What is software engineering?

The field uses a systematic approach to collect and analyze business requirements in order to design, build, and test software applications to satisfy those business requirements. The software engineering field became a discipline in the 1960s and evolved as new technologies were developed and the approach to software development became more scientific. Trends in software engineering transformed from ad hoc programming towards more formal and standardized methods. Initially, the creation of software lacked a formal development process.

The inefficiencies in the software development process made it difficult to meet the rapidly increasing demand for computing resources and complex software. This led to what is known as the "Software Crisis" which began in the mid-1960s and lasted until the mid-1980s. During this period, software development often ran over budget, behind schedule, and consisted of unmanageable, buggy code. Often software development solutions that worked for small software systems did not scale to large, complex projects.

The solution to the "Software Crisis" involved transforming unorganized coding efforts into an established engineering discipline. The mid 1980s also saw a rise in the growth of computer-aided software engineering or CASE which also helped to relieve the software crisis. CASE tools can be divided into six categories: business analysis and modeling, development tools such as debugging environments, verification and validation tools, configuration management, metrics and measurement, and project management.

The term "software engineer" is often used interchangeably with software developer but there are subtle differences. Software engineers are also developers, but the term "software developer" is usually deemed narrower in scope than that of a software engineer. A software engineer's knowledge is usually broader. Software engineers take a systematic, big picture approach in their thinking to software development whereas developers may have more creative approaches. Both software engineers and software developers have specialized knowledge, but software engineers use that knowledge to build entire systems whereas software developers use their knowledge to write code to implement specific functionality within a system.

Software engineers are tasked with designing, building, and maintaining software systems. Their Responsibilities include writing and testing code, and consulting with stakeholders such as clients, third party software vendors, security specialists, and other team members. The adoption of a measured, scientific approach to software development has influenced the way software is created and designed. Today, the development process is typically guided by the

Software Development Lifecycle or SDLC. The SDLC identifies the steps needed to develop high-quality software. Later in this module, we will discuss the SDLC, the traits of high-quality software, and roles common in the field of software engineering. In this video you learned that: Software engineering is the systematic approach to design and development of software. Responsibilities of a software engineer include: Designing, building, and maintaining software systems, writing and testing code, and consulting with stakeholders, third party vendors, security specialists, and other team members. And finally, software engineers build systems while software developers implement specific functionalities.

Software Development Life Cycle (SDLC)
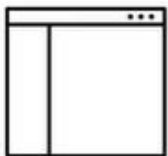
- Scientific approach to software development

- Guides the software development process

- Identifies discrete steps needed to develop software

In this video, you learned that:

- Software engineering is the systematic approach to the design and development software
- Responsibilities of a software engineer include:
  - Designing, building, and maintaining software systems
  - Writing and testing code
  - Consulting with stakeholders, third party vendors, security specialists, and other team members
- Software engineers build systems while software developers implement specific functionalities