Analysis of Software Engineering from An Engineering Perspective

Alain Abran and Kenza Meridji

Walter G. Vincenti, in his book "What engineers know and how they know it", has proposed a taxonomy of engineering knowledge. Software Engineering, as a discipline, is certainly not yet as mature as other engineering disciplines, and some authors have even challenged the notion that Software Engineering is indeed engineering. To investigate this issue, Vincenti's categories of engineering knowledge are used to analyze the SWEBOK (Software Engineering Body of Knowledge) Guide from an engineering perspective. This paper presents an overview of the Vincent'si categories of engineering knowledge, followed by an analysis of the engineering design concept in Vincenti vs. the design concept in the SWEBOK Guide: this highlights in particular the fact that Vincenti's engineering design concept is not limited to the design phase knowledge area in the SWEBOK Guide, but that it pervades many of the SWEBOK knowledge areas. Finally, the SWEBOK Software Quality knowledge area is selected as a case study, and analyzed using Vincenti's classification of engineering knowledge.

Keywords: Engineering Knowledge, ISO 19759, Software Engineering, SWEBOK, Vincenti.

1 Introduction

"Engineering is a problem-solving activity…dealing mainly with practical problems" (Vincenti).

Software engineering is defined by the IEEE (Institute of Electrical & Electronics Engineers) as "*the application of a systematic, disciplined, quantitative approach to the development, operation and maintenance of software, the application of engineering to software*" (IEEE 610.12) [1]. Of course, in comparison with mechanical and electrical engineering, Software Engineering is still an emerging engineering discipline, and one that is not as mature as other classical engineering fields.

There are millions of software professionals worldwide, and software is a ubiquitous presence in our society. However, the recognition of Software Engineering as an engineering discipline is still being challenged.

Achieving consensus by the profession on a core body of knowledge is a key milestone in all disciplines, and has been identified by the IEEE Computer Society as crucial for the evolution of Software Engineering towards professional status. The Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), written under the auspices of the IEEE Computer Society's Professional Practices Committee, was initiated in 1998 to develop an international consensus [2] in pursuing the following objectives:

- to characterize the content of the Software Engineering discipline,
- to promote a consistent view of Software Engineering worldwide,
- to provide access to the Software Engineering body of knowledge,
- to clarify the place, and set the boundary, of Software Engineering with respect to other disciplines, and
- to provide a foundation for curriculum development and individual certification material.

The SWEBOK Guide [2], also adopted as a technical report by the ISO (Internacional Organization for Standardization) [3], has been selected as the subject of a study to explore the following question: Is Software Engineering truly an engineering discipline?

The content of each knowledge area (KA) in the SWEBOK Guide was developed by domain experts and extensively reviewed by an international community of peers. This Delphi-type approach, while very extensive and paralleled by national reviews at the ISO level, did not specifically address the engineering perspective, nor did it provide a structured technique to ensure the completeness and full coverage of fundamental engineering topics. Therefore, it did not provide sufficient evidence that it had adequately tackled the identification and documentation of the knowledge expected to be present in an engineering discipline.

Authors

Alain Abran is a Professor and the Director of the Software Engineering Research Laboratory at the *École de Technologie Supérieure (ETS) – Université du Québec*, Montreal, Canada. He is the Co-executive editor of the Guide to the Software Engineering Body of Knowledge project. He is also actively involved in international Software Engineering standards and is Co-chair of the Common Software Measurement International Consortium (COSMIC). Dr. Abran has more than 20 years of industry experience in information systems development and Software Engineering. He is the co-executive editor of the IEEE project on the Guide to the Software Engineering Body of Knowledge – SWEBOK (ISO TR 19759). <aabran@ele.etsmtl.ca>

Kenza Meridji is a PhD student at the Software Engineering Research Laboratory of the *École de Technologie Supérieure – Université du Québec*, Montreal, Canada. She is pursuing research on the fundamental principles of Software Engineering. She holds a Master degree in Software engineering from Concordia University, Canada. <kenza.meridji.1@ens.etsmtl.ca>

Engineering Vocabulary	Definitions
Design	Denotes both the content of a set of plans (e.g. in the design for a new aeroplane and the process by which those plans are produced.
Normal configuration	The general shape and arrangement commonly agreed upon to best embody the operational principle.
Normal technology	According to Edward's constant that "what technological communities usually do" comprises "***the improvement of the accepted tradition*** or its application under new or more stringent conditions."
Normal design	The design involved in normal technology. The engineer working with such a design knows at the outset ***how the device in question works*** and what ***its customary features*** are, and that, if properly designed along such lines, it has a good likelihood of accomplishing the desired task. **Normal design is evolutionary rather than revolutionary.**
Operational principle	Defines the essential fundamental concept of a device,. "How its characteristic parts… fulfill their special function in combination to [sic] an overall operation which archives the purpose."
Production	Denotes the process by which these plans are translated into the concrete artifice.
Operation	Deals with the employment of the artifice in meeting the recognized need.
Radical design	How the device should be arranged, or even how it works, is largely unknown. The designer has never seen such a device before and cannot presume that it will succeed.
Engineering knowledge	The knowledge used by engineers. Engineering knowledge has to do not only with design, but also with production and operation.
Descriptive knowledge	The knowledge of how things are.
Prescriptive knowledge	The knowledge of how things should be to attain a desired end.
Device	Devices are ***single***, relatively ***compact entities***, such as aeroplanes, electric generators, turret lathes, and so forth (Laundan).
Systems	Systems are ***assemblies of devices*** brought together for a collective purpose. Examples are airlines, electric power systems and automobile factories.
Technologies	Denote ***systems and devices*** taken together.
Concepts	***May exist explicitly only*** in the designer's mind. They are unstated givens for the project, having been absorbed by osmosis, so to speak, by the engineer in the course of his development, perhaps even before entering formal engineering training. They had to be learned deliberately by the engineering community at some time, however, and form an essential part of design knowledge.

Table 1: Vincenti's Vocabulary Relating to Engineering Terms and Concepts [4].

In this paper, an approach is proposed to investigate the content of the SWEBOK Guide in a structured way to verify what engineering knowledge is included in the Guide, and what could be missing. This approach is based on Vincenti's classification of engineering knowledge. However, since this classification of engineering knowledge had not, at the time of this investigation, been used to analyze other engineering disciplines, it was felt necessary to carry out some structuring and modelling of the embedded criteria to render its use practical in the analysis of the SWEBOK Guide. In particular, the engineering design concepts had to be probed further, since at first glance there seemed to be a disconnect between the SWEBOK Guide design concept and Vincenti's description of engineering design. Finally, Vincenti's criteria are used to analyze a section of the SWEBOK Guide, the Software Quality KA.

This paper is organized as follows: Section 2 introduces Vincenti's engineering viewpoint, and Section 3 presents a set of models developed to facilitate the use of Vincenti's concepts for the analysis of an engineering discipline. Section 4 presents a mapping of Vincenti's engineering design concept to the SWEBOK Guide Software Engineering design concept. Section 5 analyzes, from an engineering viewpoint, the Software Quality KA of the SWEBOK Guide, and, finally, in Section 6, a summary and future research directions are presented.

2 Vincenti's Engineering Viewpoint

2.1 Overview and Context

Vincenti, in his book "*What engineers know and how they know it*" [4], proposed a taxonomy of engineering knowledge based on the historical analysis of five case studies in aeronautical engineering covering a roughly fifty-year period. He identified different types of engineering knowledge and classified them in six categories:

1. fundamental design concepts,
2. criteria and specifications,
3. theoretical tools,

4. quantitative data,
5. practical considerations, and
6. design instrumentalities.

Furthermore, Vincenti stated that this classification is not specific to the aeronautical engineering domain, but can be transferred to other engineering domains. However, he did not provide documented evidence of this applicability and generalization to other engineering disciplines, and no author could be identified as having attempted to do so either. In this paper, we propose some pioneering work on the use of Vincenti's categorization of engineering knowledge as constituting criteria for investigating Software Engineering from an engineering perspective.

The Vincenti categorization of knowledge was first used in a graduate seminar in 2002 at the *École de Technologie Supérieure, Université du Québec* (Canada), as an analyti-

Engineering Knowledge Category	Corresponding Criteria
Fundamental design concepts	• About the design • Designers must know the operational principle of the device • How the device works • Normal configuration • Normal design • Other features may be opened
Criteria and specifications	• Specific requirements of an operational principle • General qualitative goals • Specific quantitative goals laid out in concrete technical terms • The design problem must be "well defined" • Unknown or partially understood criteria • Assignment of values to appropriate criteria • This task takes place at the project definition level in the design hierarchy. • Definition of technical specifications
Theoretical tools	• Mathematical methods and theories for making design calculations • Intellectual concepts for thinking about the design • Precise and codifiable • They come mostly from deliberate research • They are not sufficient by themselves
Quantitative data	• Specify manufacturing processes for production • Display the detail for the device • Data essential for design • Obtained empirically • Calculated theoretically • Represented in tables or graphs • Precise and codifiable • They come mostly from deliberate research • They are not sufficient by themselves
Practical considerations	• Theoretical tools and quantitative data are not sufficient. Designers also need practical considerations derived from experience • Practical considerations are learned on the job, and not in school or from books • Practical considerations are rarely documented • Practical considerations are also derived from production and operation • This knowledge is difficult to define • This knowledge defies codification • A prototype must often be built to check the designer's work • The practical consideration learned from operation is judgment • Rules of thumb • The practices from which these rules derive include not only design, but production and operation as well
Design instrumentalities	• Knowing how • Procedural knowledge • Ways of thinking • Skills based on judgment • Knowledge on how to carry out tasks • Must be part of any anatomy of engineering knowledge

Table 2: Vincenti's Engineering Knowledge Categories and Criteria.

cal tool to tackle this issue by analyzing each of the SWEBOK KAs separately. The initial purpose of this approach was to gain insights into their content and structure, which, by definition, were expected to be of an engineering knowledge type. While it was easy for graduate students at the Master's degree and doctoral levels to use Vincenti's criteria to analyze the SWEBOK Design KA and to propose a mapping to the Vincenti categorization, this proved to be much more challenging for all the other KAs, to the point where some of these students questioned the relevance of the applicability of Vincenti's categorization to these other SWEBOK KAs, and, as a corollary to that, that these other KAs did not necessarily constitute knowledge of an engineering type. The specific vocabulary defined by Vincenti is presented in Table 1.

2.2 Vincenti's Categorization Criteria & Goals

Vincenti provides a categorization of engineering design knowledge and the activities that generate it. However, the divisions are not entirely exclusive; some items of knowledge can contain the knowledge of more than one category. From Vincenti's definitions of each engineering knowledge-type category, a number of criteria were identified and have been listed in Table 2. The goals of each category have also been identified, and these are listed in Table 3.

3 Vincenti's Classification of Engineering Knowledge Types

3.1 Relationship between The Various Categories

Since the categories are not mutually exclusive, it is important to understand the relationships between them. An initial modelling of Vincenti's categories of engineering knowledge is presented in Figure 1. This figure illustrates that, in seeking a design solution, designers move up and down within categories, as well as back and forth from one category to another.

It can also be noted that three categories (theoretical tools, quantitative data and design instrumentalities) are related in the following manner: theoretical tools guide and structure the data, while quantitative data suggest and push the development of tools for their presentation and application – see Figure 2. Furthermore, both theoretical tools and quantitative data serve as input for design instrumentalities, while appropriate theoretical tools and quantitative data are needed for technical specifications. The next section presents several models to illustrate the relationships across these engineering concepts.

3.2 Vincenti's Classification of Engineering Knowledge-type Models

We now present a detailed description of Vincenti's six categories of engineering knowledge and the related models for each. Vincenti stated that these categorizations of engineering knowledge are not exclusive, since some elements of knowledge can be found in more than one category.

3.2.1 Fundamental Design Concepts

The goal of 'fundamental design concepts', according to Vincenti, is as follows: "*Designers setting out on any normal design bring with them fundamental concepts about the device in question,*" which means the definition of fundamental concepts related to the device by the designer. Fundamental design elements are composed of four elements (Figure 3); operational principles, normal configuration, normal technology and concepts in people's minds. At first, these concepts exist only in the designer's mind:

Concepts in people's minds are inputs to the project (Figure 4).

Engineering Knowledge Category	Goals
Fundamental design concepts	Designers embarking on any ***normal design*** bring with them ***fundamental concepts*** about the device in question.
Criteria and specification	To design a device embodying a given operational principle and normal configuration, the designer must have, at some point, ***specific requirements*** in terms of hardware.
Theoretical tools	To carry out their design function, engineers use a wide range of ***theoretical tools***. These include intellectual concepts as well as mathematical methods.
Quantitative data	Even with fundamental concepts and technical specifications at hand, mathematical tools are of little use without ***data*** for the ***physical properties*** or other ***quantities*** required in the formulas. Other kinds of data may also be needed to ***lay out details of the device*** or to ***specify manufacturing processes*** for production.
Practical considerations	To complement the theoretical tools and quantitative data, which are not sufficient. Designers also need ***less sharply defined considerations*** derived from experience.
Design instrumentalities	Besides the analytical tools, quantitative data and practical considerations required for their tasks, designers need to know ***how to carry out those tasks***. How to ***employ procedures*** productively constitutes an essential part of design knowledge.

Table 3: Vincenti's Engineering Knowledge Categories and Goals.

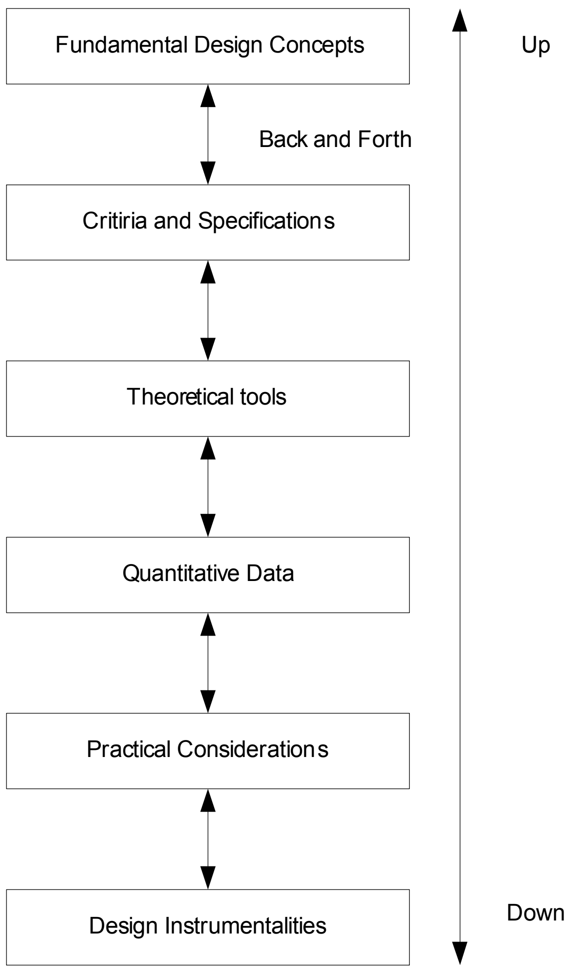


Figure 1: Vincenti's Classification of Engineering Knowledge.

Operational principles define the essential fundamental concept of a device. "*How its characteristic parts… fulfill their special functions in combination to [sic] an overall operation which archives the purpose*." The operational principles must be known by the designers first (Figure 5) and constitute the basic components for the design, whereas operational principles are abstract, and the design moves from abstract concepts to precise concepts (Figure 6).

Normal configuration is "*the general shape and arrangement that are commonly agreed to best embody the operational principle.*"

Normal technology is "*the improvement of the accepted tradition or its application under new or more stringent conditions.*" Design, in Vincenti, "denotes both the content of a set of plans (as in the design for a new aeroplane) and the process by which those plans are produced." There are two types of design: **normal design** and **radical design**. The latter is a kind of design that is unknown to the designer, and where the designer is not familiar with the device itself. The designer does not know how the device should be arranged, or even how it works. The former is a traditional design, where the designer knows how the device works. The designer also knows the traditional features of the device. This type of design is also the design involved in normal technology, which was mentioned earlier. In conclusion, "*normal design is evolutionary rather than revolutionary.*" Finally, a normal configuration and operational principles together provide a framework for normal design (Figure 7).

In Vincenti, a normal technology, or design, is part of a normal configuration and of a related operational principle.

3.2.2 Criteria and Specifications

The goal for 'criteria and specifications' can be expressed as follows: "*To design a device embodying a given operational principle and normal configuration, the designer must have, at some point,* ***specific requirements*** *in terms of hardware.*" The designer designs a device meeting specific requirements which include a given operational principle as well as a normal configuration. First, the design problem must be well defined. Then, the designer translates general quantitative goals into specific quantitative goals (Figure 8): the designer assigns values or limits to the characteristics of the device which are crucial for engineering design. This allows the designer to provide the details and dimensions of the device that will be given to the builder. Furthermore, the output at the prob-

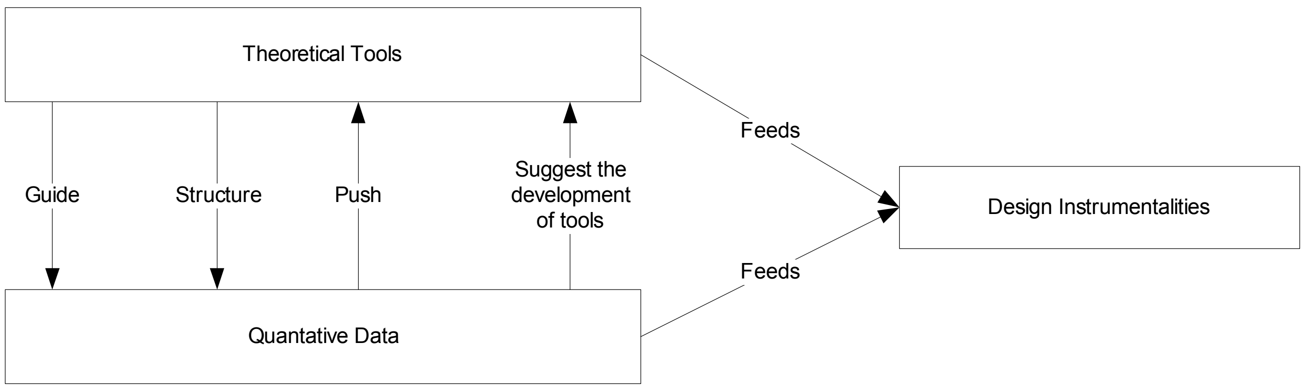


Figure 2: Relationships between Theoretical Tools & Quantitative Data.

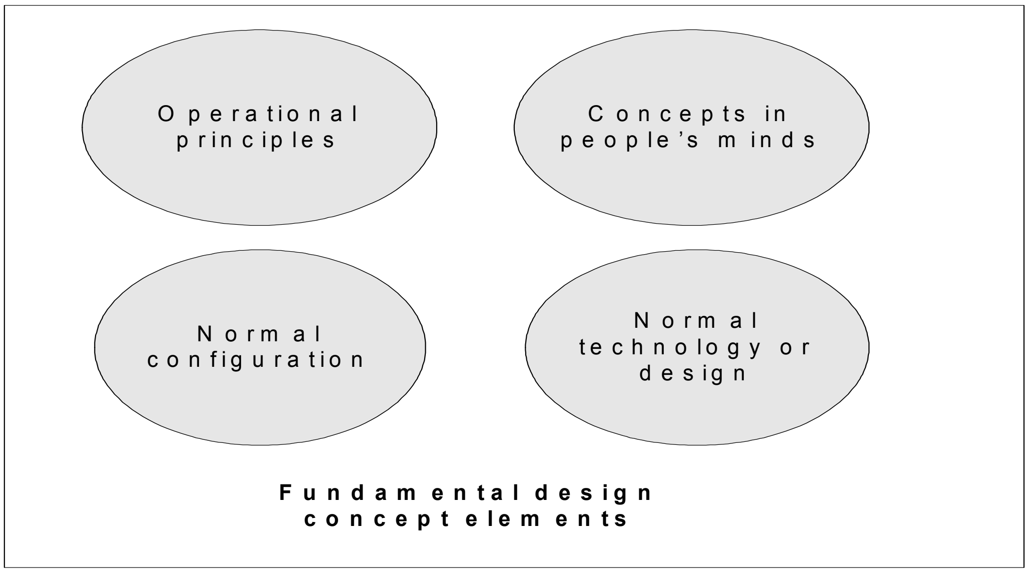

Operational principles
Concepts in people's minds
Normal configuration
Normal technology or design
Fundamental design concept elements

Figure 3: Elements of A Fundamental Design Concept.

lem definition level is used, in turn, as input to the remaining design activities that follow (Figure 9). These specifications are more important where safety is involved, as in the case of aeronautical devices. The criteria on which the specifications are based become part of the accumulating body of knowledge about how things are done in engineering. Finally, 'criteria and specifications' exists as a category of knowledge only in engineering and not in science. In science, the aim is to understand: scientists do not need to have highly specified or concrete objectives. In engineering, by contrast, to design a device, criteria and specified goals are crucial.

Design instrumentalities contain instrumentalities of the process, the procedures, judgment and ways of thinking. The latter are less tangible than procedures and more tangible than judgment; an example of ways of thinking is 'thinking by analogy'. Judgment is needed to seek out design Having the analytical tools, quantitative data and practical considerations at hand, designers also need procedural knowledge to carry out their tasks, as well as to know how carry out their tasks, as well as to know how to employ these procedures.

3.2.3 Theoretical Tools

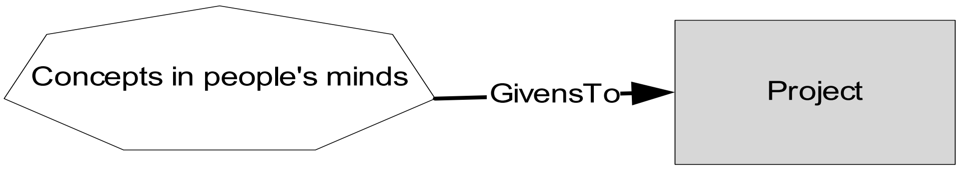


Figure 4: Project Input.

Theoretical tools are used by engineers to carry out their design. The goal of the 'theoretical tools' category is expressed by Vincenti as follows: "*To carry out their design function, engineers use a wide range of* ***theoretical tools****. These include intellectual concepts as well as mathematical methods*" (Figure 10). Intellectual concepts (such as design concepts, mathematical methods and theories) are tools for making design calculations. Both design concepts and methods are part of science.

In the first class of theoretical tools are mathematical methods and theories composed of formulas, either simple or complex, which are useful for quantitative analysis and design. This scientific knowledge must be reformulated to make it applicable to engineering. The engineering activity requires that thoughts be conceived in people's minds. In the second class of theoretical tools are intellectual concepts, which represent the language expressing those thoughts in people's minds. They are employed first in the quantitative conceptualization and reasoning that engineers have to perform before they carry out the quantitative analysis and design calculations, and then again while they are carrying them out.

3.2.4. Quantitative data

The goal of 'quantitative data' is to lay down "*the* ***physical properties*** *or other* ***quantities*** *required in the formulas. Other kinds of data may also be needed to* ***lay out details of the device*** *or to* ***specify manufacturing processes*** *for production.*" Besides fundamental concepts and technical specifications,

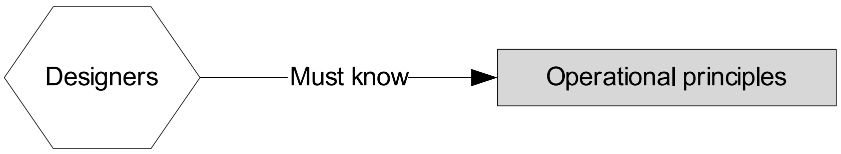


Figure 5: Designers Initial Knowledge.

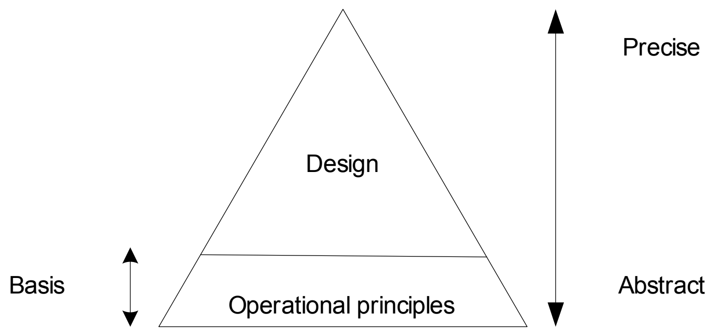


Figure 6: Design Pyramid.

the designers also need quantitative data to lay out details of the device. These data can be obtained empirically, or in some cases they can be obtained theoretically. They can be represented in tables or graphs.

These data are divided into two types of knowledge: prescriptive and descriptive:

- Descriptive knowledge is "*knowledge of how things are.*" It includes physical constants, properties of substances and physical processes. In some situations, it refers to operational conditions in the physical world. Descriptive data can also include measurement of performance.
- Prescriptive knowledge is "*knowledge of how things should be to attain a desired end.*" An example might be: "In order to accomplish this or organize this, arrange things this way."

Operational principles, normal configuration and technical specifications are prescriptive knowledge, because they prescribe how a device should satisfy its objective (Figure 11).

3.2.5 Practical Considerations

According to Vincenti, the goal of 'practical considerations' is "*to complement the role of theoretical tools and quantitative data which are not sufficient. Designers also need for their work* ***less sharply defined considerations*** *derived from experience.*" This kind of knowledge is prescriptive in the way that it shows the designers how to proceed with the design to achieve it. Vincenti refers to practical considerations as constituting non-codifiable knowledge derived from experience, unlike theoretical tools and quantitative data which are very precise and codifiable because these are derived from intentional research. This category of engineering knowledge is needed by designers as a complement to theoretical tools and quantitative data. These practical considerations are learned on the job, rather than at school or from books. They are not to be formalized or programmed. They are derived from design, as well as from production and operation. The practical consideration derived from production is not easy to define and cannot be codified, and a prototype is highly recommended to check the designer's work (Figure 12). An example of a practical consideration from operation is the judgment that comes from the feedback resulting from use.

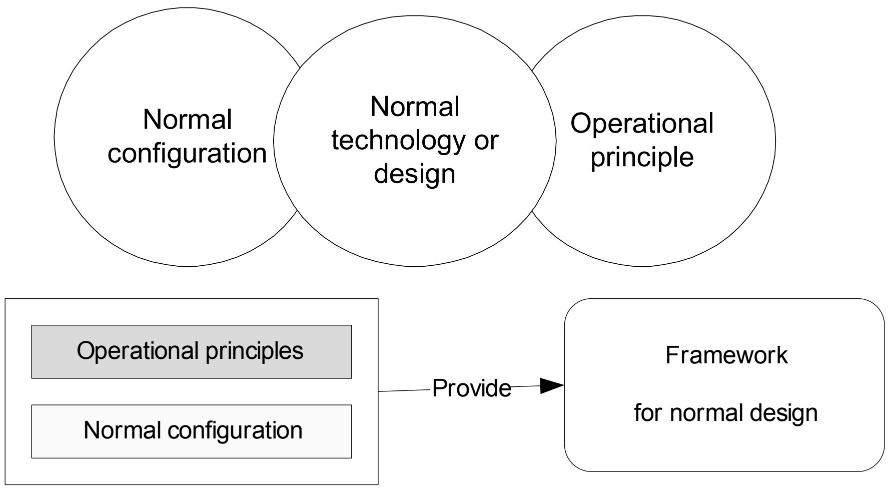


Figure 7: Relationships between Normal Configuration, Operational Principles and Normal Design.

3.2.6 Design Instrumentalities

The goal of 'design instrumentalities' in the engineering design process required for the engineer's tasks is "*to know* ***how to carry out those tasks****. How to* ***employ procedure*** *productively constitutes an essential part of design knowledge*".

Having the analytical tools, quantitative data and practical considerations at hand, designers also need procedural knowledge to

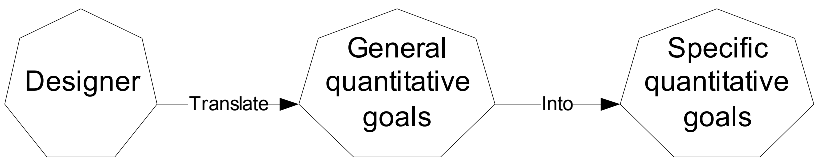


Figure 8: Designer's Goals.

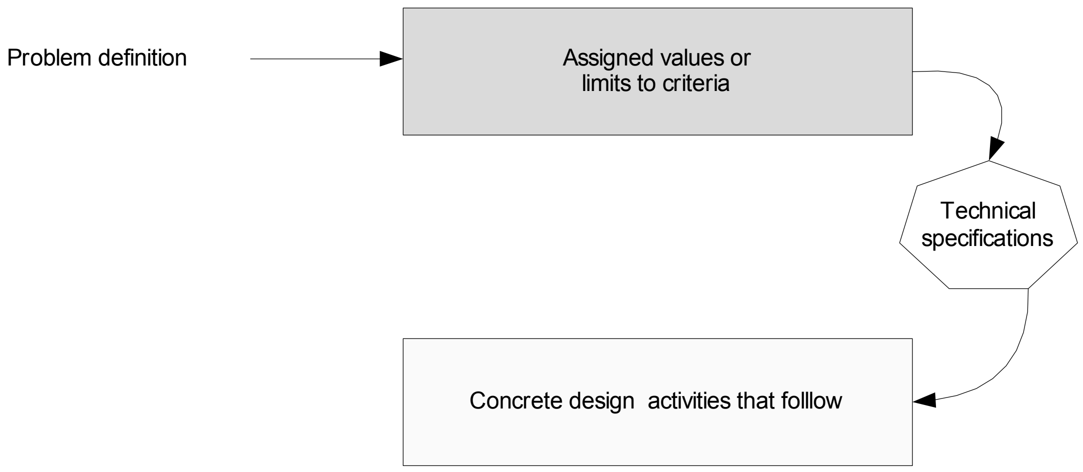

Problem definition
Assigned values or limits to criteria
Technical specifications
Concrete design activities that folllow

Figure 9: Problem Definition Level Output.

carry out their tasks, as well as to know how to employ these procedures.

Design instrumentalities contain instrumentalities of the process, the procedures, judgment and ways of thinking. The latter are less tangible than procedures and more tangible than judgment; an example of ways of thinking is 'thinking by analogy'. Judgment is needed to seek out design solutions and make design decisions (Figure 13)

4 The Design Process

4.1 TheEngineering Design Process in Vincenti

According to Vincenti, the engineering '*design*' concept "*denotes both the content of a set of plans (as in the design for a new aeroplane) and the process by which those plans are produced*". In Vincenti's view, design is an iterative and complex process which consists of plans for the production of a single entity, such as an aeroplane (device), how these plans are produced, and, finally, the release of these plans for production.

Vincenti mentions that there are two types of design in engineering, ***normal*** and ***radical***. In the former, the designer knows how the device works, how it should be arranged and what its features are. In the latter, the device is new to the engineer who is encountering it for the first time. There-

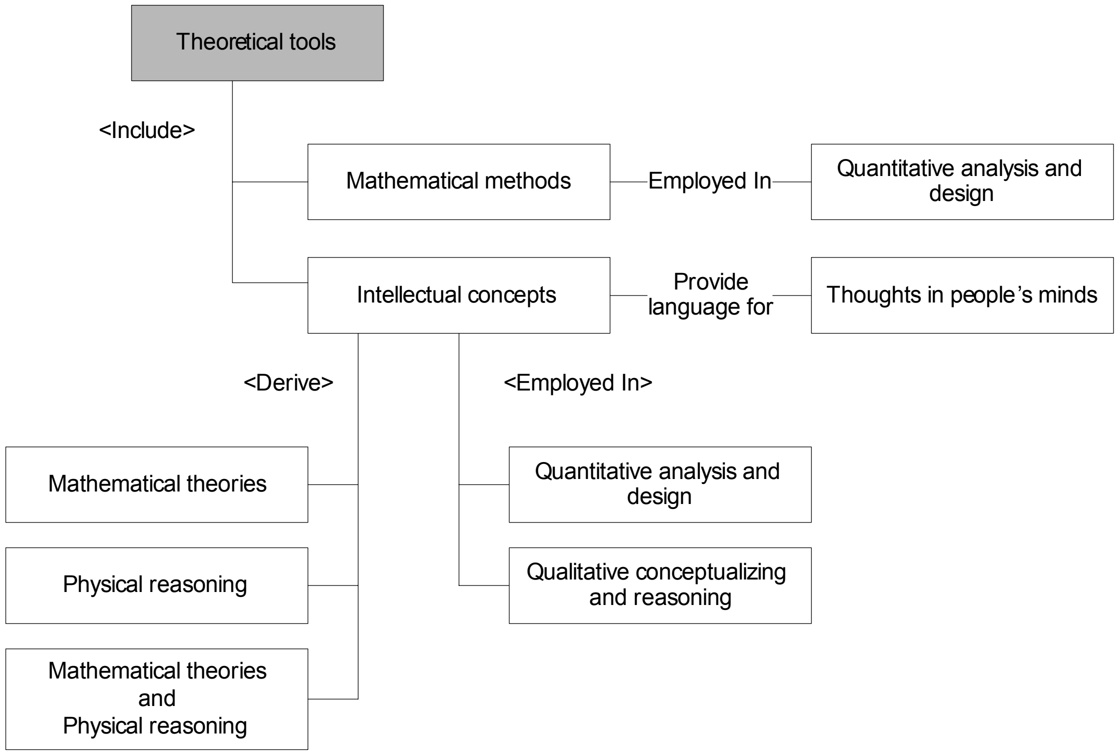

Theoretical tools
<Include>
Mathematical methods
Employed In
Quantitative analysis and design
Intellectual concepts
Provide language for
Thoughts in people's minds
<Derive>
<Employed In>
Mathematical theories
Physical reasoning
Mathematical theories and Physical reasoning
Quantitative analysis and design
Qualitative conceptualizing and reasoning

Figure 10: Theoretical Tools Model.

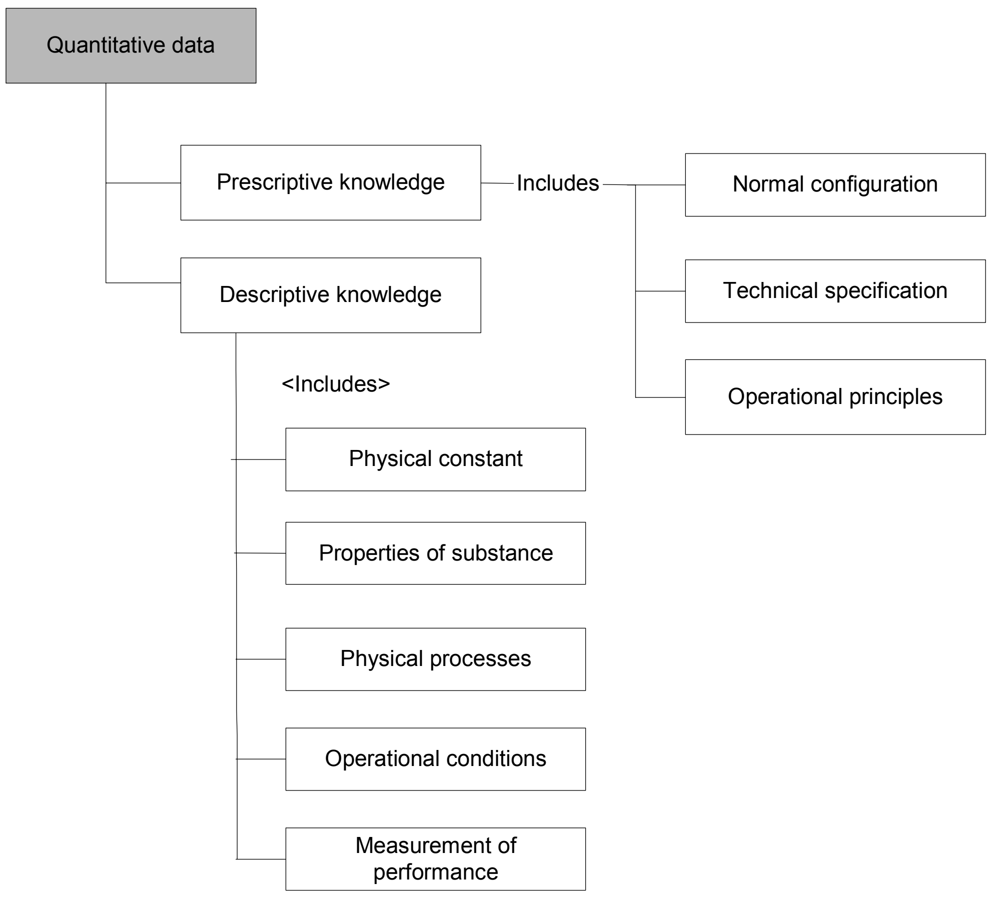


Figure 11: Quantitative Data Model.

fore, the engineer does not know how it works or how it should be organized.

He also mentions that design is a multilevel and hierarchical process. The designer starts by taking the problem as input.

The design hierarchies start from the project definition level, located at the upper level of the hierarchy where problems are abstracted and unstructured.

At the overall design level, the layout and the proportions of the device are set to meet the project definition. At level 3, the project is divided into its major components. At level 4, each component is subdivided. At level 5, the subcomponents from level 4 are further divided into specific problems.

At the lower levels, problems are well defined and structured. The design process is iterative, both up and down and horizontally throughout the hierarchy. Vincenti's view of the levels of design is modeled in Figure 14. At each level of the hierarchy, a design can be either normal or radical.

4.2 The Engineering Process in The SWEBOK Guide

The SWEBOK Guide is composed of ten KAs -- see Figure 15. Each KA is represented by one chapter in the SWEBOK Guide.

4.3 Design Motion in The SWEBOK Guide

The Software Requirements KA is composed of four phases of software requirements: elicitation, analysis, specification and validation. The elicitation phase is the process of deriving requirements through observation of existing systems. Requirements specification is the activity of transforming the requirements gathered during the analysis activity into a precise set of requirements. Software Requirements Specifications describe the software system to be delivered. In the requirements validation phase, the requirements are checked for realism, consistency and completeness.

Software design is defined in [1] as both "*the process of defining the architecture, components, interfaces, and other characteristics of a system or component*" and "*the result of [that] process*". Software ***design*** in the Software Engineering life cycle is an activity in which software requirements are taken as input to the software design phase for analysis. "*Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem.*"

The result will be the description of the software architecture, its decomposition into different components and the description of the interfaces between those components. Also described will be the internal structure of each component

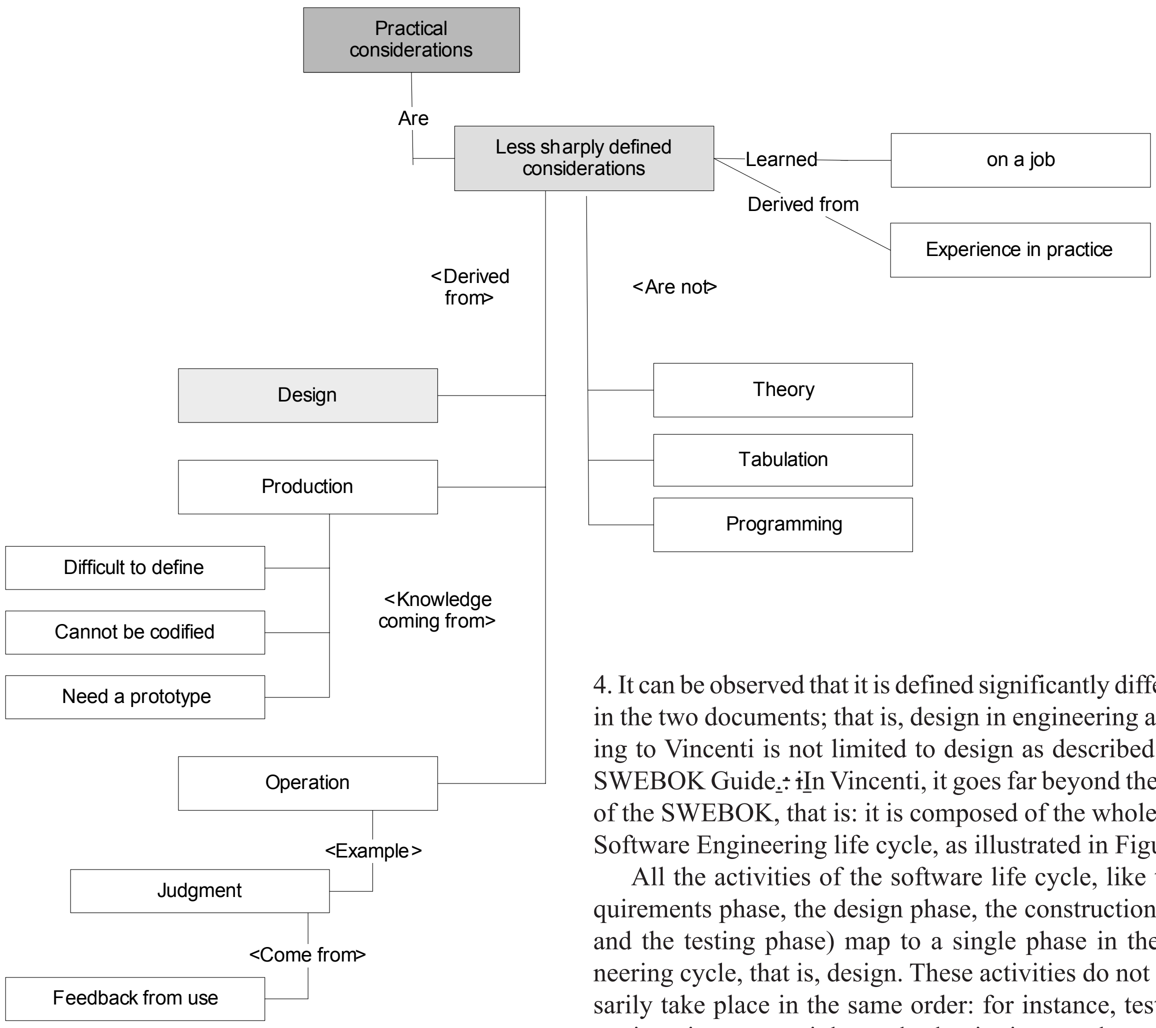


Figure 12: Practical Considerations Model.

and the related program.

4.4 Design KA: Mapping between Vincenti and The SWEBOK Guide

The analysis of the term 'design' in both Vincenti and the SWEBOK Guide is presented in Table 4: it can be observed that it is defined significantly differently in the two documents.

Is there a direct and unique mapping of this 'design' term used in both the SWEBOK Guide and Vincenti's categorization of engineering knowledge?

If there were such a direct mapping, would this mean that only the Design KA in the SWEBOK could be mapped to Vincenti's engineering knowledge? Or, alternatively, is the notion of design defined by Vincenti different from the design concept in Software Engineering as defined in the SWEBOK Guide? And, if so, what is its scope within the SWEBOK Guide?

The definitions and descriptions of this term in both Vincenti and the SWEBOK Guide are presented in Table 4. It can be observed that it is defined significantly differently in the two documents; that is, design in engineering according to Vincenti is not limited to design as described in the SWEBOK Guide.: iIn Vincenti, it goes far beyond the scope of the SWEBOK, that is: it is composed of the whole of the Software Engineering life cycle, as illustrated in Figure 16.

All the activities of the software life cycle, like the requirements phase, the design phase, the construction phase and the testing phase) map to a single phase in the engineering cycle, that is, design. These activities do not necessarily take place in the same order: for instance, testing in engineering starts right at the beginning, at the problem definition level, and goes on until the final release of the plans for production, while in the Software Engineering life cycle, as defined generically in the SWEBOK Guide, testing starts after the construction phase.

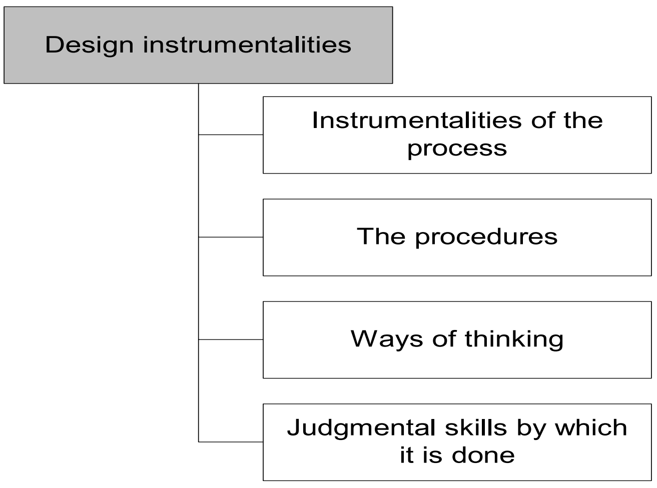

Design instrumentalities
Instrumentalities of the process
The procedures
Ways of thinking
Judgmental skills by which it is done

Figure 13: Design Instrumentalities Model.

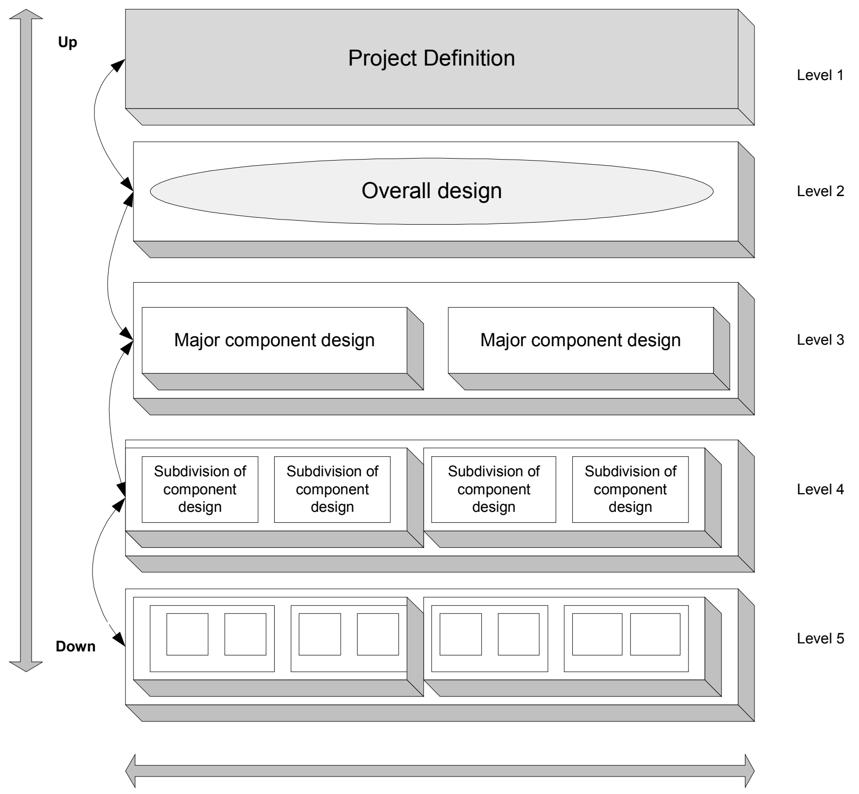

Up
Down
Project Definition
Overall design
Major component design
Major component design
Subdivision of component design
Subdivision of component design
Subdivision of component design
Subdivision of component design
Level 1
Level 2
Level 3
Level 4
Level 5

Figure 14: Modelling of The Levels of The Design Hierarchy, as Described in Vincenti.

The detailed mapping between the different design levels in engineering and in the Software Engineering life cycle is presented in Table 4.

5 Identification of The Engineering Concepts in The Software Quality KA

We present next an analysis of the engineering content within the SWEBOK Guide using one of its ten KAs as a case study, that is, Software Quality.

5.1 Software Quality

Authors and organizations have provided many different definitions of quality. For instance, Phil Crosby (Cro79) stated that it is "*conformance to user requirements*", while Watts Humphrey (Hum89) defined it as "*achieving excellent levels of fitness for use*". IBM uses the term "*market-driven quality*". Furthermore, ISO 900:2000 has described quality as "*the degree to which a set of inherent characteristics fulfills requirements*". Finally, the SWEBOK Guide introduces software quality as a separate KA, describing quality in different ways. The breakdown of software quality topics adopted in the SWEBOK Guide [2][3] is presented in Figure 17.

5.2 Analysis Using The Vincenti Classification of Engineering Knowledge

This section discusses the evaluation of the Software Quality KA of the SWEBOK Guide from an engineering perspective. To analyze the breakdown related to the Software Quality KA, the Vincenti classification of engineering knowledge is used to identify the strengths and weaknesses of this KA, and to gain further insights on the level of maturity of this topic from an engineering viewpoint.

This analysis is based on the models of engineering knowledge described earlier. These models give us a very

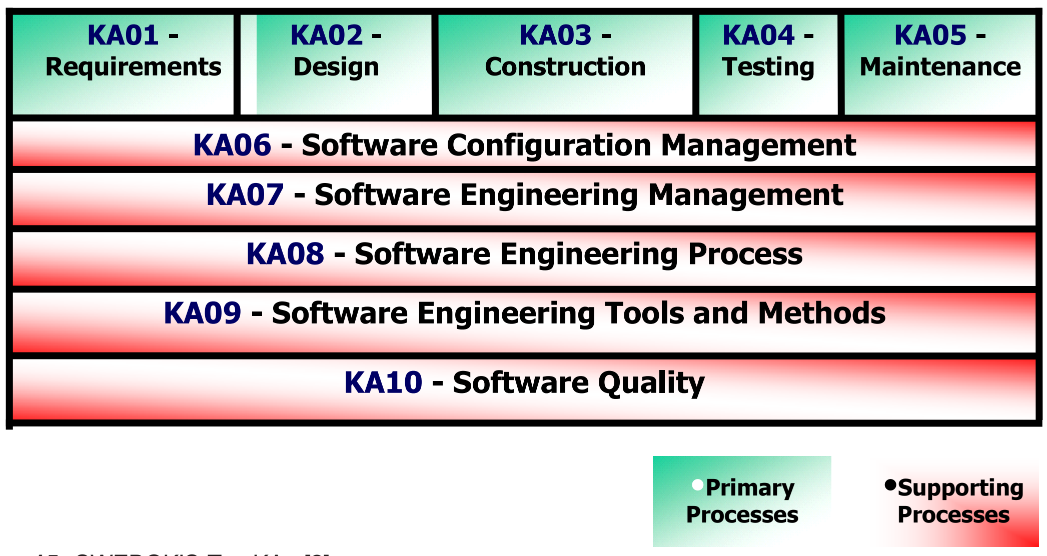

KA01 - Requirements
KA02 - Design
KA03 - Construction
KA04 - Testing
KA05 - Maintenance
KA06 - Software Configuration Management
KA07 - Software Engineering Management
KA08 - Software Engineering Process
KA09 - Software Engineering Tools and Methods
KA10 - Software Quality
Primary Processes
Supporting Processes

Figure 15: SWEBOK'S Ten KAs [2].

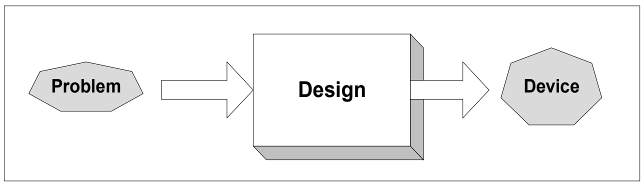
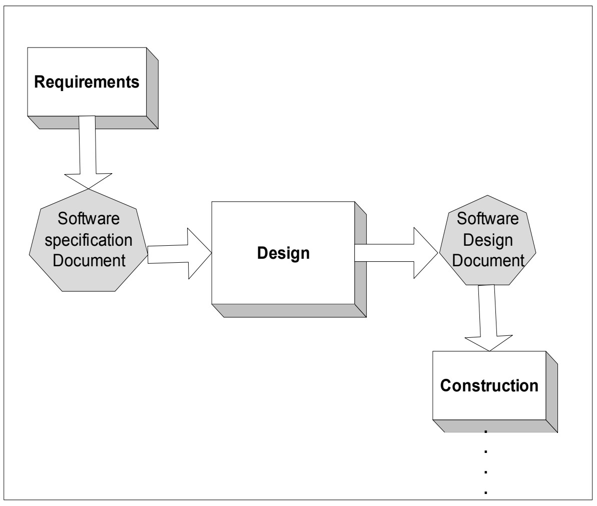
Design definition for engineering according to Vincenti	Design definition for Software Engineering
Design, as defined by Vincenti: "*The content of a **set of plans** (as in the design for a new aeroplane)*" and "*the **process** by which those plans are **produced***".	Design is defined in [IEEE 610.12-90] as both: "*The **process** of defining the **architecture**, components, interfaces, and other characteristics of a system or component*" and "*the result of [that] process*".
***Design**,* in the engineering life cycle is a process which starts by taking as input the problem, following a set of hierarchical levels. This process moves from problem definition to the production of a device as output.	Software ***design*** in the Software Engineering life cycle is an activity in which software requirements are taken as input to the software design phase for analysis. The result will be a precise description of the internal structure of the program.
	 Requirements Software specification Document Design Software Design Document Construction

Table 4: Design According to Vincenti vs. Design in The Software Engineering Life Cycle.

descriptive analysis of the various key elements contained in each of the corresponding engineering knowledge areas. This allows us to make an appropriate mapping between the different categories of the engineering knowledge area and software quality. It helps in identifying the engineering elements contained in this topic, as well as the missing ones. As a result, it looks into the software quality area from an engineering perspective. Table 6 describes the mapping between the corresponding criteria for the classification of engineering knowledge and the related software quality topics. This analysis can provide useful insights into possible strengths and weaknesses of the software quality topic. It helps categorize the knowledge contained in the Software Quality KA of the SWEBOK Guide: for instance, it covers all categories of engineering knowledge from an engineering viewpoint, but this does not mean that it is complete

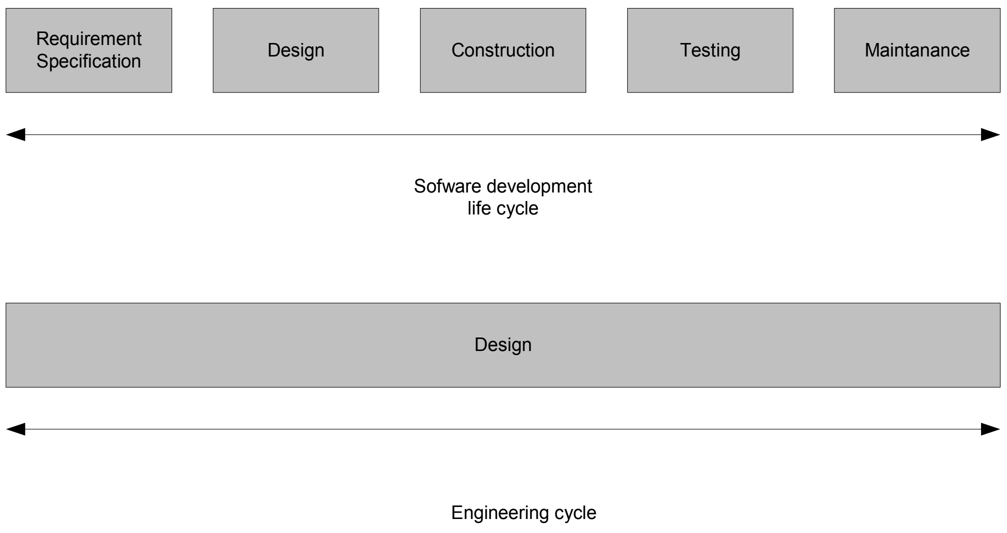


Figure 16: Design According to Vincenti vs. Design in The Software Engineering Life Cycle.

Levels	Description of the design process in engineering	Software engineering life cycle
1	Project Definition	Requirements
2	Overall design – component layout of the aeroplane to meet the project definition.	Specification
3	Major component design – division of project into wing design, fuselage design, landing gear design, electrical system design, etc.	Architecture of the system
4	Subdivision of areas of component design from level 3 according to the engineering discipline required (e.g. aerodynamic wing design, structural wing design, mechanical wing design)	Detailed design
5	Further division of the level 4 categories into highly specific problems	Construction

Table 5: Mapping of The Design Process in Engineering vs. The Software Engineering Life Cycle

and inclusive.

6 Summary

The SWEBOK Guide documents an international consensus on ten Software Engineering KAs within what is referred to as an engineering discipline. Software engineering, as a discipline, is certainly not yet as mature as other engineering disciplines, and some authors have even challenged the notion that Software Engineering is indeed engineering. The work presented here has involved investigating this engineering perspective, first by analyzing the Vincenti classification of engineering knowledge, and second by comparing the design concept in Vincenti vs. the design concept in the SWEBOK Guide.

The result of this analysis was to show that the design issue in Vincenti is not limited to the design issue in the SWEBOK Guide: it goes beyond that, in that it is composed of the whole of the Software Engineering life cycle.

Finally, the SWEBOK Software Quality KA was selected as a case study and analyzed using the Vincenti classification as a tool to analyze this KA from an engineering perspective. This analysis was carried out to identify some of the strengths and weaknesses of the breakdown of topics for the Software Quality KA. It has shown that all the cat-

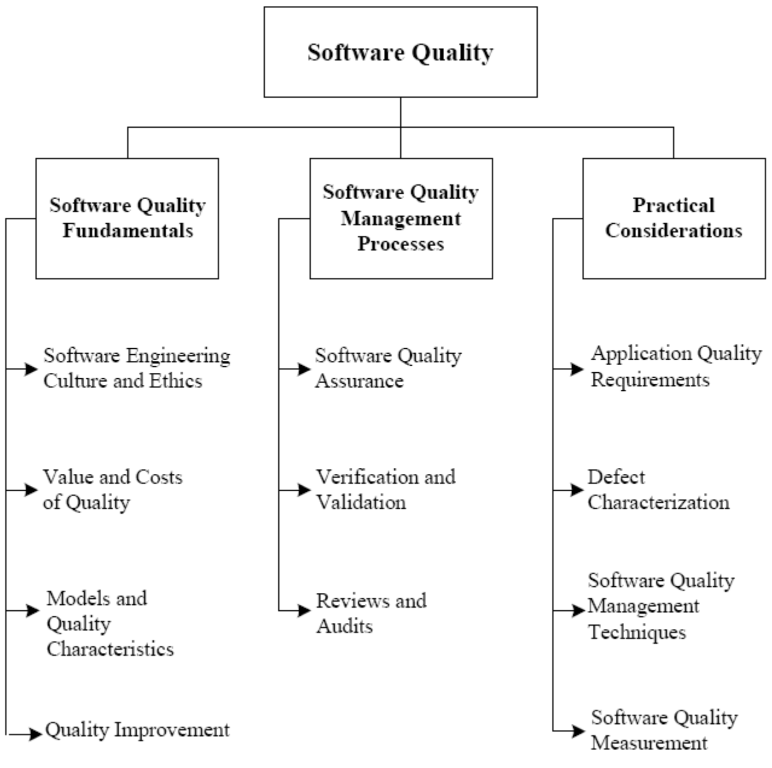

Software Quality
Software Quality Fundamentals
Software Quality Management Processes
Practical Considerations
Software Engineering Culture and Ethics
Value and Costs of Quality
Models and Quality Characteristics
Quality Improvement
Software Quality Assurance
Verification and Validation
Reviews and Audits
Application Quality Requirements
Defect Characterization
Software Quality Management Techniques
Software Quality Measurement

Figure 17: Breakdown of Topics for the Software Quality KA (SWEBOK Guide).

Engineering Knowledge Category	Corresponding Criteria	Quality Concepts Refined
Fundamental design concepts	• About the design • Designers must know the operational principle of the device • How the device works • Normal configuration • Normal design • Other features may be created	• Planning the software quality process • Quality characteristics of the software (QI), (QE), (QIU) • Software quality models • Quality assurance process • Verification process • Validation process • Review process • Audit process
Criteria and specification	• Specific requirement of an operational principle • General qualitative goals • Specific quantitative goals laid out in concrete technical terms • The design problem must be "well defined". • Unknown or partially understood criteria • Assignment of values to appropriate criteria • This task takes place at the project definition level	• Quality objective to be specified • Characteristics of quality tools • Software characteristics • Criteria for assessing the characteristics
Theoretical Tools	• Mathematical methods and theories for making design calculation • Intellectual concepts for thinking about design • Precise and codifiable	• Verification process model • Formal methods • Testing • Theory measurement • Verification/proving properties • TQM (Total Quality Management)
Quantitative data	• Specify manufacturing process for production • Display the detail for the device • Data essential for design • Obtained empirically • Calculated theoretically • Represented in tables or graphs • Descriptive knowledge • Prescriptive knowledge • Precise and codifiable	• Quality measurement • Experimental data • Empirical study • E.g. the process of requirement inspection • Value and cost of quality
Practical Considera-tions	• Theoretical tools and quantitative data are not sufficient. Designers also need considerations derived from experience • It is difficult to find them documented • They are also derived from production and operation • This knowledge is difficult to define • It defies codification • The practical consideration derived from operation is judgment • Rules of thumb	• Application quality requirements • Defect characterization
Design Instrumenta-lities	• Knowing how • Procedural knowledge • Ways of thinking • Judgment skills	• Quality assurance procedures • Quality verification procedures • Quality validation procedures • SQM process tasks & techniques • Management techniques • Measurement techniques • Project planning and tracking • Quality assurance process • Verification process • Validation process • Review process • Audit process

Table 6: Quality Concepts in The SWEBOK Guide Using Vincenti's Classification.

egories of engineering knowledge described by Vincenti are present in this KA of the SWEBOK; that is, it addresses the full coverage of all related engineering-type knowledge. This does not mean, however, that it is all-inclusive and complete, but only that the coverage extends to all categories of engineering knowledge from an engineering viewpoint.

The next stage of this R&D project will focus on investigating the application of Vincenti's engineering knowledge to the analysis of a single candidate Software Engineering principle. This analysis will be performed from an engineering viewpoint to the primary processes contained in the SWEBOK Guide with respect to the following fundamental principle: "*Manage quality throughout the life cycle as formally as possible*".

References

[1] IEEE 610.12-1990 (1990), IEEE Standard Glossary of Software Engineering Terminology, Institute of Electrical and Electronics Engineers ISBN: 155937067X. 84 pages.

[2] A. Abran, J. Moore, P. Bourque, R. Dupuis, L. Tripp. Guide to the Software Engineering Body of Knowledge – SWEBOK, IEEE Computer Society Press, Los Alamitos, 2005. <http://www.swebok.org>.

[3] ISO/IEC TR 19759-2005, Guide to the Software Engineering Body of Knowledge (SWEBOK), International Organization for Standardization - ISO, Ginebra, 2005

[4] W.G. Vincenti. What engineers know and how they know it. Baltimore, London, The Johns Hopkins University Press, 1990.