plan Stakeholder <b>Engagement</b>	.3 Manage Stakeholder Engagement	.4 <b>Monitor</b> Stakeholder Engagement	

## Knowledge Areas (chapters 4 – 13)

- 4) Project Integration Management
- 5) Project Scope Management
- 6) Project Time Management
- 7) Project Cost Management
- 8) Project Quality Management
- 9) Project Human Resource Management
- 10) Project Communications Management
- 11) Project Risk Management
- 12) Project Procurement Management
- 13) Project Stakeholder Management



„magic“ tri-/rectangle of  
project management

## PMBOK Knowledge Area Scope Management

What is **Scope Management**?

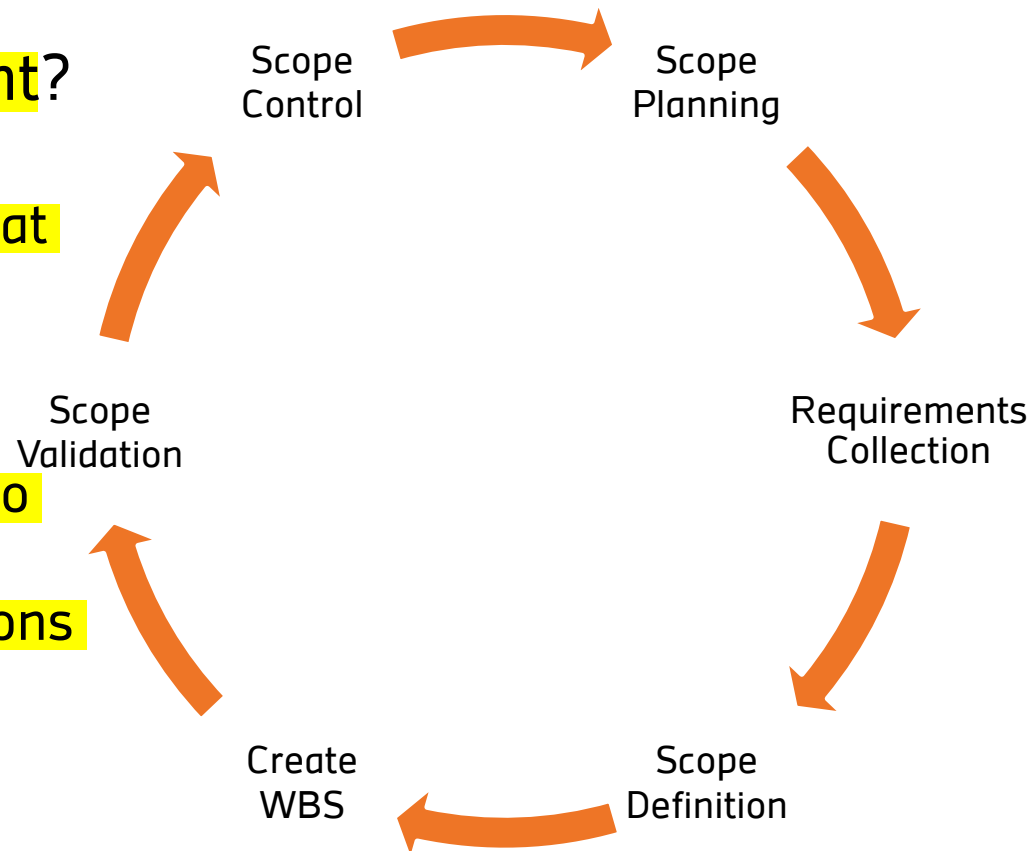
Product Scope

- the **features and functions that characterize a product or service**

Project Scope

- **the work that must be done to deliver a product with the specified features and functions**
- also: what is explicitly NOT result of the project

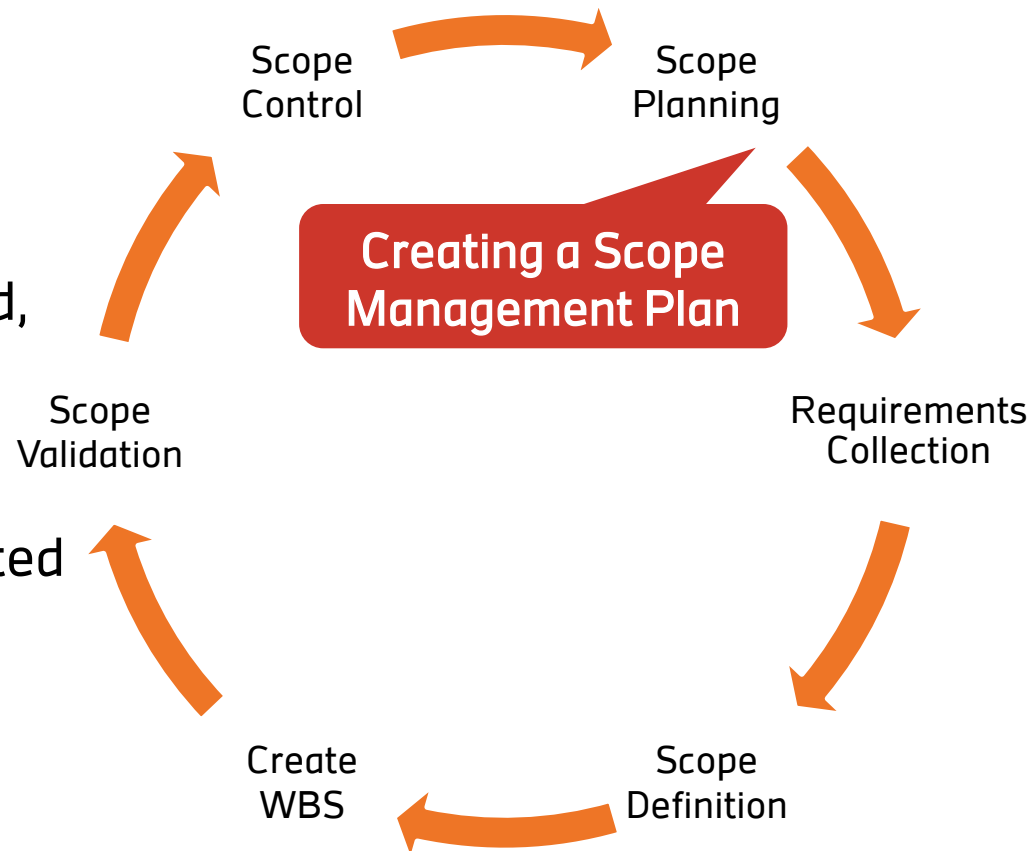
Processes



## PMBOK Scope Management: Scope Planning

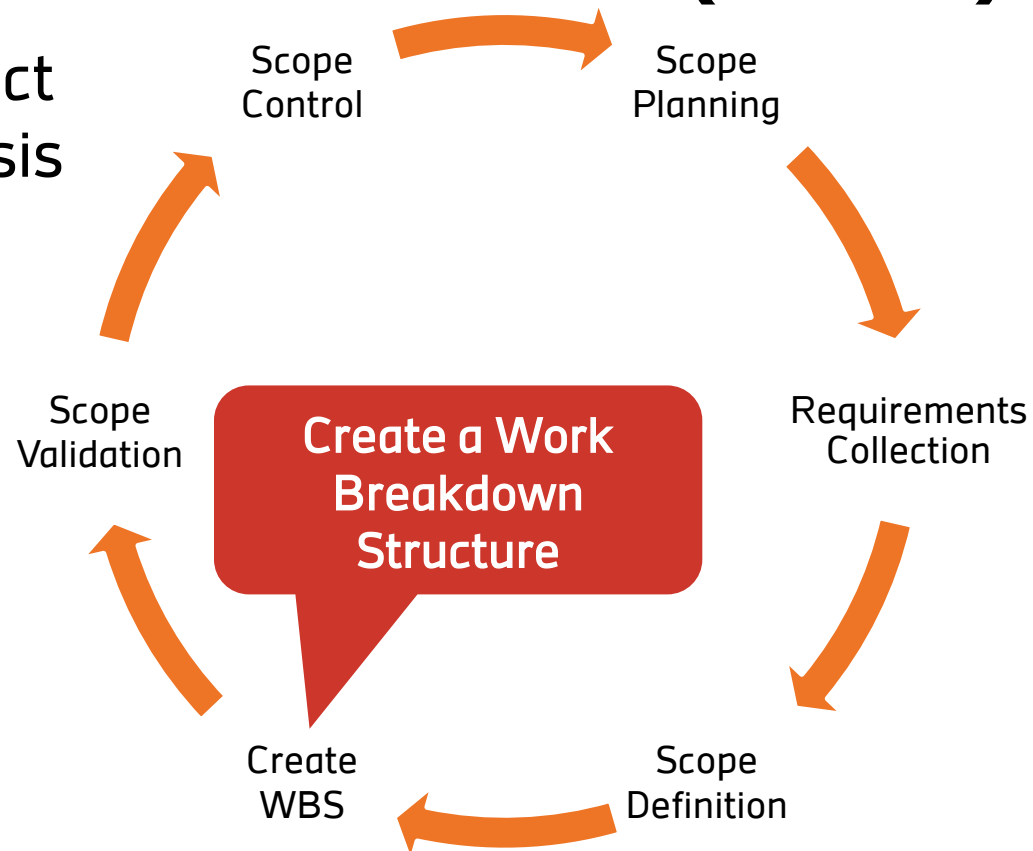
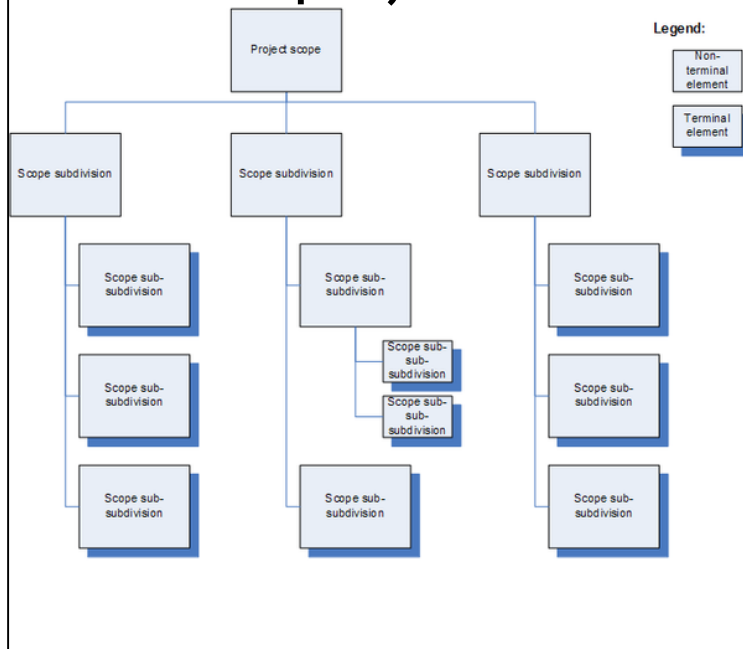
creating a Scope Management Plan that documents

- how the scope will be defined,
- validated, and
- controlled and
- how the **Work Breakdown Structure (WBS)** will be created and defined



## PMBOK Scope Management: Create a Work Breakdown Structure (WBS)

developing a detailed project scope statement as the basis for future project decisions





## Work Breakdown Structure

The WBS is the foundation of the project plan. The WBS is a hierarchical logical structure that represents all the work necessary to produce all the project deliverables. By doing so it organises and defines the total scope of the project. Work that is not in the WBS is outside the scope of the project. The WBS must be broken down to a sufficient level of detail so that one owner can be assigned responsibility for planning and managing each activity at the lowest level. By understanding the deliverables for assigned activities, by having clear completion criteria, each activity owner can successfully develop realistic and defensible time and budget estimates.

<http://www.projectsmart.co.uk/project-planning-the-first-line-of-defence-for-preventing-failed-projects.html>

## Work Breakdown Structure: Background

Developed for US Department of Defense (DoD)

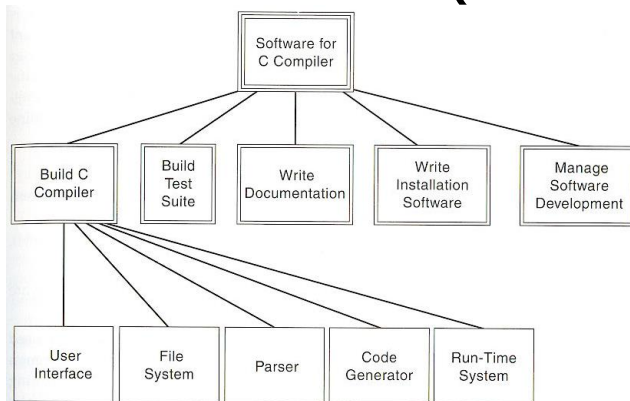
Current Definition: MIL-STD-881

- a) A product-oriented family tree composed of hardware, software, services, data, and facilities. The family tree results from systems engineering efforts during the acquisition of a defense materiel item.
- b) A WBS displays and defines the *product, or products*, to be developed and/or produced. It relates the elements of work to be accomplished to each other and to the end product. In other words the WBS is an organized method to breakdown a product into sub products at lower levels of detail.
- c) A WBS can be expressed down to *any level of interest*. Generally, the top three levels are sufficient unless the items identified are high cost or high risk. Then, is it important to take the WBS to a lower level of definition.

## Work Breakdown Structure

### Kinds of representation

- mind map (only for the initial version of the WBS)
- ✓ **diagram** (e.g. tree) representing the structure
- ✓ numbered (and indented) **list of all items**

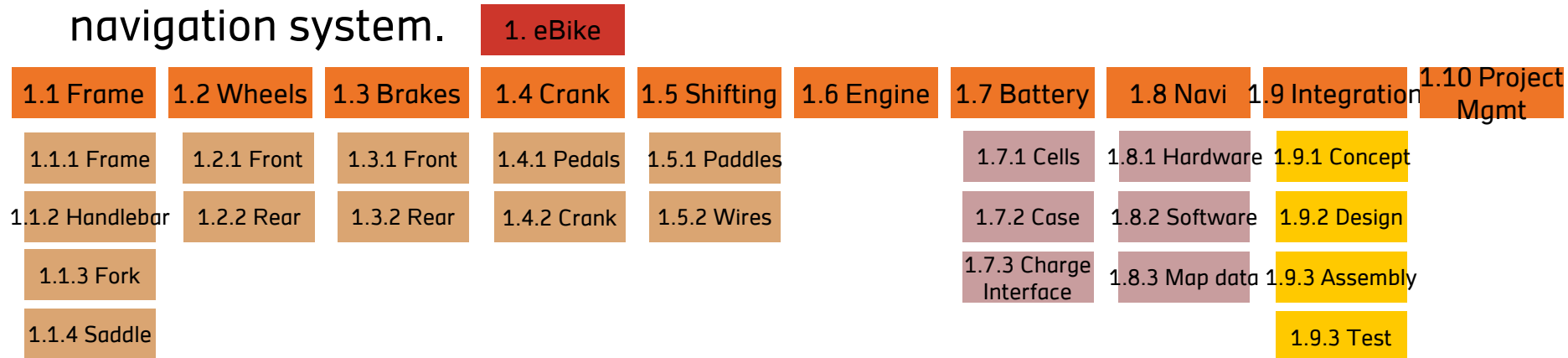


1.0 Software for C Compiler  
1.1 Build a C Compiler  
1.1.1 Build a User Interface  
1.1.2 Build a File System  
1.1.3 Build a Parser  
1.1.4 Build a Code Generator  
1.1.5 Build a Run-Time System  
1.2 Build the Test Suite for the Compiler  
1.2.1 etc.  
1.3 Write Documentation  
1.4 Write Installation Software  
1.5 Manage the Above

1.1 Build a C Compiler  
1.1.1 Build a User Interface  
1.1.1.1 Analyze Requirements for User I/F  
1.1.1.2 Design the User Interface  
1.1.1.3 Code the User Interface  
1.1.1.4 Test and Integrate the User Interface  
1.1.2 etc.

## Work Breakdown Structure – Example

Create a WBS for developing and launching a pedelec (eBike) with a navigation system.



## Work Breakdown Structure: Development

1. **Identify final project products necessary for achieving project success.** The WBS should assist the project manager in developing a clear vision of the end product. You need to answer the following question:
  - What must be delivered to achieve project success?
    - You may need to review the project scope documents for guidance.
2. **Identify the major deliverables necessary for project success.**
  - These are items that by themselves do not satisfy the project need but combined make up a successful project
  - Examples: a design completion, acceptance test completion
    - Candidates for milestones

## Work Breakdown Structure: Development (2)

3. Incorporate additional levels of detail until requirements for managing and controlling the project are met.
  - Each project is different, thus each WBS will be different
  - WBS's from previous projects can be used as templates, but remember that the management philosophy and the level of details may differ between projects
  - Understand your controlling and reporting requirements
    - Projects have different requirements; make sure you take these into consideration when developing low level details
4. Review and refine the WBS until the stakeholders agree with the level of project planning and reporting.
  - Remember that no matter how detailed a WBS is, there are planning and reporting restrictions a WBS creates.

## Work Breakdown Structure: Structuring

→ aims at stability, logical clustering and minimum interdependencies between branches

structure on the top-most level – alternatives:

- according to the deliverables of the project
  - e.g., according to the software modules
- according to the project phases
  - e.g., analysis, design, implementation, integration, installation, on-site acceptance test
- According to responsibility

in a strict sense, activities do *not* belong to the WBS hierarchical structure

- at least down to the level of work packages
- in most projects (depending on their size), significantly deeper structure (parts of work packages, individual tasks)

## WBS Component & Work Package Description Contents

- **WBS Components**
  - Milestones
  - Deliverables
  - Activities
  - Scope
  - Sometimes dates, resources, costs, quality
- Work Packages
  - Responsible/acting persons (or roles) → responsibility matrix
  - required knowledge and capabilities
  - duration and effort
  - deliverables
  - prerequisites for the WP
  - criteria for successful acceptance
  - figures (management ratios) to verify appropriate execution



WBS dictionary  
as a Best Practice



## WBS: Rules

### 100% Rule

- states that the WBS includes 100% of the work defined by the project scope and captures all deliverables – internal, external, interim – in terms of the work to be completed, including project management
- applies at all levels within the hierarchy, down to the activity: the sum of the work at the “child” level must equal 100% of the work represented by the “parent” and the WBS should not include any work that falls outside the actual scope of the project
- Is one of the most important principles guiding the development, decomposition and evaluation of the WBS

### Mutual exclusion

- no overlap in scope definition between two elements of a WBS. This ambiguity could result in duplicated work or miscommunications about responsibility and authority.
- Likewise, such overlap is likely to cause confusion regarding project cost accounting.

## WBS: Best Practices

Iterative procedure

- Finish layers completely before turning to a lower (more detailed) layer (comparable to top-down programming style)
- Test in parallel
  - Logical coherence & minimum branch interdependencies
  - nothing omitted or forgotten: use Project Management and factual Building Blocks as checklists
    - project management, quality assurance, configuration management, ...
    - basic software: procurement, installation, administration, ...
    - internal designs, prototypes (not all of them are necessarily visible for the customer 😊), ...
    - test, putting into operation, acceptance, ...

Use Hierarchical Codes/IDs for WBS elements

Use "Common sense" when creating durations of activities/groups necessary to produce a deliverable

## WBS: rules of thumb

Activity size: 40 hour/5 person days rule

- no single activity or group of activities to produce a single deliverable should be more than 40 hours of effort
- no activity/series should be longer than a single reporting period

Project management effort: 5-8% of the overall effort, 10% strict max.

For software projects of up to approx. € 1 Mio: Balanced structure

- up to a maximum of +-10 elements of the two levels of the WBS
- realistic, i.e., „limited“ number of deliverables

Top-most structure according to – among others – the kind of project

- “real” research and development (R&D) project
  - ✓ structure according to project phases often makes sense
  - less likely to require fundamental changes in the WBS
- “pure” development projects (with less risk):
  - ✓ structure according to deliverables makes more sense

## WBS: Building Approaches

Approach A



Approach B



Approach C



Approach D



*Many Others Are Possible*

Source: Futrell et al. (2002): Quality software project management

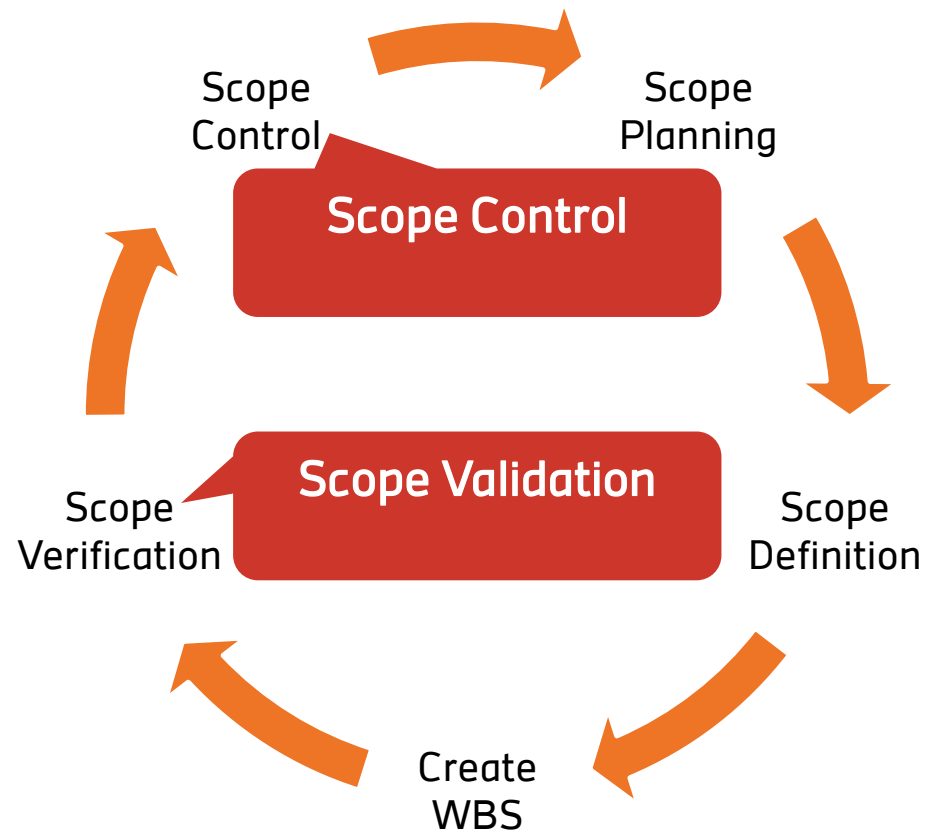
## PMBOK Scope Management: Scope Validation & Control

### Scope Validation

- formalizing acceptance of the completed project scope by the stakeholders

### Scope Control

- controlling changes to the project scope



Questions? Questions!

**THANK YOU VERY MUCH FOR  
YOUR ATTENTION!**

## Course Outline - Week Plan revisited

Agenda			Homework	HW Hrs
16.03.	Opening Ceremony			
23.03.	0 Organisation	1 Introduction, Motivation, Basics	Paper: Pick a Task & Read Starting Sources!	7
30.03.	2 Building Blocks: SWEBOK & PMBOK	Exercise Scope Management: Work Breakdown Structure	Paper: Literature Research/Exercise Scope Management: Work Breakdown Structure Video 3 Building Blocks: PRINCE2 & IPMA	4/2/2
06.04.	Project Phases: Start/Planning	Exercise Work Breakdown Structure	Paper: Literature Research & 1 <sup>st</sup> chapter	8
13.04.	Easter	Easter	Paper: Literature Research & 1 <sup>st</sup> chapter	7
20.04.	Project Phases: Controlling	Exercise Project Planning	Paper: Literature Research & 1 <sup>st</sup> chapter	7
27.04.	Exercise Project Planning	Exercise Project Controlling	Paper: extend	8
04.05.	Exercise Project Controlling	Managing Quality/Risk	Paper/Exercise Project Controlling	6/2
11.05.	Managing HR	Agile Project Management	Paper: extend	7
18.05.	Link to Portfolio Management	Advanced Topics: Your Speeches	Paper: finalize	7
25.05.	Advanced Topics: Your Speeches	Advanced Topics: Your Speeches	Paper: finalize/presentation prep & read others	7
01.06.	Pentecost		Paper: finalize/presentation prep & read others	7
08.06.	Compensation for Speech Prep.	Compensation for Speech Prep.	Paper: read others	5
15.06.	Advanced Topics: Your Speeches	Advanced Topics: Your Speeches	Paper: read others	4
22.06.	Advanced Topics: Your Speeches	Advanced Topics: Your Speeches	Exam Prep Prof. Dr. Lückemeyer	7 47
29.06.	Question Session			

## Speech Topics, Order and Groups

1) Project Classifications/Typologies (Lokappa)	21) Estimation Method Comparison (Mayur)	May 18?	June 15
2) Project Environment (Obena)	22) Project Controlling Methods (Belz)		
3) Project Complexity (Nijhawan)	23) Project Closure (Radharishan)		
4) Project Success Definitions & Factors (DiMartino)	24) Project Manager/Style Influence (Hugo)		
5) Project Failure Reasons (Upadhye)	25) Project Teams (Kumar)		
6) Project Management and Trust (Giang)	26) Managing Small Projects (Rubehn)	May 25	June 22
7) Project Stakeholder Management (Joshi)	27) Managing Megaprojects (Belkacem)		
8) Value Focus in Projects (Nayyab)	28) Managing Int./Cloud Projects (Rudresh)		
9) Contract and Location Constellation (Müller)	29) Managing Maintenance Projects (Böhm)		
10) Project Governance (Mandaogane)	30) Software Project Characteristics (Willms)		
11) Project Organisation (Venuprasad)	31) Agile Project Management Approaches in other Branches (Thomas)	May 25	June 22
12) Project Management Standards Comparison (Kharade)	32) Managing Public Sector Projects		
13) Agile Project Management (Escudé)	33) Maturity Model Comparison (Ruby)		
14) Lean Project Management (Hanumanthappa)	34) Project Sustainability (Maury-Zographos)		
15) Project Risk Management (Iqbal)	35) Project Sequence Management (Srivastava)		
16) Project Quality Management (Rron)	36) Agile/Hybrid Project Portfolio Management (Shah)	June 15	June 22
17) Managing Change (Mellihalli)			
18) Project Phases/Stages (Gauri)			
19) Project Milestones (Omerovic)			
20) Project Generation/Initiation (Mahanjan)			