Chapter 15

Engineering Foundations

Acronyms

CAD	Computer-Aided Design
CMMI	Capability Maturity Model Integrated
pdf	Probability Density Function
pmf	Probability Mass Function
RCA	Root Cause Analysis
SDLC	Software Development Life Cycle

Introduction

IEEE defines engineering as "the application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems or processes" [1]. This chapter outlines some of the engineering foundational skills and techniques that are useful for a software engineer. The focus is on topics that support other KAs while minimizing duplication of subjects covered elsewhere in this document.

As the theory and practice of software engineering matures, it is increasingly apparent that software engineering is an engineering discipline that is based on knowledge and skills common to all engineering disciplines. This Engineering Foundations Knowledge Area (KA) is concerned with the engineering foundations that apply to software engineering and other engineering disciplines. Subareas in this KA include empirical methods and experimental techniques; statistical analysis; measurement; engineering design, modeling, prototyping, and simulation; standards; and root cause analysis. Application of this knowledge, as appropriate, will allow software engineers to develop and maintain software more efficiently and effectively, which is the goal of all engineers in all engineering disciplines.

BREAKDOWN OF TOPICS FOR COMPUTING FOUNDATIONS

The breakdown of topics for the Computing Foundations KA is shown in Figure 1.

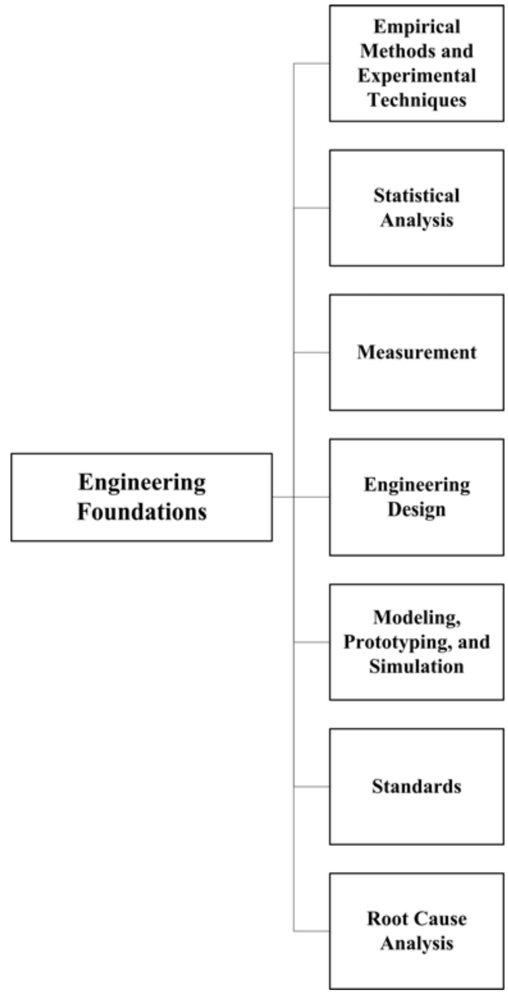

Figure 1. Breakdown of Topics for the Engineering Foundations KA.

1. Empirical Methods and Experimental Techniques [2* , c1]

An engineering method for problem solving involves proposing solutions or models of solutions and then conducting experiments or tests with that model to study the proposed solution. Thus, engineers must understand how to create an experiment and then analyze the results of the experiment in order to evaluate the proposed solution. Empirical methods and experimental techniques help the engineer to describe and understand variability in their observations, to identify the sources of variability, and to make decisions.

Three different types of empirical studies commonly used in engineering efforts are designed experiments, retrospective studies, and observational studies. Brief descriptions of the commonly used methods are given below.

Designed experiment

A designed or controlled experiment is an investigation of a testable hypothesis where one or more *independent variables* are manipulated to measure their effect on one or more *dependent variables*. A precondition for conducting an experiment is the existence of a clear hypothesis. It is important for an engineer to understand how to formulate clear hypotheses.

Designed experiments allow us to determine in precise terms how the variables are related and, specifically, whether a cause-effect relationship exists between them. Each combination of values of the independent variables is a *treatment*. The simplest experiments have just two treatments representing two levels of a single independent variable (e.g., using a tool vs. not using a tool). More complex experimental designs arise when more than two levels,, more than one independent variable, or any dependent variables are used.

Observational study

An observational or case study is an empirical inquiry that makes observations on processes or phenomenon within a real-life context. While an experiment deliberately ignores context, an observational or case study includes context as part of the observation. A case study is most useful when the focus of the study is on *how* and *why* questions, when the behavior of those involved in the study cannot be manipulated, and when contextual conditions are relevant and the boundaries between the phenomenon and context are not clear.

Retrospective Study

A retrospective study involves the analysis of historical data. Retrospective studies are also known as historical studies. This type of study uses data (regarding some phenomenon) that has been archived over time. This archived data is then analyzed in an attempt to find a relationship between variables, to predict future events, or to identify trends. The quality of the analysis results will depend on the quality of the information contained in the archived data. Historical data may be incomplete, inconsistently measured, or incorrect.

2. **Statistical Analysis** [2* , c9s1, c2s1, 3* , c10s3]

In order to carry out their responsibilities, software engineers must understand how different software product and process characteristics vary. Software engineers often come across situations where the relationship between different variables needs to be studied. An important point to note is that most of the studies are carried out on the basis of samples and so the observed results need to be generalized. Software engineers must, therefore, develop an adequate understanding of statistical techniques for collecting reliable data in terms of sampling and analysis to arrive at results that can be generalized. These techniques are discussed below.

Unit of analysis (sampling units), population, and sample

Unit of analysis: While carrying out any empirical study, observations need to be made on chosen units called the *units of analysis* or *sampling units.* The unit of analysis must be identified and be appropriate for the analysis. For example, when a software product company wants to find the perceived usability of a software product, the user forms the unit of analysis.

Population: The set of all respondents or items (possible sampling units) to be studied forms the *population.* As an example, consider the case of studying the perceived usability of a software product. In this case, the set of all possible users forms the population.

While defining the population, care must be exercised to understand the study and target population. There are cases when the population studied and the population for which the results are being generalized may be different. For example, when the study population consists of only past observations and generalizations are required for the future, the study population and the target population may not be the same.

Sample: A *sample* is a subset of the population. The most crucial issue towards the selection of a sample is its representativeness, including size. The samples must be drawn in a manner so as to ensure that the draws are independent and the rules of drawing the samples must be predefined so that the probability of selecting a particular sampling unit is known beforehand. This method of selecting samples is called *probability sampling.*

Random Variable: In statistical terminology, the process of making observations or measurements on the sampling units being studied is referred to as conducting the experiment. For example, if the experiment is to toss a coin 10 times and then count the number of times the coin lands on heads, each 10 tosses of the coin is a sampling unit and the number of heads for a given sample is the observation or outcome for the experiment. The outcome of an experiment is obtained in terms of real numbers and defines the *random variable* being studied. Thus, the attribute of the items being measured at the outcome of the experiment represents the random variable being studied; the observation obtained from a particular sampling unit is a particular realization of the random variable. In the example of the coin toss, the random variable is the number of heads observed for each experiment. In statistical studies, attempts are made to understand population characteristics on the basis of samples.

The fact that random variables are limited to real valued functions does not impose any restrictions. If we want to describe the outcomes of an *experiment* qualitatively—say by allocating a 5-point score like excellent, very good, etc. to the usability of a software product as perceived by different users—we can arbitrarily make the description real valued by coding the various ratings as numbers like 5, 4, etc.

The set of possible values of a random variable may be finite or infinite but countable (e.g., the set of all integers or the set of all odd numbers). In such a case, the random variable is called a *discrete random variable.* In other cases, the random variable under consideration may take values on a continuous scale and is called a *continuous random variable.*

Event: A subset of possible values of a random variable is called an *event.* Suppose X denotes some random variable; then, for example, we may define different events such as $X \geq x$ or $X < x$ and so on.

Distribution of a random variable: The range and pattern of variation of a random variable is given by its distribution. When the distribution of a random variable is known, it is possible to compute the chance of any event. Some distributions are found to occur commonly and are used to model many random variables occurring in practice in the context of engineering. A few of the more commonly occurring distributions are given below.

- Binomial distribution: used to model random variables that count the number of successes in n trials carried out independently of each other, where each trial results in success or failure. We make an assumption that the chance of obtaining a success remains constant [2* , c3s6].
- Poisson distribution: used to model the count of occurrence of some event over time or space [2* , c3s9].
- Normal distribution: used to model continuous random variables or discrete random variables by taking very large number of values [2* , c4s6].

Concept of parameters: A statistical distribution is characterized by some parameters. For example, the proportion of success in any given trial is the only parameter characterizing a Binomial distribution. Similarly, the Poisson distribution is characterized by a rate of occurrence. A normal distribution is characterized by two parameters: namely, its mean and standard deviation.

Once the values of the parameters are known, the distribution of the random variable is completely known and the chance (probability) of any event can be computed. The probabilities for a discrete random variable can be computed through the probability mass function, called the pmf. The pmf is defined at discrete points and gives the point mass—i.e., the probability that the random variable will take that particular value. Likewise, for a continuous random variable, we have the probability density function, called the pdf. The pdf is very much like density and needs to be integrated over a range to obtain the probability that the continuous random variable lies between certain values. Thus, if the pdf or pmf is known, the chances of the random variable taking certain set of values may be computed theoretically.

Concept of estimation: [2* , c6s2, c7s1, c7s3] The true values of the parameters of a distribution are usually unknown and need to be estimated from the sample observations. The estimates are functions of the sample values and are called statistics. For example, the sample mean is a statistic and may be used to estimate the population mean. Similarly, the rate of occurrence of defects estimated from the sample (rate of defects per line of code) is a statistic and serves as the estimate of the population rate of rate of defects per line of code. The statistic used to estimate some population parameter is often referred to as the *estimator* of the parameter.

A very important point to note is that the results of the estimators themselves are random. If we take a different sample, we are likely to get a different estimate of the population parameter. In the theory of estimation, we need to understand different properties of estimators—particularly, how much the estimates can vary across samples and how to choose between different alternative ways to obtain the estimates. For example, if we wish to estimate the mean of a population, we might use as our estimator a sample mean, a sample median, a sample mode, or the mid-range of the sample. Each of these estimators has different statistical properties that may impact the standard error of the estimate.

Types of estimates: [2* , c7s3, c8s1]

There are two types of estimates: namely, point estimates and interval estimates. When we use the value of a statistic to estimate a population parameter, we get a *point estimate*. As the name indicates, a *point estimate* gives a point value of the parameter being estimated.

Although point estimates are often used, they leave room for many questions. For instance, we are not told anything about the possible size of error or statistical properties of the point estimate. Thus, we might need to supplement a point estimate with the sample size as well as the variance of the estimate. Alternately, we might use an *interval estimate*. An *interval estimate* is a random interval with the lower and upper limits of the interval being functions of the sample observations as well as the sample size. The limits are computed on the basis of some assumptions regarding the sampling distribution of the point estimate on which the limits are based.

Properties of estimators: Various statistical properties of estimators are used to decide about the appropriateness of an estimator in a given situation. The most important properties are that an estimator is unbiased, efficient, and consistent with respect to the population.

Tests of hypothesis: [2* , c9s1]

A hypothesis is a statement about the possible values of a parameter. For example, suppose it is claimed that a new method of software development reduces the occurrence of defects. In this case, the hypothesis is that the rate of occurrence of defects has reduced. In tests of hypotheses, we decide—on the basis of sample observations—whether a proposed hypothesis should be accepted or rejected.

For testing hypotheses, the null and alternative hypotheses are formed. The null hypothesis is the hypothesis of no change and is denoted as H_0. The alternative hypothesis is written as H_1. It is important to note that the alternative hypothesis may be one sided or two sided. For example, if we have the null hypothesis that the population mean is not less than some given value, the alternative hypothesis would be that it is less than that value and we would have a one-sided test. However, if we have the null hypothesis that the population mean is equal to some given value, the alternative hypothesis would be that it is not equal and we would have a two-sided test (because the true value could be either less than or greater than the given value).

In order to test some hypothesis, we first compute some statistic. Along with the computation of the statistic, a region is defined such that in case the computed value of the statistic falls in that region, the null hypothesis is rejected. This region is called the critical region. In tests of hypotheses, we need to accept or reject the null hypothesis on the basis of the evidence obtained. We note that, in general, the alternative hypothesis is the hypothesis of interest. If the computed value of the statistic does not fall inside the critical region, then we cannot reject the null hypothesis. This indicates that there is not enough evidence to believe that the alternative hypothesis is true.

As the decision is being taken on the basis of sample observations, errors are possible; the types of such errors are summarized in the following table.

Nature	Statistical decision	
	Accept H_0	Reject H_0
H_0 is true	OK	Type I error (probability = α)
H_0 is false	Type II error (probability = β)	OK

In test of hypotheses, we aim at maximizing the power of the test (the value of $1-\beta$) while ensuring that the probability of a type I error (the value of α) is maintained within a particular value—typically 5 percent.

It is to be noted that construction of a test of hypothesis includes identifying statistic(s) to estimate the parameter(s) and defining a critical region such that if the computed value of the statistic falls in the critical region, the null hypothesis is rejected.

Concepts of correlation and regression: [2* , c11s2, c11s8]

A major objective of many statistical investigations is to establish relationships that make it possible to predict one or more variables in terms of others. Although it is desirable to predict a quantity exactly in terms of another quantity, it is seldom possible and, in many cases, we have to be satisfied with estimating the average or expected values.

The relationship between two variables is studied using the methods of correlation and regression. Both these concepts are explained briefly in the following paragraphs.

Correlation: The strength of linear relationship between two variables is measured using the *correlation coefficient.* While computing the correlation coefficient between two variables, we assume that these variables measure two different attributes of the same entity. The correlation coefficient takes a value between –1 to +1. The values –1 and +1 indicate a situation when the association between the variables is perfect—i.e., given the value of one variable, the other can be estimated with no error. A positive correlation coefficient indicates a positive relationship—that is, if one variable increases, so does the other. On the other hand, when the variables are negatively correlated, an increase of one leads to a decrease of the other.

It is important to remember that correlation does not imply causation. Thus, if two variables are correlated, we cannot conclude that one causes the other.

Regression: The correlation analysis only measures the degree of relationship between two variables. The analysis to find the relationship between two variables is called *regression analysis.* The strength of the relationship between two variables is measured using the coefficient of determination. This is a value between 0 and 1. The closer the coefficient is to 1, the stronger the relationship between the variables. A value of 1 indicates a perfect relationship.

3. Measurement [4* , c3s1, c3s2, 5* , c4s4 , 6* , c7s5]

Knowing what to measure and which measurement method to use is critical in engineering endeavors. It is important that everyone involved in an engineering project understand the measurement methods and the measurement results that will be used.

Measurands can be physical, environmental, economic, operational, or some other sort of measurand that is meaningful for the particular project. This section explores the theory of measurement and how it is fundamental to engineering. Measurement starts as a conceptualization then moves from abstract concepts to definitions of the measurement method to the actual application of that method to obtain a measurement result. Each of these steps must be understood, communicated, and properly employed in order to generate usable data. In traditional engineering, direct measures are often used. In software engineering, a combination of both direct and derived measures is necessary[6* , p273].

The theory of measurement states that measurement is an attempt to describe an underlying real empirical system. Measurement methods define activities that allocate a value or a symbol to an attribute of an entity.

Attributes must then be defined in terms of the operations used to identify and measure them— that is, the measurement methods. In this approach, a measurement method is defined to be a precisely specified operation that yields a number (called the *measurement result)* when measuring an attribute. It follows that, to be useful, the measurement method has to be well defined. Arbitrariness in the method will reflect itself in ambiguity in the measurement results.

In some cases—particularly in the physical world—the attributes that we wish to measure are easy to grasp; however, in an artificial world like software engineering, defining the attributes may not be that simple. For example, the attributes of height, weight, distance, etc. are easily and uniformly understood (though they may not be very easy to measure in all circumstances), whereas attributes such as software size or complexity require clear definitions.

Operational Definitions: The definition of attributes, to start with, is often rather abstract. Such definitions do not facilitate measurements. For example, we may define a circle as *"a line forming a closed loop such that the distance between any point on this line and a fixed interior point called the center is constant."* We may further say that the fixed distance from the center to any point on the closed loop gives the *radius* of the circle. It may be noted that though the concept has been defined, no means of measuring the radius has been proposed. The operational definition specifies the exact steps or method used to carry out a specific measurement. This can also be called the *measurement method*; sometimes a *measurement procedure* may be required to be even more precise.

The importance of operational definitions can hardly be overstated. Take the case of the apparently simple measurement of height of individuals. Unless we specify various factors like the time when the height will be measured (it is known that the height of individuals vary across various time points of the day), how the variability due to hair would be taken care of, whether the measurement will be with or without shoes, what kind of accuracy is expected (correct up to an inch, ½ inch, centimeter, etc.)—even this simple measurement will lead to substantial variation. Software engineers must appreciate the need to define measures from an operational perspective.

Levels (scales) of measurement: [4* , c3s2, 6* , c7s5] Once the operational definitions are determined, the actual measurements need to be undertaken. It is to be noted that measurement may be carried out in four different scales: namely, nominal, ordinal, interval, and ratio. Brief descriptions of each are given below.

- **Nominal scale:** This is the lowest level of measurement and represents the most unrestricted assignment of numerals. The numerals serve only as labels, and words or letters would serve as well. The nominal scale of measurement involves only classification and the observed sampling units are put into any one of the mutually exclusive and collectively exhaustive categories (classes). Some examples of nominal scales are:
    - Job titles in a company
    - The software development life cycle (SDLC) model (like waterfall, iterative, agile, etc.) followed by different software projects

  In nominal scale, the names of the different categories are just labels and no relationship between them is assumed. The only operation that can be carried out on nominal scale is that of counting the number of occurrences in the different classes. However, statistical analyses may be carried out to understand how entities belonging to different classes perform with respect to some other response variable.

- **Ordinal scale:** Refers to the measurement scale where the different values obtained through the process of measurement have an implicit ordering. The intervals between values are not specified and there is no objectively defined zero element. Typical examples of measurements in ordinal scales are:
    - Skill levels (low, medium, high)
    - Capability Maturity Model Integrated (CMMI) maturity levels of software development organizations
    - Level of adherence to process as measured in a 5-point scale of 5 indicating total adherence and 1 indicating no adherence at all

Measurement in ordinal scale satisfies the transitivity property in the sense that if A > B and B > C, then A > C. However, arithmetic operations cannot be carried out on variables measured in ordinal scales. Thus, if we measure customer satisfaction on a 5-point ordinal scale of 5 implying a very high level of satisfaction and 1 implying a very high level of dissatisfaction, we cannot say that a score of four is twice as good as a score of two. However, we can find the median, as computation of the median involves counting only. It is important to note that ordinal scale measures are commonly misused and such misuse can lead to erroneous conclusions[6* , p274].

- **Interval scales:** With the interval scale, we come to a form that is quantitative in the ordinary sense of the word. Almost all the usual statistical measures are applicable here, unless they require knowledge of a *true* zero point. The zero point on an interval scale is a matter of convention. Ratios do not make sense but the difference between levels of attributes can be computed and is meaningful. Some examples of interval scale of measurement are:
    - Measurement of temperature in different scales like Celsius and Fahrenheit. Suppose T_1 and T_2 are temperatures measured in some scale. We note that the fact that T_1 is twice T_2 does not mean that one object is twice as hot as another. We also note that the zero points are arbitrary.
    - Calendar dates. While the difference between dates to measure the time elapsed is a meaningful concept, the ratio does not make sense.
    - Many psychological measurements aspire to create interval scales. Intelligence is often measured in interval scale, as it is not necessary to define what zero intelligence would mean.

  If a variable is measured in interval scale, most of the usual statistical analyses like mean, standard deviation, correlation, and regression may be carried out on the measured values.

- **Ratio scale:** These are quite commonly encountered in physical science. These scales of measures are characterized by the fact that operations exist for determining all 4 relations: equality, rank order, equality of intervals, and equality of ratios. Once such a scale is available, its numerical values can be transformed from one unit to another by just multiplying by a constant, e.g., conversion of inch to feet or centimeter. When measurements are being made in ratio scale, existence of a non-arbitrary zero is mandatory. All statistical measures are applicable to ratio scale; logarithm usage is valid only when these scales are used, like in the case of decibels. Some examples of ratio measures are:
    - The number of statements in a software program
    - Temperature measured in the Kelvin (K) scale.

Direct and Derived Measures: [6* , c7s5]

Measures may be either direct or derived (sometimes called indirect measures). An example of a *direct measure* would be a count of how many times an event occurred, such as the number of defects found in a software product. A *derived measure* is one that combines direct measures in some way that is consistent with the measurement method. An example of a derived measure would be calculating the productivity of a team as the number of lines of code developed per developer-month. In both cases, the measurement method determines how to make the measurement.

Reliability and Validity: [4* , c3s4, c3s5]

A basic question to be asked for any measurement method is whether the proposed measurement method is truly measuring the concept with good quality. Reliability and validity are the two most important criteria to address this question.

The reliability of a measurement method is the extent to which the application of the measurement method yields consistent measurement results. Essentially, *reliability* refers to the consistency of the values obtained when the same item is measured a number of times. When the results agree with each other, the measurement method is said to be reliable. Reliability usually depends on the operational definition. It can be quantified by using the index of variation, which is computed as the ratio between the standard deviation and the mean. The smaller the index, the more reliable the resulting measurement results.

Validity refers to whether the measurement method really measures what we intend to measure. Validity of a measurement method may be looked at from three different perspectives: namely, *construct validity*, *criteria validity,* and *content validity*.

Assessing Reliability: [4* , c3s5]

There are several methods for assessing reliability; these include the test-retest method, the alternative form method, the split-halves method, and the internal consistency method. The easiest of these is the test-retest method. In the test-retest method, we simply apply the measurement method to the same subjects twice. The correlation coefficient between the first and second set of measurement results gives the reliability of the measurement method.

4. Engineering Design: [5* , c1s2, c1s3, c1s4]

A product's life-cycle costs are largely influenced by the design of the product. This is true for manufactured products as well as for software products. The design of a software product is guided by the features to be included and the quality attributes to be provided. It is important to note that software engineers use the term "design" within their own context; while there are some commonalities, there are also many differences between engineering design as discussed in this section and software engineering design as discussed in the Software Design KA. The scope of engineering design is generally viewed as much broader than that of software design. The primary aim of this section is to identify the concepts needed to develop a clear understanding regarding the process of engineering design.

Many disciplines engage in problem-solving activities where there is a single correct solution. In engineering, most problems have many solutions and the focus is on finding a feasible solution (among the many alternatives) that best meets the needs presented. The set of possible solutions is often constrained by explicitly imposed limitations such as cost, available resources, and the state of discipline or domain knowledge. In engineering problems, sometimes there are also implicit constraints (such as the physical properties of materials or laws of physics) that also restrict the set of feasible solutions for a given problem.

Engineering Design in Engineering Education:

The importance of engineering design in engineering education can be clearly seen by the high expectations held by various accreditation bodies for engineering education. Both the Canadian Engineering Accreditation Board and the ABET (Accreditation Board for Engineering and Technology) note the importance of including engineering design in education programs.

The Canadian Engineering Accreditation Board includes requirements on the amount of engineering design experience/coursework that is necessary for engineering students as well as qualifications for the faculty members who teach such coursework or supervise design projects. Their accreditation criteria states: "*Design: An ability to design solutions for complex, open-ended engineering problems and to design systems, components or processes that meet specified needs with appropriate attention to health and safety risks, applicable standards, and economic, environmental, cultural and societal considerations"* [7, p. 12].

In a similar manner, the Accreditation Board for Engineering and Technology (ABET) defines engineering design as *"…the process of devising a system, component, or process to meet desired needs. It is a decision-making process (often iterative), in which the basic sciences, mathematics, and the engineering sciences are applied to convert resources optimally to meet these stated needs"* [8, p. 4].

Thus, it is clear that engineering design is a vital component in the training and education for all engineers. The remainder of this section will focus on various aspects of engineering design.

Design as a Problem-Solving Activity: [5* , c1s4, c2s1, c3s3]

It is to be noted that engineering design is primarily a problem-solving activity. Design problems are open ended and more vaguely defined. There are usually several alternative ways to solve the same problem. Design is generally considered to be a *wicked problem*—a term first coined by Horst Rittel in the 1960s when design methods were a subject of intense interest. Rittel sought an alternative to the linear, step-by-step model of the design process being explored by many designers and design theorists and argued that most of the problems addressed by the designers are *wicked problems*. As explained by Steve McConnell, a wicked problem is one that could be clearly defined only by solving it or by solving part of it. This paradox implies, essentially, that a wicked problem has to be solved once in order to define it clearly and then solved again to create a solution that works. This has been an important insight for software designers for several decades [9* , c5s1].

Steps Involved in Engineering Design:

Engineering problem solving begins when a need is recognized and no existing solution will meet that need. As part of this problem solving, the design goals to be achieved by the solution should be identified. Additionally, a set of acceptance criteria must be defined and used to determine how well a proposed solution will satisfy the need. Once a need for a solution to a problem has been identified, the process of engineering design has the following generic steps:

a. Define the problem
b. Gather pertinent information
c. Generate multiple solutions
d. Analyze and select a solution
e. Test and implement the solution

All of the engineering design steps are iterative and knowledge gained at any step in the process may be used to inform earlier tasks and trigger an iteration in the process. These steps are expanded in the subsequent sections.

a) Define the problem At this stage, the customer's requirements are gathered. Specific information about product functions and features are also closely examined. This step includes refining the problem statement to identify the real problem to be solved; set the design goals and criteria for success.

The problem definition is a crucial stage in engineering design. A point to note is that this step is deceptively simple. Thus, enough care must be taken to carry out this step judiciously. It is important to identify needs and link the success criteria with the required product characteristics.

b) Gather pertinent information At this stage, the designer attempts to expand his/her knowledge about the problem. This is a vital, yet often neglected, stage. Gathering pertinent information can reveal facts leading to a redefinition of the problem—in particular, mistakes and false starts may be identified. While gathering pertinent information, care must be taken to identify how a product may be used as well as misused. It is also important to understand the perceived value of the product/service being offered. Included in the pertinent information is a list of constraints that must be satisfied by the solution or that may limit the set of feasible solutions.

c) Generate multiple solutions During this stage, different solutions to the same problem are developed. It has already been stated that design problems have multiple solutions. The goal of this step is to conceptualize multiple possible solutions and refine them to a sufficient level of detail that a comparison can be done among them.

d) Analyze and select a solution Once alternative solutions have been identified, they need to be analyzed to identify the solution that best suits the current situation. The analysis includes a functional analysis to assess whether the proposed design would meet the functional requirements. Physical solutions that involve human users often include analysis of the ergonomics or user friendliness of the proposed solution. Other aspects of the solution—such as product safety and liability, an economic or market analysis to ensure a return (profit) on the solution, performance predictions and analysis to meet quality characteristics, opportunities for incorrect data input or hardware malfunctions, and so on—may be studied. The types and amount of analysis used on a proposed solution is dependent on the type of problem and the needs that the solution must address as well as the constraints imposed on the design.

e) Test and implement the solution The final phase of the design process is implementation. Implementation refers to development and testing of the proposed solution. Sometimes a preliminary, partial solution called a *prototyp*e may be developed initially to test the proposed design solution under certain conditions. Feedback resulting from testing a prototype may be used either to refine the design or drive the selection of an alternative design solution. One of the most important activities in design is documentation of the design solution as well as of the tradeoffs for the choices made in the design of the solution. This work should be carried out in a manner such that the solution to the design problem can be communicated clearly to others.

The testing and verification take us back to the success criteria. The software engineer needs to devise tests such that the ability of the design to meet the success criteria is demonstrated. While designing the tests, the software engineer must think through different possible failure modes and then design tests based on those failure modes. The software engineer may choose to carry out designed experiments to assess the validity of the design.

5. Modeling, Simulation, and Prototyping [5* , c6, 10* , c13s3, 11* , c2s3.1]

Modeling is part of the abstraction process used to represent some aspects of a system. Simulation uses a model of the system and provides a means of conducting designed experiments with that model to better understand the system, its behavior, and relationships between subsystems, as well as to analyze aspects of the design. Modeling and simulation are techniques that can be used to construct theories or hypotheses about the behavior of the system; engineers then use those theories to make predictions about the system. Prototyping is another abstraction process where a partial representation (that captures aspects of interest) of the product or system is built. A prototype may be an initial version of the system but lacks the full functionality of the final version.

Modeling:

A model is always an abstraction of some real or imagined artifact. Engineers use models in many ways as part of their problem-solving activities. Some models are physical, such as a made-to-scale miniature construction of a bridge or building. Other models may be non-physical representations, such as a CAD drawing of a cog or a mathematical model for a process. Models help engineers reason and understand aspects of a problem. They can also help engineers understand what they do know and what they don't know about the problem at hand.

There are three types of models: iconic, analogic, and symbolic. An iconic model is a visually equivalent but incomplete 2-dimensional or 3-dimensional representation—for example, maps, globes, or built-to-scale models of structures such as bridges or highways.. An iconic model actually resembles the artifact modeled.

In contrast, an analogic model is a functionally equivalent but incomplete representation. That is, the model behaves like the physical artifact even though it may not physically resemble it. Examples of analogic models include a miniature airplane for wind tunnel testing or a computer simulation of a manufacturing process.

Finally, a symbolic model is a higher level of abstraction, where the model is represented using symbols such as equations. The model captures the relevant aspects of the process or system in symbolic form. The symbols can then be used to increase the engineer's understanding of the final system. An example is an equation such as $F=Ma$. Such mathematical models can be used to describe and predict properties or behavior of the final system or product.

Simulation:

All simulation models are a specification of reality. A central issue in simulation is to abstract and specify an appropriate simplification of reality. Developing this abstraction is of vital importance, as misspecification of the abstraction would invalidate the results of the simulation exercise. Simulation can be used for a variety of testing purposes.

Simulation is classified based on the type of system under study. Thus, simulation can be either continuous or discrete. In the context of software engineering, the emphasis will be primarily on discrete simulation. Discrete simulations may model event scheduling or process interaction. The main components in such a model include entities, activities and events, resources, state of the system, a simulation clock, and a random number generator. Output is generated by the simulation and must be analyzed.

An important problem in the development of a discrete simulation is that of initialization. Before a simulation can be run, the initial values of all the state variables must be provided. As the simulation designer may not know what initial values are appropriate for the state variables, these values might be chosen somewhat arbitrarily. For instance, it might be decided that a queue should be initialized as empty and idle. Such a choice of initial condition can have a significant but unrecognized impact on the outcome of the simulation.

Prototyping:

Constructing a prototype of a system is another abstraction process. In this case, an initial version of the system is constructed, often while the system is being designed. This helps the designers determine the feasibility of their design.

There are many uses for a prototype, including the elicitation of requirements, the design and refinement of a user interface to the system, validation of functional requirements, and so on. The objectives and purposes for building the prototype will determine its construction and the level of abstraction used.

The role of prototyping is somewhat different for physical systems and software. With physical systems, the prototype may actually be the first fully functional version of a system or it may be a model of the system. In software engineering, prototypes are also an abstract model of part of the software but are usually not constructed with all of the architectural, performance, and other quality characteristics expected in the finished product. In either case, prototype construction must have a clear purpose and be planned, monitored, and controlled—it is a technique to study a specific problem within a limited context[6* , c2s8].

In conclusion, modeling, simulation, and prototyping are powerful techniques for studying the behavior of a system from a given perspective. All can be used to perform designed experiments to study various aspects of the system. However, these are abstractions and, as such, may not model all attributes of interest.

6. Standards: [5* , c9s3.2, 12* , c1s2]

Moore states that a "*standard can be; (a) an object or measure of comparison that defines or represents the magnitude of a unit; (b) a characterization that establishes allowable tolerances for categories of items; and (c) a degree or level of required excellence or attainment. Standards are definitional in nature, established either to further understanding and interaction or to acknowledge observed (or desired) norms of exhibited characteristics or behavior"*[12* , p8].

Standards provide requirements, specifications, guidelines, or characteristics that must be observed by engineers so that the products, processes, and materials have acceptable levels of quality. The qualities that various standards provide may be those of safety, reliability, or other product characteristics. Standards are considered critical to engineers and engineers are expected to be familiar with and to use the appropriate standards in their discipline.

Organizations and individuals are not forced to adhere to standards but do so voluntarily because they find some benefit from doing so. Compliance or conformance to a standard lets an organization say to the public that they (or their products) meet the requirements stated in that standard. Thus, standards divide organizations or their products into those that conform to the standard and those that do not. For a standard to be useful, conformance with the standard must add value—real or perceived—to the product, process, or effort.

Apart from the organizational goals, standards are used for a number of other purposes such as protecting the buyer, protecting the business, and better defining the methods and procedures to be followed by the practice. Standards also provide users with a common terminology and expectations.

There are three internationally recognized standards-making organizations: the International Telecommunications Union, the International Electrotechnical Commission (IEC), and the International Organization for Standardization (ISO). In addition, there are regional and governmentally recognized organizations that generate standards for that region or country. For example, in the United States, there are over 300 organizations that develop standards. These include organizations such as the American National Standards Institute (ANSI), the American Society for Testing and Materials (ASTM), the Society of Automotive Engineers (SAE), and Underwriters Laboratories, Inc. (UL), as well as the US government.

There is a set of commonly used principles behind standards. Standards makers attempt to have consensus around their decisions. There is usually an openness within the community of interest so that once a standard has been set, there is a good chance that it will be widely accepted. Most standards organizations have well-defined processes for their efforts and adhere to those processes carefully. Engineers must be aware of the existing standards but must also update their understanding of the standards as those standards change over time.

In many engineering endeavors, knowing and understanding the applicable standards is critical and the law may even require use of particular standards. In these cases, the standards often represent minimal requirements that must be met by the endeavor and thus are an element in the constraints imposed on any design effort. The engineer must review all current standards related to a given endeavor and determine which must be met. Their designs must then incorporate any and all constraints imposed by the applicable standard. Standards important to software engineers are discussed in more detail in an appendix specifically on this subject.

7. Root Cause Analysis: [4* , c5, c3s7, c9s8, 5* , c9s3, c9s4, c9s5, 12* , c13s3.4.5]

Root cause analysis (RCA) is a process designed to investigate and identify why and how an undesirable event has happened. Root causes are underlying causes. The investigator should attempt to identify specific underlying causes of the event that has occurred. The primary objective of RCA is to prevent recurrence of the undesirable event. Thus, the more specific the investigator can be about why an event occurred, the easier it will be to prevent recurrence. A common way to identify specific underlying cause(s) is to ask a series of *why* questions.

Techniques for Conducting Root Cause Analysis[4* , c5, 5* , c3]

There are many approaches used for both quality control and root cause analysis. The first step in any root cause analysis effort is to identify the real problem. Techniques such as statement-restatement, why-why diagrams, the revision method, present state and desired state diagrams, and the fresh-eye approach are used to identify and refine the real problem that needs to be addressed.

Once the real problem has been identified, then work can begin to determine the cause of the problem. Ishikawa is known for the seven tools for quality control that he promoted. Some of those tools are helpful in identifying the causes for a given problem. Those tools are check sheets or checklists, Pareto diagrams, histograms, run charts, scatter diagrams, control charts, and fishbone or cause-and-effect diagrams. More recently, other approaches for quality improvement and root cause analysis have emerged. Some examples of these newer methods are affinity diagrams, relations diagrams, tree diagrams, matrix charts, matrix data analysis charts, process decision program charts, and arrow diagrams. A few of these techniques are briefly described below.

A fishbone or cause-and-effect diagram is a way to visualize the various factors that affect some characteristic. The main line in the diagram represents the problem and the connecting lines represent the factors that lead to or influenced the problem. Those factors are broken down into sub-factors and sub-sub-factors until root causes can be identified.

A very simple approach that is useful in quality control is the use of a checklist. Checklists are a list of key points in a process with tasks that must be completed. As each task is completed, it is checked off the list. If a problem occurs, then sometimes the checklist can quickly identify tasks that may have been skipped or only partially completed.

Finally, relations diagrams are a means for displaying complex relationships. They give visual support to cause-and-effect thinking. The diagram relates the specific to the general, revealing key causes and key effects.

Corrective and Preventive Actions:

Root cause analysis aims at preventing the recurrence of undesirable events. However, the main aim of a software engineer should be to prevent occurrence of undesirable events and this involves reduction of common cause variation. Reduction of variation due to common causes requires utilization of a number of techniques. An important point to note is that these techniques should be used offline and not necessarily in direct response to the occurrence of some undesirable event. Some of the techniques that may be used to reduce variation due to common causes are given below.

i. Cause-and-effect diagrams may be used to identify the sub and sub-sub causes.
ii. Fault tree analysis is a technique that may be used to understand the sources of failures.
iii. Designed experiments may be used to understand the impact of various causes on the occurrence of undesirable events (see Empirical Methods and Experimental Techniques in this KA).
iv. Various kinds of correlation analyses may be used to understand the relationship between various causes and their impact. These techniques may be used in cases when conducting controlled experiments is difficult but data may be gathered. (See Statistical Analysis in this KA.)

[1] IEEE/ISO/IEC, "IEEE/ISO/IEC 24765: Systems and Software Engineering - Vocabulary," 1st ed, 2010.

[2*] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 4th ed. Hoboken, NJ: Wiley, 2007.

[3*] L. Null and J. Lobur, *The Essentials of Computer Organization and Architecture*, 2nd ed. Sudbury, MA: Jones and Bartlett Publishers, 2006.

[4*] S. H. Kan, *Metrics and Models in Software Quality Engineering*, 2nd ed. Boston: Addison-Wesley, 2002.

[5*] G. Voland, *Engineering by Design*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[6*] R. E. Fairley, *Managing and Leading Software Projects*. Hoboken, NJ: Wiley-IEEE Computer Society Press, 2009.

[7] "Accreditation Criteria and Procedures," Canadian Engineering Accreditation Board, Engineers Canada2011.

[8] E. A. Commission, "Criteria for Accrediting Engineering Programs, 2012-2013," ABET2011.

[9*] S. McConnell, *Code Complete*, 2nd ed. Redmond, WA: Microsoft Press, 2004.

[10*] E. W. Cheney and D. R. Kincaid, *Numerical Mathematics and Computing*, 6th ed. Belmont, CA: Brooks/Cole, 2007.

[11*] I. Sommerville, *Software Engineering*, 9th ed. New York: Addison-Wesley, 2010.

[12*] J. W. Moore, *The Road Map to Software Engineering: A Standards-Based Guide*, 1st ed. Hoboken, NJ: Wiley-IEEE Computer Society Press, 2006.

[13] A. Abran, *Software Metrics and Software Metrology*: Wiley-IEEE Computer Society Press, 2010.

[14] W. G. Vincenti, *What Engineers Know and How they Know It*: John Hopkins University Press, 1990.

8. Reference Matrix

	Sommerville 2010 [1*]	IEEE/ISO/IEC 2010 [2]	Montgomery and Runger 2007 [3*]	Null and Lobur 2006 [4*]	Kan 2002 [5*]	Voland 2003 [6*]	Fairley 2009 [7*]	Canadian Engineering Accreditation [8]	EA Commission [9]	McConnell 2004 [10*]	Cheney and Kincaid 2007 [11*]	Moore 2006 [12*]
Empirical Methods and experimental techniques			C1									
Designed experiment												
Observational study												
Retrospective Study												
Statistical Analysis			C9s1, C2s1	C10s3								
Concept of unit of analysis (sampling units), sample and population												
Unit of analysis												
Population												
Sample												
Random Variable												
Event												
Distribution of a random variable			C3s6, C3s9, C4s6									
Concept of												

	Sommerville 2010 [1*]	IEEE/ISO/IEC 2010 [2]	Montgomery and Runger 2007 [3*]	Null and Lobur 2006 [4*]	Kan 2002 [5*]	Voland 2003 [6*]	Fairley 2009 [7*]	Canadian Engineering Accreditation [8]	EA Commission [9]	McConnell 2004 [10*]	Cheney and Kincaid 2007 [11*]	Moore 2006 [12*]
parameters												
Concept of estimation			C6s2, C7s1, C7s3									
Types of estimates			C7s3, C8s1									
Properties of estimators												
Tests of hypothesis			C9s1									
Concepts of correlation and regression			C11s2, C11s8									
Correlation												
Regression												
Measurements and measures					C3s1, C3s2	C4s4,	C7s5					
Major theories of measurement												
Operational definitions												
Levels (scales) of measurement					C3s2		C7s5					
Nominal scale												
Ordinal scale												
Interval scale												
Ratio scale												

	Sommerville 2010 [1*]	IEEE/ISO/IEC 2010 [2]	Montgomery and Runger 2007 [3*]	Null and Lobur 2006 [4*]	Kan 2002 [5*]	Voland 2003 [6*]	Fairley 2009 [7*]	Canadian Engineering Accreditation [8]	EA Commission [9]	McConnell 2004 [10*]	Cheney and Kincaid 2007 [11*]	Moore 2006 [12*]
Direct and Derived measures							C7s5					
Reliability and validity					C3s4, C3s5							
Assessing Reliability					C3s5							
Engineering Design						C1s2, C1s3, C1s4						
Design in Engineering Education								p12	General Criteria 3.			
Design as a problem solving activity						C1s4, C2s1, C3s3				C5s1		
Steps involved in Engineering Design												
Modeling, prototyping and Simulation	C2s3.1 ,					C6					C13s3	
Modeling												
Simulation												
Prototyping												
Standards						C9s3.2						C1s2
Root Cause					C5,	C9s3,						C13s3.4.

	Sommerville 2010 [1*]	IEEE/ISO/IEC 2010 [2]	Montgomery and Runger 2007 [3*]	Null and Lobur 2006 [4*]	Kan 2002 [5*]	Voland 2003 [6*]	Fairley 2009 [7*]	Canadian Engineering Accreditation [8]	EA Commission [9]	McConnell 2004 [10*]	Cheney and Kincaid 2007 [11*]	Moore 2006 [12*]
Analysis					C3s7, C9s8	C9s4, C9s5						5
Techniques for conducting Root Cause Analysis					C5	C3						
Corrective and Preventative Actions												

9. Further Readings

Software Metrics and Software Metrology.[13]

This book provides very good information on the proper use of the terms measure, measurement method and measurement outcome. It provides strong support material for the entire section on Measurement.

What Engineers Know and How They Know It. [14]

This book provides an interesting introduction to engineering foundations through a series of case studies that show many of the foundational concepts as used in real world engineering applications.