SWEBOK

Guide to the Software Engineering Body of Knowledge

A Straw Man Version

Pierre Bourque, Université du Québec à Montréal
Robert Dupuis, Université du Québec à Montréal
Alain Abran, Université du Québec à Montréal
James W. Moore, The MITRE Corporation
Leonard Tripp, IEEE Computer Society
Karen Shyne, The Boeing Company
Bryan Pflug, The Boeing Company
Marcela Maya, Université du Québec à Montréal
Guy Tremblay, Université du Québec à Montréal

September 1998

UQÀM

Executive Summary

Consensus on a Core Body Knowledge Is Crucial

Software engineering has not reached the status of a legitimate engineering discipline and a recognized profession. Since 1993, the IEEE Computer Society and the ACM have been actively promoting software engineering as a profession, notably through their involvement in the Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession.

Achieving consensus by the profession on a core body of knowledge is a key milestone in all disciplines and has been identified by the Steering Committee as crucial for the evolution of software engineering toward a professional status. This report, written under the auspices of this committee, is the first step in a four-year project designed to reach this consensus.

Focus on Generally Accepted Knowledge

The software engineering body of knowledge is an all-inclusive term that describes the sum of knowledge within the profession of software engineering. Since it is usually not possible to put the full body of knowledge of even an emerging discipline, such as software engineering, into a single document, there is a need for a Guide to the Software Engineering Body of Knowledge. This Guide will seek to identify and describe that subset of the body of knowledge that is generally accepted, even though software engineers must not only be knowledgeable in software engineering, but also of course in other, related disciplines.

Guide to the Software Engineering Body of Knowledge Project

The objectives of the Guide to the Software Engineering Body of Knowledge project are therefore to:

- characterize the contents of the Software Engineering Body of Knowledge
- provide a topical access to the Software Engineering Body of Knowledge;
- promote a consistent view of software engineering worldwide;
- clarify the place of, and set the boundary of, software engineering with respect to other disciplines such as computer science, project management, electrical engineering and mathematics;
- provide a foundation for curriculum development and individual certification and licensing material.

The intended audience for the Guide to the Software Engineering Body of Knowledge includes: private and public organizations, practicing software engineers, makers of public policy, professional societies, students and educators, as well as researchers.

A three-phase approach is proposed to develop the Guide to the Software Engineering Body of Knowledge. These three phases will respectively produce the “Straw Man”, “Stone Man” and “Iron Man” versions of the Guide.

Phase 1: Straw Man Version

The objectives of the first phase are to define the strategy, to deliver what is referred to as the Straw Man version of the Guide, and to gather momentum in the profession for the project. The present report constitutes this Straw Man version.

The main goal of this initial report is to propose a list of Knowledge Areas for the Guide to the Software Engineering Body of Knowledge (SWEBOK). This report also proposes a draft list of the disciplines that interact with software engineering. As its name implies, this Straw Man version is intended to be challenged and to stimulate a vigorous debate.

Knowledge Areas are the major components of a discipline, or sub-fields of study. Related Disciplines are the other disciplines with which software engineering has a non-empty intersection or shares a common boundary.

In order to propose Knowledge Areas and Related Disciplines for "generally accepted" knowledge and to do so based on recognized, public and verifiable sources of information, it was decided that the tables of contents of general software engineering textbooks, the curricula of undergraduate and graduate programs in software engineering, and the admission criteria for graduate programs would constitute the input to our analysis. A total of 24 textbooks and 29 programs were examined.

For the purposes of this Straw Man version, a potential knowledge area had to be mentioned in the table of contents of at least one quarter of the textbooks sampled to qualify as a proposed Knowledge Area.

The ISO/IEC 12207 standard on Software Life Cycle Processes is used as the basis and vocabulary for the classification of the different topics related to the life cycle. A number of other topics not related to the life-cycle were also considered.

The list of proposed Knowledge Areas based on ISO/IEC 12207 is:

- Development Process
- Requirements Analysis
- Detailed Design
- Coding
- Testing
- Maintenance Process
- Configuration Management
- Quality Assurance
- Verification and Validation
- Improvement Process

The list of proposed Knowledge Areas that do not converge well with ISO/IEC 12207 is:

- Software Development Methods
  - Object Oriented
  - Formal Methods
  - Prototyping
- Software Development Environments
- Software Engineering Overview & Definition
- Measurement/Metrics
- Software Reliability

The list of proposed Related Disciplines is:

- Computer Science
- Project Management
- Electrical Engineering
- Mathematics
- Telecommunications/Networks
- Management
- Science
- Other Engineering Disciplines
- Cognitive Sciences

Phase 2: Stone Man Version

The deliverables of the second phase (Stone Man) under the stewardship of the Industrial Advisory Board are:

- an approved list of Knowledge Areas of software engineering;
- an approved list of topics and relevant reference materials for each Knowledge Area;
- an approved list of disciplines related to Software Engineering, and the Knowledge Areas and topics lying at the junction of Software Engineering and one or more of these Related Disciplines.

To ensure relevance of the Guide, to continue building consensus and momentum for the Guide and to encourage its quick uptake in the marketplace, three components are key to the proposed strategy of the Stone Man phase: an Industrial Advisory Board, a series of specialized subcommittees and a broad comment-gathering and consensus-building process.

The Industrial Advisory Board will include key representatives from industry, major professional societies, international standards-setting bodies and academia, as well as authors of widely sold textbooks on

software engineering. It will be responsible, among other things, for the overall strategy of the project, for the selection criteria for the Knowledge Areas and the Related Disciplines, and the selection criteria for topics included in each Knowledge Area, for the selection of the subcommittee chairs for each Knowledge Area and for promoting the Guide to the SWEBOK.

Phase 3: Iron Man Version

A subsequent Iron Man version should be completed roughly two years after the Stone Man version. The development of this version will once again probably involve an Industrial Advisory Board and various expert panels. However, an even more exhaustive review and consensus-building process to gather comments and insights from members of the profession will have to be defined for this phase of the project.

Involvement By All Parties is Critical

Many long hours of work, debate and consensus-building will be required to develop the Stone Man and subsequent Iron Man versions of the Guide to the Software Engineering Body of Knowledge. Achieving consensus on the core body of knowledge is a key milestone in all disciplines and is pivotal for the evolution of software engineering toward a professional status. Involvement by all parties, industry, professional societies, standards-setting bodies and academia, is critical to ensure the relevancy and the credibility of results, and for a quick uptake of the results.

TABLE OF CONTENTS

Acknowledgments

The authors would like to thank the following reviewers for providing us with insightful comments and suggestions based on a draft version of this document. These reviewers are in alphabetical order: Denis Bourdeau (Bell Canada), Gilles Gauthier (Université du Québec à Montréal), John Harauz (Ontario Hydro), David Longstreet (Longstreet Consulting), Stephen MacDonnell (University of Otago), Serge Oligny (Université du Québec à Montréal), Lyn VanHoozer (The MITRE Corporation), Dolores Wallace (National Institute of Standards and Technology) and Laurie Werth (University of Texas at Austin). This does not imply, however, that these reviewers or their organizations agree with the positions and proposals put forward in this document.

Funding for this work was provided by the IEEE Computer Society and the Software Engineering Management Research Laboratory of the Université du Québec à Montréal. This laboratory is supported through a partnership with Bell Canada. Additional funding for the laboratory is provided by the Natural Sciences and Engineering Research Council of Canada.

1. Introduction

In spite of the millions of software professionals worldwide and the ubiquitous presence of software in our society, software engineering has not reached the status of a legitimate engineering discipline and a recognized profession.

Since 1993, the IEEE Computer Society and the ACM have been actively promoting software engineering as a profession and a legitimate engineering discipline, notably through its involvement in the Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession. A draft version of accreditation criteria for software engineering university programs [1] and a draft Code of Ethics for software engineers [2] have already been produced.

Achieving consensus by the profession on a core body of knowledge is a key milestone in all disciplines and has been identified by the Steering Committee as crucial for the evolution of software engineering toward a professional status. This report, written under the auspices of this committee, is the first step in a four-year project designed to reach this consensus.

In other engineering disciplines, the accreditation of university curricula and the licensing and certification of practicing professionals are taken very seriously[1]. These activities are seen as critical to the constant upgrading of professionals and, hence, the improvement of the level of professional practice. Recognizing a core body of knowledge is pivotal to the development and accreditation of university curricula and the licensing and certification of professionals.

The main goal of this initial report is to propose a draft list of Knowledge Areas for the Guide to the Software Engineering Body of Knowledge (SWEBOK). This report also proposes a draft list of the disciplines that interact with software engineering. As its name implies, this Straw Man version is intended to be challenged and to stimulate a vigorous debate.

The report begins with a statement of the objectives of the project, its intended audience and the proposed three-phase development and consensus-building for producing the deliverables. Chapter 3 discusses in more detail the problem being addressed and the reasoning leading up to it, other body of knowledge proposals, as well as the intended impact of the deliverables downstream. It is followed by a description in Chapter 4 of the methodology used to identify the proposed lists of Knowledge Areas and Related Disciplines. Knowledge Areas and Related Disciplines are then proposed in Chapters 5 and 6. The report closes with some brief concluding remarks and a discussion on the next steps

[1] For a more detailed discussion on the accreditation of university engineering curricula and the licensing and certification of practicing engineers, see the websites of the Accreditation Board for Engineering and Technology at www.abet.org or the Canadian Council of Professional Engineers at www.ccpe.ca

2. The Guide to the Software Engineering Body of Knowledge Project

Body of Knowledge

The software engineering body of knowledge is an all-inclusive term that describes the sum of knowledge within the profession of software engineering. As with other professions such as law, medicine and accounting, the body of knowledge rests with the practitioners and academics who apply and advance it.

Guide to a Body of Knowledge

Since it is usually not possible to put the full body of knowledge of even an emerging discipline, such as software engineering, into a single document, there is a need for a Guide to the Software Engineering Body of Knowledge. This Guide will seek to identify and describe that subset of the body of knowledge that is generally accepted or, in other words, the core body of knowledge of the discipline.

Software engineering body of knowledge and curriculum are not the same

Software engineers must not only be knowledgeable in what is specific to their discipline, but they also, of course, have to know a lot more. The goal of this initiative is not, however, to inventory everything that software engineers should know, but to identify what forms the core of software engineering.

It is the responsibility of other organizations and initiatives involved in the licensing and certification of professionals and the development and accreditation of curricula to define what a software engineer must know outside software engineering. We believe that a very clear distinction must be made between the software engineering body of knowledge and the contents of software engineering curricula.

Project Objectives

The objectives of the Guide to the Software Engineering Body of Knowledge project are therefore to:

- characterize the contents of the Software Engineering Body of Knowledge;
- provide a topical access to the Software Engineering Body of Knowledge;
- promote a consistent view of software engineering worldwide;
- clarify the place of, and set the boundary of, software engineering with respect to other disciplines such as computer science, project management, electrical engineering and mathematics;
- provide a foundation for curriculum development and individual certification and licensing material.

Intended Audience

The intended audience for the Guide to the Software Engineering Body of Knowledge includes:

- public and private organizations wishing to use and promote a consistent view of software engineering internally, notably when defining education and training, job classification and performance evaluation policies;
- practicing software engineers wishing to enhance their professional skills;
- makers of public policy engaged in defining software engineering licensing rules and guidelines for professionals: consensus on a Guide to the Software Engineering Body of Knowledge is crucial to ensure the coherence of licensing and accreditation guidelines and policies across national and state boundaries;

- professional societies engaged in defining software engineering university program accreditation guidelines, and certification rules and guidelines for professionals;
- software engineering students learning the discipline;
- educators and trainers engaged in defining curricula and course content;
- researchers looking for an agreed-upon framework when discussing their work.

A Three-Phase Development and Consensus-Building Approach

The three-phase approach outlined in Figure 1 is proposed to develop the Guide to the Software Engineering Body of Knowledge. Total duration of the three phases is expected to be four years. These three phases will respectively produce the "Straw Man", "Stone Man" and "Iron Man" versions of the Guide.

Two principles underlie this three-phase approach:

- transparency: the development process is itself published and fully documented;
- consensus-building: the development process is designed to build, over time, consensus in industry, among professional societies and standards-setting bodies and in academia.

It is in this spirit that communication channels are constantly kept open between our project and the Joint Task Force on Software Engineering Curriculum which is also under the auspices of the Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession.

A startup phase to develop an initial version of the Guide began at the outset of 1998. The objectives of this phase are to define the strategy, to deliver what is referred to as the Straw Man version of the Guide, and to gather momentum in the profession for the project. The present report constitutes this Straw Man version. As will be described in detail in Chapter 4, the adopted methodology used for this version is based on an analysis of a large number of software engineering textbooks, undergraduate and graduate software engineering curricula and graduate admission requirements. Additionally, the framework used to analyze these textbooks and academic programs is a joint ISO/IEC and IEEE standard which has itself been adopted through a rigorous and international consensus building, review and balloting process. In essence, one can say that the Straw Man version tries to identify where there is already consensus.

The Straw Man version will serve as the primary input for the subsequent "Stone Man" phase of the project expected to end by mid-1999. The deliverables of the Stone Man phase of this project will be:

- a list of Knowledge Areas of software engineering (Knowledge Areas are the major components of a discipline, or subfields of study).
- a list of topics and relevant reference materials for each Knowledge Area;
- a list of disciplines related to Software Engineering, and the Knowledge Areas and topics at the junction of Software Engineering and one or more of these Related Disciplines; however, the Stone Man version will not point to any reference materials from a Related Discipline unless it is specifically adapted to software engineering.

To ensure relevance of the Guide, to continue building consensus and momentum for the Guide and to encourage its quick uptake in the marketplace, three components are key to the proposed strategy of this Stone Man phase. They are, as shown in Figure 2, an Industrial Advisory Board, a series of specialized subcommittees and a broad comment-gathering and consensus-building process.

The Industrial Advisory Board will include key representatives from industry, major professional societies, international standards-setting bodies and academia, as well as authors of widely sold textbooks on software engineering. The draft definition of the responsibilities for the Industrial Advisory Board consists of the following:

- Review and approve the scope and development strategy of the Guide to the Software Engineering Body of Knowledge;

- Review and approve the selection criteria for Knowledge Areas;
- Review and approve the list of proposed of Knowledge Areas;
- Review and approve the selection criteria for Related Disciplines;
- Review and approve the proposed list of Related Disciplines;
- Review and approve the selection criteria and the list of topics for each Knowledge Area;
- Review and approve the reference material selection criteria;
- Review and approve the list of subcommittee Chairs;
- Review and approve a broad comment-gathering and consensus-building process for the Stone Man version;
- Oversee the broad comment-gathering and consensus-building process for the Stone Man version;
- Assist in promoting the Guide to the Software Engineering Body of Knowledge.

A number of subcommittees made up of subject matter experts will be established during the Stone Man phase and these will be responsible for selecting key reference material in the existing software engineering literature based on predefined reference selection criteria. These references could be book chapters, journal articles, public reports from industry, etc. The incorporation of these subcommittees in the design of the approach is an additional step in building consensus.

To inform software engineering professionals about the Guide, to promote it, to continue building consensus and to gather comments from a broad sample of professionals, a broad comment-gathering and consensus-building process will also be completed electronically during the Stone Man phase among the membership of the Computer Society and possibly other professional societies. The Industrial Advisory Board will ensure that due process is followed regarding this consultation and comment-gathering step.

A subsequent Iron Man version should be completed roughly two years after the Stone Man version. The development of this version will once again probably involve an Industrial Advisory Board and various expert panels. However, an even more exhaustive review and consensus-building process to gather comments and insights from members of the profession will have to be defined for this phase of the project. This review and consensus-building process should be somewhat akin to the already existing software engineering standards development and review process.

To facilitate its wide dissemination, all versions will be available at no cost on the Internet.

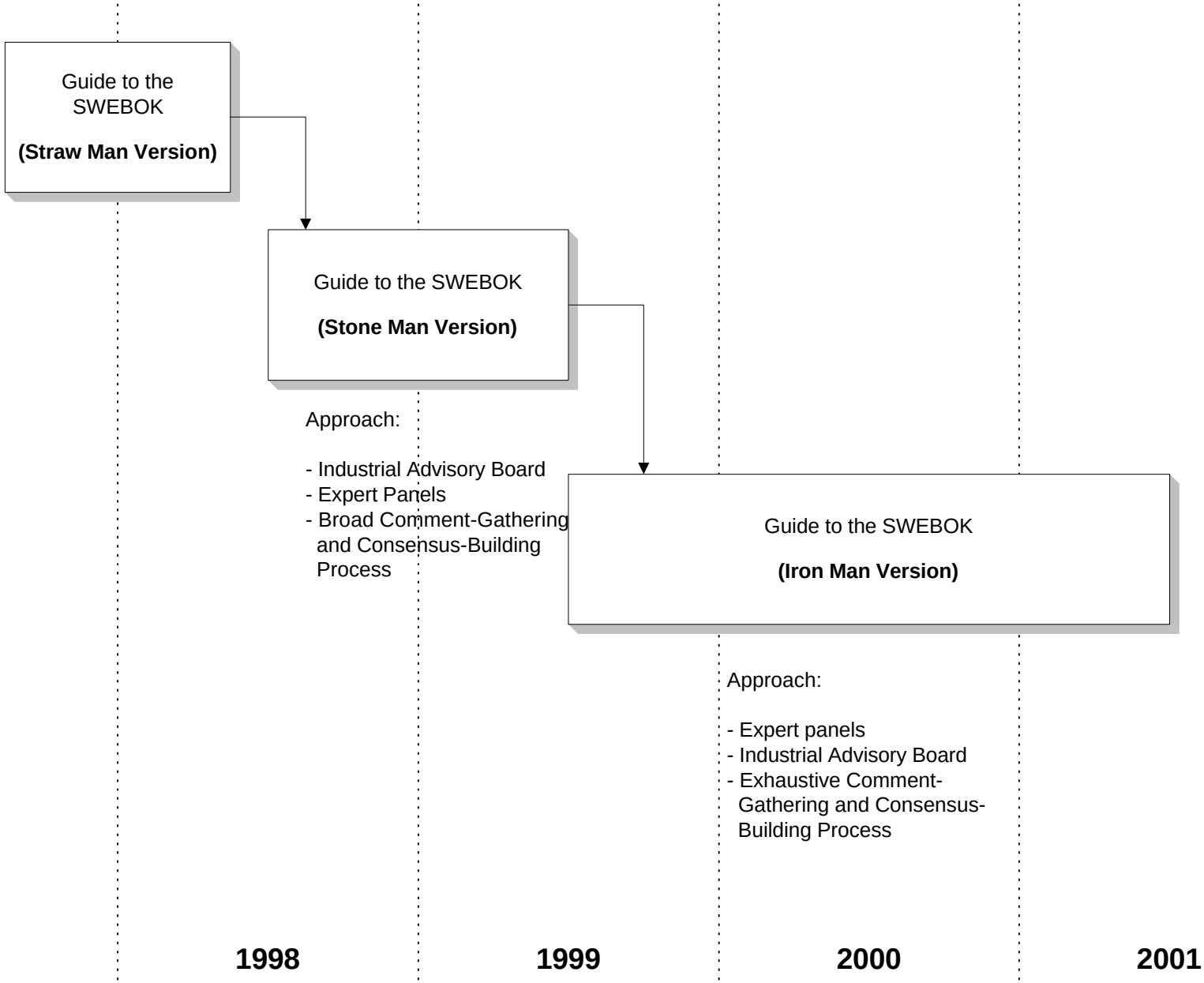


Figure 1 A Three-Phase Approach for Developing the Guide to the SWEBOK

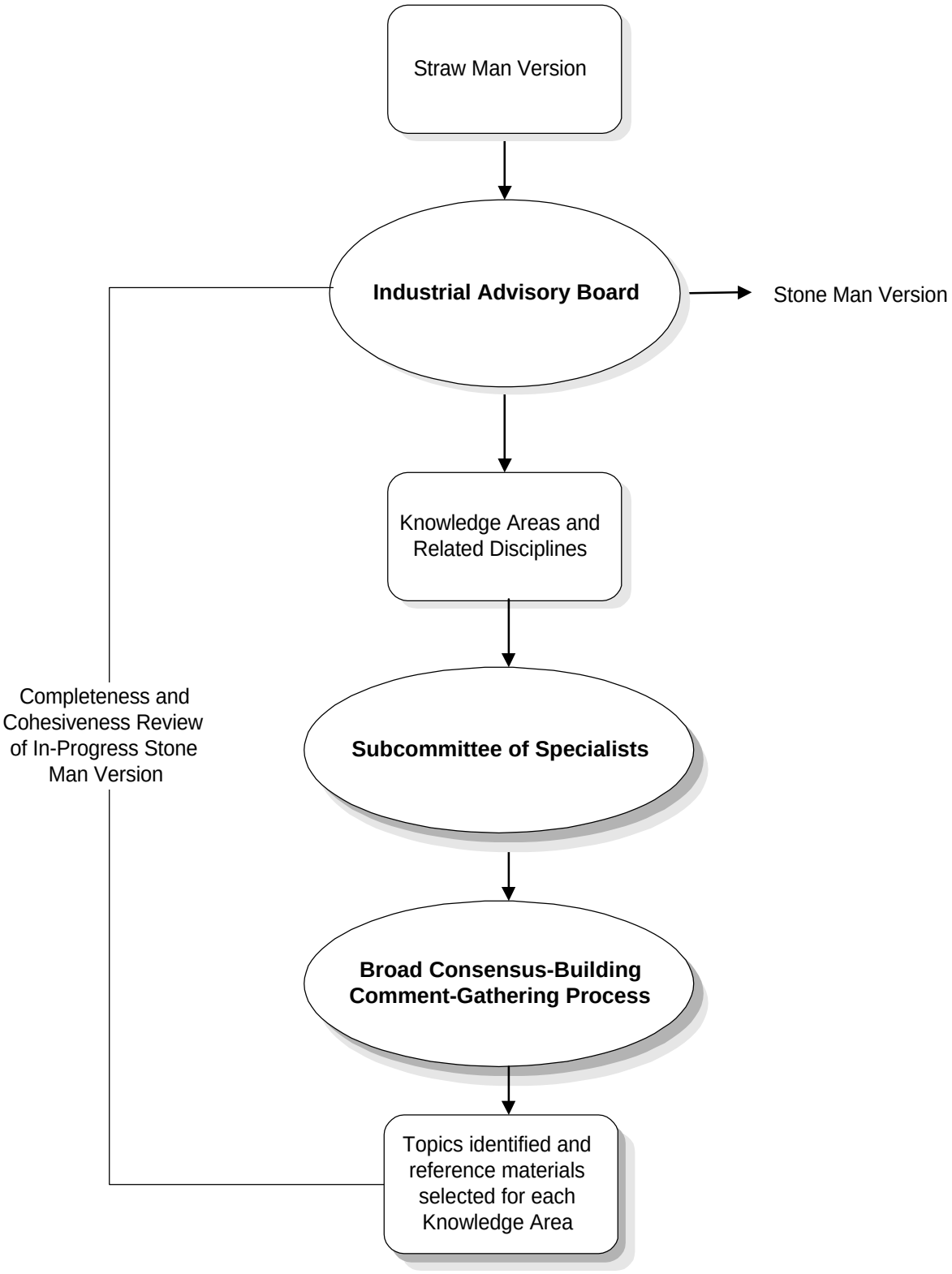


Figure 2 Proposed Strategy for Developing the Stone Man Version

3. Context and Relationships

What is software engineering?

The IEEE Computer Society defines software engineering as[2]:

> "(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)." [3]

What is a recognized profession?

For software engineering to be known as a legitimate engineering discipline and a recognized profession, consensus on a core body of knowledge is imperative. This fact is well illustrated by Starr [4] when he defines what can be considered a legitimate discipline and a recognized profession. In his Pulitzer-prize-winning book on the history of the medical profession in the USA, he states that:

> "the legitimation of professional authority involves three distinctive claims: first, that the knowledge and competence of the professional have been validated by a community of his or her peers; second, that this consensually validated knowledge rests on rational, scientific grounds; and third, that the professional's judgment and advice are oriented toward a set of substantive values, such as health. These aspects of legitimacy correspond to the kinds of attributes — collegial, cognitive and moral — usually cited in the term "profession."[3]

The software engineering profession is still immature

The term "software engineering" has now been in use for 30 years, since it was officially coined at an October 1968 conference held in Garmisch, Germany [5]. Since then, considerable progress has been made. Evidence of this progress can be found in the list of 24 general software engineering textbooks found in Appendix A. Additionally, Appendix B lists 5 undergraduate and 24 graduate programs now being offered in software engineering and that were found described on the World Wide Web. A multitude of conferences and workshops are given on the topic of software engineering yearly. As well, the discipline has now accumulated a significant number of national and international standards [6].

This progress does not, of course, imply that software engineering is, as currently practiced by individuals or by organizations, at a level sufficient to ensure consistent and reliable outcomes. The industry is still plagued by significant cost and schedule overruns. Unreliable software continues to be delivered, often with dire consequences. Projects are regularly canceled or deliver only a subset of the expected benefits. Maintenance costs, best exemplified by the Year 2000 bug, are very often prohibitive .

In 1996, Ford and Gibbs [7] wrote an in-depth report on the level of maturity of the software engineering profession. In order to discuss the maturity of a profession in a more objective and constructive manner

---

[2] Of course, there are many other definitions of software engineering. Since this effort originates from a joint committee of the ACM and the IEEE Computer Society and since this definition was agreed upon by a wide consensus within the Computer Society, it seems reasonable to start from it. The Industrial Advisory Board may find it inadequate for the purposes of the Guide to the Software Engineering Body of Knowledge or this definition may prove to be insufficient later on in the project

[3] p. 15.

and to better predict its future evolution, they begin by proposing a model of its maturity in terms of eight infrastructure components. These components are:

- Initial professional education system;
- Accreditation of professional education programs;
- Skills development mechanisms for professionals entering the practice;
- Certification of professionals administered by the profession;
- Licensing of professionals administered by government authorities;
- Professional development programs to maintain currency of knowledge and skills;
- Code of ethics;
- Professional society or societies.

Their report states that nearly all these components have existed for many years and are being continually improved for more established professions such as medicine, law, engineering, architecture and accounting. They then analyze the software engineering profession using this eight-component taxonomy and conclude that only the professional development and professional society components have advanced past the ad hoc level. They therefore infer that the software engineering profession is still immature.

Increasing interest in program accreditation, certification and licensing

There is without any doubt increasing interest in university program accreditation and the licensing and the certification of software professionals. The Cutter IT Journal published by Ed Yourdon recently devoted an entire issue to the certification and licensing of software professionals [8]. In 1996, the Institution of Engineers, Australia, began granting full accreditation to undergraduate software engineering programs [9]. Some authors have even stated recently that we in the software industry had better take these issues very seriously, otherwise government officials will do it themselves. The following citations from these authors illustrate their point of view well:

> "If the profession does not provide an effective mechanism such as certification to assure that its practitioners are doing everything possible to promote safety and security, then government will try to do it with licensing." [7]
>
> "In my opinion, the licensing or certification of at least some software engineering specialties (e.g. safety-critical systems, secure systems) is inevitable. In the current climate, licensing will probably emerge first. The only decision that we need to make is whether we want to be part of the solution or part of the problem" [10]
>
> "If the software community cannot organize itself to become a recognized profession, we will have this done for us by legislatures and others without the necessary technical expertise and understanding of the issues" [11]
>
> "If the software engineering community cannot rise to the level of becoming a recognized profession and an engineering discipline, we face an uncertain future with ever-mounting prospects of unfriendly legislation and harmful government actions." [12]
>
> "… but society might believe that severe regulation and licensing of software activities are the only way to avoid a repetition of the Year 2000 catastrophe ."[13]

It is important to note that on June 17, 1998, the Texas Board of Professional Engineers unanimously approved a proposal to recognize software engineering as a legitimate engineering discipline and to begin licensing professional engineers in this area[4].

Consensus on a core body of knowledge is an inescapable first step

Figure 3 shows that to correctly address the development of software engineering curricula, the accreditation of professional education programs and the licensing and certification of professionals, consensus by the profession on a core body of knowledge is an inescapable step. The necessity of a consensus on a core body of knowledge when discussing professional education program accreditation and the licensing or certification of professionals is well illustrated by these two citations:

> "The discipline of software engineering is still immature, but pressures from regions where engineers are licensed will add urgency to this issue. Clearly, some judgment about core material is required to perform an accreditation…" [9]
>
> "If we accept that licensing is inevitable, then we believe it is important that the profession be prepared to advise the state legislatures about the nature of software engineering and the appropriate contents of a licensing examination." [7].

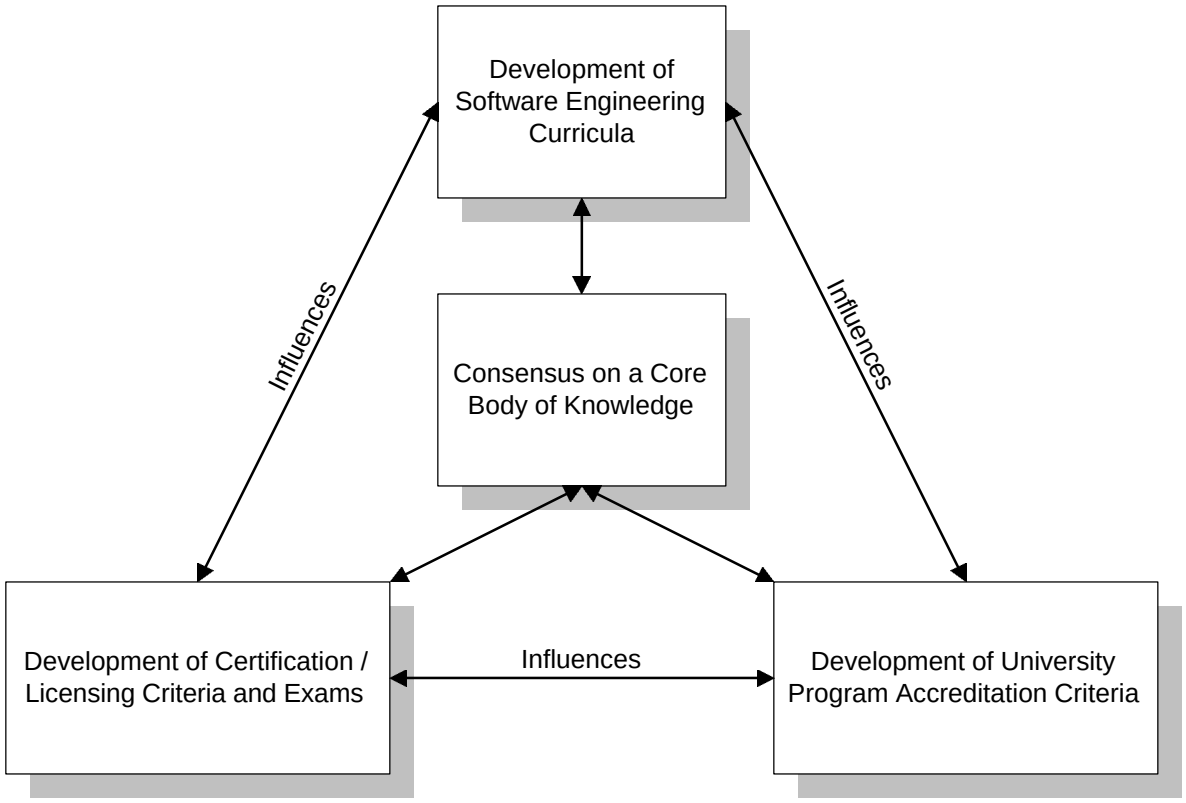


Figure 3 Key interrelationships for a core body of knowledge

[4] See http://www.main.org/peboard/sofupdt.htm

Computer science is the underlying discipline of software engineering

History has shown that a professional engineering discipline emerges when there is a sufficient scientific basis to enable a core of educated professionals to apply not only craft, experience and skill, but also theory to the analysis of problems and synthesis of solutions [14]. For most engineering disciplines, this emergence occurred in the 18th and 19th centuries with the increased scientific understanding of our physical world. Based on this criterion, Shaw [14] argues that, though software engineering has not yet matured to the state of a professional engineering discipline, this is an achievable goal. Baber [15] argues that software engineering is currently in a "pre-engineering" phase of its development, in many ways similar to the "pre-engineering" phases of shipbuilding, bridge construction and electrical technology. The availability and regular use by professionals of predictive models that have a scientific and mathematical basis is a distinguishing characteristic of "engineering" from "pre-engineering" practice.

Computer science has also evolved significantly over the past decades. Advances in the areas of algorithm design, compilers, data structures, database management systems, operating systems and programming languages, among others, testify to the ever-increasing depth and breadth of knowledge in computer science.

Parnas [16] is of the opinion that it is notably because of the maturity of computer science that we can now offer software engineering university programs. In fact, he argues that due to distinct fundamental goals, it is in the interests of both communities to separate the disciplines. Precedents for this position have been established in other engineering disciplines, such as in the separation of physics and electrical engineering.

When discussing the relationship of software engineering to its underlying science, Maibaum [17] states:

> "It is clear that the important symbiotic relationship between analysis, physics and engineering that we have experienced over more than 200 years will be repeated in the next century between logic, theoretical computer science and software engineering."

Distinct fundamental goals of computer science and software engineering

The fundamental goals of computer science and software engineering differ, as do the fundamental goals of science and engineering.[5] Science as a whole seeks to better understand and explain various phenomena. In essence, knowledge is the product of science. In his seminal book entitled "What Engineers Know and How They Know It" [20], Vincenti declares "For engineers, in contrast to scientists, knowledge is not an end in itself or the central objective of its profession."[6] He then goes on to say that, for engineering, science is "a means to a utilitarian end." Brooks describes this difference in goals very clearly by stating: "A scientist builds in order to learn; an engineer learns in order to build."[7] In essence, artifacts rather than knowledge are therefore the product of engineering, be they bridges, ships, airplanes, oil refineries, computer chips or software.

An illustration of this difference in goals is the importance given to professional education program accreditation in engineering and in science. On this issue, Parnas [16] states:

> "The work of scientists will be usually judged by other scientists, but engineers often deal with customers who are neither engineers nor scientists. Thus, while nobody has ever felt it necessary to hold science

[5] As one can debate the true engineering underpinnings of software engineering by discussing, for instance, its use of quantitative methods, one could also debate the true scientific underpinnings of computer science by discussing, for instance, its use of the scientific method. This report will leave these worthy debates to others. For an excellent discussion of the application of the scientific method in computer science and in software engineering, see [18] and [19].

[6] p. 6

[7] Cited on p. 21 of [7].

> programmes to rigid standards, accreditation has always been an important consideration for engineering programmes."

Computer science therefore seeks to better understand and to extend our knowledge in the area of computing. Extending our knowledge of software or computing is not the fundamental goal of software engineering, but rather applying this knowledge to building software.

Shaw [14] declares that although there are many definitions of engineering they all share these common elements: creating cost-effective solutions to practical problems by applying scientific knowledge to building things in the service of mankind. She then states that, for software, the problem is, appropriately, an engineering problem.

The definition given by the ACM/IEEE Computer Society Task Force on the Core of Computer Science for computing [21] is different from the definition of software engineering[8]. The Task Force states that:

> "The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation and application. The fundamental question underlying all of computing is "What can be (efficiently) automated?"

Engineering is much more than applied science

Vincenti in [20] argues at length that engineering is much more than applied science. The following quote from the opening paragraph of his book illustrates his argument:

> "Engineering knowledge, though pursued at great length and expense in schools of engineering, receives little attention from scholars in other disciplines. Most such people, when they pay heed to engineering at all, tend to think of it as applied science. Modern engineers are seen as taking over their knowledge from scientists and, by some occasionally dramatic but probably intellectually uninteresting process, using this knowledge to fashion material artifacts. From this point of view, studying the epistemology of science should automatically subsume the knowledge content of engineering. Engineers know from experience that this view is untrue…"[9]

Vincenti categorizes the elements of engineering design knowledge in the following manner:

- Fundamental design concepts;
- Criteria and specifications;
- Theoretical tools;
- Quantitative data;
- Practical considerations;
- Design instrumentalities.

He then goes on to classify engineering knowledge-generating activities into seven categories, of which only one is directly linked to the underlying science. The categories are:

- Transfer from science;

---

[8] As cited on p. 8, the IEEE Computer Society definition of software engineering is "(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)."

[9] p. 3

- Invention;
- Theoretical engineering research;
- Experimental engineering research;
- Design practice;
- Production;
- Direct trial.

These two lists illustrate well that an engineering discipline has a body of knowledge different from the body of knowledge of its underlying science. They obviously interact heavily and often have overlapping content. However, their respective categories of knowledge and types of knowledge-generating activities differ greatly. As discussed in Chapter 7, an adaptation of Vincenti's classification schema for engineering design knowledge is proposed as a common framework for structuring topics and reference materials within Knowledge Areas.

Other body of knowledge proposals

A number of groups, professional societies and individuals have proposed a number of views regarding the software engineering body of knowledge. These proposals are described in more detail in Appendix J.

The Joint Steering Committee of the IEEE Computer Society and the ACM for the Establishment of Software Engineering as a Profession established a task force in 1996 to conduct exploratory work on the issue of a software engineering body of knowledge. The task force designed and conducted a pilot survey on a sample of tasks that could be considered to be within the scope of software engineering[10]. The survey asked whether each task described would be expected to be performed by a "novice software engineer", an "expert software engineer", a "software engineering specialist" or a "manager" in the organization.

Certain proposals are incorporated into certification programs, either for a broader field as is the case with the Certified Computing Professional program of the Institute for Certification of Computer Professionals, or a more specialized field related to software engineering such as the Software Quality Engineers program of the American Society for Quality, and the Certified Quality Analyst and Certified Software Test Engineer programs of the Quality Assurance Institute.

Other proposals are being made within the context of developing software engineering curricula. Parnas for instance, while describing a new undergraduate program in the field, proposes a list of knowledge areas related to tasks performed by software engineers. The Working Group on Software Engineering Education and Training, which includes members from industry and academia, proposed a set of guidelines last spring for software education which included their view of the software engineering body of knowledge areas and knowledge components. The Australian Computer Society also includes 'Software Engineering and Methodologies' as a Knowledge Area within its Core Body of Knowledge for Information Technology Professionals. Finally, a collaborative effort of the ACM and other organizations recently published a model for undergraduate degree programs in information systems entitled IS'97 which included many software engineering elements.

These proposals regarding the software engineering body of knowledge cannot be used in their totality either, because:

- the focus is more on curriculum development than on the core body of knowledge of software engineering itself;
- the focus is not directly on software engineering but rather on perhaps broader or narrower disciplines such as computing, information technology, information systems, software quality engineering and test engineering;
- the consensus building process is not documented.

[10] The report on the survey's results can be found at computer.org/tab/seprof/survey.htm

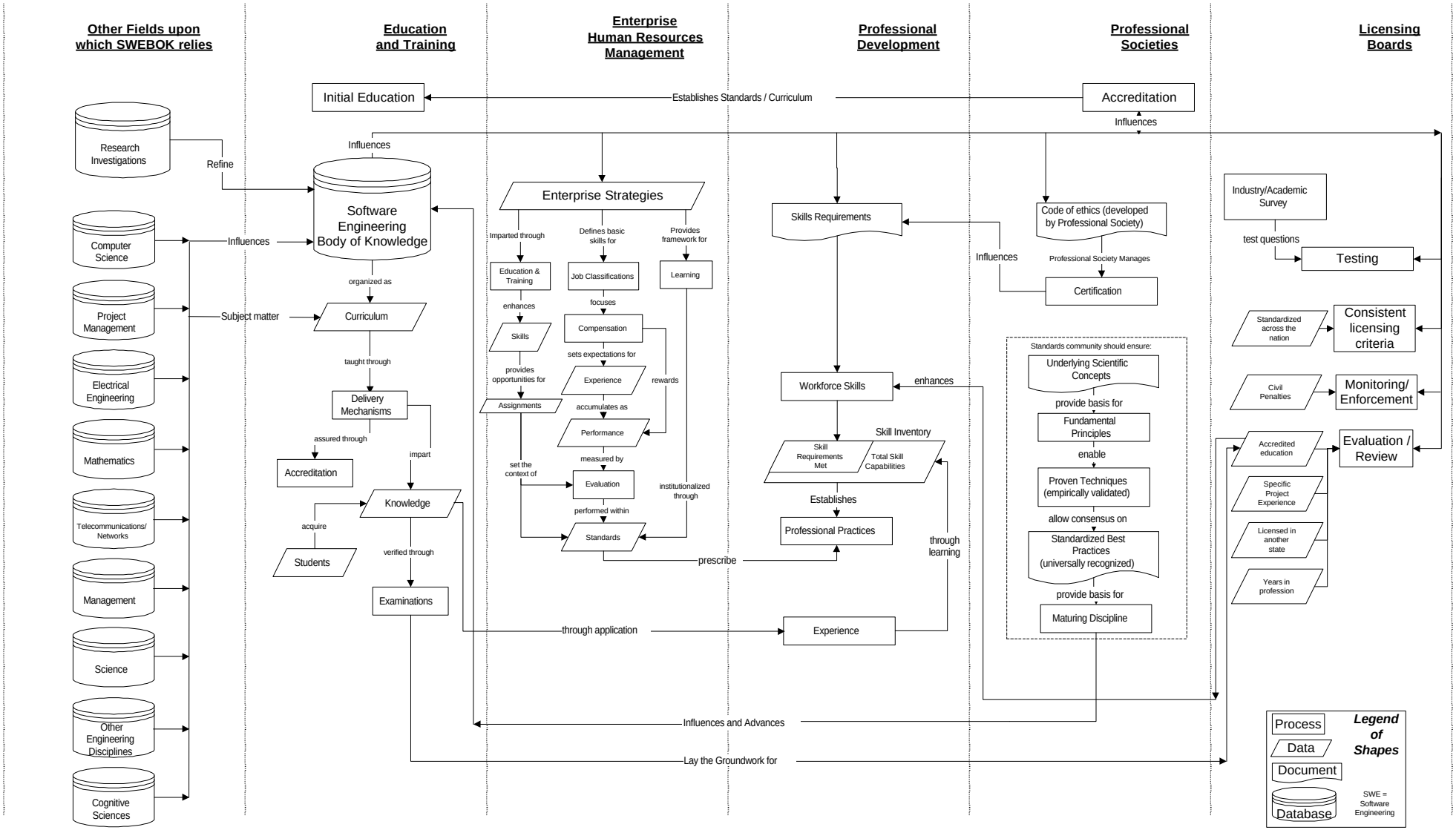
However, these proposals are an excellent input to the Industrial Advisory Board to ensure the soundness of the list of Knowledge Areas and Related Disciplines proposed in the Straw Man version of the Guide to the Software Engineering Body of Knowledge. Additionally, they should be examined when identifying topics and selecting reference materials for the Stone Man version.

Consensus on a body of knowledge is a must

The Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession has recognized that the body of knowledge of the emerging software engineering discipline, as for other disciplines of engineering, is an autonomous body of knowledge distinct from that of computer science. This committee has identified that consensus by the profession on this body of knowledge is key to the maturation of the discipline, and improvement in the level of professional practice.

Offering a much more detailed and complete view, Figure 4 enables us to understand better how this Guide to the Software Engineering Body of Knowledge may eventually impact education and training, enterprise human resources management, professional development, professional societies and licensing boards in the field of software engineering.

The Interrelationships of the Software Engineering Body of Knowledge Stakeholders Context Diagram



4. Development Methodology for Identifying Knowledge Areas and Related Disciplines

Introduction

This chapter describes the methodology used for identifying the Knowledge Areas and Related Disciplines. Knowledge Areas are the major components of a discipline, or sub-fields of study. Related Disciplines are the other disciplines with which software engineering has a non-empty intersection or shares a common boundary.

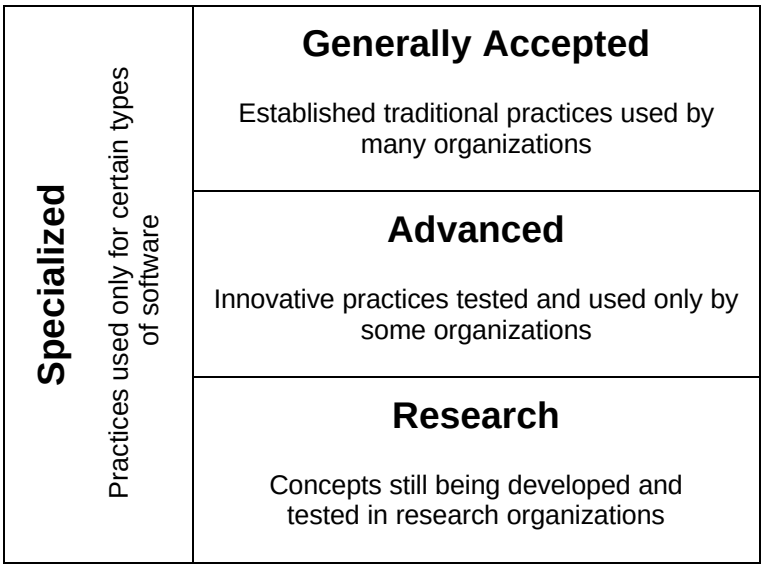
Criteria used in selecting our identification methodology

The following criteria were used in defining the methodology for identifying Knowledge Areas and Related Disciplines:

- The identification methodology had to be based on public and verifiable sources of information and had to follow a well-documented and reproducible procedure. The authors have tried to make as few editorial decisions as possible.
- The identification methodology had to be as inclusive as possible. For this Straw Man version, it was deemed better to suggest too many Knowledge Areas and Related Disciplines and for them to be abandoned later than the reverse.

Focus is on generally accepted knowledge

As stated earlier, the software engineering body of knowledge is an all-inclusive term that describes the sum of knowledge within the profession of software engineering. However, the Guide to the Software Engineering Body of Knowledge seeks to identify and describe that subset of the body of knowledge that is generally accepted or, in other words, the core body of knowledge. To better illustrate what "generally accepted knowledge" is relative to other types of knowledge, Figure 5 proposes a draft four-category schema for classifying knowledge. As an example, Appendix I proposes a classification of the knowledge on formal methods based on this schema.



» Figure 5 Categories of knowledge in the SWEBOK

The Project Management Institute in its Guide to the Project Management Body of Knowledge[11] [22] defines "generally accepted" knowledge for project management in the following manner:

> "Generally accepted means that the knowledge and practices described are applicable to most projects most of the time, and that there is widespread consensus about their value and usefulness. Generally accepted does not mean that the knowledge and practices described are or should be applied uniformly on all projects; the project management team is always responsible for determining what is appropriate for any given project."[12]

[11] This guide is currently adopted as an IEEE standard. See Chapter 7 of [6].

[12] p. 3

Knowledge Area and Related Discipline identification methodology

In order to propose Knowledge Areas and Related Disciplines for "generally accepted" knowledge and to do so based on recognized, public and verifiable sources of information, it was decided that the tables of contents of general software engineering textbooks, the curricula of undergraduate and graduate programs in software engineering, and the admission criteria for graduate programs would constitute the input to our analysis. Certainly, no one can question that general textbooks and academic curricula are an excellent source of information for better understanding any discipline.

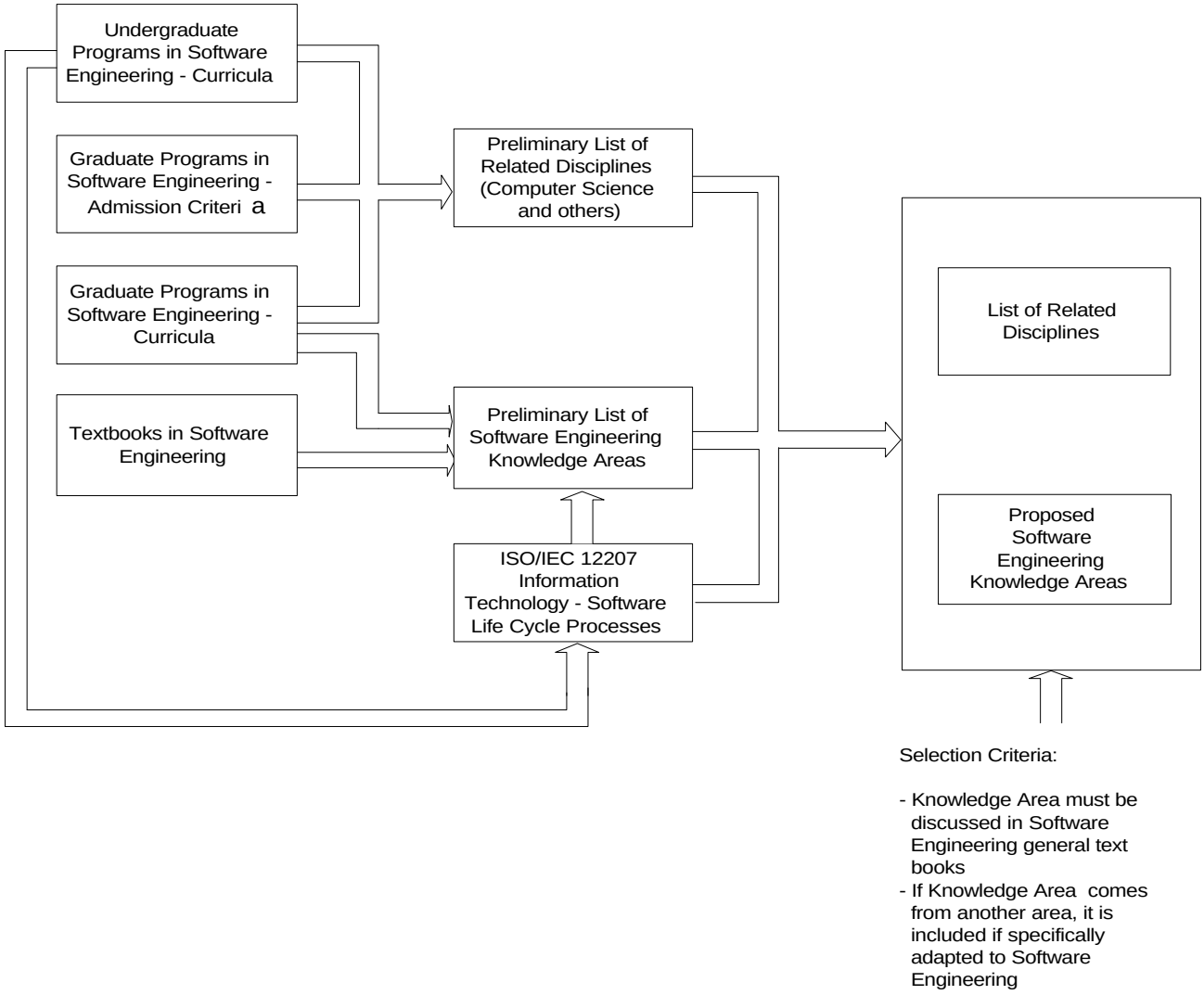


Figure 6 Knowledge Area and Related Discipline Identification Methodology

In fact, it is proposed that a Knowledge Area must be discussed in a significant number of general software engineering textbooks to be considered as "generally accepted".

General software engineering textbooks may not be the only source of generally accepted knowledge. However, textbooks are expected to contain a synthesis of what is currently considered to be the best thinking in a given field. The Industrial Advisory Board will decide if this collection of documents is sufficient or if a wider spectrum should be considered. Additionally, the various subcommittees involved in producing the Stone Man version will certainly not limit themselves to general software engineering textbooks when identifying topics and selecting reference materials within each Knowledge Area.

Additionally, it is proposed that if a Knowledge Area and subsequent subtopics are related to a discipline other than software engineering, they have to be specifically adapted to software engineering to be included in the Guide to the Software Engineering Body of Knowledge. For example, statistics are used in software engineering, but there is no specific type of statistics for this discipline. By contrast, project cost estimation is different in software engineering from that in electrical engineering. Another example would be formal methods, which have a strong relationship with computer science and even mathematics, but are specifically created to solve software engineering problems.

The general methodology used to identify the software engineering Knowledge Areas and Related Disciplines is illustrated in Figure 6. The left-hand portion shows that information was first gathered from recognized, public and verifiable source: general software engineering textbooks and academic programs. The middle portion shows that this material was synthesized into lists of Knowledge Areas and Related Disciplines. The right hand portion shows that these were organized using another well recognized standard, ISO/IEC 12207 [23], wherever applicable.

Collected information

Using the Internet, the following information was collected:

- The tables of contents of general textbooks[13] in software engineering, which present the authors' opinions on what the boundary of software engineering is and on how to classify the topics into candidate Knowledge Areas. However, these books seldom explicitly identify the Related Disciplines, even though these implicitly reveal the authors' opinions of where the discipline ends.
- The curricula of undergraduate and graduate programs in software engineering, which are another public source of information for identifying Knowledge Areas. They also provide an excellent basis for setting the boundary of software engineering and identifying the Related Disciplines. Software engineering curricula include not only courses in software engineering, but also courses in the other disciplines in which a software engineer should be educated. This is especially true at the undergraduate level. Graduate programs are generally much more focused on the discipline itself. Information on compulsory and elective courses was collected separately, since it was believed that compulsory courses would be a better basis for identifying the core body of knowledge than electives and because of the widely varying nature of elective courses due notably to which faculty and department offered the program.
- Admission criteria to graduate software engineering programs also indicate what the institutions think students should know outside software engineering. This information is useful for identifying the Related Disciplines and for setting the boundary of software engineering.

A total of twenty-four (**24**) general textbooks, five (**5**) undergraduate programs and twenty-four (**24**) graduate programs in software engineering were found and examined. For the undergraduate and graduate programs, only those having the list of required courses on the website were retained. There is no reason to believe that there would be any substantial differences between programs which have a website and those which don't. These programs are offered by universities in the United States, Canada, the United Kingdom, Australia and Sweden. Appendix A lists the general textbooks used for this report and Appendix B lists the URLs of the retained undergraduate and graduate programs.

ISO/IEC 12207

Initially, it was expected that various approaches – even paradigms – would be found by analyzing the tables of contents of general software engineering textbooks, but such was not the case. It was found that textbooks generally present most of the subject matter of software engineering around a life-cycle model. Often, more advanced material or material pertinent to the entire life cycle and not to one particular phase is presented in additional chapters.

[13] These table of contents were gathered from www.amazon.com

However, since these textbooks do not necessarily share a common life-cycle model, it was decided that the ISO/IEC 12207 standard on Software Life Cycle Processes [23] be used as the basis and vocabulary for the classification of the different topics related to the life cycle[14].

ISO/IEC 12207 was chosen for the following reasons:

- It is considered the key standard regarding the definition of life cycle process and has been adopted by the two main standardization bodies in software engineering: ISO/IEC JTC1 SC7 and the IEEE Computer Society Software Engineering Standards Committee[15].
- It has been designated as the pivotal standard around which the Software Engineering Standards Committee (SESC) is currently harmonizing its entire collection of standards [6].
- It is designed to be independent of any specific software development method or life-cycle model. Regarding ISO/IEC 12207, Moore states in [6] that:

> "The standard is intended to be independent of development technologies and methodologies and useful for any form of life cycle model, for example, waterfall, incremental, spiral, etc. In fact, one of the specified responsibilities of the supplier's role is to select the life cycle model and map the requirements of the standards to that model."[16]

- It covers the entire life cycle from concept to retirement.
- It provides roles for the acquirer, supplier, developer, maintainer and operator.

Intermediate Steps: Inventory of prepared tables

In the course of writing this report, several tables were prepared:

- *General textooks on software engineering*. First, a list of the topics covered by the various authors was produced. The majority of the books present the different topics using a software life-cycle approach. Within each category the specific topics were listed according to the number of books covering the given topic. The resulting table is presented in Appendix C.
- *Undergraduate and graduate programs in software engineering*. First, a list of the courses offered by the different programs was produced. A differentiation between required courses (those which the institutions consider to be the core knowledge) and optional courses was made. The courses were then classified according to two criteria: the software life-cycle processes, as described by ISO/IEC 12207, and the disciplines to which the courses were related. A total of five (5) tables were produced:
  - Undergraduate Programs in Software Engineering - Classification of Courses According to Potential Knowledge Areas in Appendix D.

---

[14] The authors are aware that ISO/IEC TR-15504 Information Technology Software Process Assessment also defines life cycle processes. However, ISO/IEC 12207 was preferred over ISO/IEC 15504 for the purposes of this Straw Man version since it has been adopted by both IEEE/EIA and ISO/IEC and because it has the status of a standard while ISO/IEC TR-15504 has the status of a Technical Report Type 2. However, the Industrial Advisory Board may wish to consider as potential Knowledge Areas the following additional processes defined in 15504 but not included in 12207[24]:

- Primary processes: Requirements elicitation process
- Support processes: Measurement process
  Reuse process
- Organizational processes: Quality management
  Risk management process
  Organizational alignment process

[15] IEEE/EIA is an adaptation of ISO/IEC 12207 with the same number and name.

[16] p. 197

- Undergraduate Programs in Software Engineering -
  Classification of Courses by Related Discipline in Appendix E
- Graduate Programs in Software Engineering -
  Admission Requirements by Related Discipline in Appendix F
- Graduate Programs in Software Engineering -
  Classification of Courses According to Potential Knowledge Areas
  in Appendix G
- Graduate Programs in Software Engineering -
  Classification of Courses by Related Discipline in Appendix H.

5. Proposed Knowledge Areas

Using the methodology described in the previous chapter, a compilation of topics included in the tables of contents of general textbooks and in university software engineering curricula is presented in Tables 1, 2 and 3. These tables, compiled from Appendix C, Appendix D and Appendix G, show the number of textbooks that cover a given topic at the *Table of Contents* level, and the number of programs that include required and elective courses on this topic.

As stated earlier, a proposed Knowledge Area must be covered in a *significant number* of textbooks to be considered as "generally accepted". For the purposes of this Straw Man version, this *significant nu*mber is set at 6, or one quarter of the textbooks listed in Appendix A. Potential Knowledge Areas meeting this requirement and that converge well with the ISO/IEC 12207 standard are shaded in Table 1.

However, a number of topics do not converge well with ISO/IEC 12207. The list of potential Knowledge Areas that do not converge well with the ISO/IEC 12207 standard but that meet the requirement for "generally accepted" are shown in Table 2. The list of potential Knowledge Areas that do not converge well with the ISO/IEC 12207 standard and that do not meet the requirement for "generally accepted" are shown in Table 3.

Since university programs of different types (undergraduate and graduate, professional and research, etc.) were surveyed, it was decided not to include this information in the selection criterion for considering a Knowledge Area as "generally accepted". Additionally, a limited number of courses are often offered in one program, especially the graduate level. This information, however, may be useful to the Industrial Advisory Board in their review and approval of Knowledge Areas.

The following two elements where taken into consideration in setting the *significant* number at 6, or one-quarter of the textbooks:

- As discussed earlier, the identification methodology had to be as inclusive as possible and it was deemed better to suggest too many Knowledge Areas than too few.
- many topics are covered in a textbook without them being included in the table of contents; a more detailed analysis of the textbooks would surely be most insightful and would better represent each book.

Also, these limitations must be kept in mind in interpreting the proposed list of Knowledge Areas:

- The survey of the textbooks only considered tables of contents of textbooks written in English and accessible through the Internet. This means that it is quite possible that many excellent textbooks not listed on the website of the online library were omitted from this analysis, especially textbooks in languages other than English. This is because the objective here was not to exhaustively survey all general software engineering textbooks, but rather to collect a representative sample of them.
- Many excellent university programs not found or not described on the Internet have surely been omitted from this analysis, especially programs taught in languages other than English.
- Analysis of university software engineering programs was based on course titles only. Once again, a more detailed analysis of the course syllabuses would surely be most insightful.
- There is occasionally overlap, and some topics belonging to more than one category are counted more than once. For instance, Formal Methods is sometimes presented in the appendices as a separate topic, and sometimes it is included within the life-cycle classification (e.g. Formal methods/specification languages, Object-oriented topics, etc.). This information is presented this way to better evaluate the coverage of these topics not to create redundancy.
- The importance of some specialized topics (e.g. Fault-tolerant software, Real-time software, etc.) may be underestimated since the analysis is based on general software engineering textbooks.

Process Class	Life Cycle Processes and Activities	Textbooks (24)	Programs(29) with Required Courses[17]	Programs(29) with Optional Courses
Primary	**Software Acquisition**	2		2
	Software Supply			
	Development Process		11 (general)[18]	9 (general)
	Requirements Analysis[19] [20]	22	22	10
	Architectural Design	2		
	Detailed Design[21]	23	14	14
	Coding	18	4	12
	Integration	4		
	Testing	16	9	7
	Installation	3		
	Acceptance Support	3		
	Operation Process			
	System Operation	2		
	User Support			
	Maintenance Process[22]	14	4	5
Supporting	**Documentation**			
	Configuration Management	10		
	Quality Assurance	15	11	7
	V&V	12	9	7
	Joint Review	5		
	Audits	3		
	Problem Resolution Processes			
Organizational	**Management Process**	20	20	10
	Infrastructure Process			
	Improvement Process	16	5	2
	Training Process			

Table 1 Proposed Knowledge Areas based on ISO/IEC 12207[23]

[17] Includes 24 graduate and 5 undergraduate programs

[18] Courses discussing the development process in general

[19] The activity entitled Process Implementation is omitted from this table since no book chapters or courses refer directly to this activity.

[20] Since in this report we did not wish to engage in the worthy debate of distinguishing "systems engineering" from "software engineering", the activity entitled "systems requirements analysis" is not listed in this table.

[21] Please note that many of the topics in the tables of contents and course titles that we assigned to detailed design could arguably be assigned to architectural design. A more detailed analysis of the textbook chapters themselves and the course syllabuses would enable a better assignment of these topics and often resolve the differences in vocabulary.

[22] The material in the textbooks is never organized according to the ISO/IEC 12207 classification of activities for maintenance and therefore no analysis is performed at the activity level for maintenance.

[23] Proposed Knowledge Areas considered as generally accepted are shaded. A proposed Knowledge Area must be covered in a *significant number* of textbooks to be considered as "generally accepted". For the purposes of this Straw Man version, this *significant nu*mber is set at 6, or one quarter of the textbooks listed in Appendix A.

Potential Knowledge Areas	Textbooks (24)	Programs(29) with Required courses	Programs(29) with Optional courses
Software Development Methods	14		2
Object Oriented	14	4	14
Formal Methods	9	11	7
Prototyping	9		
Software Development Environments	13	1	3
Software Engineering Overview & Definition	11	11	5
Measurement/Metrics	9	4	9
Software Reliability	6	1	5

» Table 2 Potential Knowledge Areas for non-ISO 12207 Topics That Meet the Selection Criteria for "Generally Accepted"

Potential Knowledge Areas	Textbooks (24)	Programs(29) with Required courses	Programs(29) with Optional courses
Software Products	5		
Software Reuse	5	2	4
Real-Time/Embedded Software	4	2	7
Reengineering	4	2	
Human Factors	3	5	10
Standards	3		
Fault-Tolerant Software	2		
Ethics	1	1	
Legal Aspects			2
Software Security/Safety		2	7

» Table 3 Potential Knowledge Areas for non-ISO 12207 Topics That Do Not Meet The Selection Criteria for "Generally Accepted"

6. Proposed Related Disciplines

Based on a synthesis of the courses taught in undergraduate and graduate programs in software engineering and on the admission criteria for graduate programs, Table 4 proposes a list of Related Disciplines for software engineering. The complete list of courses and the details of the admission criteria are listed in Appendix E, Appendix H and Appendix F.

Table 4 is sorted in order of number of required courses in the discipline, then by number of graduate programs requiring knowledge in the discipline as an admission criterion, followed by the number of programs containing optional courses in the discipline.

Table 4 shows a strong bias toward Computer Science in the list of elective courses. This is probably explained by the fact that most software engineering programs are offered by Computer Science departments. It follows that the electives offered are very often a subset of the courses offered by these departments. Consequently, the relatively small number of courses in electrical engineering and in "other engineering disciplines" is probably due to the fact that few of these programs are taught in engineering schools.

When interpreting the list of proposed Related Disciplines, the reader must always keep in mind the following limitations:

- Many excellent university programs were not an input to this survey, especially those not taught in English. The purpose of this survey is to build a representative sample of university programs in software engineering, not to establish a definitive list of university programs.
- Analysis of software engineering university programs was based on course titles only, and an analysis of their syllabuses would surely provide additional insight. However, such a further analysis was not within the scope of this Straw Man phase.

Proposed Related Disciplines	Core Courses Number of programs/Number of courses (out of 29 under-graduate and graduate programs)	Admission Criteria Number of programs /24 (graduate programs only)	Elective Courses Number of programs/Number of courses (out of 29 undergraduate and graduate programs)
Computer Science[24]	19/67	17	23/190
Project Management	19/24		10/14
Electrical Engineering	9/13	3	7/12
Mathematics	8/21	11	4/9
Telecommunications/Networks	7/12	1	11/29
Management	4/11		9/25
Science	1/4		
Other Engineering Disciplines		1	1/4
Cognitive Sciences			2/2

» Table 4 Proposed List of Related Disciplines

[24] The list of topics included in Computer Science is listed in the Appendices. Although there is always room for interpretation, this list is similar to the one used by Glass [25], for instance, which is derived from the CS Curriculum of the ACM/IEEE-CS *Joint Task Curriculum Task Force*. For some, Software Engineering includes everything related to the development of software, including programming languages, for example. This is not the view here, the precise goal being rather to distinguish between Computer Science and Software Engineering.

7. Summary and Next Steps

Given the pervasive presence of software in our society and the increased concerns over the necessity for certification and licensing, consensus on a Guide to the Software Engineering Body of Knowledge is a must. It is critical that leadership on this important issue be on a worldwide scale, otherwise future university program accreditation guidelines and certification and licensing rules for professionals will differ widely.

A three-phase project has been initiated to develop the Guide to the Software Engineering Body of Knowledge. This report is the result of the first phase and was written with the premise that such a Guide must contain "consensually validated" knowledge and practices and rest on rational grounds. Consequently, it is based on the analysis of general software engineering textbooks and university programs offered in the field. The compilation was carried out as objectively as possible and in a reproducible manner. The process produced a list of potential Knowledge Areas and Related Disciplines.

The list of proposed Knowledge Areas based on ISO/IEC 12207 is:

- Development Process
  - Requirements Analysis
  - Detailed Design
  - Coding
  - Testing
- Maintenance Process
- Configuration Management
- Quality Assurance
- Verification and Validation
- Improvement Process

The list of proposed Knowledge Areas that do not converge well with ISO/IEC 12207

- Software Development Methods
  - Object Oriented
  - Formal Methods
  - Prototyping
- Software Development Environments
- Software Engineering Overview & Definition
- Measurement/Metrics
- Software Reliability

The list of proposed Related Disciplines is:

- Computer Science
- Project Management
- Electrical Engineering
- Mathematics
- Telecommunications/Networks

- Management
- Science
- Other Engineering Disciplines
- Cognitive Sciences

This report, which is intended to jump start the second, or Stone Man phase, will most certainly stimulate a lively debate within the Industrial Advisory Board. The deliverables of the Stone Man phase are:

- a list of Knowledge Areas of software engineering;
- a list of topics and relevant reference materials for each Knowledge Area;
- a list of disciplines related to Software Engineering, and the Knowledge Areas and topics lying at the junction of Software Engineering and one or more of these Related Disciplines.

To ensure the completeness and cohesiveness of the Stone Man version, a common framework is required for structuring Knowledge Areas. The identification methodology used in the Straw Man version for proposing Knowledge Areas and Related Disciplines must be expanded to be appropriate for identifying topics and selecting reference materials within each Knowledge Area. This is due notably to:

- the varying level of granularity of the tables of contents of textbooks;
- the widely ranging age of these textbooks;
- the widely varying types of university programs surveyed for this report;
- the different number of courses offered within each program;
- the fact that course titles and table of contents entries were analyzed rather than course syllabuses and textbooks chapters.

It is therefore suggested that a list of topics be drafted for each subcommittee based on an synthesis of the six most recent general software engineering textbooks listed in Appendix A[25]. These draft lists of topics would be classified using an adapted version of the schema proposed by Vincenti [20]for engineering design knowledge[26]. Each subcommittee would then be asked to review and improve the list of proposed topics and select reference materials for each topic. The subcommittees would return an updated version of the list of proposed topics for a given Knowledge Area and pertinent reference materials classified using the adapted Vincenti categories.

The Vincenti categories of engineering design knowledge are proposed as a framework for organizing topics and reference materials because:

- they are based on a detailed historical analysis of an established branch of engineering: areonautical engineering;
- they are viewed by Vincenti as applicable to all branches of engineering[27];

---

[25] These textbooks are Behforooz and Hudson, 1996, Jalote, 1997, Pfleeger, 1998, Pressman, 1996, Sommerville, Ian, 1995 and Dorfman and Thayer, a general tutorial on software engineering.

[26] As cited in Chapter 3 and proposed by Vincenti, the categories of engineering design knowledge are:
- fundamental design concepts;
- criteria and specifications;
- theoretical tools;
- quantitative data;
- practical considerations;
- design instrumentalities.

[27] In the introduction to the chapter that proposes the categories of engineering design knowledge, Vincenti states on p. 200: "Although the cases all come from aeronautics, the generalizations of this chapter are intended to be more universal. Design in other branches of engineering (mechanical, electrical, etc.) though different in detail, proceeds in much the same fashion. It therefore involves the same broad categories of knowledge and activities that generate it. The specifics from my experience and the studies of others supply illustrative evidence for this fact. As stated in chapter 1, I believe the generalizations to the other branches will call for addition and modification rather than fundamental revision."

- gaps in the software engineering body of knowledge within certain categories as well as efforts to reduce these gaps over time would be made apparent;
- due to generic nature of the categories, knowledge within each knowledge area could evolve and progress significantly while the framework itself would remain stable;

Many long hours of work, debate and consensus building will be required to develop the Stone Man and subsequent Iron Man versions of the Guide to the Software Engineering Body of Knowledge. Achieving consensus on the core body of knowledge is a key milestone in all disciplines and is pivotal for the evolution of software engineering toward a professional status. Involvement by all parties, industry, professional societies, standard setting bodies and academia, is critical to ensure the relevancy and the credibility of results, and for a quick uptake of the results.

Later on p. 236, he states, after presenting a summary table of knowledge categories and knowledge-generating activities: "I believe the table and the ideas behind it apply to design in all branches (aeronautical, mechanical, electrical, etc.), of modern engineering. I believe, in addition, though I haven't thought about the matter in depth, that they can also be adapted without major difficulty to the engineering that occurs in production and operation."

8. References

[1] "Draft Software Engineering Accreditation Criteria," *IEEE Computer*, vol. 31, pp. 73-75, 77, 1998.

[2] D. Gotterbarn, K. Miller, and S. Rogerson, "Software Engineering Code of Ethics, version 3.0," *IEEE Computer*, pp. 88-92, 1997.

[3] "IEEE Standard Glossary of Software Engineering Terminology," IEEE, Piscataway, NJ std 610.12-1990, 1990.

[4] P. Starr, *The Social Transformation of American Medicine*: BasicBooks, 1982.

[5] P. Naur and B. Randell, "Software Engineering," presented at Report on a Conference sponsored by the NATO Science Committee, Garmisch, Germany, 1968.

[6] J. W. Moore, *Software Engineering Standards, A User's Road Map*. Los Alamitos: IEEE Computer Society Press, 1998.

[7] G. Ford and N. E. Gibbs, "*A Mature Profession of Software Engineering*," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical CMU/SEI-96-TR-004, January 1996.

[8] "Special Issue on IT Licensing and Certification," in *Cutter IT Journal*, vol. 11, 1998, pp. 30.

[9] P. Dart, L. Johnston, C. Schmidt, and L. Sonenberg, "Developing an Accredited Software Engineering Program," *IEEE Software*, vol. 14, 1997.

[10] N. R. Mead, "Are We Going to Fish or Cut Bait? Licensing and Certification of Software Professionals," *Cutter IT Journal*, vol. 11, pp. 4-8, 1998.

[11] L. Werth, "Certification and Licensing for Software Professionals and Organizations," presented at *11th Conference on Software Engineering Education and Training (CSEE&T '98)*, Atlanta, Georgia, 1998.

[12] C. Jones, "Software Challenge - Legal Status of Software Engineering," *Computer*, vol. 28, pp. 98-99, 1995.

[13] E. Yourdon, "Why Do We Need Licensing? It's Not As If We've Killed Anyone...," *Cutter IT Journal*, vol. 11, pp. 26-30, 1998.

[14] M. Shaw, "Prospect for an Engineering Discipline of Software," *IEEE Software*, pp. 930-940, 1990.

[15] R. L. Baber, "Comparison of Electrical "Engineering" of Heaviside's Times and Software "Engineering" of Our Times," *IEEE Annals of the History of Computing*, vol. 19, pp. 5-17, 1997.

[16] D. L. Parnas, "*Software Engineering Programmes are not Computer Science Programmes*," McMaster University, Hamilton, Ontario CRL Report no. 361, April 1998.

[17] T. Maibaum, "What We Teach Software Engineering in the University: Do we Take Engineering Seriously?," *ACM SIGSOFT, Software Engineering Notes*, vol. 22, pp. 40-50, 1997.

[18] W. F. Tichy, "Should Computer Scientists Experiment More?," *Computer*, vol. 31, pp. 32-40, 1998.

[19] M. V. Zelkowitz and D. Wallace , "Experimental Models for Validating Technology," *Computer*, vol. 31, pp. 23-31, 1998.

[20] W. G. Vincenti, *What Engineers Know and How They Know It - Analytical Studies from Aeronautical History*. Baltimore and London: Johns Hopkins, 1990.

[21] P. J. Denning, D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner, and P. R. Young, "Computing as a Discipline," *Communications of the ACM*, vol. 32, pp. 9-23, 1989.

[22] W. R. Duncan, "*A Guide to the Project Management Body of Knowledge*," Project Management Institute, Upper Darby, PA 1996.

[23] "*Information Technology - Software Life Cycle Processes*," International Standard, Technical ISO/IEC 12207:1995(E), 1995.

[24] T. P. Rout, "Issues in the Development of an International Standard for Software Process Assessment," *Software Process Newsletter*, vol. 10, pp. 1-6, 1997.

[25] R. Glass, "A Comparative Analysis of the Topic Areas of Computer Science, Software Engineering, and Information Systems," *Journal of Systems Software*, vol. 19, pp. 277-289, 1992.

[26] T. B. Hilburn, D. J. Bagert, S. Mengel, and D. Oexmann, "Software Engineering Across Computing Curricula," *3rd Annual Conference on Integrating Technology into Computer Science Education - ITiCSE'98*, pp. 4, 1998.

9. Appendices

Appendix A. List of General Textbooks and Tutorials on Software Engineering

Behforooz, Ali and Frederick J. **Hudson,** 1996, *Software Engineering Fundamentals*, Oxford University Press.

Bell, Dough, Ian **Morrey** and John **Pugh,** 1992, *Software Engineering, 2nd Edition,* Prentice Hall.

Blum, Bruce I., 1992, *Software Engineering: A Holistic View,* Oxford University Press.

Conger, Sue A., 1993, *The New Software Engineering*, Course Technology.

Dorfman, Merlin and Ricahrd H. **Thayer,** Editors, 1996, *Software Engineering,* IEEE Computer Society Press, Los Alamitos, California.

Fairclough, Jon, Editor, 1995, *Software Engineering Guides*, Prentice Hall.

Fairley, Richard E., 1985, *Software Engineering Concepts,* McGraw Hill.

Ford, Neville J. and Mark **Woodroffe**, 1993, *Introducing Software Engineering*, Prentice Hall.

Ghezzi, Carlo**,** Mehdi **Jazayeri** and Dino **Mandrioli,** 1991, *Fundamentals of Software Engineering*, Prentice Hall.

Humphrey, Watts S., 1995, *A Discipline for Software Engineering*, Addison-Wesley.

Ince, D., 1989, *Software Engineering,* International Thomson Computer Press.

Jalote, Pankaj, 1997, *An Integrated Approach to Software Engineering,* Springer Verlag, New York.

Jones, Gregory W., 1990, *Software Engineering*, John Wiley & Sons.

Mazza, C., J. **Faircoulgh**, B. **Melton**, D. **de Pablo**, A. **Sheffer** and R. **Stevens,** 1994, *Software Engineering Standards*, Prentice Hall.

Pfleeger, Shari Lawrence, 1998, *Software Engineering: Theory and Practice,* Prentice Hall, New Jersey.

Pressman, Roger S., 1988, *Software Engineering, A Beginner's Guide,* McGraw Hill.

Pressman, Roger S., 1996, *A Manager's Guide to Software Engineering*, McGraw Hill.

Pressman, Roger S., 1996, *Software Engineering: A Practitioner's Approach, 4th Edition,* McGraw Hill.

Sage, Andrew P. and James D. **Palmer**, 1990, *Software Systems Engineering*, John Wiley & Sons.

Sallis, Philip, Tate **Graham** and Stephen **McDonnell**, 1995, *Software Engineering: Practice, Management, Improvement*, Addison-Wesley.

Schach, Stephen R, 1993, *Software Engineering, 2nd Edition*, McGraw-Hill.

Sommerville, Ian, 1995, *Software Engineering, 5th Edition*, Addison-Wesley.

Thayer, Richard H. and Andrew D. McGettrick, Editors, 1993, *Software Engineering: A European Perspective*, IEEE Computer Society Press, Los Alamitos, California.

Van Vliet, Hans and Vrije **Van Vliet**, 1993, *Software Engineering: Principles and Practice,* John Wiley & Sons.

Appendix B. URLs of Undergraduate and Graduate Programs in Software Engineering

Undergraduate Programs in Software Engineering

We found five undergraduate programs in software engineering at four universities:

- The University of Birmingham offers two distinct programs, one entitled *Software Engineering* and the other *Software Engineering with Business Studies* - Birmingham, United Kingdom.

  www.cs.bham.ac.uk/degreeregs/

- University of London - Imperial College of Science, Technology and Medicine Birmingham, United Kingdom

  www.doc.ic.ac.uk/teaching/under/comp/regulations/mengse.html

- University of New South Wales, Australia

  www.cse.unsw.edu.au/school/teaching/courses/bese.html

- University of Ottawa - Ottawa, Canada

  No URL available

Graduate Programs in Software Engineering

We found 24 graduate programs at 23 universities:

- Andrews University - Berrien Springs, MI, USA
  MSc in Software Engineering

  http://www.andrews.edu/CS/cis-ms.html

- Carnegie Mellon University - Pittsburgh, PA,USA
  Master of Software Engineering

  http://www.cs.cmu.edu/afs/cs/project/mse/www/

- Concordia - Montreal, QC
  Master in Computer Science - Software Engineering Option

  http://www.cs.concordia.ca/Graduate_Info/Graduate_Programs_M.html

- DePaul University - Chicago, IL,USA
  MSc in Software Engineering (One concentration in Software Development, the other in Software Management)

  http://www.cs.depaul.edu/programs/Segrad.html

- Embry-Riddle University - Daytona Beach, FL,USA
  Master of Software Engineering

  http://www.db.erau.edu/catalog/graduate/mse.html

- Flinders University of South Australia, Australia
  Master of Software Engineering

  http://www.cs.flinders.edu.au/

- Kansas State University - Manhattan, KS, USA
  Master of Software Engineering

  http://www.ksu.edu/grad/catalog/cis.htm

- Monmouth University, West Long Branch, NJ, USA
  MS in Software Engineering

  http://www.monmouth.edu/muse/stinfc97.html

- National Technological University - Fort Collings, CO, USA
  MS in Software Engineering

  http://www.ntu.edu/2/software.htm

- National University - La Jolla, CA, USA
  MS in Software Engineering

  http://www.nu.edu/catalog/somt/msse.html

- Seattle University - Seattle, WA, USA
  Master of Software Engineering

  http://www.seattleu.edu/~mse/mse97.html

- Southern Methodist University - Dallas, TX, USA
  MS in Software Engineering

  http://www.seas.smu.edu/disted/se/

- Texas Christian University - Fort Worth, TX, USA
  Master of Software Engineering

  http://www.cs.tcu.edu/grad/grad.html

- Université du Québec à Montréal - Montreal, QC, Canada
  M.Sc.A. in Software Engineering

  http://www.regis.uqam.ca/Programmes/3821.html

- University of Calgary - Calgary, AL, Canada
  MSc with Specialization in Software Engineering

  http://ksi.cpsc.ucalgary.ca/SERN/SEMSc.html

- University of Colorado - Colorado Springs, CO, USA
  Master of Engineering - Option in Software Systems Engineering

  http://mepo-b.uccs.edu/software.html

- University of Houston - Clear Lake - Houston, TX, USA
  MS in Software Engineering

  http://www.cl.uh.edu/nas/applied/graduate/MSoftEngg.html

- University of Karlskrona/Ronneby - Sweden
  MS in Software Engineering

  http://www.hk-r.se/for/internationell/master.htm

- University of Maryland - College Park, Maryland, USA
  Master of Software Engineering

  http://www.cs.umd.edu/Grad/mswe.html

- University of Missouri-Kansas City, USA
  MS in Computer Science - Software Engineering Concentration

  http://www.umkc.edu/umkc/catalog/html/cmp-sc/0000.html

- University of Scranton - Scranton, PA, USA
  MS in Software Engineering

  http://academic.uofs.edu/department/gradsch/gsofteng.htm

- University of St. Thomas - Minneapolis, Minnesota, USA
  MS in Software Engineering

  http://www.gps.stthomas.edu/ms.html

- University of Stirling - Stirling, Scotland, United Kingdom
  MS in Software Engineering

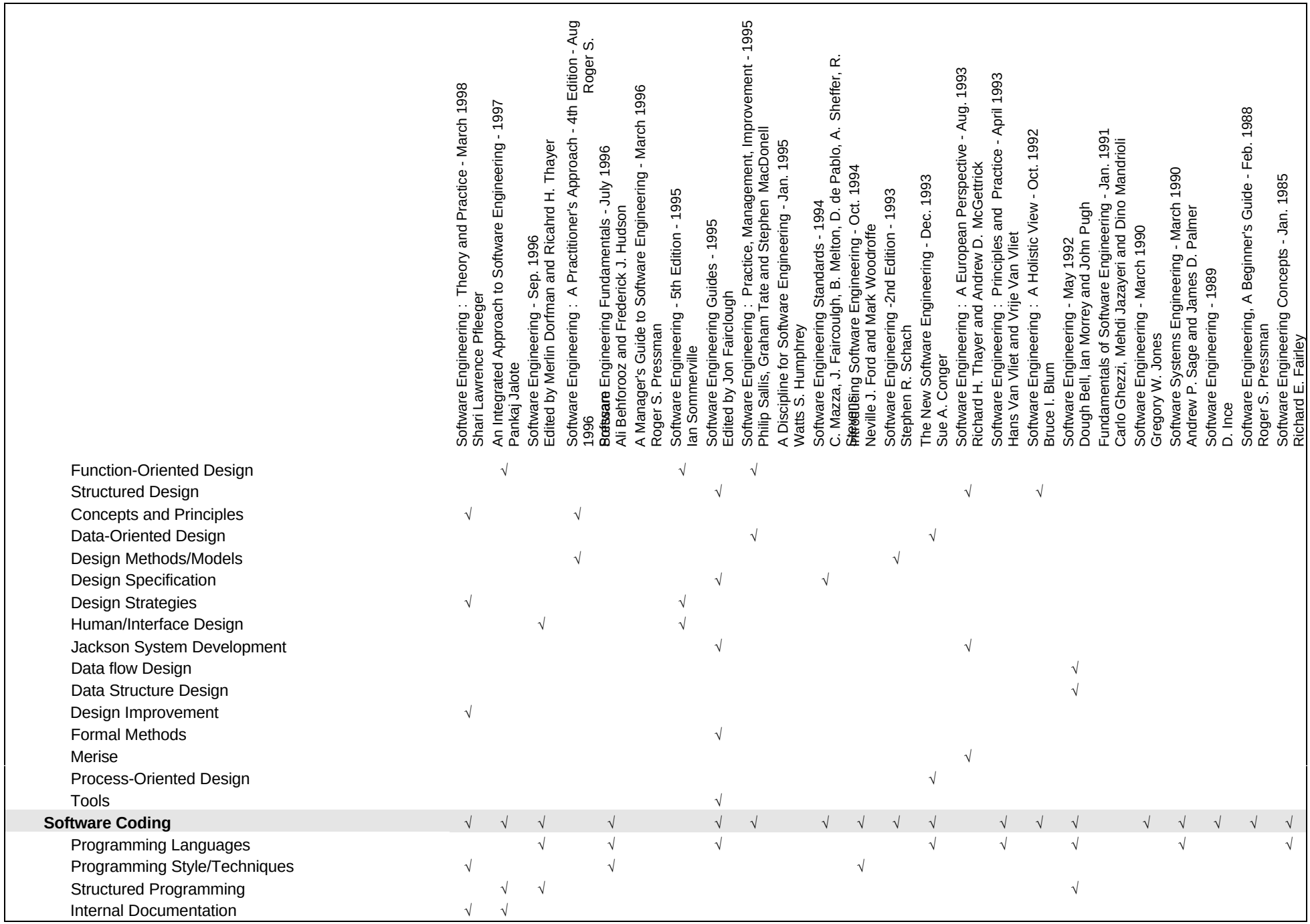
  http://www.cs.stir.ac.uk/~sbj/se-leaflet.html

Appendix C.

General Textbooks and Tutorials on Software Engineering - Classification of Table of Contents Entries According to Potential Knowledge Areas

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Introduction to Software Engineering																									
Software Engineering Overview	√	√	√									√		√	√		√			√	√	√		√
Software Problem/Crisis		√	√										√												
Software Engineering Principles																			√				√		
ISO/IEC 12207 Primary Processes																									
Acquisition Process							√							√											
Software/System and Hardware Procurement							√							√											
Development Process	√	√	√	√	√		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
System/Software Requirements Analysis	√	√	√	√	√		√	√	√		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Requirements/Problem/Systems Analysis - Analysis		√		√	√		√	√	√		√	√		√	√	√	√			√			√		
Object-Oriented Analysis		√		√	√			√						√											
Analysis Modeling				√			√													√					
Structured Analysis		√						√												√					
Data-Oriented Analysis									√					√											
Function-Oriented Analysis									√																
Informal Approach		√																							
Method-Based Analysis							√																		
Process-Oriented Analysis														√											
View Point-Oriented Analysis							√																		
Other		√																							
Requirements Specification	√	√			√		√	√	√		√	√	√		√	√			√	√	√	√	√		

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Using Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Formal Specification/Specification Languages		√					√	√								√		√					
Methods/techniques	√							√															
Specification Attributes					√			√															
Specification Tools					√			√															
Algebraic Specification							√																
Animation of Requirements Specification															√								
Real-Time Software Specification					√																		
Requirements Document	√	√			√		√	√			√												
Prototyping	√	√						√	√						√								
Requirements Process/Activities	√	√					√				√												
Requirements Identification/Capture	√							√														√	
Non-Functional Requirements							√																
Jackson System Development								√															
Types of Requirements	√																						
Requirements Evolution							√																
Software Architectural Design	√						√																


Software Engineering : Theory and Practice - March 1998
Shari Lawrence Pfleeger
An Integrated Approach to Software Engineering - 1997
Pankaj Jalote
Software Engineering - Sep. 1996
Edited by Merlin Dorfman and Ricahrd H. Thayer
Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996
Roger S.
Software Engineering Fundamentals - July 1996
Ali Behforooz and Frederick J. Hudson
A Manager's Guide to Software Engineering - March 1996
Roger S. Pressman
Software Engineering - 5th Edition - 1995
Ian Sommerville
Software Engineering Guides - 1995
Edited by Jon Fairclough
Software Engineering : Practice, Management, Improvement - 1995
Philip Sallis, Graham Tate and Stephen MacDonell
A Discipline for Software Engineering - Jan. 1995
Watts S. Humphrey
Software Engineering Standards - 1994
C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R.
Software
Software Engineering - Oct. 1994
Neville J. Ford and Mark Woodroffe
Software Engineering -2nd Edition - 1993
Stephen R. Schach
The New Software Engineering - Dec. 1993
Sue A. Conger
Software Engineering : A European Perspective - Aug. 1993
Richard H. Thayer and Andrew D. McGettrick
Software Engineering : Principles and Practice - April 1993
Hans Van Vliet and Vrije Van Vliet
Software Engineering : A Holistic View - Oct. 1992
Bruce I. Blum
Software Engineering - May 1992
Dough Bell, Ian Morrey and John Pugh
Fundamentals of Software Engineering - Jan. 1991
Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli
Software Engineering - March 1990
Gregory W. Jones
Software Systems Engineering - March 1990
Andrew P. Sage and James D. Palmer
Software Engineering - 1989
D. Ince
Software Engineering, A Beginner's Guide - Feb. 1988
Roger S. Pressman
Software Engineering Concepts - Jan. 1985
Richard E. Fairley
Function-Oriented Design
Structured Design
Concepts and Principles
Data-Oriented Design
Design Methods/Models
Design Specification
Design Strategies
Human/Interface Design
Jackson System Development
Data flow Design
Data Structure Design
Design Improvement
Formal Methods
Merise
Process-Oriented Design
Tools
Software Coding
Programming Languages
Programming Style/Techniques
Structured Programming
Internal Documentation

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircough, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Tools								√					√											
User Manual								√			√													
Code Generators									√															
Functional Programming																√								
Logic Programming																√								
Object-Oriented Programming																√								
Parallel Programming																√								
Principles													√											
Top-Down/Bottom-Up Programming		√																						
System/Software Integration					√			√					√										√	
Integration Plan					√																			
System/Software Testing	√	√	√	√	√		√	√			√		√	√	√	√				√	√	√	√	
Unit/Module Testing	√	√			√			√			√													
System Testing	√				√			√			√													
Testing Fundamentals/Basic Concepts/Principles	√	√			√								√											
Functional Testing	√	√						√																
Integration Testing	√							√			√													
Object-Oriented Testing	√	√		√																				
Other Tests	√				√			√																
Test Plan	√				√		√																	
Interface Testing							√	√																

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Structural Testing		√					√																		
Testing Process		√					√																		
Testing Strategies				√			√																		
Tools	√							√																	
White-Box/Black-Box Testing							√	√																	
Debugging					√																				
Performance Testing	√																								
Real-Time Software Testing					√																				
Testing Techniques				√																					
Software Installation	√							√			√														
Transfer Document	√							√			√														
Planning								√																	
Software Acceptance Support	√							√			√														
Acceptance Testing								√			√														
Training	√																								
Operation Process								√															√		
System Operation								√															√		
Maintenance Process	√		√		√		√	√			√	√	√	√		√	√						√	√	√	√
Documentation							√	√			√														
Maintainability	√						√																	√	
Maintenance Process					√		√																		

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S.	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Managing Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Tools	√							√																
Cost							√																	
ISO/IEC 12207 Supporting Life Cycle Processes																								
Configuration Management Process		√	√	√		√	√	√			√				√	√	√							
Process Implementation		√					√	√			√													
Configuration Management Plans		√					√	√			√													
Tools								√																
Configuration Identification		√						√			√													
Configuration Control		√					√	√			√													
Change Control		√						√			√													
Item control/storage								√			√													
Change Management							√																	
Configuration Status Accounting		√						√			√													
Status Accounting and Auditing		√						√			√													
Configuration Release Management and Delivery							√																	
Version and Release Management							√																	
System Building							√																	
Quality Assurance Process		√	√	√	√	√	√	√	√	√	√	√		√	√	√						√		
Process Implementation		√			√		√	√		√	√				√	√								
Quality Assurance Plans		√			√			√			√													

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Quality Management							√			√					√	√									
Quality Assurance Process/Activities					√		√				√														
Product/Process Assurance					√																				
Quality Attributes					√																				
Verification and Validation Processes	√	√	√				√	√	√	√	√								√	√	√				√
Design Verification/Validation	√	√						√		√	√														
Requirements Validation	√	√					√																		
Software/Program Inspections			√				√				√														
Verification and Validation Plans		√						√			√														
Walkthroughs		√						√			√														
Cleanroom Method								√	√																
Traceability			√					√																	
Static Verification							√																		
Tools								√																	
Joint Review Process		√	√					√		√	√														
Design Reviews		√								√	√														
Reviews			√					√			√														
Code Reviews		√								√															
Requirement Reviews		√																							
Audit Process			√					√			√														
Audits			√					√			√														

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
ISO/IEC 12207 Organizational Life Cycle Processes																							
Management Process	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	√		√	√	√		√
Initiation and Scope Definition																							
Concepts/Principles				√													√						
Activities/Framework							√		√		√												
Planning	√	√	√	√	√	√	√	√	√	√	√		√		√	√							√
Estimation	√	√	√		√	√	√	√	√	√					√	√							√
Cost/Effort Estimation	√	√	√		√		√	√							√	√							√
Duration/Schedule Estimation		√			√		√	√		√													
Resource Estimation							√	√		√													
Building Estimation Models/Estimation Techniques		√					√																
COCOMO		√						√															
Size Estimation		√								√													
FPA								√															
Non-labor cost Estimation								√															
Risk	√	√	√	√	√	√		√	√		√												
Risk Management	√	√	√	√	√				√														
Risk Assessment/Analysis		√			√	√																	
Risk Control		√			√																		

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Handbook of Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Risk Areas					√																		
Risk Model					√																		
Risk Table/Matrix								√															
Project Scheduling		√		√		√	√				√												
Process/Activities								√		√													
Cost-benefit Analysis					√																		
Planning Tools					√																		
Project Budgeting											√												
Staff/Personnel/Resources Planning		√																					
Execution and Control	√	√		√		√	√	√	√		√				√	√							
Project Control/Monitoring/Tracking	√	√		√		√			√						√	√							
Managing People	√					√	√									√							
Leading the project/Manager's Role								√	√		√												
Reporting								√			√												
Technical Management								√			√												
Management of contracted software									√														
Management of In-house software development									√														
Tools								√															
Review and evaluation	√						√		√														
Productivity							√		√														
Resources Evaluation and Improvement	√																						

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Improvement Process	√	√	√	√	√	√	√		√	√	√		√		√	√	√		√		√			
Process Establishment	√	√	√		√		√		√	√	√		√		√	√	√		√		√			
Life-Cycle Processes	√	√	√		√		√		√		√				√	√	√				√			
Models/Approaches		√	√		√		√				√						√				√			
Prototyping		√					√				√													
Spiral Model		√	√				√																	
Waterfall Model		√									√													
Evolutionary Model											√													
Incremental Model											√													
Iterative Enhancement Model		√																						
Others			√													√								
Process Models/Structure	√								√				√		√		√							
Process Analysis/Modeling							√		√															
Process Selection									√	√														
Process Characteristics		√																						
Process Assessment	√					√																		
Process Evaluation/Assessment	√					√																		
Process Improvement	√	√	√				√		√										√					
CMM			√				√		√															
Process Maturity		√							√															
ISO 9000									√															

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Spice									√															
Special Topics																								
Metrics/Measurement		√	√	√	√		√	√	√	√					√									
Management Metrics		√		√	√			√	√															
Complexity Metrics		√						√	√															
Quality Metrics		√					√								√									
Size Metrics		√							√	√														
Process Metrics								√	√															
Product Metrics								√	√															
Reliability Metrics							√		√															
Data Collection					√																			
Object Oriented Metrics				√																				
Software Attributes					√																			
Technical Metrics				√																				
Tools			√																					
Other Metrics		√						√	√															
Real-Time Systems			√	√	√		√																	
Real-Time Systems Design			√	√	√		√																	
Real-Time Systems Requirements Specification					√																			
Real-Time Systems Testing					√																			
Re-engineering/Reengineering			√	√	√		√																	

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Reverse Engineering					√		√																	
Software Development Environments					√		√								√			√			√			
CASE			√	√	√	√	√						√		√									
CASE Classification							√																	
Integrated CASE							√																	
CASE Workbenches							√																	
Integrated Environments							√																	
Tool Integration									√															
Tools/Toolsets									√															
Unix Environment									√															
Software Development Methodologies/Paradigms	√	√	√	√	√	√	√	√	√			√		√	√			√			√			
Object-Oriented Approach	√	√	√	√		√		√	√			√						√			√			
Object-Oriented Design	√		√	√		√												√			√			
Object-Oriented Analysis	√		√	√																				
Definitions, Concepts, Principles			√	√																				
Object-Oriented Development		√																						
Object-Oriented Programming																		√						
Formal Methods	√	√	√				√	√				√		√	√									
Formal Specification/Specification Languages	√						√	√							√									
Definition															√									
Formal Design								√																

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Managing Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Prototyping		√	√				√	√				√									√		
Prototyping in the Software Development Process		√					√																
Design and Prototyping								√															
Prototyping Techniques							√																
Requirements Analysis and Prototyping		√																					
User Interface Prototyping							√																
Jackson System Development								√							√		√						
Data-centered Approach									√														
Function-Oriented Approach									√														
Knowledge-based Approach									√														
Structured Systems Analysis and Design Method (SSADM)			√																				
Software Products	√			√		√	√		√														
Software Products Evaluation	√																						
Software Products Improvement	√																						
Software Reliability	√		√		√		√										√					√	
Defensive Programming							√																
Definitions					√																		
Design Rules					√																		
Exception handling							√																
Fault Avoidance							√																
Fault Tolerance					√		√																

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircough, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Introducing Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Models					√																			
Redundancy					√																			
Specification							√																	
Software Reuse			√	√			√		√														√	
Code Reuse									√															
Software Development for Reuse							√																	
Software Development with Reuse							√																	
Standards			√				√								√									
Evaluating Software Engineering Standards			√												√									
Quality Standards															√									
Other Topics																								
Cleanroom Software Engineering				√			√												√					
Safety-Critical Software	√						√								√									
Software Tools												√					√			√				
Fault-Tolerant Software															√					√				
Human Factors					√		√																	
Client/Server				√																				
Ethics												√												
Expert Systems																						√		
Software Engineering Education			√																					
Software Psychology																		√						

	Software Engineering : Theory and Practice - March 1998 Shari Lawrence Pfleeger	An Integrated Approach to Software Engineering - 1997 Pankaj Jalote	Software Engineering - Sep. 1996 Edited by Merlin Dorfman and Ricahrd H. Thayer	Software Engineering : A Practitioner's Approach - 4th Edition - Aug 1996 Roger S. Pressman	Software Engineering Fundamentals - July 1996 Ali Behforooz and Frederick J. Hudson	A Manager's Guide to Software Engineering - March 1996 Roger S. Pressman	Software Engineering - 5th Edition - 1995 Ian Sommerville	Software Engineering Guides - 1995 Edited by Jon Fairclough	Software Engineering : Practice, Management, Improvement - 1995 Philip Sallis, Graham Tate and Stephen MacDonell	A Discipline for Software Engineering - Jan. 1995 Watts S. Humphrey	Software Engineering Standards - 1994 C. Mazza, J. Faircoulgh, B. Melton, D. de Pablo, A. Sheffer, R. Stevens	Managing Software Engineering - Oct. 1994 Neville J. Ford and Mark Woodroffe	Software Engineering -2nd Edition - 1993 Stephen R. Schach	The New Software Engineering - Dec. 1993 Sue A. Conger	Software Engineering : A European Perspective - Aug. 1993 Richard H. Thayer and Andrew D. McGettrick	Software Engineering : Principles and Practice - April 1993 Hans Van Vliet and Vrije Van Vliet	Software Engineering : A Holistic View - Oct. 1992 Bruce I. Blum	Software Engineering - May 1992 Dough Bell, Ian Morrey and John Pugh	Fundamentals of Software Engineering - Jan. 1991 Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli	Software Engineering - March 1990 Gregory W. Jones	Software Systems Engineering - March 1990 Andrew P. Sage and James D. Palmer	Software Engineering - 1989 D. Ince	Software Engineering, A Beginner's Guide - Feb. 1988 Roger S. Pressman	Software Engineering Concepts - Jan. 1985 Richard E. Fairley
Technology Transition						√																		

Appendix D.
Undergraduate Programs in Software Engineering -
Classification of Courses According to Potential Knowledge Areas

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	University of New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Compulsory Courses					
Introduction Software Engineering	2	2		6	1
Software Engineering/IS Engineering	√ √	√ √		√√√√√	√
ISO/IEC 12207 Primary Processes	1	1	2	3	7
Development Process	1	1	2	3	7
General Subjects					2
Software Development					√
Foundation of software Development					√
System/Software Requirements Analysis				2	1
Analysis				√	
Object-Oriented Analysis					√
Requirements Engineering				√	
Software Detailed Design	1	1	2	1	4
Design			√ √	√	√ √
Human/User Interface	√	√			√
Object-Oriented Design/Modeling					√
ISO/IEC 12207 Supporting Life Cycle Processes					1
Quality Assurance Process					1
Quality					√
ISO/IEC 12207 Organizational Life Cycle Processes				1	1
Management Process				1	1
Project/Software Management					√
Information Management				√	
Special Topics					3
Real-Time Software/Embedded Systems					√
Reengineering					√
Software Security/Safety					√
Other Courses	11	15	18	20	28
Accounting					√
Algebra					√
Algorithms			√	√	√
Artificial Intelligence			√		
Business Management		√			√ √
C++	√ √	√ √			
Calculus					√ √
Chemistry					√
Communication Skills and Professional Issues	√	√			
Compilers			√		

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	University of New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Computational Methods/Computing			√	√ √	
Computer Architecture	√	√	√ √		
Computer Graphics	√	√	√		
Computer Science	√	√			
Concurrent Systems/Programming			√		
Data Bases/Data Management	√	√	√	√ √	√
Data Strucutres/Data Organization				√	√
Digital Computer Organization				√	√
Discret Mathematics	√		√	√	√ ½
Economy		√ √			√
Ethics				√	
File Management					√
Foundations of Computer Science					√
Hardware			√		√
Human/User Interface				√	
Human Factors/Human Resources		√			
Logic	√	√	√	½	½
Marketing		√			
Mathematics				√ √	
Mechanics					√
Microprocessors					√
Networks/Networking			√	√ √	
Operating Systems			√ √	√	√
Physics					√ √
Programming				√ √	
Reuse				½	
Semantics			√		
Simulation			√		
Software Workshop	√ √	√ √			
Statistics			√	√	√
Technical Communication and Writing					√
Telecommunications/Communication Systems					√ √ √
Optional Courses					
ISO/IEC 12207 Primary Processes	7	6			1
Development Process	7	6			1
System/Software Requirements Analysis	1				
Formal Methods/specification languages	√				
Software Detailed Design	1	1			

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	University of New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Human/User Interface	√	√			
Software Coding	4	4			1
Commercial Programming	√	√			
Comparison of Programming Languages	√	√			
Programming					√
Programming Languages Principles	√	√			
Programming Methods	√	√			
System/Software Testing	1	1			
Verification, Validation and Testing	√	√			
ISO/IEC 12207 Supporting Life Cycle Processes	1	1			
Verification and Validation Process	1	1			
Verification and Validation	√	√			
ISO/IEC 12207 Organizational Life Cycle Processes	2	2			
Management Process	2	2			
Project Planning	√	√			
Strategic Management	√	√			
Special Topics	1				
Real-Time Software/Embedded Systems	√				
Other Courses	29	21	2		14
Accounting	√	√			
Algebra					√
Artificial Intelligence	√√√√√ ½	√ √ ½			√
Automata Theory	√				
Calculus	√				
Cognitive Science	√	√			
Compilers	√	√			√
Computer Graphics	√	√			√
Computer Structures					√
Trends in Computing			√		
Data Bases/Data Management	√	√			
Distributed Systems					√
Evolutionary Computation	√	√			
Expert Systems	½	½			
Foreign Language			√		
Image Processing	√	√			√
Internet					√
International Business	√	√			
Logic	√	√			

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	University of New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Marketing	√ √	√			
ML (Programming Language)	√	√			
Networks/Networking	√				
Numerical Methods					√
Operating Systems	√ √	√ √			
Combinatorial Optimization	√	√			
Pattern Recognition					√
Parallel Systems	√	√			√
Prolog	√	√			
Robotics					√
Simulation					√
Statistics	√				
Telecommunications/Communication Systems					√
Virtual Reality	√	√			
Optional specified (number)			1		3
Optional specified (credits)	120	120			
Optional non specified (number)			16	5	3
Optional non specified (credits)	20				
Project/Studio (credits)	√(40)	√(40)	√	√	√

Summary	University of Birmingham, UK BSc in Computer Science/Software Engineering		University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies		University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)		University of New South Wales, Sydney, Australia Bachelor of Software Engineering		University of Ottawa, Ontario B.A.Sc. in Software Engineering	
	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available
Introduction Software Engineering	2		2				6		1	
ISO/IEC 12207 Primary Processes	1	7	1	6	2		4		7	1
Development Process	1	7	1	6	2		3		7	1
General Subjects									2	
System/Software Requirements Analysis		1					2		1	
Software Detailed Design	1	1	1	1	2		1		4	
Software Coding		4		4						1
Software Testing		1		1						
ISO/IEC 12207 Supporting Life Cycle Processes		1		1					1	
Quality Assurance Process									1	
Verification and Validation Process		1		1						
ISO/IEC 12207 Organizational Life Cycle Processes		2		2			1		1	
Management Process		2		2			1		1	
Special Topics		1							3	
Other Courses	11	29	15	21	18	2	20		28	14
Optional non specified (number)					16		5		13	
Optional non specified (credits)	20									

Appendix E.
Undergraduate Programs in Software Engineering - Classification of Courses by Related Discipline

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	Universityof New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Compulsory Courses					
Communication	1	1			1
Communication Skills and Professional Issues	√	√			
Technical Communication and Writing					√
Computer Science	8	8	11	10	7
Algorithms			√	√	√
Artificial Intelligence			√		
C++	√ √	√ √			
Compilers			√		
Computational Methods/Computing			√	√ √	
Computer Graphics	√	√	√		
Computer Science	√	√			
Concurrent Systems/Programming			√		
Data Bases/Data Management	√	√	√	√ √	√
Data Structures/Data Organization				√	√
File Management					√
Foundations of Computer Science					√
Operating Systems			√ √	√	√
Programming				√ √	
Semantics			√		
Simulation			√		
Software Workshop	√ √	√ √			
Electrical Engineering	1	1	3	1	3
Computer Architecture	√	√	√ √		
Digital Computer Organization				√	√
Hardware			√		√
Microprocessors					√
Management		5			4
Accounting					√
Business Management		√			√ √
Economy		√ √			√
Marketing		√			
Mathematics	2	1	3	5	7
Algebra					√
Calculus					√ √
Discrete Mathematics	√		√	√	√ ½
Logic	√	√	√	½	½
Mathematics				√ √	
Statistics			√	√	√
Project Management					1
Project/Software Management					√
Science					4
Chemistry					√
Mechanics					√
Physics					√ √

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	Universityof New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Telecommunications/Networks			1	2	3
Networks/Networking			√	√ √	
Telecommunications/Communication Systems					√ √ √
Optional Courses					
Application Domains	1				
Real-Time Software/Embedded Systems	√				
Cognitive Science	1	1			
Cognitive Science	√	√			
Communication			1		
Foreign Language			√		
Computer Science	24	20	1		13
Artificial Intelligence	√√√√√ ½	√ √ ½			√
Automata Theory	√				
Commercial Programming	√	√			
Comparison of Programming Languages	√	√			
Compilers	√	√			√
Computer Graphics	√	√			√
Computer Structures					√
Trends in Computing			√		
Data Bases/Data Management	√	√			
Distributed Systems					√
Evolutionary Computation	√	√			
Expert Systems	½	½			
Human/User Interface	√	√			
Image Processing	√	√			√
Internet					√
ML (Programming Language)	√	√			
Operating Systems	√ √	√ √			
Parallel Systems	√	√			√
Pattern Recognition					√
Prolog	√	√			
Programming					√
Programming Languages Principles	√	√			
Programming Methods	√	√			
Robotics					√
Simulation					√
Virtual Reality	√	√			
Management	5	4			
Accounting	√	√			
International Business	√	√			
Marketing	√ √	√			
Strategic Management	√	√			
Mathematics	4	2			2
Algebra					√
Calculus	√				

	University of Birmingham, UK BSc in Computer Science/Software Engineering	University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies	University of London, UK Imperial College of Science, Technology and Medicine MEng Computing (Software Engineering)	Universityof New South Wales, Sydney, Australia Bachelor of Software Engineering	University of Ottawa, Ontario B.A.Sc. in Software Engineering
Combinatorial Optimization	√	√			
Logic	√	√			
Numerical Methods					√
Statistics	√				
Project Management	1	1			
Project Planning	√	√			
Software Engineering	2	1			
Formal Methods/Specification languages	√				
Verification, Validation and Testing	√	√			
Telecommunications/Networks	1				1
Networks/Networking	√				
Telecommunications/Communication Systems					√
Optional specified (number)			1		3
Optional specified (credits)	120	120			
Optional non specified (number)			16	5	3
Optional non specified (credits)	20				
Project/Studio (credits)	√(40)	√(40)	√	√	√

Summary Table

	University of Birmingham, UK BSc in Computer Science/Software Engineering		University of Birmingham, UK BSc in Computer Science/Software Engineering with Business Studies		University of London, UK Imperial College of Science, Technology and Medicine Eng Computing (Software Engineering)		Universityof New South Wales, Sydney, Australia Bachelor of Software Engineering		University of Ottawa, Ontario B.A.Sc. in Software Engineering	
	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available	Compulsory Courses	Optional Courses Available
Cognitive Science		1		1						
Communication	1		1			1			1	
Computer Science	8	24	8	20	11	1	10		7	13
Electrical Engineering	1	1	1	1	3		1		3	
Ethics							1			
Management		5	5	4					4	
Mathematics	2	4	1	2	3		5		7	2
Project Management		1		1					1	
Science									4	
Telecommunications/Networking		1			1		2		3	1
Optional non specified (number)					16		5		13	
Optional non specified (credits)	20									

Appendix F.
Graduate Programs in Software Engineering -
Admission Requirements by Related Discipline

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Degree in Computer Science or equivalent	√		√	√		√	√	√		√ or	√	√	√	√	√		√	√	√		√	√	√
Grades							3.0	2.5 to 3.0				3.0/ 4.0	3.0	3.0/ 4.3	3.0				3.0/ 4.0	3.0	3.0/ 4.0	2.7	
Graduate Record Exam											√		√							√	√		
Experience Required																		√					
Software Development	√	√						√			√ or	√ or	√	√	√				√		√ or	√	
Software Maintenance											√	√									√		
Experience (Can be used for acceptance instead of degree)																							
Non Specified						√		√						√	√					√			√
Computer Science	2			1			3																
Algorithms							√																
Requirements Analysis	√																						
Data Structures							√																
Programming	√			√			√																
Mathematics				1			1																
Mathematics				√			√																
Software Engineering							1																
Software Engineering							√																
Specific Undergraduate Courses/Knowledge Required																							
Computer Science	3	7	3	1				4		8	1		2	2	2	2	1		1	7	3		1
Algorithms		√						√		√				√									
Comparative Programming Languages		O*																					
Compiling techniques		O*																					
Computer Organization	√	√																					

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Computer Science	√																√						
Data Analysis				√																			
Data Structures		√						√		√			√	√		√				√	√		
Data Bases		O*								√													
Discrete Structures																				√ √			
File Processing																					√		
Human-Computer Interaction															√								
Operating Systems		O*						√		√										√			
Programming								√												√ √			
Programming Languages	√	√	√ √ √							√ √	√		√		√	√			√	√	√		√
Programming Methods		√								√													
Unix										√													
Electrical Engineering										2										4	1		
Computer Architecture																				√ √			
Circuits and Devices																				√			
Digital Computing										√													
Principles of Hardware Organization										√											√		
Probability/Stochastic Processes																				√			
General Engineering																	1						
General Engineering																	√						
Mathematics	3	1		1				1				1		3		1		1		2	1		
Calculus	√			√								√		√						√			
Discrete Mathematics	√	√						√						√		√		√			√		
Linear Algebra														√									
Mathematics																				√			
Statistics	√																						

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Software Engineering				1				1							1	1	1						
Formal Methods				√																			
Object-Oriented Design															√								
Software Development								√															
Software Engineering																√	√						
Telecommunications/Networks														1									
Networks														√									

*two required

Summary	Andrews University - Michigan MS in Software Engineering		Carnegie Mellon University - Pittsburgh, PA Master of SE		Concordia - Montreal, QC Master in Computer Science - SE Option		DePaul University - Chicago, IL MS in SE		Embry-Riddle University - Daytona Beach, FL Master of SE		Flinders University of South Australia, Australia Master of SE		Kansas State University - Manhattan, KS Master of SE		Monmouth University, West Long Branch, NJ MS in SE		National Technological University - Fort Collings, CO MS in SE		National University - La Jolla, CA MS in SE		Seattle University - Seattle, WA Master of SE		Southern Methodist University - Dallas, TX MS in SE		Texas Christian University - Fort Worth, TX Master of SE		Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE		University of Calgary - Calgary, AL MSc with Specialization in in SE		University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE		University of Houston - Clear Lake - Houston, TX MS in SE		University of Karlskrona/Ronneby - Sweeden MS in SE		University of Maryland - College Park, Maryland Master of SE		University of Missouri-Kansas City MS in Computer Science - SE Concentration		University of Scranton - Scranton, PA MS in SE		University of St. Thomas - Minneapolis, Minnesota MS in SE		University of Stirling - Stirling, Scotland MS in SE	
	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience	Required Courses/Knowledge	Optional Experience
Computer Science	3	2	7		3		1	1						3	4				8		1				2		2		2		2		1				1		7		3				1	
Electrical Engineering																			2																				4		1					

Appendix G.
Graduate Programs in Software Engineering -
Classification of Courses According to Potential Knowledge Areas

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Credits	48 (quarter credits)	?	45	?	?	36 (credit hours)		33 (credit hours)	36	33 (sem. credits)	60 (quarter units)	45 (grad. credits)	30 (sem. credit hrs.)	31 (sem. hours)	45	?		36 (credit hours)	40 pts	36		36	42	
Grades				2.5/ 4.0	2.5/ 4.0						3.0		3.0/ 4.0				3.0				3.0			
Core Courses																								
Introduction to Software Engineering	2			1	1	1		1			2		1	1						1	1			
Software Engineering Principles				√	√						√													
Software Engineering/IS Engineering	√ √					√		√			√		√	√						√	√			
ISO/IEC 12207 Primary Processes	2	2	4	6	5	2		1	4	7	1	3	2	3	5	2	4	4	3	2	2	4	3	
Development Process	2	2	3	6	5	2		1	4	7	1	3	2	2	4	2	3	4	3	2	2	4	3	
General Subjects		1		1	1					2								2	1	1	1	1	1	
Development Methods/Methodologies		√		√						√													√	
Object-Oriented Development					√														√					
Software Development																		√				√		
Software Development Environments and Tools										√								√[4]		√	√			
System/Software Requirements Analysis	2	1	1	2	1	1		1	2	2		1	1		2	2	2	1	1	1	1	2	1	
Formal Methods/specification languages		√		√				√	√	√					√		√	√√[2]√[3]		√	√	√		
Requirements Analysis/Specification	√ √		√	½	½	√			√	√		√	√		√	√ √	√		√			√	½	
Software Detailed Design			1	2	2	1			1	1		1		1	1		1					1	1	
Design				½	½	½			√	√		√		½	√		√					√	½	
Human/User Interface			√[1]																					
Object-Oriented Design/Modeling				√	√																			
Software Coding				1	1					1		1												
C++				√	√																			

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Implementation										√														
Programming Methods												√												
System/Software Testing			1						1	1	1		1	1	1			1	1					
Testing			√[2]						√	√	√		½	½	½			√[1] √[4]	√					
Maintenance Process			1											1	1		1							
Maintenance			√											½	√		√							
ISO/IEC 12207 Supporting Life Cycle Processes			3			1			1	3	1	1	2	1	2	1	1	1	2				1	
Quality Assurance Process			2			1				2		1	1		1	1	1		1				1	
Quality Assurance			√ √[2]			√				√		√	½		½ ½[1]	√	√		½				√	
Reliability										√														
Verification and Validation Process			1						1	1	1		1	1	1			1	1					
Verification and Validation			√[2]						√	√	√		½	½	½			√[1] √[4]	√					
ISO/IEC 12207 Organizational Life Cycle Processes	1	1	2		2	1		1	1	2	1	1	2	1	1	1	2	2	2	3		1	1	
Management Process	1	1	2		2	1		1		1	1	1	2	1	1		2	1	2	1		1	1	
Estimation					½																			
IT Management																	√							
Productivity			√[2]												½[1]									
Project Planning													½											
Project/Software Management	√	√	√		√	√		√		√	½	√	½	√			√	√	½	√		√	√	
Software Economics																			√					
Improvement Process									1	1						1		1		2				
Software/Systems Process									√							√		√		√				
Life Cycle Models										√										√				

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Special Topics			1		1					1								2-3	1					
CASE																							√	
Measurement/Metrics					½					√								√[2] √[4]	√					
Real-Time Software/Embedded Systems																		√[1]						
Reengineering			√[1]															½[2] ½[4]						
Reuse																		½[2] ½[4]						
Software Security/Safety																		√[1]						
Other Courses	4	2	1-2	4	3	1			3		7	5		1	1-3	2	2	2-3		2	4	2	2	
Algorithms												½									√			
Analysis of Software Artifacts		√																						
Artificial Intelligence											√													
Computer Architecture	√																√							
Concurrent Systems																		√[1]						
Current Trends in Software Engineering																√								
Current/Special/Advanced Topics in SE											√													
Data Bases/Data Management	√										√ √						√	√[2] √[3]					√	
Data Analysis and Regression					√																			
Data Structures	√											½												
Distributed Systems			√												√			√[3]						
Foundations of Computer Science/SI				√ √ √	√ √										√[1]								√	
Hardware and Software Integration											√													
Information Security																				√				
Mathematics												√											√	
Networks/Networking									√[1] √[1]		√										√			

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Operating Systems	√																							
Organizational Management															√[1]									
Protocols																					√			
Quantitative Approach to Engineering Software																				√				
Research Process				√							½					√								
Software Architecture/IS Architectures		√				½						√						√						
Technical Communication and Writing												√		√									√	
Telecommunications/Communication Systems			√[1]																		√			
Wireless									√[1]															
Electives - Courses specified																								
Introduction to Software Engineering				1	1				1				1									1		
Software Engineering/IS Engineering				√	√				√				√									√		
ISO/IEC 12207 Primary Processes			6	13	13	3		3	1			5	5	3		8	1			2	2	1	6	6
Acquisition Process													1							1				
Software Acquisition													√							½				
Development Process			6	12	12	3		3	1			4	3	3		8	1			1	2		6	6
General Subjects			1	3	3	1		1								2	1						2	1
Object-Oriented Development			√	√		√		√[1]								√	√						√ √	√
Software Development				√	√											√								
Software Development Methods				√	√ √																			
System/Software Requirements Analysis			2	3	3	1						1	1			1							1	2
Data Analysis				√ √	√																			
Formal Methods/specification languages			√[2]	√	√ √	√						√												√
Object-Oriented Analysis													√			½								

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Requirements Analysis/Specification			√1																				√	√
Software Detailed Design			2	2	2	1		1				2	2	2		4							2	2
Design			√1													√								√
Human/User Interface				√	√	√						√ √	√	√		√ √							√ √	√
Object-Oriented Design/Modeling			√1	√	√			√1					√	√		½								
Software Coding				2	2				1			1		1		1						2		1	1
Implementation									½															
Object-Oriented Programming				√	√							√		√		√								
Programming Languages				√	√																	√ √	√	½
System/Software Testing			1	2	2			1													1			
Testing			√2	√ √	√ √			√2													√			
Maintenance Process				1	1							1	1									1		
Maintenance				√	√							√	√									√		
ISO/IEC 12207 Supporting Life Cycle Processes			1	3	3			1	1					2			1			2				1
Quality Assurance Process				1	1				1					2			1			1				1
Quality Assurance									√					½										√
Reliability				√	√									½			√			½				
Verification and Validation Processes			1	2	2			1												1				
Verification and Validation			√2	√ √	√ √			√2												√				
ISO/IEC 12207 Organizational Life Cycle Processes				3	1	3			1							6	1			4				1
Management Process				3	1	1			1							4	1			4				1
Estimation				½																				
Human Factors/Human Resources						√										√				√ √				
IT Management																	√							

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Productivity Tools																								√	
Project Control																½									
Project Planning																½									
Project/Software Management				√					√							√									√
Risk/Cost-benefit Analysis				√	√																				
Software Economics																				√					
Strategic Management																				√					
Improvement Process						2										2									
Software/Systems Process						√																			
Software Process Improvement						√																			
Software Process Modeling																√ √									
Special Topics			1	2	1	5		2	4			3	1	2		1				1	1	3	2	2
CASE																						√		√
Client-Server Systems																							√	√
Measurement/Metrics				½		√		$√_1$				√	√	½								√		
Real-Time Software/Embedded Systems						√ √ √		$√_2$	√ √			√										√	√	
Reuse									½			√				√				½				
Software Security/Safety			√	√	√	√			√					½							√			
Other Courses			7	4	7	5		4	8	7		6	1	10	4	2	31			7	19	1	14	7
ADA													√	√										
Algorithms										√														
Artificial Intelligence			√			√						√		√			√ √			√	√ √ √		√ √	√
Compilers																	√			√				
Computational Geometry																	√ √							
Computational Theory										√														

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Computer Architecture										√				√			√ √				√		√	
Computer Graphics			√									√					√				√		√	
Computer Language Processing			√																					
Computer Performance									√								√				√			
Computer Technology				√	√																			
Computer Vision																	√							
Computer/Engineering Optimization																	√ √							
Control Systems																	√							
Current Trends in Software Engineering						√																		
Current/Special/Advanced Topics in SE						√						√					√			√	√			
Data Bases/Data Management			√	√	√ $\sqrt{_1}$			$\sqrt{_1}$ $\sqrt{_2}$				√		√	√[1]		√ √			√	√ √	√	√	½
Data Structures										√														
Digital Systems										√														
Distributed Systems				√	√	√						√		√										√
Domain Analysis																√								
Expert Systems			√											√										√
Hardware Acquisition																				½				
Knowledge-based Systems			√												√[1]									
Legal Aspects of Software												√			√[1]									
Management and Behavioral Science										√														
Mathematics										√														
Microprocessors																							√ √	
Multimedia					$\sqrt{_1}$												√						√ √	
Networks/Networking								$\sqrt{_1}$	√ √ √					√			√ √ √				√√√√ √			½
New Technologies															√[1]									

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Object-Oriented Databases																								√	
Operating Systems																	√ √			√	√			
Organizational Management														√		√								
Parallel Systems			√														√							
Protocols								$√_2$													√			
Robotics																	√						√	
Semiconductors																	√							
Simulation						√								√			√ √ √						√	
Software Architecture/IS Architectures				√	√				√															
Telecommunications/Communication Systems					$√_1$				√ √ √								√ √ √			√	√ √ √		√	½
Unix														√									√	√
Number to choose			√:? - $√_i$:1	5	√:4 – $√_1$:1			$√_1$:2 - $√_2$:1	5	1		9 hrs.	3	4	1	2	?			3	*	4	E:5	E:8
Credits to meet						6-9																		
Electives - Disciplines specified																								
Computer Science	√	√						√		√														
Chemical Engineering								√																
Design (not software design)		√																						
Electrical Engineering		√						√																
Industrial Engineering								√																
Information Systems	√																							
Languages		√																						
Management/Administration		√																						
Mechanical Engineering								√																
Nuclear Engineering								√																

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Philosophy/Ethics		√																							
Psychology/Cognitive Science		√																							
Social Science		√																							
Software Engineering		√								√															
Number of courses to choose								2		2															
Credits to meet	4 - 24	30%																							
Electives - Non specified																									
Number to choose				2	2					2			3												
Credits to meet						6-9		6																	
Project - Thesis																									
Project/Studio (credits)	√ √ (8)	√ (40 %)	√ (9)	√ (1 course)	1 course + 1 Masters	√ (3) or		√ (6)	√ (6) or		√ √ (?)	√ (9)		√ (2 courses)	√² √² (12)	√ (?)	√ (?)	√ (6 hrs) or	√ (10-20 pts)	√ (3)	√ (?) or	√ (6)	√ (6)	√3 months
Thesis (credits)				√ (?)		√ (9)			√ (6)									√ (6 hrs)			√ (?)			

√¹: System Artchitecture Specialization
√²:Quality Control Specialization
√¹: Telecommunications option
√¹: SI option
√²:Integration Option
√¹: Safety Track
√²:Reuse/Reengineering Track
√³:Information Management Track
*Depend on concentration chosen

√: Required
½: Part of a course
MS: Master of Science
M.Sc.A.: Master in Applied Science
IS: Information Systems
IT: Information Technology

Summary	Andrews University - Berrien Springs, MI MS in Software Engineering		Carnegie Mellon University - Pittsburgh, PA Master of SE		Concordia - Montreal, QC Master in Computer Science - SE Option		DePaul University - Chicago, IL MS in SE - Software Development Concentration		DePaul University - Chicago, IL MS in SE - Project Management Concentration		Embry-Riddle University - Daytona Beach, FL Master of SE		Flinders University of South Australia, Australia Master of SE		Kansas State University - Manhattan, KS Master of SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Introduction to Software Engineering	2						1	1	1	1	1				1	
ISO/IEC 12207 Primary Processes	2		2		4	6	6	13	5	13	2	3			1	3
Acquisition Process																
Development Process	2		2		3	6	6	12	5	12	2	3			1	3
General Subjects			1			1	1	3	1	3		1				1
System/Software Requirements Analysis	2		1		1	2	2	3	1	3	1	1			1	
Software Detailed Design					1	2	2	2	2	2	1	1				1
Software Coding							1	2	1	2						
System/Software Testing					1	1		2		2						1

Summary	Monmouth University, West Long Branch, NJ MS in SE		National Technological University - Fort Collings, CO MS in SE		National University - La Jolla, CA MS in SE		Seattle University - Seattle, WA Master of SE		Southern Methodist University - Dallas, TX MS in SE		Texas Christian University - Fort Worth, TX Master of SE		Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE		University of Calgary - Calgary, AL MSc with Specialization in SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Introduction to Software Engineering		1			2				1	1	1					
ISO/IEC 12207 Primary Processes	4	1	7		1		3	5	2	5	3	3	5		2	8
Acquisition Process										1						
Development Process	4	1	7		1		3	4	2	3	2	3	4		2	8
General Subjects			2													2
System/Software Requirements Analysis	2		2				1	1	1	1			2		2	1
Software Detailed Design	1		1				1	2		2	1	2	1			4
Software Coding		1	1				1	1				1				1
System/Software Testing	1		1		1				1		1		1			

Summary	University of Colorado - Colorado Springs, CO Master of Engineering - option SE		University of Houston - Clear Lake - Houston, TX MS in SE		University of Karlskrona/Ronneby - Sweeden MS in SE		University of Maryland - College Park, Maryland Master of SE		University of Missouri-Kansas City MS in Computer Science - SE Concentration		University of Scranton - Scranton, PA MS in SE		University of St. Thomas - Minneapolis, Minnesota MS in SE		University of Stirling - Stirling, Scotland MS in SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Introduction to Software Engineering							1		1			1				
ISO/IEC 12207 Primary Processes	4	1	4		3		2	2	2	2	4	1	3	6		6
Acquisition Process								1								
Development Process	3	1	4		3		2	1	2	2	4		3	6		6
General Subjects		1	2		1		1		1		1		1	2		1
System/Software Requirements Analysis	2		1		1		1		1		2		1	1		2
Software Detailed Design	1										1		1	2		2
Software Coding										2				1		1
System/Software Testing			1		1			1								

Summary	Andrews University - Berrien Springs, MI MS in Software Engineering		Carnegie Mellon University - Pittsburgh, PA Master of SE		Concordia - Montreal, QC Master in Computer Science - SE Option		DePaul University - Chicago, IL MS in SE - Software Development Concentration		DePaul University - Chicago, IL MS in SE - Project Management Concentration		Embry-Riddle University - Daytona Beach, FL Master of SE		Flinders University of South Australia, Australia Master of SE		Kansas State University - Manhattan, KS Master of SE		Monmouth University, West Long Branch, NJ MS in SE		National Technological University - Fort Collings, CO MS in SE		National University - La Jolla, CA MS in SE		Seattle University - Seattle, WA Master of SE		Southern Methodist University - Dallas, TX MS in SE		Texas Christian University - Fort Worth, TX Master of SE		Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE		University of Calgary - Calgary, AL MSc with Specialization in SE		University of Colorado - Colorado Springs, CO Master of Engineering - option SE		University of Houston - Clear Lake - Houston, TX MS in SE		University of Karlskrona/Ronneby - Sweeden MS in SE		University of Maryland - College Park, Maryland Master of SE		University of Missouri-Kansas City MS in Computer Science - SE Concentration		University of Scranton - Scranton, PA MS in SE		University of St. Thomas - Minneapolis, Minnesota MS in SE		University of Stirling - Stirling, Scotland MS in SE	
Maintenance Process					1			1		1												1				1										1								
ISO/IEC 12207 Supp. Life Cycle Processes					3	1		3		3	1					1	1	1	2		1		1		2		1	2	2		1		1	1	1		2			2			1		1
Quality Assurance Process					2			1		1	1							1	2				1		1			2	1		1		1	1			1			1			1		1
Verification and Validation Process					1	1		2		2						1	1		1		1				1		1		1				1				1								
ISO/IEC 12207 Org. Life Cycle Processes	1		1		2			3	2	1	1	3			1		1	1	2		1		1		2		1		1		1	6	2	1	2		2		3	4			1		1
Management Process	1		1		2			3	2	1	1	1			1			1	1		1		1		2		1		1			4	2	1	1		2		1	4			1		1
Improvement Process												2					1		1													1	2			1				2						
Special Topics					1	1		2	1	1		5				2		4	1					3		1		2				1			2-3		1			1		3		2		1
Other Courses	4		2		1-2	7	4	4	3	7	1	5				4	3	8		7	7		5	6		1	1	10	1-3	4	2	2	2	31	2-3				2	7	4	19	2	1	2	14		7

Appendix H.
Graduate Programs in Software Engineering - Classification of Courses by Related Discipline

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Credits	48 (quarter credits)	?	45	?	?	36 (credit hours)	72 units	33 (credit hours)	36	33 (sem. credits)	60 (quarter units)	45 (grad. credits)	30 (sem. credits)	31 (sem. hours)	45	?		36 (credit hours)	40 pts	36		36	42	
Grades				2.5/ 4.0	2.5/ 4.0						3.0		3.0/ 4.0				3.0				3.0			
Core Courses																								
Communication												1		1									1	
Technical Communication and Writing												√		√									√	
Computer Science	3		2-4	4	3					1	3	3			1 - 2		1	1 - 3		2	3	1	1	
Algorithms												½									√			
Artificial Intelligence											√													
C++				√	√																			
Concurrent Systems																		√[1]						
Data Bases/Data Management	√										√ √						√	√[2] √[3]					√	
Data Structures	√											½												
Distributed Systems			√												√			√[3]						
Foundations of Computer Science/SI				√ √ √	√ √										√[1]							√		
Human/User Interface/Interaction			√[1]																					
Implementation										√														
Information Security																				√				
Operating Systems	√																							
Programming Methods												√												
Protocols																					√			
Real-Time Software/Embedded Systems																		√[1]						
Software Security/Safety																		√[1]						
Electrical Engineering	1								1		1						1							

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Computer Architecture	√																√							
Hardware and Software Integration											√													
Wireless									√[1]															
Management															1		1							
IT Management																	√							
Organizational Management															√[1]									
Mathematics					1							1										1		
Data Analysis and Regression					√																			
Mathematics												√										√		
Project Management	1	1	2		2	1		1		1	1	1	2	1	2		1	1	2	1		1	1	
Estimation					½																			
Productivity			√[2]												½[1]									
Project Planning													½											
Project/Software Management	√	√	√		√	√		√		√	½	√	½	√	1		√	√	½	√		√	√	
Software Economics																			√					
Software Engineering	4	4	4-5	6	6	5		2	5	10	4	4	4	4	6 - 7	5	5	4 - 8	5	5	2	4	5	
Analysis of Software Artifacts		√																						
CASE																							√	
Current Trends in Software Engineering																√								
Current/Special/Advanced Topics in SE											√													
Design				½	½	½			√	√		√		½	√		√					√	½	
Formal Methods/Specification languages		√		√				√	√	√					√		√	√√[2]√[3]		√	√	√		
Life Cycle Models										√										√				
Maintenance			√											½	√		√							
Measurement/Metrics					½					√								√[2] √[4]	√					

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Object-Oriented																			√					
Object-Oriented Design/Modeling				√	√																			
Object-Oriented Development					√																			
Quality Assurance			√ √[2]			√				√		√	½		½ ½[1]	√	√		½				√	
Quantitative Approach to Engineering																				√				
Reengineering			√[1]															½[2] ½[4]						
Reliability										√														
Requirements Analysis/Specification	√ √		√	½	½	√			√	√		√	√		√	√ √	√		√			√	½	
Reuse																		½[2] ½[4]						
Software Architecture/IS Architectures		√				½						√						√						
Software Development																		√					√	
Development Environments and Tools										√								√[4]		√	√			
Software Engineering Principles				√	√						√													
SE Methods/Methodologies		√		√						√													√	
Software Engineering/IS Engineering	√ √					√		√			√		√	√						√	√			
Software/Systems Processes									√							√		√		√				
Verification, Validation and Testing			√[2]						√	√	√		½	½	½			√[1] √[4]	√					
Telecommunication/Networks			1						2		1										2			
Networks/Networking									√[1] √[1]		√										√			
Telecommunications/Comm. Systems			√[1]																		√			
Others				1							1					1								
Research Process				√							½					√								
Electives - Courses Specified																								

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Application Domains																1								
Domain Analysis																√								
Computer Science			8	9	9	9	8	5	4	3		8	3	10	3	2	21			4	12	3	15	8
ADA													√	√										
Algorithms										√														
Artificial Intelligence			√			√						√		√			√ √			√	√ √ √		√ √	√
Client-Server Systems																							√	√
Compilers																	√			√				
Computational Geometry																	√ √							
Computational Theory										√														
Computer Graphics			√				√					√					√				√		√	
Computer Language Processing			√																					
Computer Performance									√								√				√			
Computer Technology				√	√																			
Computer Vision																	√							
Computer/Engineering Optimization																	√ √							
Control Systems																	√							
Data Analysis and Statistical Software				√ √	√																			
Data Bases/Data Management			√	√	√ $√_1$		√ √ √	$√_1 √_2$				√		√	√[1]		√ √			√	√ √	√	√	½
Data Structures										√														
Distributed Systems				√	√	√						√		√										√
Expert Systems			√											√										√
Human/User Interface				√	√	√	√					√ √	√	√		√ √							√ √	√
Knowledge-based systems			√												√[1]									
Measurement/Metrics				½		√		$√_1$				√	√	½								√		

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Multimedia					√1												√						√ √	
New Technologies															√1									
Object-Oriented Databases																							√	
Operating Systems																	√ √			√	√			
Parallel Systems			√				√										√							
Programming							√ √																	
Programming Languages				√	√		√														√ √		√	½
Protocols								√2													√			
Real-Time Software/Embedded Systems						√ √ √		√2	√ √			√										√	√	
Robotics																	√						√	
Simulation						√								√			√ √ √						√	
Software Security/Safety			√	√	√	√			√					½							√			
Unix														√									√	√
Electrical Engineering										2				1			3				1		3	
Computer Architecture										√				√			√ √				√		√	
Digital Systems										√														
Microprocessors																							√ √	
Semiconductors																	√							
Ethics/Legal Aspects												1			1									
Legal Aspects of Software												√			√1									
Management						1				1				1		2	1			3				
Human Factors/Human Resources						√										√				√ √				
IT Management																	√							
Management and Behavioral Science										√														
Organizational Management														√		√								

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Strategic Management																				√				
Mathematics										1														
Mathematics										√														
Project Management				3	1				1						1	3				1			1	1
Estimation				½																				
Productivity Tools																							√	
Project Control																½								
Project Planning																½								
Project/Software Management				√					√							√								√
Risk/Cost-benefit Analysis				√	√																			
Software Economics																				√				
Software Engineering			6	12	13	6	6	3	5			5	2	4		9	3			4	1	3	3	6
CASE																						√		√
Current Trends in Software Engineering						√																		
Current/Special/Advanced Topics in SE						√	√					√					√			√	√			
Design			$√_1$													√								√
Formal Methods/Specification languages			$√_2$	√	√√	√						√												√
Implementation									½															
Maintenance				√	√							√	√									√		
Object-Oriented			√				√									√							√	
Object-Oriented Analysis													√			½								
Object-Oriented Design/Modeling			$√_1$	√	√			$√_1$					√	√		½								
Object-Oriented Development				√		√		$√_1$									√						√	√
Object-Oriented Programming				√	√							√		√		√								
Quality Assurance							√		√					½										√

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Reliability				√	√									½			√			½				
Requirements Analysis/Specification			$\surd_1$				√																√	√
Reuse									½			√				√				½				
Research Topics in Software Engineering							√																	
Software Architecture/IS Architectures				√	√				√															
Software Development				√	√											√								
Software Development Methods				√	√ √																			
Software Engineering/IS Engineering				√	√				√				√									√		
Software Process Improvement						√																		
Software Process Modeling																√ √								
Software/Systems Process						√																		
Tools for Software Engineering							√																	
Verification, Validation and Testing			$\surd_2$	√ √	√ √			$\surd_2$												√				
Telecommunication/Networks					1			1	6					1			6			1	8		1	2
Networks/Networking								$\surd_1$	√ √ √					√			√ √ √				√√√√ √			½
Telecommunications/Comm.					$\surd_1$				√ √ √								√ √ √			√	√ √ √		√	½
Others													1							2				
Hardware Acquisition																				½				
Software Acquisition													√							½				
Number to choose			√:? - $\surd_i$:1	5	√:4 – $\surd_1$:1			$\surd_1$:2 - $\surd_2$:1	5	1		9 hrs.	3	4	1	2	?			3		4	5	8
Credits to meet						6-9	27 units																	
Electives - Disciplines specified																								
Computer Science	√	√						√		√														

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Information Systems	√																							
Electrical Engineering		√						√																
Other Engineering Disciplines																								
Chemical Engineering								√																
Industrial Engineering								√																
Mechanical Engineering								√																
Nuclear Engineering								√																
Design (not software design)		√																						
Languages		√																						
Management		√																						
Ethics/Legal Aspects		√																						
Psychology/Cognitive Science		√																						
Social Science		√																						
Software Engineering		√								√														
Number of courses to choose								2		2														
Credits to meet	4 - 24	30%																						
Electives - Non specified																								
Number to choose				2	2					2			3											
Credits to meet						6-9		6																
Project - Thesis																								
Project/Studio (credits)	√√ (8)	√(40%)	√ (9)	√ (1 course)	1 course + 1 Master	√ (3) or	√√ (?)	√ (6)	√ (6) or		√√ (?)	√ (9)		√ (2 course)	√² √² (12)	√ (?)	√ (?)	√ (6 hrs) or	√ (10-20 pts)	√ (3)	√ (?) or	√ (6)	√ (6)	√ 3 months
Thesis (credits)				√ (?)		√ (9)	√ √ (?)		√ (6)									√ (6 hrs)			√ (?)			

√¹: System Architecture Specialization

√¹: Telecommunications option

√¹: SI option

√¹: Safety Track √²:Reuse/Reengineering Track

	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineering with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
			√2:Quality Control Specialization												√2:Integration Option			√3:Information Management Track		√4:Software Processes Track					

½: Part of a course

MS: Master of Science

M.Sc.A.: Master in Applied Science

IS: Information Systems

IT: Information Technology

Summary

	Andrews University - Michigan MS in Software Engineering		Carnegie Mellon University - Pittsburgh, PA Master of SE		Concordia - Montreal, QC Master in Computer Science - SE Option		DePaul University - Chicago, IL MS in SE - Software Development Concentration		DePaul University - Chicago, IL MS in SE - Project Management Concentration		Embry-Riddle University - Daytona Beach, FL Master of SE		Flinders University of South Australia, Australia Master of SE		Kansas State University - Manhattan, KS Master of SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Application Domains																
Communication																
Computer Science	3	D		D	2-4	8	4	9	3	9		9		8		5 D
Design				D												
Electrical Engineering	1			D	1											D
Ethics/Legal Aspects				D												
Languages				D												
Management				D								1				
Mathematics									1							
Other Engineering Disciplines																D
Project Management	1		1		2			3	2	1	1				1	
Psychology/Cognitive Science				D												
Social Science				D												
Software Engineering	4		4	D	4-5	6	6	12	6	13	5	6		6	2	3
Telecommunications/Networks					1					1						1
Others							1									

	Monmouth University, West Long Branch, NJ MS in SE		National Technological University - Fort Collings, CO MS in SE		National University - La Jolla, CA MS in SE		Seattle University - Seattle, WA Master of SE		Southern Methodist University - Dallas, TX MS in SE		Texas Christian University - Fort Worth, TX Master of SE		Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE		University of Calgary - Calgary, AL MSc with Specialization in in SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Application Domains																1
Communication							1				1					
Computer Science		4	1	3 D	3		3	8		3		10	1-2	3		2
Design																
Electrical Engineering	1		1	2								1			1	
Ethics/Legal Aspects								1						1		
Languages																
Management				1								1	1			2
Mathematics				1			1									
Other Engineering Disciplines																
Project Management		1	1		1		1		2		1		2	1		3
Psychology/Cognitive Science																
Social Science																
Software Engineering	5	5	10	D	4		4	5	4	2	4	4	6-7		5	9
Telecommunications/Networks	2	6			1							1				
Others					1					1					1	

	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE		University of Houston - Clear Lake - Houston, TX MS in SE		University of Karlskrona/Ronneby - Sweeden MS in SE		University of Maryland - College Park, Maryland Master of SE		University of Missouri-Kansas City MS in Computer Science - SE Concentration		University of Scranton - Scranton, PA MS in SE		University of St. Thomas - Minneapolis, Minnesota MS in SE		University of Stirling - Stirling, Scotland MS in SE	
	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available	Required Courses	Elective Courses Available
Application Domains																
Communication													1			
Computer Science	1	21	1-3				2	4	3	12	1	3	1	15		8
Design																
Electrical Engineering		3							2	1				3		
Ethics/Legal Aspects																
Languages																
Management	1	1						3								
Mathematics											1					
Other Engineering Disciplines																
Project Management	1		1		2		1	1			1		1	1		1
Psychology/Cognitive Science																
Social Science																
Software Engineering	5	3	4-8		5		5	4	2	1	4	3	5	3		6
Telecommunications/Networks		6						1	2	8				1		2
Others								2								

Summary	Andrews University - Michigan MS in Software Engineering	Carnegie Mellon University - Pittsburgh, PA Master of SE	Concordia - Montreal, QC Master in Computer Science - SE Option	DePaul University - Chicago, IL MS in SE - Software Development Concentration	DePaul University - Chicago, IL MS in SE - Project Management Concentration	Embry-Riddle University - Daytona Beach, FL Master of SE	Flinders University of South Australia, Australia Master of SE	Kansas State University - Manhattan, KS Master of SE	Monmouth University, West Long Branch, NJ MS in SE	National Technological University - Fort Collings, CO MS in SE	National University - La Jolla, CA MS in SE	Seattle University - Seattle, WA Master of SE	Southern Methodist University - Dallas, TX MS in SE	Texas Christian University - Fort Worth, TX Master of SE	Université du Québec à Montréal - Montreal, QC M.Sc.A. in SE	University of Calgary - Calgary, AL MSc with Specialization in in SE	University of Colorado - Colorado Springs, CO Master of Engineerig with option in SE	University of Houston - Clear Lake - Houston, TX MS in SE	University of Karlskrona/Ronneby - Sweeden MS in SE	University of Maryland - College Park, Maryland Master of SE	University of Missouri-Kansas City MS in Computer Science - SE Concentration	University of Scranton - Scranton, PA MS in SE	University of St. Thomas - Minneapolis, Minnesota MS in SE	University of Stirling - Stirling, Scotland MS in SE
Research Process				1							1					1								
Hardware Acquisition																				1				
Software Acquisition													1							1				

Appendix I.
Draft Classification of Knowledge on Formal Methods Based on the Proposed Four-Category Schema

Introduction

As an illustration of how the subject matter of a Knowledge Area could be broken down into *Generally Accepted*, *Advanced*, *Research* and *Specialized*, we present in the following pages some key areas of formal methods. For this presentation, these topics were also broken down according to the main life cycle phases, without using the 12207 vocabulary. The various topics included were identified from an informal literature survey done over the last few years (see below for the references which were examined) and which gave rise to an annotated bibliography (the annotations are in French) currently containing over 500 entries. This bibliography on formal methods is available at the following URL, where it can be searched:

http://www.info.uqam.ca/~tremblay/chercher-reference.cgi

The categorization into Generally accepted, Advanced, Research and Specialized was obtained*, grosso modo*, as follows:

- **Generally accepted**: A topic discussed in a number of (mainstream) software engineering textbooks.
- **Advanced**: A topic discussed in numerous formal methods related books or papers. Note that this category also includes a topic (program derivation and verification) discussed in numerous books, even basic programming ones, but rarely used in practice.
- **Research**: A topic discussed in a few (more than 1) formal methods research papers.
- **Specialized:** A topic relevant to only certain types of software.

The references were obtained mainly, but not exclusively, from the followings:

- **Books**: Mainstream software engineering books and books specifically targeted to formal methods.
- **Journals**: ACM Computing Surveys, CACM, ACM Sigplan Notices, ACT TOPLAS, Computer Networks and ISDN Systems, IEEE Computer, IEEE Software, IEEE Trans. on Soft. Eng., Journal of Systems and Software, Science of Computer Programming, Software -- Practice and Experience, The Computer Journal.
- **Conferences**: CONCUR, FME, VDM, AMAST, Computer-Aided Verification, Intl. Conf. on Soft. Eng., Protocol Specification, Testing and Verification, TAPSOFT, ZUM.

Requirements analysis and specification

Generally Accepted

- Formal specification of the abstract behavior of a system (black box functional specification) using an abstract model or axiomatic specifications, with pre/post-conditions (e.g., VDM, Z, Larch two-tier approach) [Lam88, Pre92, GH93, Som95, Pfl98].

Advanced

- Verification of the internal consistency of a specification by generating and discharging appropriate proof obligations (using rigorous inspection and/or formal proofs) [Jon86,Sha95,BDMW97].

Research

- Formal specifications of the abstract behavior of a system using various approaches, e.g., assertions on traces [BP78,Jan97], Petri nets [Rei87,Fur93, BOP97], Statecharts [Har88,HG97], etc.

- Animation of formal specifications and/or use of formal specifications for prototyping in order to validate the requirements [HI88,BM93,WP94,BDMW97].
- Formal specification of (concrete) person-machine interfaces [Ale90,KB97].
- Integration of formal methods with existing requirements and analysis approaches (e.g., OO approaches [CHB92,Ca93,AS97,HG97], structured analysis [PvKP91, SFD92,GP95]).

Specialized

- Telecommunication protocol design, telephony, hardware design: Formal specification of the abstract, external behavior of a (finite state) system (e.g. SDL, Lotos, CCS/CSP) + Formal specification of some important properties required and/or expected of the system using modal, temporal logic + Verification of those properties using model-checking [CES86,Tur93,CWa96,Bru97].

Architectural design

Generally accepted

- Formal specification of the behavior of modules using model-based or abstract machine approaches (e.g., VDM, Z, B) [Lam88,ALN+91,Pre92,Som95,Pfl98].

  Specification of abstract data types using algebraic approaches (e.g., Larch, ACT-ONE) [Som95,Lam88,dMRV92,GH93].

Advanced

- Verification of the internal consistency of a module specification by generating and discharging appropriate proof obligations (using rigorous inspection and/or formal proofs) [Jon86,Sha95,BDMW97].

Research

- Formal specification of architectural styles and patterns [AG94,Gar95,CM97].

Detailed design

Advanced

- Verification of the refinement of modules by generating and discharging appropriate proof obligations [Jon86,Sha95,BDMW97,TTOV97].

Coding and testing

Advanced

- Program derivation and formal (*in-the-small*) program verification [Gri81,Dro89,AI91].

Research

- Derivation of test cases based on the formal specification of a module (black-box unit testing) [DF93,SC96,NB92,FJJ+96,Den96,BDMW97].
- Automatic or semi-automatic transformation of specification to synthesize software and/or generate executable code [Par90,Jul93,SH94].

Qualification testing

Research

- Derivation of test cases based on the formal (functional) specification of a system (black-box testing) [DF93,SC96,NB92,FJJ+96,Den96,BDMW97].

References

AG94

R. Allen and D. Garlan.
Formalizing architectural connection.
In *Proc. 16th Int'l Conf. Software Eng.*, pages 71-80. IEEE Computer Society Press, 1994.

AI91

D. Andrews and D. Ince.
Practical formal methods with VDM.
The McGraw-Hill International Series in Software Engineering, 1991.

Ale90

H. Alexander.
Structuring dialogues using CSP.
In *Formal Methods in Human-Computer Interaction*, chapter 9, pages 273-295. Cambridge University Press, 1990.

ALN+91

J.-R. Abrial, M.K.O Lee, D.S. Neilson, P.N. Scharbach, and I.H. Sorensen.
The B-method.
In *VDM '91: Formal Software Development Methods*, pages 398-405. Springer-Verlag, LNCS-552, 1991.

AS97

K. Achatz and W. Schulte.
A formal OO method inspired by Fusion and Object-Z.
In *ZUM '97: The Z Formal Specification Notation*, pages 92-111. Springer-Verlag, LNCS-1212, 1997.

BDMW97

J. Bicarregui, J. Dick, B. Matthews, and E. Woods.
Making the most of formal specification through animation, testing and proof.
Science of Computer Programming, 29(1):53-78, 1997.

BM93

P. Borba and S. Meira.
From VDM specifications to functional prototypes.
J. Systems Software, 21(3):267-278, Mar. 1993.

BOP97

L. Baresi, A. Orso, and M. Pezzè.
Introducing formal specification methods in industrial practice.
In *ICSE '97 (Intl. Conf. on Soft. Eng.)*, pages 56-66, 1997.

BP78

W. Bartussek and D.L. Parnas.
Using assertions about traces to write abstract specifications for software modules.
In *European Cooperation in Informatics*, pages 211-236. Springer-Verlag, LNCS-65, 1978.

Bru97

G. Bruns.
Distributed Systems Analysis with CCS.
International Series in Computer Science. Prentice-Hall, 1997.

Ca93

E. Casais and al.
Formal Methods and Object-Orientation.
Tutorial at TOOLS Europe 1993, 1993.

CES86

E.M. Clarke, E.A. Emerson, and A.P. Sistla.
Automatic verification of finite-state concurrent systems using temporal logic specifications.
ACM TOPLAS, 8(2):244-263, 1986.

CHB92

D. Coleman, F. Hayes, and S. Bear.
Introducing objectcharts or how to use statecharts in object-oriented design.
IEEE Trans. on Soft. Eng., 18(1):9-18, Jan. 1992.

CM97

P. Ciancarini and C. Mascolo.
Analyzing and refining an architectural style.
In *ZUM '97: The Z Formal Specification Notation*, pages 349-368. Springer-Verlag, LNCS-1212, 1997.

CWa96

E.M. Clarke, W.M. Wing, and al.
Formal methods: State of the art and future directions.
ACM Computing Surveys, 28(4):626-643, 1996.

Den96

R. Denney.
A comparison of the model-based & algebraic styles of specification as a basis for test specification.
Soft. Eng. Notes, 21(5):60-65, 1996.

DF93

J. Dick and A. Faivre.
Automating the generation and sequencing of test cases from model-based specifications.
In *FME '93: Industrial-Strength Formal Methods*, pages 268-284. Springer-Verlag, LNCS-670, 1993.

dMRV92

Jan de Meer, Rudolf Roth, and Son Vuong.
Introduction to algebraic specifications based on the language ACT ONE.
Computer Networks and ISDN Systems, 23(5):363-392, 1992.

Dro89

G. Dromey.
Program Derivation -- The Development of Programs From Specifications.
Addison-Wesley Publishers Ltd., 1989.

FJJ+96

J.-C. Fernandez, C. Jard, T. Jéron, L. Nedelka, and C. Viho.
An experiment in automatic generation of test suites for protocols with verification technology.
Technical Report 2923, INRIA, Rocquencourt, Juin 1996.

Fur93

U. Furbach.
Formal specification methods for reactive systems.
J. Systems Software, 21(2):129-139, Feb. 1993.

Gar95

D. Garlan.
Research directions in software architecture.
ACM Computing Surveys, 27(2):257-261, 1995.

GH93

J.V. Guttag and J.J. Horning.
Larch: Languages and Tools for Formal Specification.
Springer-Verlag, 1993.

GP95

C. Gaskell and R. Phillips.
A structured analysis formalism with execution semantics to allow unambiguous model interpretation.
In *Software Engineering -- ESEC '95*, pages 235-253, 1995.

Gri81

D. Gries.
The Science of Programming.
Springer-Verlag, 1981.

Har88

D. Harel.
On visual formalisms.
Comm. of the ACM, 31(5):514-530, May 1988.

HG97

D. Harel and E. Gery.
Executable object modeling with Statecharts.
IEEE Computer, 30(7):31-42, 1997.

HI88

S. Hekmatpour and D. Ince.
Software Prototyping, Formal Methods and VDM.
Addison-Wesley Publishing Co., 1988.

Jan97

R. Janicki.
Foundations of the trace assertion method of module interface specification.
Technical Report CRL Report 348, McMaster University, 1997.

Jon86

C.B. Jones.
Systematic Software Development using VDM.
Prentice-Hall International Series in Computer Science, 1986.

Jul93

R.K. Jullig.
Applying formal software synthesis.
IEEE Software, 10(3):11-22, May 1993.

KB97

J.C. Knight and S.S. Brilliant.
Preliminary evaluation of a formal specification to user interface specification.
In *ZUM '97: The Z Formal Specification Notation*, pages 329-346. Springer-Verlag, LNCS-1212, 1997.

Lam88

D.A. Lamb.
Software Engineering: Planning for Change.
Prentice-Hall, 1988.

NB92

K. Naik and Sarikaya. B.
Testing communication protocols.
IEEE Software, 9(1):27-37, Jan. 1992.

Par90

H.A. Partsch.
Specification and transformation of programs: a formal approach to software development.
Springer, 1990.

Pfl98

S.L. Pfleeger.
Software Engineering -- Theory and Practice.
Prentice-Hall, Inc., 1998.

Pre92

R.S. Pressman.
Software Engineering -- A Practitioner's Approach (Third Edition).
McGraw-Hill, Inc., 1992.

PvKP91

N. Plat, J. van Katwijk, and K. Pronk.
A case for structured analysis/formal design.
In *VDM '91: Formal Software Development Methods*, pages 81-105. Springer-Verlag, LNCS-551, 1991.

Rei87

W. Reisig.
Petri nets in software engineering.
In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, pages 63-96. Springer-Verlag, LNCS-255, 1987.

SC96

P. Stocks and D. Carrington.
A framework for specification-based testing.
IEEE Trans. on Soft. Eng., 22(11):777-793, 1996.

SFD92

L.T. Semmens, R.B. France, and T.W.G. Docker.
Integrated structured analysis and formal specification techniques.
The Computer Journal, 35(6):600-610, 1992.

SH94

A.C. Storey and H.P. Haughton.
A strategy for the production of verifiable code using the B method.
In *FME '94: Industrial Benefits of Formal Methods*, pages 346-365. Springer-Verlag, LNCS-873, 1994.

Sha95

N. Shankar.
Computer-aided computing.
In Bernhard M"oller, editor, *Mathematics of Program Construction '95*, number 947 in Lecture Notes in Computer Science, pages 50-66. Springer-Verlag, 1995.

Som95

I. Sommerville.
Software Engineering (Fifth Edition).
Addison-Wesley, 1995.

TTOV97

S. Taouil-Traverson, P. Ozello, and S. Vignes.
Développement formel de logiciel de sécurité: utilisation de la méthode B à la SNCF.
Technique et Sciences Informatique, 16(9):1187-1209, 1997.

Tur93

K.J. Turner.
Using formal description techniques: an introduction to Estelle, LOTOS, and SDL.
Wiley series in communication and distributed systems, 1993.

WP94

Y. Wang and D.L. Parnas.
Simulating the behavior of software modules by trace rewriting.
IEEE Trans. on Soft. Eng, 20(10):750-759, 1994.

Appendix J.
Additional Information on Other Body of Knowledge Proposals

The Joint Steering Committee of the IEEE Computer Society and the ACM for the Establishment of Software Engineering as a Profession established a task force in 1996 to conduct exploratory work on the issue of the software engineering body of knowledge. The task force designed and conducted a pilot survey on a sample of tasks that could be considered within the scope of software engineering[28]. The survey asked whether each task described would be expected to be performed by a "novice software engineer", an "expert software engineer", a "software engineering specialist" or a "manager" in the organization.

The Institute for Certification of Computer Professionals (ICCP)[29], a non-profit organization, offers a certification program for software practitioners entitled Certified Computing Professional (CCP). The ICCP states that there are currently 50,000 certificate holders. To obtain this certificate, a candidate must have at least 48 months of direct full-time experience in computer-based information systems. A portion of this experience requirement may be substituted with post-secondary education. Additionally, candidates must successfully pass three exams, one of which is to be chosen from among various different topics, including software engineering. The topics covered in the software engineering exam are: computer systems engineering, software project planning, software requirements, software design, programming languages and coding, software quality assurance, software testing techniques, software maintenance and configuration management.

The Software Quality Engineers program (SQE) is a certification program of the American Society for Quality (ASQ)[30]. Obtaining this certificate also requires experience, which can be partly waived with post-secondary education and by passing a 4-hour, 160-question exam. The "exam body of knowledge" follows this table of contents:

- general knowledge, conduct and ethics;
- software quality management;
- software processes;
- software project management;
- software metrics, measurement and analytical methods;
- software inspection, testing, verification and validation;
- software audits;
- software configuration management.

The Quality Assurance Institute offers two specialized certification programs related to software engineering: Certified Quality Analyst (CQA) and Certified Software Test Engineer (CSTE)[31]. Obtaining the CQA certificate requires a bachelor's degree, which can be waived with an Associate's degree and/or experience, a character reference and successfully passing a four hour, four part exam. The "Common Body of Knowledge for the Information Systems Quality Assurance Profession" is provided as study material for this exam. It describes knowledge in the following areas:

[28] The report on the survey results can be found at computer.org/tab/seprof/survey.htm
[29] See http://www.iccp.org/profess.html
[30] See http://www.asq.org/about/divtech/softdiv/topcert.htm
[31] See www.qaiusa.com

- Auditing and Control
- Change Management
- Communications
- Disaster Recovery
- Human Resource Principles
- Management Techniques
- Principles of I/S
- Quality Assurance
- Quality Control Techniques
- Quality Management
- Quantitative Methods
- Reviews
- Standards
- Testing
- Training and Development
- Vendor Control

To obtain the Certified Software Test Engineer (CSTE) certificate, candidates must have direct experience in software testing and must be able to show proficiency in six software testing skills via a resume and other supporting documents. Candidates must, as of January 1999, successfully pass an exam on the CSTE common body of knowledge. This body of knowledge includes sixteen knowledge domains grouped into four categories:

- Test management: communication, professional development, testing concepts and test environments;
- Test planning: risk analysis, development methods and environment, test methods and techniques, and planning process;
- Test execution: verification methods, test tools, test-case design and performing tests;
- Test results analysis and reporting: defect tracking and management, evaluating test results, quantitative methods and test reporting.

Parnas proposes in [16] that the development of a body of knowledge in software engineering must begin with the identification of tasks performed by software engineers. He then goes on to propose a list of nine tasks:

- Analyze the intended application to determine the requirements that must be satisfied, and record these requirements in a precise, well-organized and easy-to-use document.
- Participate in the design of the computer system configuration, determining which functions will be implemented in hardware and which functions will be implemented in software, and selecting the basic hardware and software components.
- Analyze the performance of a proposed design (either analytically or by simulation) to make sure that the proposed system can meet the application's requirements.
- Design the basic structure of the software, its division into modules, the interfaces between these modules and the structure of individual programs, while precisely documenting all software design decisions.

- Analyze the software structure for completeness, consistency and suitability for the intended application.
- Implement the software as a set of well-structured and well-documented programs.
- Integrate new software with existing or "off the shelf" software.
- Perform systematic and statistical testing of the software and integrated computer system.
- Revise and enhance software systems, maintaining their conceptual integrity and keeping documents complete and accurate.

Parnas also points out that many other topics important to software engineers, such as project management, are at the core of engineering as a whole and hence should not be included in the software engineering body of knowledge.

Hilburn *et al.* in [26] recently proposed a body of knowledge for software engineering divided into four major knowledge areas, which are then divided into knowledge components. These are:

- Core knowledge area:
  - Software requirements
  - Software design
  - Software construction
  - Software project management
  - Software evolution
- Foundations area:
  - Computing fundamentals
  - Human factors
  - Application domains
- Recurring area:
  - Ethics and professionalism
  - Software processes
  - Software quality
  - Software modeling
  - Software metrics
  - Tools and environments
  - Documentation
- Supporting area: this area includes other fields of study which complete the education of software engineers such as "general education", mathematics, natural sciences and business studies.

"Software Engineering and Methodologies" has also been incorporated as a "knowledge area" or unit in the Core Body of Knowledge for Information Technology Professionals[32] published by the Australian Computer Society. The topics covered in this unit are:

- Fundamentals of Software Engineering
  - requirements analysis

[32] http://www.acs.org.au/national/pospaper/bokpt1.htm

- functional and technical specifications
- process, data and object orientation models
- documentation standards
- software testing
- software maintenance
- software quality assurance
- formal specification methods
- software configuration management

• Project Management
- project planning, estimation and control
- project evaluation and control techniques
- team construction and management
- principles of software project management
- prototyping

A model curriculum and guidelines for undergraduate degree programs in information systems entitled IS'97[33] has recently been published after going through a very serious comment-gathering and review process. This model curriculum was produced through a collaborative effort of the Association for Computing Machinery (ACM), the Association for Information Systems (AIS) and the Association of Information Technology Professionals (AITP). The draft curriculum was reviewed at eleven national and international meetings involving over 1,000 individuals from industry and academia. A body of knowledge for information systems that includes many software engineering elements is proposed in IS'97.

[33] See http://webfoot.csom.umn.edu/faculty/gdavis/curcomre.pdf