https://doi.org/10.48009/1_iis_2015_28-30

Issues in Information Systems
Volume 15, Issue I, pp. 28-30, 2015

THE DILEMMA BETWEEN SOFTWARE ENGINEERING AND SYSTEMS ANALYSIS & DESIGN

Mohammad A. Rob, University of Houston-Clear Lake, rob@uhcl.edu

ABSTRACT

Software Engineering (SE) and Systems Analysis & Design (SA&D) are two subjects that are taught in two different academic disciplines. SE is focused on developing a software while SA&D is focused on developing an information system. An information system cannot be built without building its software. So why do we need to cover so many courses in SE as compared to just one in SA&D? What is the relationship between the two subjects and what are differences? Analyzing various textbooks, we found that "Quality" is the major focus in software engineering and the method of developing an information system is the main focus in systems analysis & design. We have also identified, that in addition to selecting a process model, software engineers focus on five Quality Management Metrics (5QMMs) to develop a quality software; however, only one of them, namely the project management is included in the SA&D texts. On the other hand, the identity of SE and SA&D texts are getting blurred, due to the market demand of large and complex business systems over engineering and technical systems that dominated before.

Keywords: Systems Analysis and Design, Software Engineering, SDLC, Process Model, Quality.

INTRODUCTION

Software Engineering (SE) is typically a stand-alone program in the computer science area, while Systems Analysis & Design (SA&D) is a course typically taught in a Management Information Systems (MIS) program. Software engineering is focused on developing a software and SA&D is focused on developing an information system. However, software is a major part of an information system that includes hardware as well as other resources needed to operating an information system. So why do we need a whole software engineering program with many courses as compared to just one course on the systems analysis and design? Where is the connection between them and where do they meet? This paper tries to address some of the intertwined issues relating the two subject areas.

METHODOLOGY

First two major textbooks from two subject areas are compared chapter by chapter, then by the important topics covered. A list of topics is created from the texts. Then other books are included to find the inclusion or exclusion of each of the topics. Thus a matrix is created with rows as topics and columns as textbooks, and then check marks are used for the inclusion of the topics. Studying the matrix, a comparative knowledge is developed on the two subject areas. Further study included detailed understanding of some key topics from research papers and web search. Comparative results are briefly presented in the following and a model is developed based on our findings.

RESULTS

The Meeting and Departing Areas in SE and SA&D

Both SE and SA&D are focused on the phased approach of developing a software or system, commonly known as Software or Systems Development Life Cycle (SDLC). The various approaches of SDLC: waterfall, iterative, spiral, Agile, and others are applicable to both. There are not many textbooks on SE [1-3] as a subject, but there are about ten major textbooks in the on SA&D [4-6]. However, there are about ten books on individual topics in SE such as Requirements Engineering, Software Architecture & Design, Software Construction, Software Testing, Software Maintenance, Software Configuration Management, Software Engineering Management, Software Engineering

Process, Software Engineering Tools and Methods, and Software Quality Management. Recently, some traditional SE topics such as COCOMO, Functions Points, CMM, Use Cases, UML, and Object-Oriented Analysis are being included in the SA&D textbooks.

While most SA&D textbook chapters are highly organized by the SDLC phases and activities following the Waterfall model, a basic SE text is focused on software quality and project management. While SE texts clearly focus on software architecture and quality, there is a clear focus on business process, database and network in SA&D texts. Knowledge gathered from a SE text prepares a Software Engineer or Software Architect, while SA&D prepares a Systems Analyst.

While software engineering traditionally focuses on technical, real-time, embedded and mission critical systems, SA&D focuses on business information systems. According to Wikipedia [7], "Software engineering is the study and application of engineering methods to the design, development, and maintenance of software. It applies a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." The discipline of software engineering was created to address the poor quality of software, get projects exceeding time and budget under control, and ensure that software is built systematically, rigorously, measurably, on time, on budget, and within specification. However, these attributes are addressed in all SA&D texts as well. So what is the "big deal" about "engineering" here? In fact managing a software or system development project can be defined as an art and science of managing scope, time, cost and quality, as two software projects can never be considered same. Early researchers like Barry Boehm [8] and Alan Albrecht [9] came up with cost estimation models for software, which became the de facto standards for software size calculation; but still they are estimations, not perfect science or engineering. Similar arguments apply for process models. So many models have been proposed over time, and still continues to manage software developmental metrics.

The Quality Focus in Software Engineering Texts

In project management, we typically focus on three inter-related constraints: scope, time and cost. Many times quality is considered as the fourth constraint, but for software or rather any kind of consumer product, quality is the most important factor. Software Engineering texts have a major focus on "quality," and the quality is ensured by the management of five software developmental metrics (5QMMs): quality management, project management, requirements management, risk Management, and configuration management. Management of these metrics ensures quality software production. Each of these metrics is a subject or course by itself. However, only the project management is included in the SA&D texts. To summarize, in order to develop a quality software, we adapt a particular process model and focus on 5QMMs to develop that quality product.

One of the most important topic in quality assurance is the adaptability of Capability Maturity Model (CMM). It identifies a quality culture is an organizational environment where quality is viewed as everyone's responsibility. It suggests that the development and maintenance of software should be systematic, disciplined, and quantified [10]. Software development is not just about developing a piece of quality software, it is a continuous improvement process of an organization. Only certain organizations who can attain a certain level of process maturity can develop certain quality of software. Five levels of CMM developed by Watts Humphrey becomes the de facto standard to evaluate the capability of a software development company. To achieve a level beyond the first chaotic level, it requires an organization to focus on certain Key Process Areas (KPAs). Adapting an SDLC process model and the Five Quality Management Metrics (5QMMs) mentioned before can be broken down into Level 2 and 3 key process areas. A Software Engineering text or program typically covers CMM in detail, but how much of that is covered in a SA&D text? Recent editions of some SA&D textbooks started to mention or add a section on CMM, and rightfully mentioned by one of the authors that effective Project Management is essential for an organization to achieve CMM Level 2 and application of a repeatable process model can lead to CMM Level 3 [3].

Software Engineer vs. Systems Analyst

Most SA&D books are very clear in defining the role of a systems analyst, the knowledge required by a systems analyst, and that the information gathered from the textbook will prepare a systems analyst as a professional.

However, Software Engineering texts do not define the role of a software engineer, and the term "Engineer" generally means that an individual holds a professional license, which are controlled by individual states and they have to be renewed at regular intervals. Software engineers typically do not hold a professional license. So, can we say that software engineers are true engineers? That used to be a question to be answered; however, starting from 2013, National Council of Examiners for Engineering and Surveying (NCEES) started a professional licensure exam on Software, thereby allowing Software Engineers to be licensed and recognized [12].

CONCLUSIONS

This paper tries to address some interweaved issues relating two subject areas: Software Engineering (SE) and Systems Analysis & Design (SA&D). Based on the textbook topics in the two subject areas, our analysis shows that "Quality" focus is the key factor in the area of software engineering, while the process models are same in both subjects. We have identified, that in addition to selecting a process model, software engineers need to focus on five Quality Management Metrics (5QMMs) to develop a quality software; however, only one of them, namely the project management is included in the SA&D texts. We have also found that over time, many software engineering concepts are being adapted to the Systems Analysis & Design texts. This trend should continue, and the focus of quality should be broadened in the SA&D texts to bring them in par with the SE texts. Whether software development is an engineering process or not, which might be a debatable issue, but Software Engineers are being recognized as licensed professionals. Furthermore, Software Engineering and Systems Analysis & Design are verging close to each other. It is not the field of engineering or business, but the market demand that is making their identity getting blurred. Thus the Software Engineering texts should continue to broaden their topics that relate to large business systems including databases and web technologies.

REFERENCES

1. Kendall, K. E., & Kendall, J. E. (2008). *Systems analysis and design*, Pearson-Prentice Hall.
2. Shelly, G. B., & Rosenblatt, H. J. (2012). *Systems analysis and design, course technology*. Cengage Learning.
3. Whitten, J. L., Bentley, L. D., & Dittman, K. C. (2004). *Systems analysis and design methods,* McGraw-Hill.
4. Pressman, R. S., & Maxim*,* B. (2014). *Software engineering: A practitioner's approach*, McGraw-Hill Publishing.
5. Sommerville, I. (2010). *Software engineering* (9th ed.), Harlow, England: Pearson Education.
6. Tsui, F., Karam, O., & Bernal, B. (2014). *Essentials of software engineering*, Jones & Bartlett Learning.
7. *Software Engineering*. (2015, March). Available: http://en.wikipedia.org/wiki/Software_engineering
8. Boehm, B. (1981). *Software engineering economics*, Prentice-Hall Publishers, Englewood Cliffs, NJ.
9. Albrecht, A. J. (1979). Measuring application development productivity. *IBM Application Development Symposium*, pp. 83-92.
10. Humphrey, W. S. (1981). *Managing the software process*. Addison-Wesley Publishing.
11. *Software Engineering Body of Knowledge.* (2015). IEEE Computer Society. Available: http://en.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge
12. *NCEES introduces PE exam for software*. (2015). Available: http://ncees.org/about-ncees/news/ncees-introduces-pe-exam-for-software-engineering.