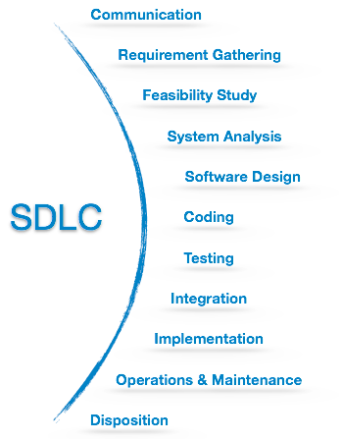
Software Engineering’s foundations is comprised of the fundamental mathematics and engineering concepts. Mathematics is the study of formal systems. The word “formal” is associated with preciseness, so there are no ambiguous or erroneous interpretations (Mathematical Foundations, Bourque). Engineering is defined as the application of a systematic, disciplined, quantifiable approach to structures, machines, product's, systems or processes by the IEEE (Engineering Foundation, Bourque). Software engineers need to obtain and maintain foundational skills and knowledge from both areas.

Software Engineering is a detailed study of engineering to the design, development, and maintenance of software (What Is Software Engineering, 2018). Software engineering was introduced to address the issue of low-quality in software projects. Software can have problems to arise when timelines and budgets are exceeded, and reduced quality is a factor. Software engineering ensures that applications are built consistently, correctly, and within the timeline and budget assigned to them (What Is Software Engineering, 2018). Software products are usually judged by how easily they can be used by the end-user (). The features it offers to the user are also a basis of judgment (What Is Software Engineering, 2018). Applications must score in the areas of operation or how well software works on budget, usability, efficiency, correctness, functionality, dependability, as well as security and safety (What Is Software Engineering, 2018). The application must also score well on transition, or how well it can be and is shifted from one platform to another (portability, reusability, and adaptability are important in this area). Maintenance is also important for software to rank highly. It needs to work in the changing environment with modularity, maintainability, flexibility, and scalability (What Is Software Engineering, 2018).

The Software Development Lifecycle (SDLC) is also an important part in software engineering and the workings of an application (see figure below).



(www.tutorialspoint.com/software\_engineering/software\_development\_life\_cycle.htm)

The first step taken is typically a user-request initiation for a specific task or a specific output. The user submits the requirements and the software development team sifts through the user requirements to categorize and differentiate between system requirements and functional requirements (What Is Software Engineering, 2018). The software engineers can then analyze if the software can be made to fulfill all of the asked requirements of the user (What Is Software Engineering, 2018). The developer then decides to map out the course of action for the project. The analysis also includes a software products limitation. Based upon the requirements and the analysis a software design can be constructed, the code can be written, and testing can be conducted at the various levels of code (What Is Software Engineering, 2018). Thus, ensuring that all levels of code work efficiently, effectively, and properly. User’s engagement and feedback can then be sought out and the cycle repeated if necessary (What Is Software Engineering, 2018). However, in order to do all of this, software engineers need to have a strong foundational understanding of mathematics and engineering.

In order for software engineers to be successful, they need to understand the foundations of mathematics. Mathematics is a key fundamental in computing as well as many coding functions. Therefore, in order to understand key coding concepts and functions, the mathematics behind them need to be understood. Discrete mathematics and logic are foundations for computer-based disciplines like that of Computer Science, Information Systems, and Software Engineering (Satyaseel, 2018). Mathematics demands that logical reasoning considering the facts and universal truth be used instead of simply just pouring out what was fed into a system. The concept of abstraction is also heavily emphasized in software engineering (Satyaseel, 2018). Every concept from implementing classes, construct, and method is based on some level of abstraction (Satyaseel, 2018). Therefore, the mathematics behind computing and coding (necessary for software engineering) is important not only for the concepts but for the practicing of rigorous reasoning with abstract objects and structures (Satyaseel, 2018). Other important mathematical concepts for software developers include working with the number system and geometry (especially with web design) (Satyaseel, 2018). There are layouts and coding the core logic as well as the procedural programs, all of which need the understanding of mathematics.

The foundational concepts of engineering are also important to software engineers. Knowledge areas that are included in both engineering and software engineering include that of empirical methods, experimental techniques, statistical analysis, measurement, engineering design, modeling, prototyping, simulations, standards and the root cause analysis (Engineering Foundation, Bourque). Maintaining these areas of knowledge and carrying these over to software engineering allows for work to be done efficiently and effectively, as is the goal of all engineers. Like engineers, software engineers need to be able to design and conduct complete experiments. They need to understand how different product and process characteristics vary and how to properly analyze these characteristics. Knowing what to measure and which measurement method to use is also critical in all engineering fields (Engineering Foundation, Bourque). The engineering design process is also necessary to carry over as a product's life cycle costs are largely impacted and influenced by the design (also true for software products) (Engineering Foundation, Bourque). Engineering standards are also very important to maintain and follow in software engineering. Standards provide requirements, specification, guidelines, or characteristics that must be observed by engineering to ensure that the product's, processes, and materials have acceptable levels of quality (Engineering Foundation, Bourque). Software engineers also need to be able to investigate and identify why and how an undesirable event occurred through the use of root cause analysis (RCA) (Engineering Foundation, Bourque). RCA needs to be maintained as a check for quality control of a product. Overall, the fundamental concepts and disciplines of engineers are necessary for software engineers to adopt and follow in order to be successful, efficient, and effective.

Engineering and mathematical foundational concepts are important to the field of software engineering. The use of rigorous logic and reasoning along with the concept of abstraction are important within the coding and understanding of code in software engineering. The disciplines applied in engineering like that of experimentation, analysis, measurement, as well as the design process are important engineering foundations that also need to be the root of software engineering. With the combination of these foundational concepts, software engineers can be successful, effective, and efficient.

Works Cited

Celebrating 40++ years of service to computing education communities. “ACM CCECC.” *Computer Engineering - ACM CCECC*, 2018, ccecc.acm.org/guidance/computer-engineering.

“Chapter 14: Mathematical Foundations.” *Guide to the Software Engineering Body of Knowledge Version 3.0 ; SWEBOK ; a Project of the IEEE Computer Society*, by Pierre Bourque, IEEE Computer Soc., 2014.

“Chapter 15: Engineering Foundations.” *Guide to the Software Engineering Body of Knowledge Version 3.0 ; SWEBOK ; a Project of the IEEE Computer Society*, by Pierre Bourque, IEEE Computer Soc., 2014.

Satyaseel, Harshit. “How Important Is Maths For Software Engineering?” *Technotification*, 20 Aug. 2018, www.technotification.com/2018/08/maths-in-software-engineering.html.

“Software Engineering - What Is Software Engineering? Software Engineering Meaning, Software Engineering Definition.” *The Economic Times*, Economic Times, 2018, economictimes.indiatimes.com/definition/software-engineering.