International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

Software System Engineering: Quantification of software using Measures of Effectiveness

Dr.M.Karthika, Assistant Professor

Department of Information Technology
N M S S. Vellaichamy Nadar College
Nagamalai,Madurai, Tami Nadu, India
mkpartha@yahoo.com

K.S. Mohanasundaram, Assistant Professor

Department of Information Technology
N M S S. Vellaichamy Nadar College
Nagamalai,Madurai – 625 019
mohanasundaramks@yahoo.com

ABSTRACT- A system-engineering principle specifically to the development of large, complex software systems provides a powerful tool for process and product management. Software engineering has its early roots in system engineering which is reflected in their many common terms. System engineering looks at controlling the total system development including software. Software engineering looks at controlling just software development. This type of Application can be called as component Engineering. The application of system engineering to the development of software gives a large measure of control software development. Quantification of the software can be done on various factors.

The proposed work describes novel and quantitative software readiness criteria to support objective and effective decision-making at process level. The system performance can be quantified using evaluation criteria, technical performance measures. The performance criteria will show how the system satisfies its requirement. Evaluation criteria are often called the measures of effectiveness. This effectiveness of the complex problem can be attained using optimization techniques which acts as a tool for the measures. The comparison can done through an optimizing technique.

Keywords—System engineering, Optimizing Technique, Performance criteria , Evaluation Criteria, System performance.

Introduction

Systems engineering signifies a formalized approach to identify new methods and research opportunities similar to the way it occurs in other fields of engineering. Systems engineering focuses on analyzing and eliciting customer needs that are required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem, the system lifecycle.The goal of the system engineering is to support some business function or to develop a product that can be sold to generate business revenue. To accomplish the goal, a computer-based system makes use of a variety of system elements like, software, Hardware, people, database,

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

documentation, procedures. Computer programs, data structures, and related documentation that serve to effect the logical method, procedure, or control that is required. Hardware are Electronic devices that provide computing capability, the interconnectivity devices (e.g., network switches, telecommunications devices)that enable the flow of data, and electromechanical devices (e.g., sensors, motors, pumps) that provide external world function. People are the Users and operators of hardware and software .Database are large, organized collection of information that is accessed via software.

Documentation are the descriptive information (e.g., hardcopy manuals, on-line help files, Web sites) that portrays the use and/or operation of the system. Procedures involve the steps that define the specific use of each system element or the procedural context in which the system resides. The elements combine in a variety of ways to transform information. For example, a marketing department transforms raw sales data into a profile of the typical purchaser of a product; a robot transforms a command file containing specific instructions into a set of control signals that cause some specific physical action. Creating an information system to assist the marketing department and control software to support the robot both require system engineering [8].

Software And System Engineering

Software engineering has helped shape modern systems engineering practice. The techniques used in the handling of complexes of large software-intensive systems has had a major effect on the shaping and reshaping of the tools, methods and processes of SE. The integration have been done in models[14].

Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem with operations, cost & schedule, performance, training & support, test, disposal and manufacturing [13].

Systems engineering with assistance from the other engineering disciplines establishes the baseline system design; allocates system requirements; establishes measures of effectiveness for ranking alternative designs; and integrates the design among the design disciplines. Systems engineering is responsible for verifying that the system developed meets all requirements defined in the system specification and for providing the analysis which assures that all requirements will be met.

SwSE begins after the system requirements have been partitioned into hardware and software subsystems. SwSE establishes the baseline for all project software development. Like software engineering, it is both a technical and a management process. The SwSE technical process is the analytical effort necessary to transform user operational needs [3].

A software system is described as Software system requirements and design specifications. Also Provides Necessary procedures to verify, test, and accept the finished software product. They also have, the necessary documentation to use, operate,

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

and maintain it. SwSE is not a job description. It is a process that many people and organizations perform: system engineers, managers, software engineers, programmers, and – not to be ignored – acquirers and users [7].

Microsoft Word is a classic example: A product that would fit on a 360-kilobyte diskette 20 years ago now requires a 600-megabyte compact disc. But there are other reasons for increased size and complexity. Specifically, software has become the dominant technology in many if not most technical systems. It often provides the cohesiveness and data control that enable a complex system to solve problems.

Software provides the system's major technical complexity.

Because of the increase in size and complexity, the vast majority of large software systems do not meet their projected schedule or estimated cost, nor do they completely fulfill the system acquirer's expectations[1]. This phenomenon has long been known as the software crisis [1]. In response to this crisis, software developers have introduced different engineering practices into product development.

As large system solutions become increasingly dependent on software, a system engineering approach to software development can help avoid the problems associated with the software crisis.

System engineering provides the tools the technical management task requires. The application of system engineering principles to the development of a computer software system produces activities, tasks, and procedures called software system engineering (SwSE). Many practitioners consider SwSE to be a special case of system engineering and others consider it to be part of software engineering. However, it can be argued that SwSE is a distinct and powerful tool for managing the technical development of large software projects.

The following are examples of what makes the mechanics of software engineering different than computer science: Dividing the project into phases such as life-cycle development methods. Managing software as a separate project. Using intermediate products (specifications), e.g., requirements specifications, design specifications. Reviewing, testing, and auditing. Using configuration management and quality (process) assurance. Prototyping and the reuse of existing components.

Both SwSE and software engineering are technical and management processes, but software engineering produces software components and their supporting documentation. Specifically, software engineering is the following:

The practical application of computer science, management, and other sciences to the analysis, design, construction, and maintenance of software and its associated documentation.

An engineering science that applies the concepts of analysis, design, coding, testing, documentation, and management to the successful completion of large, custom-built computer programs under time and budget constraints.

The systematic application of methods, tools, and techniques that achieve a stated requirement or objective for an effective and efficient software system.

Figure 1 illustrates the engineering relationships between system engineering, SwSE, and software engineering. Traditional system engineering does

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

initial analysis and design as well as final system integration and testing.

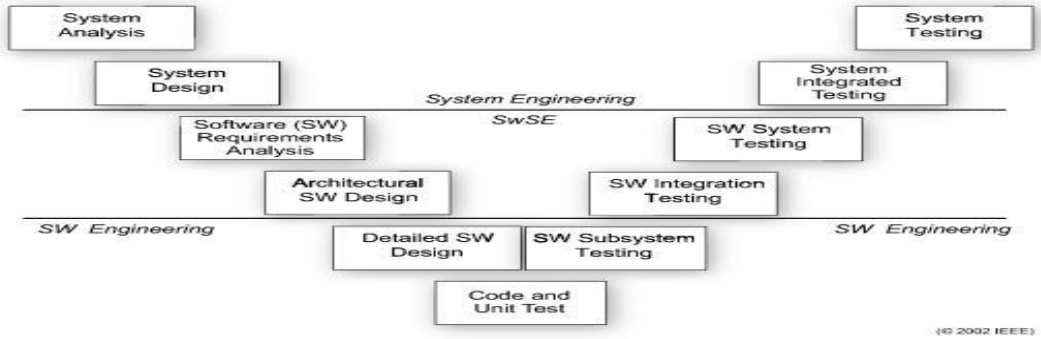

Figure 1: Engineering Relationships

During the initial stage of software development, SwSE is responsible for software requirements analysis and architectural design. SwSE also manages the final testing of the software system component engineering.

THE SYSTEM

A *system* is a collection of elements related in a way that allows a common objective to be accomplished. In computer systems, these elements include hardware, software, people, facilities, and processes.

System engineering is the practical application of scientific, engineering, and management skills necessary to transform an operational need into a description of a system configuration that best satisfies that need. It is a generic problem-solving process that applies to the overall *technical* management of a system development project. This process provides the mechanism for identifying and evolving a system's product and process definitions.

IEEE Std. 1220-1998 describes the system engineering process and its application throughout the product life cycle [3]. *System engineering produces documents, not hardware.* These documents are associated with the developmental processes within the project's life-cycle model. They also define the expected process

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

environments, interfaces, products, and risk management tools throughout the project.

System engineering involves five functions:

- *Problem definition*
- *Solution analysis*
- *Process planning*
- *Process control*
- *Product evaluation*

System engineering provides the baseline for all project development, as well as a mechanism for defining the *solution space*. The solution space describes the product at the highest level – before the system requirements are partitioned into the hardware and software subsystems.

This approach is similar to the software engineering practice of specifying constraints as late as possible in the development process. The further into the process a project gets before defining a constraint, the more flexible the implemented solution will be.

SwSE

The term *software system engineering* dates from the early 1980s and is credited to Dr. Winston Royce [4], an early leader in software engineering. SwSE is responsible for the overall technical management of the system and the verification of the final system products. As with system engineering, SwSE produces documents, not components. This differentiates it from software engineering, which produces computer programs and user manuals.

SwSE begins after the system requirements have been partitioned into hardware and software subsystems. SwSE establishes the baseline for all project software development. Like software engineering, it is both a technical and a management process. The SwSE technical process is the analytical effort necessary to transform user operational needs into the following:

A software system description. Software system requirements and design specifications. Necessary procedures to verify, test, and accept the finished

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

software product. Necessary documentation to use, operate, and maintain it.

SwSE is not a job description. It is a process that many people and organizations perform: system engineers, managers, software engineers, programmers, and – not to be ignored – acquirers and users.

Software developers often overlook system engineering and SwSE in their projects. They consider systems that are all software or that run on commercial off-the-shelf (COTS) computers to be just software projects, not system projects. Ignoring the systems aspects of software development contributes to our long-running software crisis.

Functions of SwSE

Table 1 lists the five main functions of system engineering correlated to SwSE, along with a brief general description of each SwSE function.

Table I System Engineering Functions correlated to SwSE

System Engineering Function	SwSE Function	SwSE Function Description
Problem Definition	Requirements Analysis	Determine needs and constraints by analyzing system requirements allocated to software.
Solution Analysis	Software Design	Determine ways to satisfy requirements and constraints, analyze possible solutions, and select the optimum one.
Process Planning	Process Planning	Determine product development tasks, precedence, and potential risks to the project.
Process Control	Process Control	Determine methods for controlling project and process, measure progress, and take corrective action where necessary.
Product Evaluation	Verification, Validation, and Testing (VV&T)	Evaluate final product and documentation.

(© 2002, IEEE)

RequirementAnalysis

The first step in any software development activity is to determine and document the system-level requirements in either a system requirements specification (SRS) or a software requirements specification or both. Software requirements include capabilities that a user needs to solve a problem or achieve an objective as well as capabilities that a system or component needs to satisfy a contract, standard, or other formally imposed document [6].

We can categorize software requirements as follows [7]:

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

Functional requirements specify functions that a system or system component must be capable of performing.

Performance requirements specify performance characteristics that a system or system component must possess such as speed, accuracy, and frequency.

External interface requirements specify hardware, software, or database elements with which a system or component must interface, or set forth constraints on formats, timing, or other factors caused by such an interface.

Design constraints affect or constrain the design of a software system or software system component, for example, language requirements, physical hardware requirements, software development standards, and software quality assurance standards.

Quality attributes specify the degree to which software possesses attributes that affect quality, such as correctness, reliability, maintainability, and portability.

Software requirements analysis begins after system engineering has defined the acquirer and user system requirements. Its functions include identification of all – or as many as possible – software system requirements, and its conclusion marks the established requirements baseline, sometimes called the allocated baseline.

SoftwareDesign

Software design is the process of selecting and documenting the most effective and efficient system elements that together will implement the software system requirements [8]. The design represents a specific, logical approach to meet the software requirements.

Software design is traditionally partitioned into two components:

Architectural design is equivalent to system design, during which the developer selects the system-level structure and allocates the software requirements to the structure's components. Architectural design – sometimes called *top-level design* or *preliminary design* – typically defines and structures computer program components and data, defines the interfaces, and prepares timing and sizing estimates. It includes information such as the overall processing architecture, function allocations

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

(but not detailed descriptions), data flows, system utilities, operating system interfaces, and storage throughput.

Detailed design is equivalent to component engineering. The components in this case are independent software modules and artifacts.

The methodology proposed here allocates architectural design to SwSE and detailed design to software engineering.

Process Planning

Planning specifies the project goals and objectives and the strategies, policies, plans, and procedures for achieving them. It defines in advance what to do, how to do it, when to do it, and who will do it.

Planning a software engineering project consists of SwSE management activities that lead to selecting a course of action from alternative possibilities and defining a program for completing those actions.

There is an erroneous assumption that project management performs all project planning. In reality, project planning has two components – one accomplished by project management and the other by SwSE – and the bulk of project planning is an SwSE function.

Table 2 shows an example partitioning of planning functions for a software system project.

SwSE Planning Activities	Project Management Planning Activities
Determines tasks to be done.	Determines skills necessary to do the tasks.
Establishes order of precedence between tasks.	Establishes schedule for completing the project.
Determines size of the effort.	Determines cost of the effort (in staff time).
Determines technical approach to solving the problem.	Determines managerial approach to monitoring the project's status.
Selects analysis and design tools.	Selects planning tools.
Determines technical risks.	Determines management risks.
Defines process model.	Defines process model.
Updates plans when the requirements or development environment change.	Updates plans when the managerial conditions and environment change.

(© 2002, IEEE)

Table 2: Process Planning Versus Project Planning

VV&T

The *VV&T* effort determines whether the engineering process is correct and the products are in compliance with their requirements [9]. The following critical definitions apply:

Verification determines whether the products of a given phase of the software development cycle fulfill the requirements established during the

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

previous phase. Verification answers the question, *am I building the product right?*

Validation determines the correctness of the final program or software with respect to the user's needs and requirements. Validation answers the question, *am I building the right product?*

Testing is the execution of a program or partial program, with known inputs and outputs that are both predicted and observed for the purpose of finding errors. Testing is frequently considered part of validation.

Verification and Validation (V&V) is a continuous process of monitoring system engineering, SwSE, software engineering, and project management activities to determine that they are following the technical and managerial plans, specifications, standards, and procedures. V&V also evaluates the software engineering project's interim and final products. Interim products include requirements specifications, design descriptions, test plans, and review results. Final products include software, user manuals, training manuals, and so forth.

Any individual or function within a software development project can do V&V. SwSE uses V&V techniques and tools to evaluate requirements specifications, design descriptions, and other interim products of the SwSE process. It uses testing to determine if the final product meets the project requirements specifications.

The last step in any software development activity is to validate and test the final software product against the software requirements specification and to validate and test the final system product against the SRS. System engineering and SwSE are disciplines used primarily for technical planning in the front end of the system life cycle and for verifying that the plans were met at the project's end. Unfortunately, a project often overlooks these disciplines, especially if it consists entirely of software or runs on COTS computers.

THE PROCESS

A systems engineering process is a process for applying systems engineering techniques to the development of all kinds of systems. Systems

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

engineering processes are related to the stages in a system life cycle. The systems engineering process usually begins at an early stage of the system life cycle and at the very beginning of a project [10].

The purpose of system engineer is to produce system that satisfy the customer's need , increase the probability of the system success, reduce risk and reduce total life cycle cost.

The systems engineering process can be decomposed into,

(i) Systems Engineering Technical Process,

(ii) Systems Engineering Management Process

The goal of the Management Process is to organize the technical effort in the lifecycle, while the Technical Process includes assessing available information, defining effectiveness measures, to create a behavior model, create a structure model, perform trade-off analysis, and create sequential build & test plan [11].

Depending on their application, although there are several models that are used in the industry, all of them aim to identify the relation between the various stages mentioned above and incorporate feedback. Examples of such models include the Waterfall model and the VEE model. The system lifecycle in systems engineering is an examination of a system or proposed system that addresses all phases of its existence to include system design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out and disposal.

CONCLUSION

Ignoring the systems aspects of any software project can result in software that will not run on the hardware selected or will not integrate with other software systems.Conducting software engineering without conducting SwSE puts a project in jeopardy of being incomplete or having components which do not work together, and/or exceeding the project's scheduled budget.

Software engineering and SwSE are primarily disciplines used in the front end of the system life cycle for technical planning and at the very late part of the life cycle to verify if the plans have been met. A review of the emphasis in this paper will show that much of the work of planning and SwSE is

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

done during the top-level requirements analysis and top-level design phases. The other major activity of SwSE is the final validation and testing of the completed system.

Software engineering principles, activities, tasks, and procedures can be applied to software development. This paper has summarized, in broad steps, what is necessary to implement SwSE on either a hardware-software system (that is primarily software) or on an almost total software system.The above represented can be quantified using a quantification tool called optimizing technique.

REFERENCES

Gibbs, W.W. "Software's Chronic Crisis." *Scientific American* Sept. 1994: 86-95.

IEEE. *Software Engineering Standards Collection*. Vol. 1-4. Piscataway: IEEE Press, 1999.

IEEE. *Standard for Application and Management of the System Engineering Process*. Std. 1220-1998, Piscataway: IEEE Press, 1998.

Royce, W.W. "Software Systems Engineering." *Management of Software Acquisition*. Fort Belvoir, VA: Defense Systems Management College, 1981-1988.

IEEE. *Standard for Software Project Management Plans*. Std. 1058-1998. Piscataway: IEEE Press, 1998.

IEEE. *Standard Glossary of Software Engineering Terminology*. Std. 610. 12-1990. Piscataway: IEEE Press, 1990.

IEEE. *Recommended Practice for Software Requirements Specifications*. Std. 830-1998. Piscataway: IEEE Press, 1998.

IEEE. *Recommended Practice for Software Design Descriptions*. Std. 1016-1998. Piscataway: IEEE Press, 1998.

IEEE. *Standard for Software Verification and Validation*. Std. 1012-1998. Piscataway: IEEE Press, 1998.

Bahill, A. T. and Briggs, C, The systems engineering started in the middle process: a consensus of system engineers and project managers, *Systems Engineering*, 4(2), 156-167, 2001.

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-191 PAGE NO: 1167-1178
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

David W. Oliver, Timothy P. Kelliher & James G. Keegan, Jr. *Engineering Complex Systems with Models and Objects. McGraw-Hill*, 1997.

Operations Research: An Introduction (7th Edition) Hamdy A. Taha

Software Engineering A Practioner's Approach, Fifth Edition, Roger S.Pressman.

Boehm, Barry. "Integrating Software Engineering and Systems Engineering." *1st Annual Conference on Systems Integration, 12-14 March 2003.*